

GIOVANA TRINDADE DA SILVA OLIVEIRA

**ESTUDO E APLICAÇÕES DA EVOLUÇÃO
DIFERENCIAL**



**UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA MECÂNICA**

2006

GIOVANA TRINDADE DA SILVA OLIVEIRA

ESTUDO E APLICAÇÕES DA EVOLUÇÃO DIFERENCIAL

Dissertação apresentada ao Programa de Pós-graduação em Engenharia Mecânica da Universidade Federal de Uberlândia, como parte dos requisitos para a obtenção do título de **MESTRE EM ENGENHARIA MECÂNICA**.

Área de concentração: Mecânica dos Sólidos e Vibrações.

Orientadora: Prof^a. Dr^a. Sezimária F. Pereira Saramago.

UBERLÂNDIA – MG

2006

Dados Internacionais de Catalogação na Publicação (CIP)

O48e Oliveira, Giovana Trindade da Silva, 1976-
 Estudo e aplicações da evolução diferencial / Giovana Trindade da
 Silva Oliveira. - 2006.
 126 p. : il.

 Orientadora: Sezimária F. Pereira Saramago.
 Dissertação (mestrado) – Universidade Federal de Uberlândia, Progra-
 ma de Pós-Graduação em Engenharia Mecânica.
 Inclui bibliografia.

 1. Vibração - Teses. 2. Mecânica dos sólidos - Teses. 3. Algoritmos
 genéticos - Teses. I. Saramago, Sezimária de Fátima Pereira. II. Univer-
 sidade Federal de Uberlândia. Programa de Pós-Graduação em Engenha-
 ria Mecânica. III. Título.

CDU: 621:534

GIOVANA TRINDADE DA SILVA OLIVEIRA

ESTUDO E APLICAÇÕES DA EVOLUÇÃO DIFERENCIAL

Dissertação **APROVADA** pelo Programa de Pós-graduação em Engenharia Mecânica da Universidade Federal de Uberlândia.

Área de concentração: Mecânica dos Sólidos e Vibrações.

Banca Examinadora:

Prof^a. Dr^a. Sezimária de Fátima Pereira Saramago – UFU – Orientadora

Prof Dr Valder Steffen Jr – UFU

Prof Dr Paulo Roberto Bergamaschi – UFG

UBERLÂNDIA, 29 de agosto de 2006

AGRADECIMENTOS

A Deus por estar presente em todos os momentos da minha vida.

À Prof^a Dr^a. Sezimária F. P. Saramago pela orientação durante a realização das pesquisas que, com muito profissionalismo e paciência, sempre indicou a melhor forma de conduzir o trabalho.

À minha família pelo apoio e incentivo na busca desta realização e principalmente ao meu marido Marco Aurélio pela compreensão.

A todos os colegas do laboratório computacional que de alguma forma contribuíram para a realização deste trabalho.

À Universidade Federal de Uberlândia e ao programa de Pós-Graduação em Engenharia Mecânica pela oportunidade de realizar este trabalho.

Aos membros da banca examinadora pelas contribuições dadas ao trabalho.

Ao CNPq - Conselho Nacional de Desenvolvimento Científico e Tecnológico pelo apoio financeiro.

OLIVEIRA, G. T. S. **Estudo e aplicações da Evolução Diferencial**. 2006. 126 p. Dissertação de Mestrado, Universidade Federal de Uberlândia, Uberlândia.

Resumo

Nas últimas décadas, a aplicação de otimização em problemas de engenharia tem crescido consideravelmente. Existem muitos métodos de otimização e o desempenho de cada um deles depende do tipo de problema considerado. Os métodos naturais, que se baseiam em regras probabilísticas, têm sido amplamente estudados. O objetivo deste trabalho é apresentar um estudo detalhado do método de otimização natural denominado Evolução Diferencial e suas estratégias, apresentando sua fundamentação teórica. Neste trabalho é realizada uma revisão de Algoritmos Genéticos e Recozimento Simulado. Estas técnicas são utilizadas para comparar os resultados por elas obtidos com os calculados aplicando Evolução Diferencial. Para verificar a eficiência do método estudado, são utilizados funções matemáticas clássicas e alguns problemas de engenharia. A Evolução Diferencial é aplicada com eficiência em problemas de otimização multi-objetivo, na presença ou não de restrições, incluindo dois problemas complexos em robótica.

Palavras Chave: Otimização, Algoritmos Evolutivos, Evolução Diferencial, Otimização multi-objetivo.

OLIVEIRA, G. T. S. **A Study and applications of Differential Evolution**. 2006. 126 p. M. Sc. Dissertation, Universidade Federal de Uberlândia, Uberlândia.

Abstract

In the last few decades, the application of optimization in engineering problems has grown considerably. There are various optimization methods and the performance of each one depends on the type of problem considered. The natural methods, which are based on probabilistic rules, have been widely studied. The purpose of this work is to present a detailed study of the natural optimization method called Differential Evolution and its strategies. A theoretical formulation is presented. In this work, a revision of Genetic Algorithms and Simulated Annealing is made. These techniques are employed to compare their results to those obtained with Differential Evolution. Classic mathematical functions and some problems of engineering are used in order to verify the efficiency of the studied technique. The Differential Evolution is applied with effectiveness in multi-objective optimization problems with and without the presence of constraints including two complex robotic problems.

Keywords: Optimization, Evolutionary Algorithms, Differential Evolution, Multi-objective Optimization.

LISTA DE FIGURAS

Figura	Descrição	Pág.
Figura 1.1.	Classificação de alguns métodos de otimização.	2
Figura 1.2.	Busca ponto a ponto.	3
Figura 2.1.	x^1 é o minimizador global, $f(x^1)$ é o valor ótimo, x^2 é um minimizador local estrito, $[x^3, x^4]$ é um conjunto de minimizadores locais não estritos.	10
Figura 2.2.	A restrição g_3 é inativa no ponto ótimo pois ela não define a estrutura de Ω em torno de X^* .	10
Figura 2.3.	$\max f(X) = \min -f(X)$	11
Figura 3.1.	Fluxograma do Recozimento Simulado.	17
Figura 3.2.	Representação de um cromossomo com n genes de cinco bits cada.	19
Figura 3.3.	Exemplo da seleção por roleta.	22
Figura 3.4.	Representação do operador cruzamento simples entre dois indivíduos.	23
Figura 3.5.	Representação do operador cruzamento uniforme com parâmetros contínuos.	23
Figura 3.6.	Representação do operador mutação.	24
Figura 3.7.	Fluxograma básico de um algoritmo genético binário.	25
Figura 4.1.	Processo para gerar o vetor doador $V^{(q+1)}$ de uma função bidimensional.	29
Figura 4.2.	Ilustração do processo de cruzamento binomial para $\alpha = 2$, $\beta = 4$, e $\gamma = Np$.	30
Figura 4.3.	Ilustração do processo de cruzamento exponencial para $\alpha = 2$, $\beta = 4$, e $\gamma = Np$.	31

Figura 5.1.	Função de Rastrigin bidimensional, dada pela Eq.(5.1).	38
Figura 5.2.	Erro absoluto da função objetivo em relação à variação do fator de perturbação F_p (fixando: $Np=10$, $Nger=200$ e $Pc=0,5$).	39
Figura 5.3.	Número de avaliação da função objetivo em relação à variação do fator de perturbação F_p (fixando: $Np=10$, $Nger=200$ e $Pc=0,5$).	40
Figura 5.4.	Erro absoluto da função objetivo em relação à variação da taxa de cruzamento Pc (fixando: $Np=10$, $Nger=200$ e $F_p=0,8$).	40
Figura 5.5.	Número de avaliação da função objetivo em relação à variação da taxa de cruzamento Pc (fixando: $Np=10$, $Nger=200$ e $F_p=0,8$).	41
Figura 5.6.	Erro absoluto da função objetivo em relação à variação do número de indivíduos da população Np (fixando: $F_p=0,8$, $Nger=200$ e $Pc=0,5$).	42
Figura 5.7.	Número de avaliação da função objetivo em relação à variação do número de indivíduos da população Np (fixando: $F_p=0,8$, $Nger=200$ e $Pc=0,5$).	42
Figura 5.8.	Erro absoluto da função objetivo em relação à variação do número máximo de gerações $Nger$ (fixando: $Np=10$, $Pc=0,6$ e $F_p=0,8$).	43
Figura 5.9.	Número de avaliação da função objetivo em relação à variação do número máximo de gerações $Nger$ (fixando: $Np=10$, $Pc=0,6$ e $F_p=0,8$).	44
Figura 5.10.	Representação gráfica da função dada pela Eq.(5.3).	46
Figura 5.11.	Solução de Pareto para o problema DEA, usando a estratégia 1 da Evolução Diferencial.	52
Figura 6.1.	Esquema cinemático de um robô manipulador 3R.	54
Figura 6.2.	Notação de Denavit e Hartenberg para manipuladores.	56
Figura 6.3.	(a) Espaço de trabalho de um robô 3R; (b) seção radial plana.	59
Figura 6.4.	Família de curvas contida na seção radial do espaço de trabalho de um particular manipulador 3R.	60
Figura 6.5.	Cálculo do volume do espaço de trabalho de manipuladores 3R.	60
Figura 6.6.	Discretização da seção radial usando malha retangular.	61

Figura 6.7.	Seção transversal do espaço de trabalho ótimo de um manipulador 3R de acordo com as técnicas utilizadas.	64
Figura 6.8.	(a) Arquitetura do CaPaMan; (b) Mecanismo de acionamento (perna i).	65
Figura 6.9.	Parâmetros associados à perna i .	67
Figura 6.10.	Orientação da plataforma móvel em função dos ângulos de rotação θ , φ e ψ .	68
Figura 6.11.	Forças atuantes na PM.	73
Figura 6.12.	Forças atuantes nos pontos de articulação H_i .	75
Figura 6.13.	Forças atuantes nas juntas esféricas.	77
Figura 6.14.	Forças nos paralelogramos articulados i ($i=1,2,3$).	78
Figura 6.15.	Diagrama de corpo livre para análise do efeito da força de inércia no segmento d_i .	78
Figura 6.16.	Fluxograma do procedimento numérico para otimização.	81
Figura 6.17.	Gráfico 3D da trajetória ótima do centro da plataforma móvel do CaPaMan e suas projeções no plano cartesiano usando AG (--) e ED (-*-).	83
Figura 6.18.	Gráfico das trajetórias iniciais e ótimas para o ângulo de entrada em função do tempo nos mecanismos 1, 2 e 3.	84
Figura 6.19.	Curvas iniciais e ótimas das velocidades angulares das manivelas de entrada nos mecanismos 1, 2 e 3.	84
Figura 6.20.	Gráfico das acelerações iniciais e ótimas das manivelas de entrada em função do tempo para o CaPaMan.	85
Figura 6.21.	Curva inicial e ótima do jerk das manivelas de entrada.	86
Figura 6.22.	Curvas iniciais e ótimas do torque nos atuadores.	86

LISTA DE TABELAS

Tabela	Descrição	Pág.
Tabela 3.1.	Exemplo do método de seleção Roleta	21
Tabela 4.1.	Representação das estratégias da evolução diferencial.	33
Tabela 5.1.	Resultados da otimização para a função de Rastrigin, Eq.(5.1).	44
Tabela 5.2.	Resultados ótimos para a função de Rosenbrok, Eq.(5.2).	45
Tabela 5.3.	Resultados da otimização da função dada pela Eq.(5.3).	46
Tabela 5.4.	Resultados da otimização para a função de Rastrigin, dada pela Eq.(5.4).	47
Tabela 5.5.	Resultados da otimização restrita para o problema definido nas Eqs.(5.5) e (5.6).	48
Tabela 5.6.	Resultados da otimização restrita para o problema definido nas Eqs.(5.8) e (5.9).	49
Tabela 5.7.	Coeficientes do custo e da emissão de seis geradores.	51
Tabela 5.8.	Resultados da otimização do problema DEA, com potências em MW.	52
Tabela 6.1.	Maximização do volume do espaço de trabalho dos robôs manipuladores 3R.	63
Tabela 6.2.	Parâmetros dimensionais do protótipo CaPaMan.	81
Tabela 6.3.	Valores ótimos obtidos do processo de otimização.	82

Lista de Símbolos

Letras latinas

A	Área da seção radial plana.
$\vec{a}_{Gbi}, \vec{a}_{Gci}, \vec{a}_{Gdi}$	Acelerações nos segmentos b_i , c_i e d_i respectivamente.
a_i, d_j	Parâmetros de Denavit-Hartenberg (translações).
$a_i^{\sup}, d_j^{\sup}$	Valores máximos que os respectivos parâmetros a_i e d_j podem assumir durante o processo de otimização.
a_P	Aceleração do centro da plataforma móvel PM.
$B_{k,d}$	Funções de concordância para B-splines de ordem $d-1$.
$B_{k,d}^i(t)$	Funções de concordância para B-splines de ordem $d-1$ da trajetória α_i .
c_i	Comprimento das bielas.
$C\alpha_j, S\alpha_j$	Co-seno e seno do ângulo α_j , respectivamente.
$C\theta_i, S\theta_i$	Co-seno e seno do ângulo θ_i , respectivamente.
d_i	Comprimento das manivelas de saída.
E, E_0	Energia e energia inicial respectivamente.
e_{bi}, e_{ci}, e_{di}	Offsets de inércia do centro de massa dos segmentos b_i , c_i e d_i respectivamente.
$f(X)$	Função objetivo ou função custo, ou ainda, função de adaptação (fitness).
F	Soma das forças de reação F_i ($i=1, 2,3$) nos pontos H_i da PM.
F_{dci}	Força no segmento c_i devido ao segmento d_i .
F_{ext}	Forças externas.
F_i	Forças nas juntas esféricas.
F_{in}	Força de inércia.
$F_{inbi}, F_{inci}, F_{indi}$	Forças nos segmentos b_i , c_i , d_i , respectivamente.
F_{iy}, F_{iz}	Forças atuantes nos pontos de articulação H_i .
$f_k(X)$	Vetor cujas componentes são k funções objetivo.
f_k^0	Solução ideal.
F_p	Fator de perturbação (ou taxa de perturbação) dos indivíduos da população.

G	Peso da plataforma móvel.
G_{bi}, G_{ci}, G_{di}	Centro de gravidade dos segmentos b_i , c_i e d_i respectivamente.
$g_j(X)$	Função de restrição de desigualdade ($j = 1, \dots, J$).
H	Ponto situado na extremidade do robô, com coordenadas em $X_3Y_3Z_3$.
h_i	Comprimento das barras conectoras entre as pernas do CaPaMan e a plataforma móvel.
H_i	Ponto situado na extremidade da barra conectora onde se localiza as juntas prismáticas.
H_j^x, H_j^y, H_j^z	Coordenadas do ponto H em relação aos eixos X_j , Y_j e Z_j , respectivamente.
$h_l(X)$	Função de restrição de igualdade ($l = 1, \dots, L$).
I	Matriz de inércia da plataforma móvel.
I_c	Matriz de inércia da PM com respeito ao centro de massa P .
$I_{Gbi}, I_{Gci}, I_{Gdi}$	Momento de inércia nos centros de gravidades dos segmentos b_i , c_i e d_i respectivamente.
J, J_0	Jerk e jerk inicial respectivamente.
K_b	Constante de Boltzmann.
k_p	Fator de penalidade na função pseudo objetivo.
L_i	Vetores com origem nos pontos O_i , e extremidades nas juntas esféricas H_i .
$(L_i)_{O_iX_iY_iZ_i}$	Vetores L_i em relação ao sistema cartesiano $O_iX_iY_iZ_i$.
M	Massa da PM.
m_b, m_c, m_d	Massa dos segmentos b , c e d respectivamente.
m_i	Comprimento das manivelas de entrada.
N	Torque resultante devido as forças F_i , em relação ao sistema de referência $OXYZ$ fixo na base PF.
N_{ext}	Torque externo.
N_{ger}	Número máximo de gerações.
N_{in}	Torque de inércia.
n_{iters}	Número de iterações para cada temperatura.
n_p	Pontos de controle da B-spline.
Np	Número de indivíduos da população.
n_{temp}	Número de temperaturas.

O	Centro da plataforma fixa.
O_i	Pontos médios das bases dos paralelogramos articulados.
OX_i	Segmento de mesma direção do eixo X_i e com origem no centro da plataforma fixa.
$OXYZ$	Sistema cartesiano inercial atado à plataforma fixa.
$O_iX_iY_iZ_i$	Sistemas cartesianos cujos centros são os pontos médios das bases dos mecanismos de quatro barras.
P	Centro da plataforma móvel.
$P(X)$	Função de penalidade na função pseudo objetivo.
Pa_d	Probabilidade acumulativa de cada cromossomo d .
p_c	Posição de cruzamento na cadeia de bits de cada cromossomo.
Pc	Probabilidade de cruzamento.
p_d	Probabilidade ou aptidão do indivíduo d da população de cromossomos.
Pd	Probabilidade de seleção dos cromossomos.
PF	Plataforma fixa.
PH_i	Segmentos que unem o centro da plataforma móvel a cada uma das juntas prismáticas.
p_i	Indica as posições dos bits numa cadeia binária.
P_i	vetores com origem no centro da plataforma móvel e extremidades nas juntas esféricas correspondendo aos respectivos δ_i .
P_i^*	Vetores com origem no centro da plataforma fixa e extremidade nas juntas prismáticas.
p_k^i	Pontos de controle geradores da B-spline da trajetória α_i .
P_m	Probabilidade de mutação.
PM	Plataforma móvel.
$PX_PY_PZ_P$	Sistema cartesiano não inercial atado à plataforma móvel.
q	q -ésima geração (iteração) no processo iterativo.
$[R_i]$	Matriz de rotação de um ângulo δ_i em torno do eixo Z .
$R(\theta, z)$	Matriz de rotação de um ângulo θ em torno do eixo Z .
$R(\varphi_y, y)$	Matriz de rotação de um ângulo φ_y em torno do eixo Y' .
$R(\psi, z)$	Matriz de rotação de um ângulo ψ em torno do eixo Z'' .
R	Matriz de transformação.

r, z	Coordenadas de um ponto da seção radial.
$rand$	Número aleatório gerado no intervalo de 0 a 1.
r_B	Raio da plataforma fixa.
r_g	Baricentro da seção radial.
$r_{\min}, r_{\max}, z_{\min}, z_{\max}$	limites do espaço de trabalho pré-estabelecidos.
$\mathfrak{R}^n$	Espaço Euclidiano n -dimensional.
$Rot(\alpha_{i-1}, X_{i-1})$	Matriz de rotação horária de um ângulo α_{i-1} em torno do eixo X_{i-1} .
$Rot(\theta_i, Z_i)$	Matriz de rotação anti-horária de um ângulo θ_i em torno do eixo Z_i .
r_P	Raio da plataforma móvel.
r_t	Fator de resfriamento da temperatura T .
R^T	Transposta da matriz de transformação.
S_i	Deslocamentos das juntas prismáticas.
T_0, Tt	Tempo inicial e tempo total de percurso, respectivamente.
T_0^m	Matriz de transformação homogênea que representa as coordenadas do efetuador em relação à base.
t	Tempo de percurso da trajetória α_i .
T	Temperatura, parâmetro de controle do SA.
T_f	Ordem de grandeza desejada para a precisão do ponto ótimo (SA).
t_{ii}, t_{if}	Tempo inicial e final da trajetória α_i , respectivamente.
T_i	Vetor do desvio padrão da função objetivo em um intervalo dado (SA).
T_i, T_f	Temperatura inicial e final, respectivamente (SA).
T_{i-1}^i	Matriz de transformação homogênea que representa o sistema $X_i Y_i Z_i$ em Relação ao sistema $X_{i-1} Y_{i-1} Z_{i-1}$.
$Trans(a_{i-1}, X_{i-1})$	Matriz de translação de a_{i-1} unidades sobre o eixo X_{i-1} .
$Trans(d_i, Z_i)$	Matriz de translação de d_i unidades sobre o eixo Z_i .
u_{ix}, u_{iy}, u_{iz}	Componentes dos vetores com origem em P e extremidade nas juntas esféricas H_i .
$U^{(q+1)}$	Vetor resultante ou vetor experimental (trial vector).
V	Volume do espaço de trabalho dos robôs manipuladores 3R.
$V^{(q+1)}$	Vetor modificado ou vetor doador (mutated vector, donor vector).
$v_i (i=1, 2, 3)$	Os vetores com origem no centro da plataforma fixa e extremidades nos pontos O_i centros dos mecanismos de quatro barras.

$V_i (i=1, \dots, 5)$	Expressões simplificadoras das equações de velocidade e acelerações angulares.
$V_i (i=6, \dots, 24)$	Expressões simplificadoras dos vetores de forças e torques.
X	Vetor das coordenadas do ponto P em relação ao sistema $OXYZ$, para a seção 6.2.
$X = (x_1, x_2, \dots, x_n)^T$	Vetor das variáveis de projeto.
$X_\alpha, X_\beta, X_\gamma, X_\rho$ e X_δ	Vetores escolhidos aleatoriamente na população para gerar novo indivíduo.
X_{best}	Melhor indivíduo da população.
$x_{H_i}, y_{H_i}, z_{H_i}$	Componentes das juntas esféricas.
x_i^{inf}, x_i^{sup}	Limites inferiores e superiores das variáveis de projeto, respectivamente.
$X_i Y_i Z_i$	Sistema de referência associado ao i -ésimo membro do robô manipulador 3R.
$X_s^{(q)}$	Vetor a ser substituído na população (target vector).
$\dot{x}, \dot{y}, \dot{z}$	Derivadas primeiras das componentes do vetor t .
$\ddot{x}, \ddot{y}, \ddot{z}$	Derivadas segundas das componentes do vetor t .
$W(H)$	Espaço de trabalho formado pelo ponto H situado na extremidade do robô.
w_k	Coefficientes de ponderação das funções multi-objetivos.

Letras gregas

α_i	Ângulos de entrada formados entre as manivelas de entrada e as bases de cada um dos mecanismos de quatro barras, para a seção 6.2.
$\dot{\alpha}_i$	Derivadas primeiras dos ângulos de entradas (velocidades angulares), para a seção 6.2.
$\ddot{\alpha}_i$	Derivadas segundas dos ângulos de entradas (acelerações angulares), para a seção 6.2.

α_i, θ_i	Parâmetros de Denavit-Hartenberg (ângulos de rotação), para a seção 6.1.
$\alpha_i(t)$	Trajetória da perna i num instante t , para a seção 6.2.
α_{ii}, α_{if}	Ponto inicial e final da trajetória i , respectivamente, para a seção 6.2.
$\alpha_i^{\inf}, \alpha_i^{\sup}$	Limites inferiores e superiores de α_i , para a seção 6.2.
β_i	Maior ângulo entre as bielas dos mecanismos de quatro barras e os vetores $\vec{a}_{Gci}$.
γ_i	Define a direção do vetor força de reação atuando na base do segmento d_i .
δ_i	Ângulos de rotações estruturais entre OX e OX_i , bem como entre PH_p e PH_i .
θ, φ, ψ	Ângulos de Euler.
$\dot{\theta}, \dot{\varphi}, \dot{\psi}$	Derivadas primeiras dos ângulos de Euler.
$\ddot{\theta}, \ddot{\varphi}, \ddot{\psi}$	Derivadas segundas dos ângulos de Euler.
Σm_i	Número total de bits (alelos) de um cromossomo, onde m_i é o tamanho do gene i .
τ_i	Torque total de entrada.
$\tau_i^{\inf}, \tau_i^{\sup}$	Limites inferiores e superiores de τ_i .
τ_{Mi}	Torque de entrada devido às pernas.
τ_{Pi}	Torque de entrada devido a PM.
$\phi(X)$	Função pseudo objetivo.
φ_y	Ângulo de rotação em torno do eixo Y' .
ψ	Ângulo de rotação em torno do eixo Z_P .
$\dot{\omega}$	Vetor aceleração angular.
$\dot{\omega}_{bi}, \dot{\omega}_{ci}, \dot{\omega}_{di}$	Acelerações angulares dos segmentos b_i, c_i e d_i respectivamente.
$\dot{\omega}_x, \dot{\omega}_y, \dot{\omega}_z$	Componentes do vetor aceleração angular.
ω	Vetor velocidade angular.
$\omega_x, \omega_y, \omega_z$	Componentes do vetor velocidade angular.

SUMÁRIO

Resumo.....	vii
Abstract.....	ix
Lista de Figuras.....	xi
Lista de Tabelas.....	xv
Lista de Símbolos	xvii
 CAPÍTULO 1 - Introdução.....	 1
 CAPÍTULO 2 - Formulação do problema geral de otimização.....	 9
 CAPÍTULO 3 - Revisão de alguns métodos naturais.....	 15
3.1. Recozimento Simulado (Simulated Annealing).....	15
3.2. Algoritmos Genéticos	18
3.2.1. Seleção	20
3.2.2. Cruzamento	22
3.2.3. Mutação	23
3.2.4. Considerações finais	24
 CAPÍTULO 4 - Evolução Diferencial	 27
4.1. Operadores da Evolução Diferencial	28
4.1.1. Mutação	28
4.1.2. Cruzamento	29
4.1.3. Seleção	31
4.2. Estratégias da Evolução Diferencial	32
4.3. Algoritmo da Evolução Diferencial	34
 CAPÍTULO 5 - Simulações Numéricas – Aplicações	 37
5.1. Efeito da variação dos parâmetros da Evolução Diferencial	37
5.2. Funções matemáticas sem restrições	45
5.2.1. Exemplo 1.....	45
5.2.2. Exemplo 2.....	46
5.2.3. Exemplo 3.....	47

5.3. Funções matemáticas com restrições	47
5.3.1. Exemplo 4	48
5.3.2. Exemplo 5	49
5.4. Problema de Engenharia: Problema do Despacho Econômico e Ambiental – DEA..	
.....	50
CAPÍTULO 6 - Evolução Diferencial aplicada a sistemas robóticos	53
6.1. Otimização do espaço de trabalho de manipuladores 3R.....	53
6.1.1. Cinemática de robôs manipuladores utilizando o método de Denavit-Hartenberg	55
6.1.2. Espaço de trabalho de manipuladores 3R	58
6.1.3. Formulação do problema ótimo	62
6.1.4. Simulações numéricas	62
6.2. Otimização de trajetórias do manipulador CaPaMan.....	64
6.2.1. Descrição do CaPaMan.....	64
6.2.2. Formulação da Trajetória.....	66
6.2.3. Posição das juntas esféricas e suas derivadas temporais.....	67
6.2.4. Determinação dos ângulos de Euler.....	68
6.2.5. Determinação das derivadas temporais dos ângulos de Euler	69
6.2.6. Matriz de transformação	70
6.2.7. Coordenadas do ponto central da plataforma móvel	71
6.2.8. Derivadas temporais das coordenadas do ponto central	71
6.2.9. Velocidades e acelerações angulares	72
6.2.10. Dinâmica da plataforma móvel	73
6.2.11. Dinâmica dos paralelogramos articulados	77
6.2.12. O Torque total de entrada	79
6.2.13. Formulação do Problema ótimo.....	80
6.2.14. Simulações Numéricas	82
CAPÍTULO 7 - Conclusões e Trabalhos Futuros	89
REFERÊNCIAS BIBLIOGRÁFICAS.....	91
ANEXO I	97

CAPÍTULO I

INTRODUÇÃO

Otimização é o processo de fazer algo melhor, consiste de variações experimentais sobre um conceito inicial e caminha no sentido de melhorar as informações que o caracterizam. A otimização visa determinar a melhor configuração de projeto de um dado sistema sem ter que testar todas as possibilidades possíveis. Um problema de otimização pode apresentar várias soluções, assim, a definição de “melhor” solução é relativa ao problema considerado, ao método de solução e às tolerâncias adotadas.

Em síntese, pode-se dizer que otimização é o processo de ajuste de características de um dado processo, matemático ou experimental, para se encontrar o valor máximo ou mínimo da função associada ao referido processo, que represente seu desempenho.

O problema geral de otimização consiste em minimizar ou maximizar uma *função objetivo* de n variáveis, chamadas *variáveis de projeto* ou *de decisão*, sujeita, ou não, a *restrições de igualdade*, *desigualdade* e *restrições laterais*. Estas restrições definem o *espaço de busca viável* ou *região de busca* onde se espera encontrar a melhor solução, também denominada *solução ótima*. A função objetivo, também referida como função de adaptação (fitness) ou função custo, e as funções de restrições podem ser funções lineares ou não lineares em relação às variáveis de projeto. Tais funções podem ser explícitas ou implícitas e serem calculadas por técnicas analíticas ou numéricas.

Pode-se aplicar a otimização em várias áreas do conhecimento. Nas últimas décadas, a aplicação de otimização em problemas de engenharia tem crescido consideravelmente. Existem muitos métodos de otimização e cada um deles alcança melhor desempenho dependendo do tipo de problema considerado. A escolha do método depende de uma série de características do problema a ser otimizado, principalmente do comportamento da função que o representa, podendo ser de difícil determinação.

De acordo com as características dos problemas, pode-se classificar as ferramentas de otimização em dois grandes grupos: programação linear e programação não-linear.

Na programação linear, tanto a função objetivo quanto as restrições são lineares, conseqüentemente, trata-se de problemas convexas, que possuem métodos para solução bem conhecidos (BREGALDA, et al., 1988; BERTSIMAS & TSITSIKILIS, 1997).

As técnicas de otimização também podem ser classificadas em métodos determinísticos e estocásticos, estes últimos também conhecidos como métodos naturais, conforme esquematizado na Fig.1.1.

A maioria dos métodos determinísticos são baseados no cálculo de derivadas ou em aproximações destas, necessitando de informações do vetor gradiente, seja procurando o ponto onde ele se anula ou usando a direção para a qual aponta. Estes métodos produzem bons resultados quando as funções são contínuas, convexas e unimodais (MARTÍNEZ & SANTOS, 1995; VANDERPLAATS, 1999; IZMAILOV & SOLODOV, 2005).

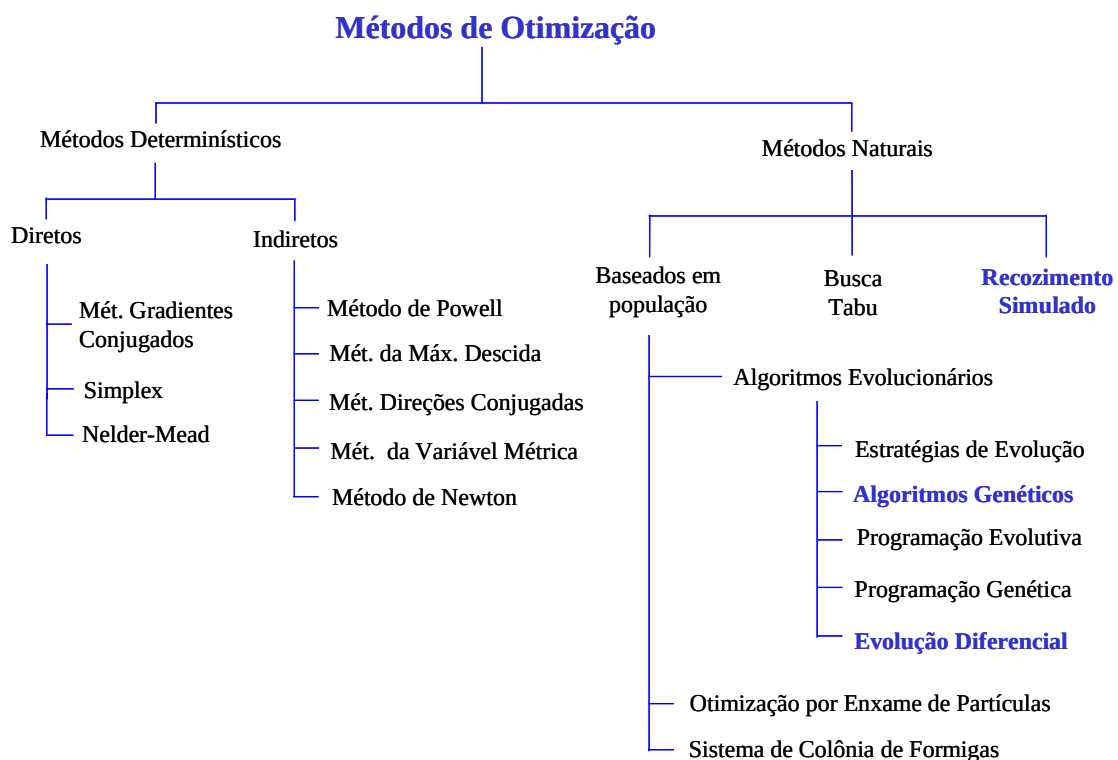


Figura 1.1 – Classificação de alguns métodos de Otimização

Sendo um campo crescente de pesquisa e aplicação, torna-se difícil e perigoso chamar qualquer método de *clássico*. Pode-se entender como métodos clássicos todo algoritmo de otimização e busca que usa apenas uma solução atualizada em cada iteração e que, principalmente, trata-se de métodos determinísticos. Existe uma ampla bibliografia

disponível sobre esta classe de métodos (DEB, 1995; FOX, 1971; HAUG & ARORA, 1989; HIMMELBLAU, 1972; REKLAITIS et al., 1983, VANDERPLAATS, 1999).

A maioria dos algoritmos clássicos ponto a ponto usa um procedimento determinístico para aproximar a solução ótima. Tais algoritmos começam de uma solução inicial. Daí, baseado em regras de transição pré-especificadas, o algoritmo indica uma direção de busca, obtida através de informações locais. A busca unidimensional, isto é, busca em apenas uma dimensão, é então realizada ao longo da direção de busca até encontrar a melhor solução. Esta melhor solução torna-se a nova solução e o procedimento é repetido por um determinado número de vezes. A Fig.1.2 ilustra este procedimento. Os algoritmos, em geral, variam no modo de escolher as direções de busca.

Segundo Deb (1995), os métodos clássicos de otimização podem ser classificados em dois grupos distintos: métodos diretos e métodos indiretos. Nos métodos de busca direta, somente os valores da função objetivo e das restrições são usados para conduzir a estratégia de busca, enquanto que os métodos indiretos usam a primeira e/ou a segunda derivada da função objetivo e/ou restrições para conduzir o processo de busca.

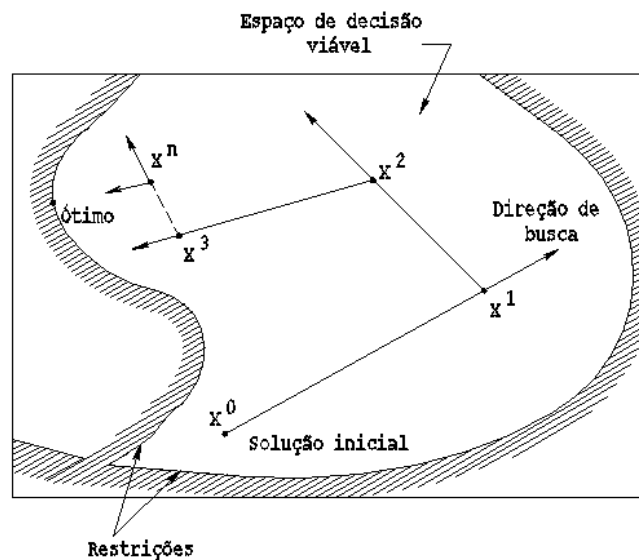


Figura 1.2 - Busca ponto a ponto

Como a informação da derivada não é usada, os métodos de busca direta são freqüentemente lentos, exigindo muitas avaliações da função objetivo para obter a convergência. Pela mesma razão, eles podem também ser aplicados a muitos problemas sem necessidade de modificações no algoritmo. Por outro lado, os métodos baseados no gradiente convergem mais rapidamente para perto da solução ótima, porém são ineficientes quando aplicados a problemas que apresentam não diferenciabilidade ou descontinuidade.

Além disso, há algumas dificuldades comuns entre as técnicas diretas e as indiretas, como segue:

- 1 – A convergência a uma solução ótima depende da escolha da solução inicial;
- 2 – Muitos algoritmos tendem a ficarem ‘presos’ em uma solução sub-ótima (mínimos locais);
- 3 – Um algoritmo eficiente na solução de um problema de otimização pode não ser eficiente na solução de outro;
- 4 – Não são eficientes em tratar problemas onde o espaço de busca é discreto;
- 5 – Não são adequados para serem usados em computação paralela.

Como não-linearidades e interações complexas entre as variáveis de projeto ocorrem freqüentemente em problemas práticos de otimização, os espaços de busca geralmente contêm mais de uma solução ótima. Assim, na solução destes problemas, os métodos clássicos podem ser atraídos por algum destes mínimos locais e não conseguem escapar.

De uma forma geral, cada algoritmo clássico de otimização é indicado para resolver um conjunto específico de problemas. Por exemplo, o método de Programação Geométrica é destinado a resolver preferencialmente problemas do tipo polinomial. Programação geométrica é eficiente na solução de tais problemas, mas não podem ser aplicadas facilmente para resolver outros tipos de funções. O Método da Direção Conjugada ou Método dos Gradientes Conjugados é eficiente para resolver funções objetivo quadráticas, mas possuem dificuldades em trabalhar com problemas multi-objetivos. O método de programação linear sucessiva de Frank-Wolfe trabalha eficientemente com funções lineares e restritas, mas para resolver problemas não lineares seu desempenho depende bastante da escolha das condições iniciais (REKLAITIS et al., 1983). Assim, um algoritmo pode ser mais adequado para um problema e não ser aplicado a outros. Isto requer do usuário conhecimento de um grande número de algoritmos de otimização, a fim de resolver diferentes problemas.

Na maioria dos problemas práticos de otimização, algumas variáveis de projeto podem assumir apenas valores discretos. Esta restrição freqüentemente surge devido a condições comerciais de alguns produtos ou componentes. Por exemplo, se o diâmetro de um componente mecânico é uma variável de projeto e este componente é um produto padronizado, o algoritmo de otimização não pode usar qualquer diâmetro arbitrário.

Uma prática usual para tentar resolver tais problemas é supor que todas as variáveis são contínuas durante o processo de otimização e, em seguida, adotar a medida padronizada mais próxima da solução ótima encontrada. No entanto, há muitas dificuldades para esta aproximação. Como valores inviáveis de uma variável de decisão são permitidos no processo de otimização, o algoritmo gasta muito tempo calculando soluções inviáveis.

Isto faz com que seja despendido um esforço de busca inútil. Além disso, durante os cálculos de pós-otimização, os tamanhos padronizados escolhidos na vizinhança daqueles valores ótimos devem ser verificados para cada variável. Assim, para cada n destas variáveis, um total de $2n$ soluções adicionais precisam ser avaliadas (restrições laterais). Deve-se considerar ainda que, verificar as duas opções para cada variável, pode não garantir a combinação ótima em relação a outras variáveis.

Todas estas dificuldades podem ser eliminadas se apenas valores viáveis, isto é, valores discretos, das variáveis de projeto forem permitidos durante o processo de otimização.

Muitos problemas práticos de otimização requerem o uso de software de simulação envolvendo o Método de Elementos Finitos, simulação computacional da mecânica dos fluidos, solução de equações não lineares, ou outros métodos computacionais complexos para calcular a função objetivo e as restrições. Devido à acessibilidade e disponibilidade de sistemas de computação paralela, tornou-se viável usar tais métodos na otimização de problemas complexos. Como muitos métodos clássicos usam aproximação ponto a ponto, as vantagens dos sistemas paralelos não podem ser totalmente exploradas.

Em engenharia, os problemas geralmente são complexos, não-lineares, de difícil representação e descritos por funções nem sempre diferenciáveis, necessitando de métodos numéricos para sua solução. A abordagem anterior sugere que métodos clássicos enfrentam dificuldades na otimização de alguns problemas deste tipo.

Os métodos estocásticos podem reduzir algumas das dificuldades apresentadas anteriormente, justificando o fato de que cada vez mais estão sendo desenvolvidas pesquisas com estes métodos, visando comparar seus resultados com os métodos clássicos na solução de problemas de otimização.

Os métodos naturais, dos quais os algoritmos evolucionários, ou evolutivos (AEs) fazem parte, se caracterizam pela busca da melhor solução através de regras de probabilidade, trabalhando de maneira “aleatória orientada”. Tais métodos utilizam apenas as informações da função de otimização, não requerendo informações sobre suas derivadas ou possíveis descontinuidades.

Estas técnicas ganharam popularidade com a evolução dos computadores, já que requerem um número elevado de avaliações do problema. Isto é necessário para que se dê chance ao método de explorar devidamente toda a região do espaço de busca em que está contida a solução ótima, resultando em grande número de avaliações da função, necessários para encontrar a solução.

Entre as técnicas mais conhecidas dos métodos naturais destacam-se os algoritmos genéticos (AGs), recozimento simulado (Simulated Annealing), estratégias evolucionárias

(EEs), programação evolutiva (PE), programação genética (PG), que podem ser encontrados na literatura especializada (HAUPT & HAUPT, 1998; DEB, 2001). Estes algoritmos se baseiam em população de indivíduos, onde cada indivíduo representa um ponto de busca no espaço de soluções potenciais de um dado problema e imitam os princípios da natureza para criar procedimentos de otimização. Os Algoritmos Evolutivos possuem alguns procedimentos de seleção baseados na aptidão dos indivíduos, e em operadores de cruzamento e mutação.

O objetivo desta pesquisa é estudar o método de otimização aleatório denominado Evolução Diferencial (ED), desenvolvido por Storn e Price (1995), bem como analisar seu desempenho em problemas de otimização considerados de difíceis soluções. A idéia principal da Evolução Diferencial é gerar novos indivíduos, denotados vetores doadores, pela adição da diferença vetorial ponderada entre dois indivíduos aleatórios da população a um terceiro indivíduo. Será mostrado que este princípio de usar diferenças de vetores para perturbar a população (indivíduos) resulta em um método de rápida convergência, fácil implementação e robusto. Serão estudadas algumas estratégias da evolução diferencial que são formuladas alterando-se a forma de obtenção do operador mutação.

Este método tem sido aplicado com sucesso em vários campos da ciência, podendo ser citado, entre muitos outros: solução de projeto de sistemas (STORN, 1999), solução de sistemas lineares (CHENG & HWANG, 2001), transferência de calor (BABU & MUNAWAR, 2001), projeto de robô manipulador (BERGAMASCHI et al., 2005). Nos últimos anos a Evolução Diferencial têm sido bastante utilizada na otimização de processos químicos (Babu et al., 2004; Babu et al., 2005; Babu et al., 2006), cujos problemas envolvem a otimização de funções multi-objetivos, não-lineares, bastante complexas. Assim, a Evolução Diferencial está sendo identificada como uma fonte potencial para otimização rápida e precisa.

Além disso, serão descritos os métodos naturais: Algoritmos Genéticos e Recozimento Simulado, com o objetivo de comparar os resultados do método estudado.

Esta dissertação está dividida em 7 capítulos. O capítulo 2 apresenta uma formulação geral do problema ótimo, bem como conceitos e definições importantes no estudo de otimização. Uma revisão dos métodos naturais Recozimento Simulado e Algoritmos Genéticos é apresentada no capítulo 3.

O capítulo 4 traz uma fundamentação teórica do algoritmo de Evolução Diferencial. No capítulo 5 é realizada uma análise estatística de influência dos parâmetros da Evolução Diferencial. Além disso, o algoritmo é aplicado para otimizar funções matemáticas, com e sem restrições, e na solução do seguinte problema de engenharia: despacho econômico e ambiental de geração de energia elétrica, cujo objetivo é selecionar as potências geradoras

ótimas, assim como encontrar a demanda de carga com custo mínimo, e emissão mínima de poluentes na atmosfera.

Já no capítulo 6, são apresentados dois problemas de otimização com aplicações em robótica industrial: a maximização do volume do espaço de trabalho de manipuladores 3R e a otimização de trajetórias de um manipulador com estrutura paralela.

No capítulo 7 tem-se as considerações finais e sugestões para trabalhos futuros.

CAPÍTULO II

FORMULAÇÃO DO PROBLEMA GERAL DE OTIMIZAÇÃO

Os métodos de busca e otimização consideram vários conceitos importantes que são descritos a seguir. Sejam dados um conjunto $\Omega \subset \mathbb{R}^n$ e uma função $f : \Omega \rightarrow \mathbb{R}$. Um problema de otimização visa determinar um minimizador de f no conjunto Ω , sendo escrito como:

$$\min f(X) \text{ onde } X \in \Omega \quad (2.1)$$

O conjunto Ω é chamado *conjunto viável* e define a região onde estão as possíveis soluções do problema. Também recebe o nome de *região viável*, *espaço de busca*, ou ainda *espaço de projeto*. Para limitar um certo espaço, utiliza-se as funções de restrição. Os pontos de Ω são chamados *pontos viáveis*.

A função $f(X)$ é a função que atinge um valor extremo, ou seja, permite avaliar o grau de otimalidade de cada elemento do espaço de busca. É chamada *função objetivo* ou *função custo*, ou ainda *função de adaptação (fitness)*. A função objetivo pode ser unidimensional, ou seja, possui apenas uma variável de projeto, ou multidimensional, isto é, possui mais de uma variável de projeto.

Variável de projeto ou *de decisão* são variáveis que se modificam durante o processo de otimização, ou seja, são quantidades que alteram o valor da função objetivo, sendo representadas por x_i , para $i = 1, \dots, n$.

Um ponto $X^* \in \Omega$ é minimizador global de (2.1) se $f(X^*) \leq f(X)$, para todo $X \in \Omega$ e é minimizador local de (2.1) se existe uma vizinhança U de X^* tal que $f(X^*) \leq f(X)$, para todo $X \in \Omega \cap U$, como pode ser observado na Fig. 2.1.

Restrições são funções que delimitam o espaço de busca, através de igualdades ou desigualdades matemáticas, que estabelecem exigências para a solução ótima.

Restrições laterais são funções que servem para limitar os valores das variáveis de projeto, sendo definidas como: limite inferior $x^{inf} = (x_1^{inf}, x_2^{inf}, \dots, x_n^{inf})^T$ e limite superior $x^{sup} = (x_1^{sup}, x_2^{sup}, \dots, x_n^{sup})^T$, onde T representa a transposição do vetor.

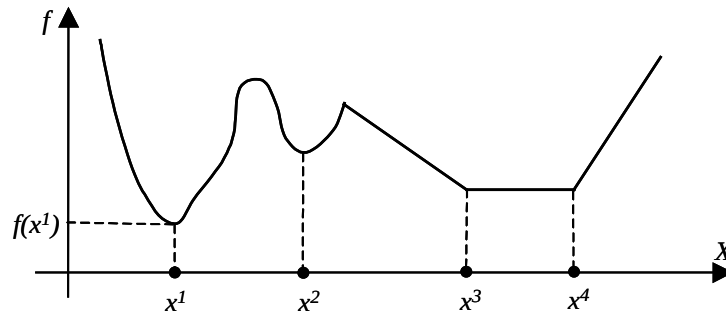


Figura 2.1 - x^1 é o minimizador global, $f(x^1)$ é o valor ótimo, x^2 é um minimizador local estrito, $[x^3, x^4]$ é um conjunto de minimizadores locais não estritos

Restrições ativas são as restrições de desigualdade que são satisfeitas por X^* na igualdade, ou seja, X^* pertencente ao conjunto viável tal que $g_j(X^*)=0$, $j = 1, \dots, J$ (MARTINEZ & SANTOS, 1995), caso contrário, a restrição é dita *inativa*.

A Figura 2.2 ilustra um exemplo onde $j = 1$ e $j = 2$ as restrições são ativas e para $j = 3$ a restrição é inativa.

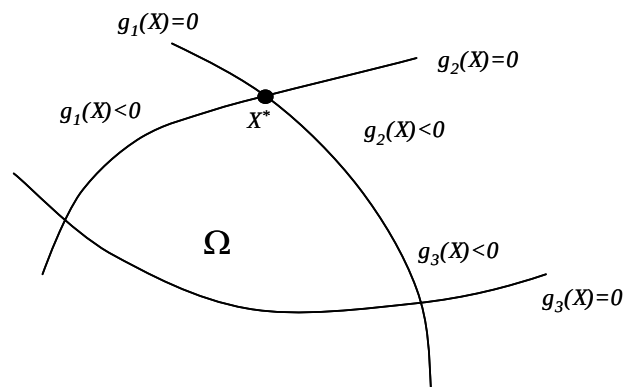


Figura 2.2 - A restrição g_3 é inativa no ponto ótimo pois ela não define a estrutura de Ω em torno de X^*

Ponto ótimo é o vetor das variáveis de projeto que satisfaz as restrições, além de corresponder a um extremo da função objetivo. Normalmente é denotado por X^* .

Valor ótimo é o valor da função objetivo no ponto ótimo.

Solução ótima é o par formado pelo ponto ótimo e pelo valor ótimo, $(X^*, f(X^*))$, podendo ser local ou global.

O problema geral de otimização consiste em minimizar ou maximizar uma função objetivo, sujeita ou não a restrições de igualdade, desigualdade e restrições laterais.

A função objetivo e as funções de restrições podem ser funções lineares ou não lineares em relação às variáveis de projeto, implícitas ou explícitas, descontínuas, não-diferenciáveis, não convexas, calculadas por técnicas analíticas ou numéricas.

Um problema de otimização pode ser formulado como o de encontrar um vetor de n variáveis de projeto $X = (x_1, x_2, \dots, x_n)^T$ que otimize uma função objetivo $f(X)$, e satisfaça as restrições de igualdade, desigualdade e laterais. O problema geral pode ser escrito conforme as Eqs.(2.2) e (2.3).

$$\text{Minimizar } f(X), \quad X \in \mathfrak{R}^n \quad (2.2)$$

$$\text{Sujeito a } \begin{cases} g_j(X) \leq 0, & j=1, \dots, J \\ h_l(X) = 0, & l=1, \dots, L \\ x_i^{\inf} \leq x_i \leq x_i^{\sup}, & i=1, \dots, n \end{cases} \quad (2.3)$$

Onde $g_j(X)$ representa as restrições de desigualdade e $h_l(X)$ as restrições de igualdade, se houver.

Caso o problema seja maximizar a função objetivo, basta transformar o problema multiplicando a função por -1 , i.e., minimiza-se a função $-f(X)$, pois as soluções locais e globais de ambos os problemas são as mesmas, com sinais opostos para os valores ótimos, conforme a Fig. 2.3. Por isso, do ponto de vista matemático, não existe nenhuma diferença relevante entre os problemas de minimização e de maximização: todos os resultados obtidos para uma classe de problemas podem ser estendidos para a outra classe sem dificuldade (IZMAILOV & SOLODOV, 2005).

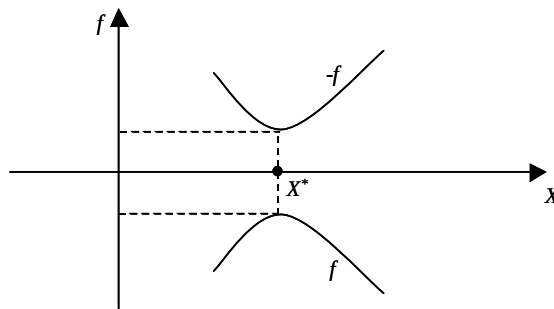


Figura 2.3 - $\max f(X) = \min -f(X)$

Um problema de otimização que possui mais de uma função objetivo é denominado problema *multi-objetivo* ou *multi-critério*. Em muitas situações, várias funções objetivo precisam ser minimizadas ou maximizadas simultaneamente. Isto ocorre devido à complexidade dos problemas reais que, na maioria das vezes, apresentam objetivos conflitantes entre si, ou seja, a melhoria de um objetivo provoca a piora de outro (ESCHENAUER et al., 1990). Neste caso, as soluções do problema são chamadas de *soluções ótimas de Pareto* (HAJELA & LIN, 1992). O problema pode ser escrito matematicamente conforme as Eqs.(2.2) e (2.3), sendo que $f(X)$ é dada como um vetor cujas componentes são k funções objetivo. Esta função vetorial pode ser escrita como:

$$f(X) = [f_1(X), f_2(X), \dots, f_k(X)] \quad (2.4)$$

Existem várias técnicas para resolver este tipo de problema (OLIVEIRA, 2005a). A técnica aqui utilizada é o método dos objetivos ponderados. Neste método, transforma-se o vetor de funções objetivo em uma função escalar. Assim, um problema de otimização multi-objetivo é transformado em um problema de otimização escalar (OSYCZKA, 1990), criando uma função da forma:

$$f(X) = \sum_{k=1}^K w_k \frac{f_k(X)}{f_k^0} \quad (2.5)$$

onde, os coeficientes de ponderação w_k são usualmente assumidos como:

$$\sum_{k=1}^K w_k = 1 \quad (2.6)$$

Seria desejável que os coeficientes de ponderação w_k representassem a importância relativa de cada critério. Para que w_k possa refletir de forma aproximada a importância dos objetivos, todas as funções devem ser expressas em unidades que aproximem os mesmos valores numéricos (DEB, 2001). Assim, na Eq.(2.5), a função vetorial foi normalizada pelo uso da solução ideal f_k^0 . Esta solução é determinada obtendo-se separadamente o mínimo viável, para todas as funções objetivo. Em outras palavras:

$$\begin{cases} f_k^0 = \min f_k(X), & k=1, \dots, K \\ \text{sujeito as restrições (2.3)} \end{cases} \quad (2.7)$$

Muitos algoritmos evolutivos de otimização são desenvolvidos para resolver problemas irrestritos. Assim, no caso de problemas com restrições, torna-se necessário introduzir modificações no método. Neste estudo, é utilizado o conceito de *Função de Penalidade* (MARTINEZ & SANTOS, 1995; VANDERPLAATS, 1999). Nesta técnica, os problemas com restrições são transformados em problemas sem restrições adicionando uma função de penalidade $P(X)$ à função objetivo original para limitar as violações das restrições. Esta nova função objetivo, chamada *pseudo objetivo*, é penalizada toda vez que encontrar uma restrição ativa. Seja a função pseudo objetivo, ϕ , dada conforme a Eq.(2.8).

$$\phi(X) = f(X) + k_p P(X) \quad (2.8)$$

$$P(X) = \left[\sum_{j=1}^J \left\{ \max[0, g_j(X)] \right\}^2 + \sum_{l=1}^L [h_l(X)]^2 \right] \quad (2.9)$$

onde $f(X)$ é a função objetivo original dada na Eq.(2.2) ou (2.5), $P(X)$ é uma função de penalidade imposta dada pela Eq.(2.9), $g_j(X)$ e $h_l(X)$ são funções de restrições de desigualdade e igualdade, respectivamente, conforme a Eq.(2.3). O escalar k_p é um multiplicador que quantifica a magnitude da penalidade. Para a eficiência do método, deve-se procurar um valor adequado para o fator de penalidade de forma a garantir que todas as restrições sejam obedecidas.

CAPÍTULO III

REVISÃO DE ALGUNS MÉTODOS NATURAIS

Os métodos naturais são procedimentos iterativos que tentam simular os processos usados na natureza para resolver problemas difíceis. Estes algoritmos também são referidos na literatura como algoritmos estocásticos, entre os mais conhecidos pode-se citar Simulated Annealing, Busca Tabu e um grupo de métodos baseados em população. Neste último grupo destacam-se os Algoritmos Evolutivos ou Evolucionários (Algoritmos Genéticos, Estratégias de Evolução, Evolução Diferencial, etc.) e os algoritmos baseados na inteligência coletiva (otimização por enxame de partículas, colônia de formigas, etc.). Diversas técnicas vêm sendo desenvolvidas e aprimoradas, dentre elas, neste capítulo serão considerados o Recozimento Simulado (Simulated Annealing) e os Algoritmos Genéticos (AGs). A seguir tem-se uma revisão destas técnicas, uma vez que serão utilizadas para comparar os resultados obtidos com a Evolução Diferencial objeto de estudo deste trabalho.

3.1. Recozimento Simulado (Simulated Annealing)

O algoritmo Recozimento Simulado surgiu em 1983, tendo como precursor Kirkpatrick et al. (1983) que se baseou nas idéias de Metropolis et al. (1953). Consiste em uma útil conexão entre a mecânica (comportamento de um sistema em equilíbrio térmico a uma temperatura finita) e a otimização combinatória (encontrar o mínimo de uma dada função dependendo de vários parâmetros). Alguns resultados publicados utilizando esta técnica, em particular devido aos esquemas de resfriamento rápido (MASTERS, 1993), têm merecido a atenção de físicos e engenheiros.

Este método de otimização faz uma analogia com o processo de recozimento (annealing) da metalurgia. Sabe-se da Metalurgia que, se o metal é resfriado em condições apropriadas, o cristal simples pode ser obtido (KIRKPATRICK et al., 1983). No recozimento, o metal é aquecido a altas temperaturas, causando um choque violento nos átomos. Se o

metal for resfriado de forma brusca, a microestrutura tende a um estado randomicamente instável, porém, se o metal é resfriado de forma suficientemente lenta, o sistema procurará um ponto de equilíbrio caracterizado por uma microestrutura ordenada e estável.

Metropolis et al. (1953) introduziram um método numérico simples que representa o estado de equilíbrio de um conjunto de átomos a uma dada temperatura.

Cada configuração é definida por um conjunto de átomos de energia E e temperatura T . A cada passo do algoritmo, os átomos sofrem um pequeno deslocamento aleatório provocando uma pequena alteração de energia ΔE no sistema. Se esta nova energia ΔE é menor ou igual a zero ($\Delta E \leq 0$), o deslocamento é aceito e esta configuração é usada como ponto de partida do próximo passo. Caso contrário, ($\Delta E > 0$), o deslocamento ainda pode ser aceito dependendo da probabilidade da configuração dada por:

$$P(\Delta E) = e^{\left(-\Delta E / K_b T\right)} \quad (3.1)$$

onde K_b é a constante de Boltzmann.

Um número aleatório $rand$, uniformemente distribuído, deve ser gerado no intervalo $[0, 1]$. Se $rand \leq P(\Delta E)$ a nova configuração é aceita. Se $rand > P(\Delta E)$ a configuração anterior é utilizada para iniciar um novo passo.

A escolha da função de probabilidade $P(\Delta E)$, conforme descrito acima, deve-se ao fato de que o sistema evolui segundo uma distribuição de Boltzmann, mas outras funções de distribuição podem ser testadas.

Os parâmetros do algoritmo são: a função custo, que representa a energia do sistema; as variáveis de projeto, que descrevem sua configuração e a temperatura, que é um parâmetro de controle (CORONA et al., 1987).

Se T tiver magnitude muito superior ao desvio padrão da função no intervalo, quase todos os pontos são aceitos. Ao passo que se T for igual a zero, o método se torna uma busca aleatória do mínimo. Assim, adota-se: T_i como o valor do desvio padrão da função objetivo no intervalo estudado e T_f com a ordem de grandeza desejada para a precisão do ponto ótimo (SARAMAGO & FARIA, 2001).

Na otimização via Recozimento Simulado considera-se a perturbação aleatória das variáveis de projeto e a manutenção do melhor valor da função objetivo. As variáveis de projeto são perturbadas aleatoriamente e armazena-se o melhor valor da função objetivo a cada perturbação. A temperatura é então reduzida e novas tentativas executadas. Este

procedimento continua até escapar de um mínimo local. Ao final do processo pretende-se obter um mínimo global.

Após algumas tentativas o melhor valor da função é chamado de *centro* (*center*), em torno do qual ocorrerão as perturbações na próxima temperatura.

O algoritmo do método, desenvolvido por Saramago et al. (1999), pode ser acompanhado através da Fig. 3.1.

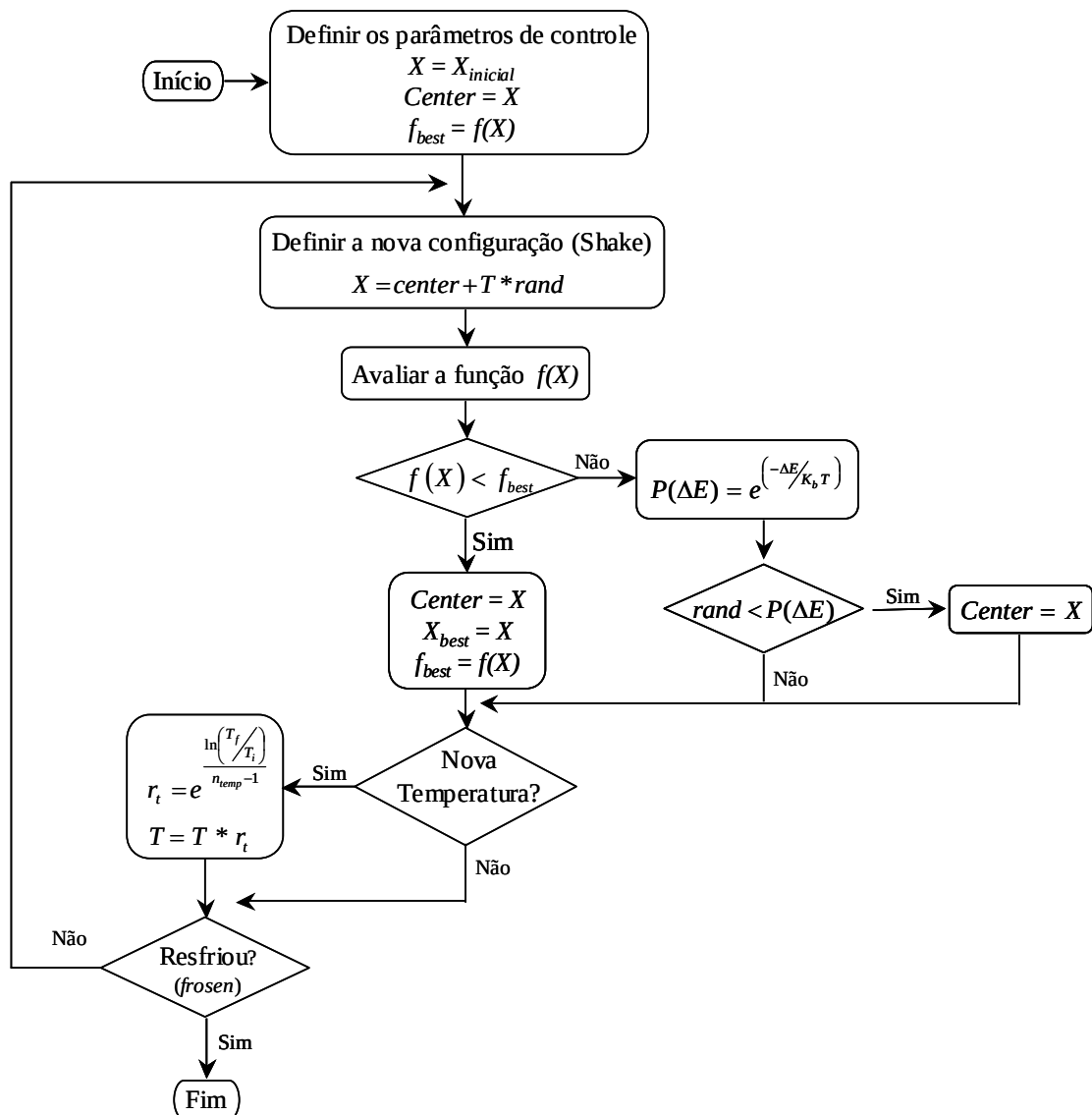


Figura 3.1 - Fluxograma do Recozimento Simulado

Os parâmetros de controle para iniciar o procedimento são: a função objetivo $f(X)$; as variáveis de projeto iniciais, X ; o número de variáveis de projeto, n ; a temperatura inicial T_i ; a temperatura final T_f ; o número de iterações para cada temperatura, n_{iters} ; o número de temperaturas, n_{temp} e o critério de parada.

A configuração inicial das variáveis de projeto é adotada como centro. O valor inicial da função objetivo é adotado como o melhor valor.

No próximo passo, o número aleatório *rand* é gerado e as variáveis são modificadas (“shake”). Na Eq.(3.2) observa-se que quatro valores randômicos são gerados, de forma que a variável *rand* seja adotada como média zero (MASTERS, 1993). Assim, uma nova configuração é obtida pela Eq.(3.3) e um novo valor da função objetivo pode ser calculado.

$$rand = rand_1 + rand_2 - rand_3 - rand_4 \quad (3.2)$$

$$X = center + T * rand \quad (3.3)$$

O esquema inicia-se com uma temperatura alta, que é reduzida discretamente, usando o fator r_t , $0 < r_t < 1$, conforme Eqs.(3.4) e (3.5):

$$r_t = e^{\frac{\ln\left(\frac{T_f}{T_i}\right)}{n_{temp}-1}} \quad (3.4)$$

$$T = T * r_t \quad (3.5)$$

Após a temperatura ser reduzida, novas tentativas são executadas. Este procedimento é repetido até escapar de um mínimo local. Ao final do processo pretende-se obter um mínimo global.

3.2. Algoritmos Genéticos (AGs)

São métodos de busca semi-aleatória baseada na teoria de evolução de Charles Darwin. Esses métodos operam com conjuntos de candidatos, chamados de *população*, que são constantemente modificados utilizando dois princípios básicos da evolução natural das espécies: *seleção* e *variação*. Estes princípios tentam representar a competição onde os indivíduos mais aptos reproduzem e repassam seu material genético às gerações futuras, e os indivíduos menos aptos tendem a desaparecer da população (DEB, 2001).

O método foi desenvolvido por John Holland (1975) na Universidade de Michigan e popularizado por um de seus alunos, David Goldberg (1989), que apresentou a solução de problemas complexos de engenharia usando os algoritmos genéticos.

Fazendo uma analogia com o processo da evolução natural, as soluções candidatas são denominadas *indivíduos*, também referidos na literatura como *cromossomos*, *cadeia de valores de parâmetros* ou *cadeias binárias* (HAUPT & HAUPT, 1998; DEB, 2001).

Resumidamente, o procedimento é o seguinte: define-se os parâmetros de otimização e a função objetivo, em seguida faz-se uma codificação dos parâmetros, procede-se uma sequência repetitiva de procedimentos que consistem em avaliar, selecionar, recombinar e modificar os indivíduos, gerando, assim, novas populações. Esses procedimentos são efetuados com base na *aptidão* de cada solução, em que indivíduos com melhores aptidões terão uma maior probabilidade de serem selecionados para o processo de cruzamento, passando assim parte dos seus genes a seus descendentes. Cada iteração desse processo recebe o nome de *geração*. A finalização é realizada através de testes de convergência, sendo que um número máximo de gerações deve sempre ser estabelecido. O melhor indivíduo será adotado como solução do problema.

A codificação dos parâmetros da função objetivo pode ser expressa por *bits* (binária) ou por *parâmetros contínuos* (*números reais* ou *pontos flutuantes*). Neste capítulo, é apresentado o algoritmo genético binário, a fim de facilitar a compreensão dos algoritmos genéticos clássicos, sendo citados alguns aspectos que os diferem dos algoritmos com parâmetros contínuos. Na codificação real, a representação de um cromossomo d com n parâmetros, dados por $p_1, p_2, \dots, p_n$, é escrito como um vetor com $1 \times n$ elementos, tal que:

$$\text{cromossomo}_d = [p_1, p_2, \dots, p_n] \quad (3.6)$$

Na Eq.(3.6) cada parâmetro é representado por um número real. No algoritmo genético binário, a codificação de um cromossomo d que possui n parâmetros com, por exemplo, 5 bits cada é apresentada na Fig. 3.2.

$$\text{cromossomo}_d = \left[\begin{array}{ccccccc} \overset{x_1}{\underbrace{11010}_{\text{gene}_1}} & \overset{x_2}{\underbrace{01100}_{\text{gene}_2}} & \dots & \overset{x_i}{\underbrace{01010}_{\text{gene}_i}} & \dots & \overset{x_n}{\underbrace{00101}_{\text{gene}_n}} \end{array} \right]$$

Figura 3.2 - Representação de um cromossomo com n genes de 5 bits cada

Cada *gene* representa uma variável da função objetivo. Os bits *zeros* e *uns* da cadeia binária correspondem aos *alelos* da genética. O número total de bits de um cromossomo é dado por $\sum m_i, i=1, \dots, n$, onde m_i é o comprimento do gene i . Este comprimento depende da

precisão requerida para o problema e da amplitude do intervalo definido pelas restrições laterais, podendo ser obtido, usando a Eq.(3.7).

$$m_i = \log_2 \left(\frac{x_i^{\sup} - x_i^{\inf}}{\text{precisão}} \right) \quad (3.7)$$

A decodificação pode ser feita conforme a Eq.(3.8) generalizada. Os tamanhos dos genes podem ser diferentes, pois dependem das restrições laterais.

$$x_i = x_i^{\inf} + \text{decimal}(\text{gene}_i)_2 \times \frac{x_i^{\sup} - x_i^{\inf}}{2^{m_i} - 1}, i = 1, \dots, n \quad (3.8)$$

onde $\text{decimal}(\text{gene}_i)_2 = \sum_{p_i=0}^{m_i-1} \text{bit}_{p_i} \times 2^{p_i}$ é o valor decimal da variável, sendo bit_{p_i} o bit correspondente à posição p_i do gene i .

Tanto a evolução biológica quanto os algoritmos genéticos começam com uma população inicial constituída por Np indivíduos gerados aleatoriamente. Com a população inicial definida, calcula-se a função objetivo, que é geralmente referida na literatura, como *função de avaliação (fitness)* ou *função custo*. Esta função associa um valor numérico, conhecido como grau de adaptação, a cada indivíduo da população. Quando possível, esta função deve ser de cálculo rápido, uma vez que ela deve ser avaliada para todos os indivíduos da população, o que tende a elevar o esforço computacional.

Os procedimentos básicos que transformam a população ao longo das gerações em busca da solução do problema de otimização caracterizam os operadores genéticos: *seleção*, *cruzamento* e *mutação*. Na prática, tem-se verificado que a consistência de um AG é garantida por estes três operadores (HAUPT & HAUPT, 1998). Uma descrição destes operadores é dada a seguir.

3.2.1 Seleção (Reprodução)

Este operador determina quais indivíduos serão escolhidos para o cruzamento. O grau de adaptação de cada cromossomo é caracterizado de acordo com o valor da função custo. Em um problema de maximização, por exemplo, quanto maior este valor, maior a probabilidade de contribuir à geração seguinte. Existem vários mecanismos para executar o operador seleção (DeJONG, 1975; GOLDBERG, 1989).

Na seleção elitista os melhores indivíduos de uma população intermediária são escolhidos. Para obter a população intermediária escolhe-se a metade da população de uma geração que corresponde aos indivíduos mais aptos, sendo a outra metade eliminada. Desse modo o cruzamento é realizado com os melhores indivíduos e espera-se que a convergência do algoritmo se torne mais rápida.

Na seleção aleatória, como indica o próprio nome, os indivíduos de uma população intermediária são selecionados aleatoriamente para o posterior cruzamento.

Na seleção por torneio dois indivíduos são escolhidos aleatoriamente, a seguir gera-se um número aleatório $rand \in [0, 1]$. Se $rand$ for menor que um parâmetro previamente definido, que determina o quanto este operador será elitista, o melhor indivíduo é escolhido, senão o outro sobreviverá.

Outra forma de fazer a seleção é através do método da Roleta, no qual a probabilidade de seleção de um indivíduo é diretamente proporcional ao valor da função custo. Por exemplo, sejam quatro cromossomos com suas respectivas aptidões dados conforme a Tab. 3.1. A probabilidade de seleção é a razão entre a aptidão de cada cromossomo e o somatório da aptidão de todos os outros conforme a Eq.(3.9):

$$P_d = \frac{P_d}{\sum_{j=1}^{Np} P_j}, \quad d = 1, \dots, Np \quad (3.9)$$

Tabela 3.1 - Exemplo do método de seleção Roleta

nº	Cromossomo	Aptidão (p_d)	P_d	Pa_d	Graus
1	1100101010	10,0	0,500	0,500	180
2	1001010101	5,0	0,250	0,750	90
3	1110100101	2,5	0,125	0,875	45
4	0010101001	2,5	0,125	1,000	45
Σ		20,0	1,00	1,000	360

Na Tabela 3.1, Pa_d são as probabilidades acumulativas dos cromossomos, sendo dada por:

$$Pa_d = \sum_{z=1}^d P_z, \quad d = 1, \dots, Np \quad (3.10)$$

Para a representação gráfica na roleta, conforme Fig. 3.3, multiplica-se esta razão por 360 graus.

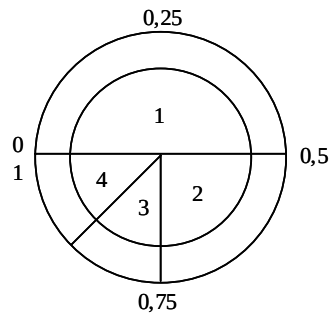


Figura 3.3 - Exemplo da seleção por roleta

Quanto maior a participação do cromossomo na área da roleta, maior a probabilidade dele continuar na população.

A seguir, deve-se selecionar as cadeias que irão contribuir para a geração seguinte. Gera-se um conjunto de números aleatórios $rand \in [0, 1]$ em quantidade igual ao número de indivíduos da população. Por exemplo, se $rand < Pa_1$, seleciona-se o primeiro cromossomo, senão, passa-se para o indivíduo subsequente e faz-se a análise novamente.

No processo de seleção, observa-se que alguns cromossomos podem ser selecionados mais de uma vez, ou seja, os melhores serão copiados mais vezes, enquanto que os piores não sobreviverão.

3.2.2 Cruzamento

Nos sistemas biológicos o cruzamento pode ocorrer durante a reprodução sexuada permitindo a troca de material genético entre dois indivíduos. O algoritmo genético utiliza os cromossomos determinados no processo de seleção para gerarem novos descendentes. Este operador é a primeira forma do algoritmo explorar o espaço de busca e evitar uma convergência prematura, ou seja, evitar ótimos locais (HAUPT & HAUPT, 1998; BRAGA, 1998).

A forma mais simples de cruzamento envolve dois pais que irão produzir dois descendentes. A quantidade de cromossomos da população a ser submetida ao cruzamento é definida através da *probabilidade de cruzamento* P_c , fornecida pelo usuário. É bastante usual adotar para esta probabilidade $P_c = 60\%$.

O processo de escolha dos indivíduos que serão cruzados deve ser feito em pares, gerando números aleatórios $rand_d \in [0, 1]$, $d = 1, \dots, N_p$. Por exemplo, se $rand_1$ for menor que a probabilidade P_c , então o primeiro cromossomo da população será selecionado. Desta forma, um cromossomo d é selecionado para o cruzamento se $rand_d < P_c$. A posição de cruzamento é selecionada entre o primeiro e o último bit dos cromossomos paternos. Uma maneira de selecionar esta posição é descrita a seguir:

Seja p_c a posição de cruzamento na cadeia binária de cada cromossomo, dada conforme a Eq.(3.11).

$$p_c = 1 + rand [(\sum m_i - 1) - 1] \quad (3.11)$$

onde $rand$ é um número aleatório no intervalo $[0, 1]$ e $\sum m_i, i = 1, \dots, n$, é a quantidade de bits do cromossomo, que também representa a posição do último bit no cromossomo.

Cada cadeia é quebrada na posição p_c e todas as informações de um cromossomo, compreendidas entre as posições p_c+1 e $\sum m_i$, são copiadas para o outro cromossomo e vice-versa, como ilustrado na Fig. 3.4.



Figura 3.4 - Representação do operador cruzamento simples entre dois indivíduos

No caso contínuo, o modo mais simples é escolher um ou mais pontos no cromossomo e indicá-los como posições de cruzamento. As posições onde deve ocorrer o cruzamento são selecionadas aleatoriamente e conforme a Fig. 3.5, percebe-se nos descendentes uma combinação dos parâmetros de ambos os pais.

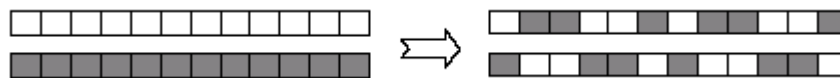


Figura 3.5 - Representação do operador cruzamento uniforme com parâmetros contínuos

Existem outros tipos de cruzamento que podem ser vistos na literatura (HAUPT & HAUPT, 1998; MICHALEWICZ, 1996; ESHELMAN & SHAFFER, 1993).

3.2.3 Mutação

É uma modificação aleatória do material genético dos indivíduos, ou seja, é a alteração de pequenas percentagens nos bits dos cromossomos. Sendo uma outra forma do algoritmo genético explorar a região de busca. Este operador pode introduzir características que não dependem da população original aumentando a diversidade da população, podendo evitar que o algoritmo convirja prematuramente.

Uma estratégia usual é a mutação de *ponto único*, que modifica o bit “1” para “0” e vice versa em determinadas posições do cromossomo, conforme Fig. 3.6. Os pontos de mutação são selecionados aleatoriamente em uma matriz $Np \times \sum m_i, i=1, \dots, n$, que representa o número

total de bits da população. Aumentar o número de mutações aumenta a liberdade do algoritmo em buscar soluções fora do espaço de busca, para evitar isso, usualmente faz-se mutação de 1% a 5% do total de bits por iteração. Esta mutação não se aplica aos melhores indivíduos, pois são considerados de *elite*.

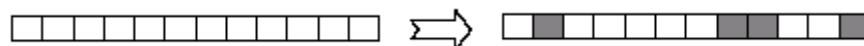


Figura 3.6 - Representação do operador mutação

Uma forma de realizar a mutação é gerar pares aleatórios (A , B), onde A representa o cromossomo a sofrer a mutação e B representa a posição do bit a ser mudado. Outra forma é selecionar aleatoriamente a posição em um cromossomo, obedecendo a uma *probabilidade de mutação* P_m (também conhecida como *taxa de mutação*), e mudar o valor do bit. Neste caso, é necessário gerar números aleatórios *rand* no intervalo $[0, 1]$ em mesma quantidade de bits total da população, representando a posição seqüencial de cada bit na matriz populacional. Para os casos onde *rand* for menor que a probabilidade P_m serão feitas as mutações nos bits correspondentes. Geralmente, recomenda-se usar P_m igual a 1%. (HAUPT & HAUPT, 1998).

Na representação real, a probabilidade de mutação adequada está entre 1% e 20%. O parâmetro que sofre mutação é substituído por um novo parâmetro gerado aleatoriamente.

Depois de realizadas as mutações, os custos associados aos descendentes e aos cromossomos que sofreram mutações são calculados, compondo a próxima geração.

3.2.4 Considerações finais

Como o algoritmo é iterativo, o procedimento acima é repetido até obter uma solução desejável. Usualmente é usado como critério de parada o número máximo de gerações, porém existem outros critérios tais como: estagnação da população (os cromossomos e seus respectivos custos se repetem); existência de um indivíduo com qualidades satisfatórias, entre outros. A Fig. 3.7 mostra um fluxograma básico de um algoritmo genético binário.

O melhor indivíduo, ou seja, aquele que possuir a melhor aptidão será adotado como solução do problema de otimização.

Os algoritmos genéticos apresentam algumas vantagens quando comparados aos métodos de otimização determinísticos, podendo citar: facilidade em sua implementação; eficiência ao evitarem ótimos locais; não requerem o cálculo de derivadas, o que favorece a otimização de funções complicadas; são robustos e apresentam desempenho satisfatório na

resolução de vários tipos de problemas, entre outras. Como desvantagem do método, tem-se o elevado custo computacional, e além disto, nem sempre a solução ótima global é precisa (HAUPT & HAUPT, 1998).

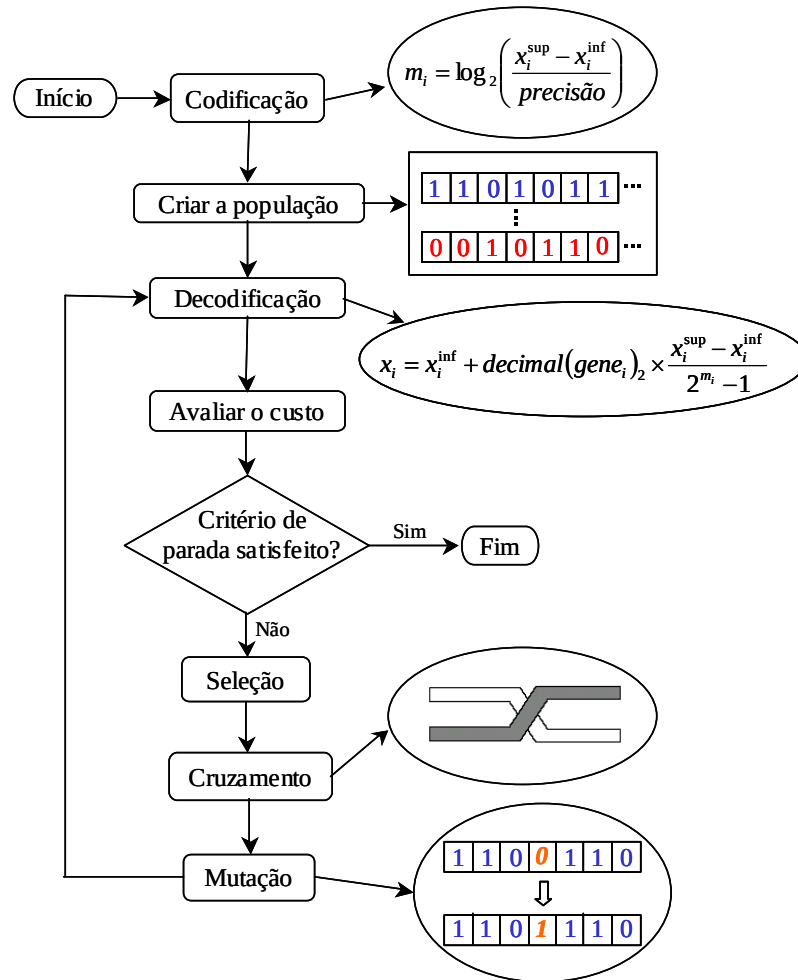


Figura 3.7 - Fluxograma básico de um algoritmo genético binário

CAPÍTULO IV

EVOLUÇÃO DIFERENCIAL (ED)

Dentre os algoritmos evolucionários, destaca-se o algoritmo de Evolução Diferencial (Differential Evolution), desenvolvido por Storn e Price em 1995, visando a busca por melhores resultados com uma abordagem um pouco diferente da utilizada nos algoritmos genéticos e nas estratégias de evolução. Trata-se de um método de busca direta estocástica que surgiu de tentativas de resolver o problema de ajuste polinomial de Chebychev. Kenneth Price introduziu a idéia de usar diferenças de vetores para perturbar a população de vetores (indivíduos) resultando em um método que requer poucas variáveis de controle, é de rápida convergência, fácil de usar e robusto (STORN, 2006).

A escolha do algoritmo de Evolução Diferencial para otimização numérica, conforme Cheng e Hwang (2001), está baseada nas seguintes características:

- É um algoritmo de busca estocástica, originado dos mecanismos de seleção natural;
- Dificilmente torna-se preso em ótimos locais, pois busca a solução ótima global manipulando uma população de soluções, ou seja, buscando simultaneamente a solução em diferentes regiões do espaço de busca;
- É muito eficaz para resolver problemas de otimização com função objetivo descontínua, pois não requer informação sobre suas derivadas;
- Permite que os parâmetros de entrada e saída sejam manipulados como números ordinários reais (pontos flutuantes) sem processamento extra, e, portanto, utiliza eficientemente os recursos do computador;
- Trabalha bem como otimizador local porque os diferenciais gerados por uma população convergente eventualmente tornam-se infinitesimais;
- É eficaz trabalhando com uma população pequena.

Seja uma população inicial escolhida aleatoriamente composta por N_p indivíduos, chamados vetores, cobrindo todo o espaço de busca. Para um problema com n variáveis de projeto, cada vetor possui n componentes. Geralmente, esta população é criada por uma distribuição de probabilidade uniforme, quando não há nenhum conhecimento sobre o

problema. Assim, a população segue uma evolução natural, porém o número de indivíduos permanece fixo durante o processo de minimização.

A idéia principal da evolução diferencial é gerar novos indivíduos, denotados vetores modificados ou doadores, pela adição da diferença vetorial ponderada entre dois indivíduos aleatórios da população a um terceiro indivíduo. Esta operação é chamada *mutação*.

As componentes deste novo indivíduo doador são misturadas com as componentes de um indivíduo escolhido aleatoriamente (denotado vetor alvo ou vetor a ser substituído), para resultar o chamado vetor tentativa, ou vetor experimental. O processo de misturar os parâmetros é referido freqüentemente como *cruzamento* na comunidade dos algoritmos evolutivos.

Se o custo do vetor experimental for menor que o custo do vetor alvo, então o vetor experimental será o vetor alvo da geração seguinte. Esta última operação é chamada *seleção*. O procedimento é finalizado através de algum critério de parada.

4.1. Operadores da Evolução Diferencial

Os operadores da evolução diferencial se baseiam no princípio da evolução natural cujos objetivos são manter a diversidade da população, evitar convergências prematuras e obter a melhor solução.

4.1.1 Mutação

Para a obtenção do vetor doador $V^{(q+1)}$, considere os vetores $X_\alpha^{(q)}$, $X_\beta^{(q)}$ e $X_\gamma^{(q)}$ distintos entre si e escolhidos aleatoriamente em uma população com Np indivíduos. Np deve ser maior ou igual a 4 para garantir uma quantidade suficiente de indivíduos para a execução do método. Os índices aleatórios $\alpha, \beta, \gamma \in \{1, \dots, Np\}$ são inteiros distintos entre si. Utilizando o par de vetores (X_β, X_γ) da q -ésima geração define-se o vetor diferença $(X_\beta - X_\gamma)$. Esta diferença é multiplicada por F_p , sendo denotada diferença vetorial ponderada ou apenas diferença ponderada e será usada para perturbar o terceiro vetor X_α . O fator de perturbação F_p é um número real, positivo pertencente ao intervalo $[0, 2]$ e controla a amplitude do vetor diferença.

O processo de mutação pode ser escrito como:

$$V^{(q+1)} = X_\alpha^{(q)} + F_p (X_\beta^{(q)} - X_\gamma^{(q)}) \quad (4.1)$$

A Figura 4.1 mostra um exemplo bidimensional que ilustra os diferentes vetores que geram o vetor doador $V^{(q+1)}$.

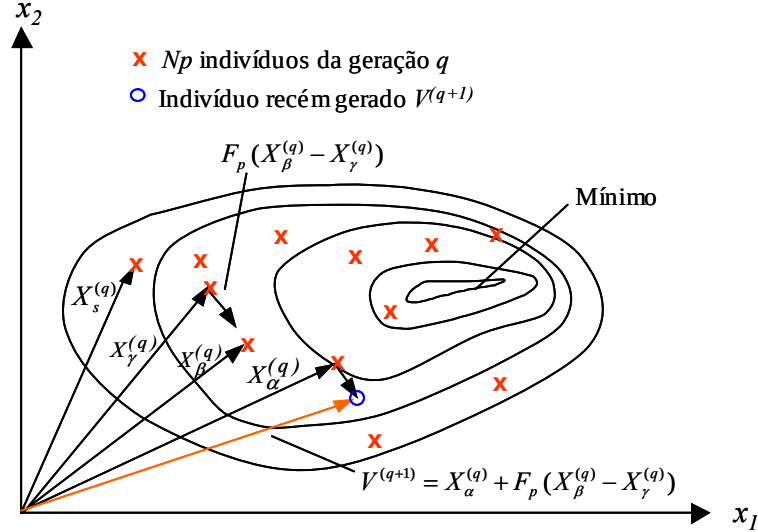


Figura 4.1 - Processo para gerar o vetor doador $V^{(q+1)}$ de uma função bidimensional

4.1.2 Cruzamento

Considere que para cada vetor alvo $X_s^{(q)}$, $s \in \{1, \dots, Np\}$, e diferente dos índices α , β e γ , foi gerado um vetor doador. O cruzamento é introduzido para aumentar a diversidade dos indivíduos que sofreram a mutação. Assim, utilizando o vetor doador e o vetor alvo, as componentes do vetor experimental $U^{(q+1)}$ são escolhidas pela seguinte comparação:

$$u(i)^{(q+1)} = \begin{cases} v(i)^{(q+1)}, & \text{se } rand_i \leq Pc. \\ x_s(i)^{(q)}, & \text{se } rand_i > Pc, \quad i = 1, \dots, n. \end{cases} \quad (4.2)$$

sendo $v(i)^{(q+1)}$ a $(q+1)$ -ésima componente do vetor doador $V^{(q+1)}$, $x_s(i)$ a do vetor alvo $X_s^{(q)}$ e $rand_i$ um número gerado aleatoriamente no intervalo $[0, 1]$, $Pc \in [0, 1]$ é a probabilidade do cruzamento e representa a probabilidade do vetor experimental herdar os valores das variáveis do vetor doador, devendo ser fornecida pelo usuário. Quando $Pc = 1$, por exemplo, todas as componentes do vetor experimental virão do vetor doador $V^{(q+1)}$. Por outro lado, se $Pc = 0$, todas as componentes do vetor experimental virão do vetor alvo $X_s^{(q)}$.

Este tipo de cruzamento, apresentado por Storn e Price (1995) é denominado operador cruzamento binomial (devido aos experimentos binomiais independentes), sendo executado em cada variável sempre que um número aleatório $rand \in [0, 1]$ for menor que a

probabilidade de cruzamento P_c . A Fig 4.2 mostra o processo de cruzamento binomial para uma função de 7 variáveis.

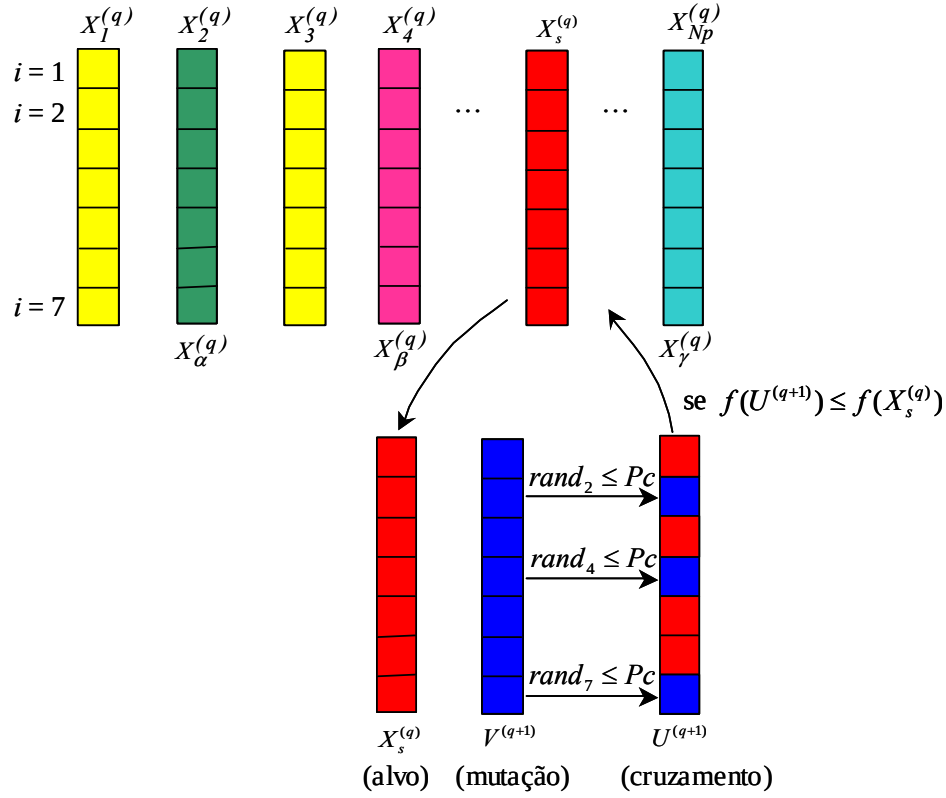


Figura 4.2 - Ilustração do processo de cruzamento binomial para $\alpha = 2$, $\beta = 4$ e $\gamma = Np$

Alguns anos mais tarde, Storn e Price (1997) desenvolveram o operador cruzamento exponencial, em que o cruzamento é executado nas variáveis enquanto o número aleatório $rand \in [0,1]$ for menor que a probabilidade de cruzamento P_c . A primeira vez que este número aleatório ultrapassar o valor de P_c , nenhum cruzamento é executado e as variáveis restantes são deixadas intactas, ou seja:

$$\begin{aligned} \text{Enquanto } rand_i \leq P_c, u(i)^{(q+1)} &= v(i)^{(q+1)}, \\ \text{Se } rand_i > P_c, u(j)^{(q+1)} &= x_s(j)^{(q)}, j = (i+1), \dots, n \end{aligned} \quad (4.3)$$

A Figura 4.3 mostra o processo de cruzamento exponencial para uma função de 7 variáveis.

Se após o cruzamento uma ou mais componentes do vetor experimental estiver fora da região de busca, definida pelas restrições laterais das variáveis de projeto, as seguintes correções devem ser feitas:

$$\begin{cases} \text{Se } u(i) < x(i)^{\inf} \text{ então } u(i) = x(i)^{\inf} \\ \text{Se } u(i) > x(i)^{\sup} \text{ então } u(i) = x(i)^{\sup} \end{cases} \quad i = 1, \dots, n \quad (4.4)$$

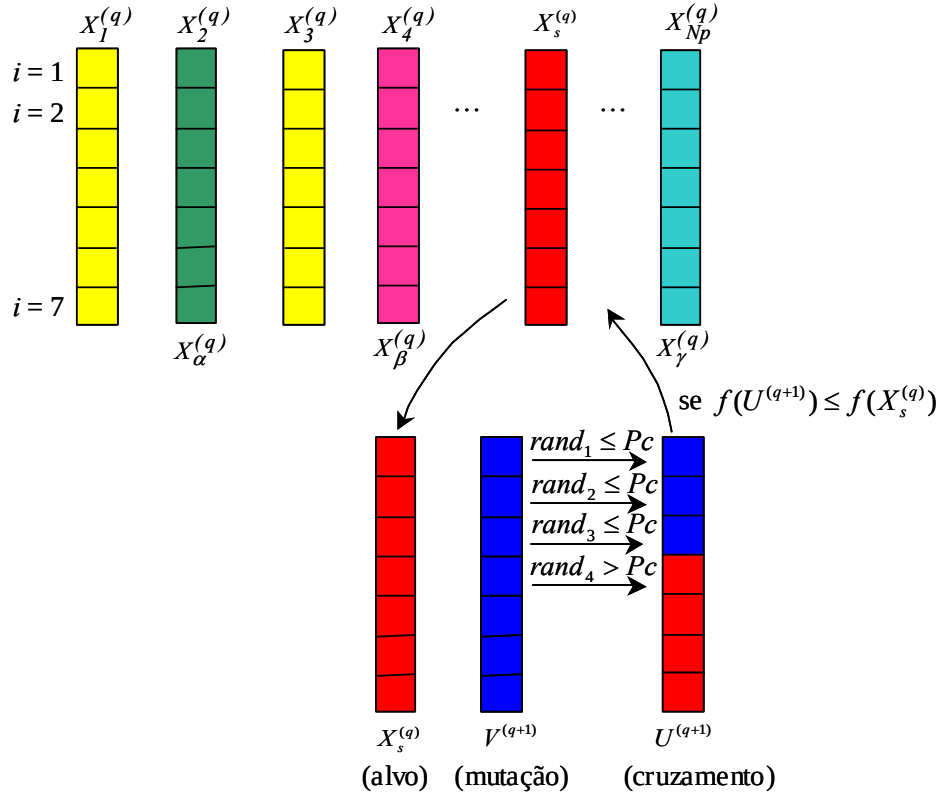


Figura 4.3 - Ilustração do processo de cruzamento exponencial para $\alpha = 2$, $\beta = 4$ e $\gamma = Np$

4.1.3 Seleção

A seleção é o processo de produzir filhos melhores. Diferentemente de outros algoritmos evolutivos, a evolução diferencial não usa hierarquia (elitismo) nem seleção proporcional. Ao invés disso, o custo do vetor experimental $U^{(q+1)}$ é calculado e comparado com o custo do vetor alvo $X_s^{(q)}$. Se o custo do vetor experimental for menor que o custo do vetor alvo, o vetor alvo da próxima geração será o vetor experimental. Caso contrário, o vetor alvo da próxima geração será o vetor alvo da geração atual.

Em outras palavras, este processo pode ser escrito como:

$$\begin{cases} \text{Se } f(U^{(q+1)}) \leq f(X_s^{(q)}) \text{ então } X_s^{(q+1)} = U^{(q+1)} \\ \text{Se } f(U^{(q+1)}) > f(X_s^{(q)}) \text{ então } X_s^{(q+1)} = X_s^{(q)} \end{cases} \quad (4.5)$$

O procedimento acima é finalizado através de algum critério de parada, sendo que o número máximo de gerações deve ser estabelecido.

Usualmente, o desempenho do algoritmo de ED depende principalmente do tamanho da população Np , da região de busca, da taxa de cruzamento e também do fator de perturbação F_p .

4.2. Estratégias da Evolução Diferencial

Diferentes estratégias da evolução diferencial podem ser obtidas alterando-se a forma de obtenção dos operadores mutação e cruzamento, de acordo com:

- o tipo de indivíduo (X_α) a ser modificado na formação do vetor doador;
- o número de vetores diferenciais (diferenças ponderadas) considerados;
- o tipo de cruzamento a ser utilizado.

Para classificar as diferentes variações, Storn e Price (1995) introduziram a seguinte notação:

Estratégia ED/ $a/b/c$

onde,

a – especifica o vetor (X_α) a ser perturbado. Se este vetor for escolhido aleatoriamente entre os indivíduos da população, este parâmetro da estratégia é representado como “ $a = \text{rand}$ ”. Caso o vetor X_α seja adotado como o vetor de menor custo da população (X_{best}), representa-se este parâmetro por “ $a = \text{best}$ ”.

b – determina o número de vetores diferença usados para a perturbação de X_α na obtenção do vetor doador. Por exemplo, o operador mutação, dado na Eq.(4.1) utiliza um vetor diferença, assim “ $b = 1$ ”.

c – representa o tipo de cruzamento adotado. Se for utilizado o cruzamento binomial tem-se o parâmetro “ $c = \text{bin}$ ”, se for o cruzamento exponencial “ $c = \text{exp}$ ”.

Usando esta notação e supondo cruzamento binomial, a Eq.(4.1) pode ser escrita como: ED/ $\text{rand}/1/\text{bin}$.

Considere agora que o vetor doador seja obtido por:

$$V^{(q+1)} = X_{best}^{(q)} + F_p (X_\beta^{(q)} - X_\gamma^{(q)}) \quad (4.6)$$

Neste caso, onde o melhor indivíduo X_{best} foi perturbado, tem-se a estratégia ED/ $\text{best}/1/\text{bin}$ ou ED/ $\text{best}/1/\text{exp}$, dependendo do cruzamento utilizado.

Se o número de indivíduos da população for grande o suficiente, a diversidade da população pode ser melhorada usando duas diferenças ponderadas para perturbar um vetor existente, ou seja, cinco vetores distintos são escolhidos aleatoriamente na população atual. O vetor diferencial usa dois pares de diferenças ponderadas para perturbar X_α , que pode ser aleatório ou o melhor vetor da população atual. Este processo pode ser escrito como:

$$V^{(q+1)} = X_\alpha^{(q)} + F_p (X_\rho^{(q)} - X_\beta^{(q)}) + F_p (X_\gamma^{(q)} - X_\delta^{(q)}) \quad (4.7)$$

ou

$$V^{(q+1)} = X_{best}^{(q)} + F_p (X_\alpha^{(q)} - X_\beta^{(q)}) + F_p (X_\gamma^{(q)} - X_\delta^{(q)}) \quad (4.8)$$

Os índices aleatórios $\alpha, \beta, \gamma, \rho, \delta \in \{1, 2, \dots, Np\}$, são inteiros mutuamente distintos e diferentes do índice s , tal que $Np \geq 6$. De acordo com o cruzamento adotado, a Eq.(4.7) representa a estratégia ED/rand/2/bin ou ED/rand/2/exp. De forma similar, a Eq.(4.8) representa a estratégia ED/best/2/bin ou ED/best/2/exp

Tabela 4.1 - Representação das estratégias da evolução diferencial

Número	Mutação	Notação
1	$V^{(q+1)} = X_\alpha^{(q)} + F_p (X_\beta^{(q)} - X_\gamma^{(q)})$	ED/rand/1/bin
2	$V^{(q+1)} = X_{best}^{(q)} + F_p (X_\beta^{(q)} - X_\gamma^{(q)})$	ED/best/1/bin
3	$V^{(q+1)} = X_\alpha^{(q)} + F_p (X_\rho^{(q)} - X_\beta^{(q)}) + F_p (X_\gamma^{(q)} - X_\delta^{(q)})$	ED/rand/2/bin
4	$V^{(q+1)} = X_{best}^{(q)} + F_p (X_\alpha^{(q)} - X_\beta^{(q)}) + F_p (X_\gamma^{(q)} - X_\delta^{(q)})$	ED/best/2/bin
5	$V^{(q+1)} = X_{old}^{(q)} + F_p (X_{best}^{(q)} - X_{old}^{(q)}) + F_p (X_\gamma^{(q)} - X_\delta^{(q)})$	ED/rand-to-best/2/bin
6	$V^{(q+1)} = X_\alpha^{(q)} + F_p (X_\beta^{(q)} - X_\gamma^{(q)})$	ED/rand/1/exp
7	$V^{(q+1)} = X_{best}^{(q)} + F_p (X_\beta^{(q)} - X_\gamma^{(q)})$	ED/best/1/exp
8	$V^{(q+1)} = X_\alpha^{(q)} + F_p (X_\rho^{(q)} - X_\beta^{(q)}) + F_p (X_\gamma^{(q)} - X_\delta^{(q)})$	ED/rand/2/exp
9	$V^{(q+1)} = X_{best}^{(q)} + F_p (X_\alpha^{(q)} - X_\beta^{(q)}) + F_p (X_\gamma^{(q)} - X_\delta^{(q)})$	ED/best/2/exp
10	$V^{(q+1)} = X_{old}^{(q)} + F_p (X_{best}^{(q)} - X_{old}^{(q)}) + F_p (X_\gamma^{(q)} - X_\delta^{(q)})$	ED/rand-to-best/2/exp

Existem ainda outras duas estratégias, onde o vetor doador tem contribuições do melhor indivíduo da população (X_{best}) e de algum indivíduo da geração anterior (X_{old}), além dos seguintes vetores diferença:

$$V^{(q+1)} = X_{old}^{(q)} + F_p (X_{best}^{(q)} - X_{old}^{(q)}) + F_p (X_{\gamma}^{(q)} - X_{\delta}^{(q)}) \quad (4.9)$$

Dependendo do tipo de cruzamento adotado, a Eq.(4.9) pode representar as estratégias ED/rand-to-best/2/bin ou ED/rand-to-best/2/exp.

Resumidamente, as dez estratégias podem ser descritas de acordo com a Tab. 4.1.

Cabe ressaltar que uma estratégia que funciona bem para um dado problema pode não funcionar bem quando aplicada a outro problema, sendo recomendável que várias estratégias sejam testadas para o mesmo problema.

4.3. Algoritmo da Evolução Diferencial

(i) Definir a estratégia a ser usada (vide Tab. 4.1), escolher o fator de perturbação F_p e a probabilidade de cruzamento P_c ;

(ii) **(inicialização)** Gerar uma população inicial aleatória, utilizando, por exemplo, a distribuição uniforme:

$$x(i)_d^{(q)} = x(i)^{\inf} + rand_i (x(i)^{\sup} - x(i)^{\inf}), \quad i = 1, \dots, n; \quad d = 1, \dots, Np$$

onde $x(i)^{\inf} \leq x(i) \leq x(i)^{\sup}$ são as restrições laterais, $rand_i \in [0, 1]$ e neste caso, $q=1$.

(iii) Escolher um indivíduo aleatório $X_s^{(q)}$ ou adotar $X_{best}^{(q)}$, de acordo com a estratégia escolhida, a ser substituído (alvo);

(iv) Escolher outros três indivíduos $X_{\alpha}^{(q)}$, $X_{\beta}^{(q)}$ e $X_{\gamma}^{(q)}$, $\alpha \neq \beta \neq \gamma \neq s$ ou cinco indivíduos distintos, dependendo da estratégia adotada em (i);

(v) **(mutação)** Gerar um indivíduo doador $V^{(q+1)}$ de acordo com a estratégia escolhida, por exemplo:

$$V^{(q+1)} = X_{\alpha}^{(q)} + F_p (X_{\beta}^{(q)} - X_{\gamma}^{(q)}) \quad \text{ou} \quad V^{(q+1)} = X_{best}^{(q)} + F_p (X_{\beta}^{(q)} - X_{\gamma}^{(q)})$$

(vi) **(cruzamento)** Gerar um indivíduo $U^{(q+1)}$ a ser comparado com $X_s^{(q)}$ ou $X_{best}^{(q)}$ através da equação dada por:

$$u(i)^{(q+1)} = \begin{cases} v(i)^{(q+1)}, & \text{se } rand_i \leq Pc. \\ x_s(i)^{(q)} \text{ (ou } x_{best}(i)^{(q)}), & \text{se } rand_i > Pc, \quad i = 1, \dots, n. \end{cases}$$

Se após o cruzamento uma ou mais componentes de $U^{(q+1)}$ estiver fora da região de busca, faz-se a seguinte correção:

Se $u(i) < x(i)^{\inf}$, então faz-se $u(i) = x(i)^{\inf}$,

Se $u(i) > x(i)^{\sup}$, então faz-se $u(i) = x(i)^{\sup}$, $i = 1, \dots, n$.

(vii) **(seleção)** Escolher o melhor indivíduo analisando a função objetivo:

Se $f(U^{(q+1)}) \leq f(X_s^{(q)})$ então $X_s^{(q+1)} = U^{(q+1)}$;

Se $f(U^{(q+1)}) > f(X_s^{(q)})$ então $X_s^{(q+1)} = X_s^{(q)}$.

(viii) **(critério de parada)** Se um critério de parada é satisfeito, fim. Senão, passar para a próxima geração ($q = q+1$) e voltar ao passo (iii).

CAPÍTULO V

SIMULAÇÕES NUMÉRICAS – APLICAÇÕES

Neste capítulo é feita uma comparação entre os resultados obtidos com Algoritmos Genéticos, Recozimento Simulado e as diversas estratégias do algoritmo de Evolução Diferencial, a fim de observar o comportamento do último método citado. Primeiramente é apresentado um estudo cujo objetivo é analisar o efeito dos parâmetros fornecidos pelo usuário para a utilização da Evolução Diferencial, para isto utilizou-se a função de Rastrigin (MENDES, 2004; OLIVEIRA, 2004) bidimensional. Em seguida são examinadas funções matemáticas com ou sem a imposição de restrições. Finalmente, apresenta-se o problema do despacho econômico e ambiental de geração de energia elétrica, cujo objetivo é selecionar as potências geradoras ótimas, assim como encontrar a demanda de carga com custo mínimo e emissão mínima de poluentes na atmosfera.

Os métodos de otimização foram executados utilizando códigos computacionais implementados em Matlab7, sendo utilizado um microcomputador com as seguintes configurações: microprocessador Intel(R) Pentium; (R) 4 CPU 2.40 GHz, AT/AT compatível, 261664 KB RAM. Os Algoritmos Genéticos foram aplicados usando o código computacional GAOT (HOUCK et al., 1995), adotando-se os parâmetros padrões: $Np=80$; precisão de 10^{-6} ; número de gerações igual a 100; seleção tipo ranqueamento com probabilidade de 80% e probabilidade de mutação de 1%. Os parâmetros utilizados pelo Recozimento Simulado (SARAMAGO & FARIA, 2001) foram: quantidade de temperaturas: $n_{temps}=100$; quantidade de iterações para cada temperatura $n_{iters}=300$; temperatura inicial igual a 2,5 e desvio padrão final de 0,001.

5.1. Efeito da variação dos parâmetros da Evolução Diferencial

Seja a função conhecida como função de Rastrigin, que possui 10^n mínimos locais, dada pela Eq.(5.1). Considerando $n=2$, tem-se uma função com 100 mínimos locais, conforme Fig. 5.1, e solução analítica $f(X^*)=0$ e $X^*=(0,0)^T$.

$$f(x_1, \dots, x_n) = 10n + \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i)], \quad x_i \in [-10, 10], \quad i=1,2 \quad (5.1)$$

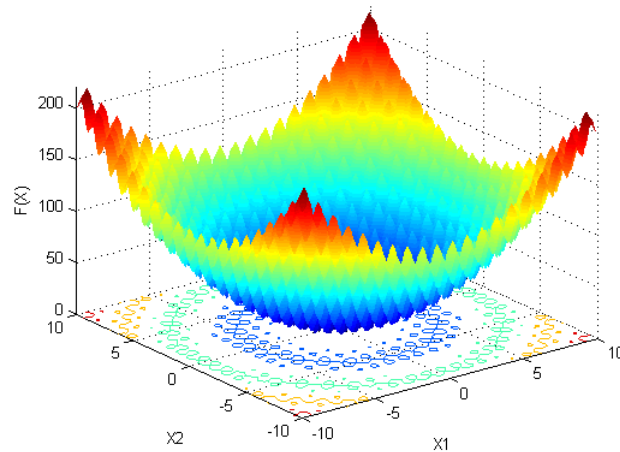


Figura 5.1 - Função de Rastrigin bidimensional, dada pela Eq.(5.1)

Por se tratar de uma função amplamente estudada, a função de Rastrigin foi escolhida para um estudo cujo objetivo é analisar o efeito dos parâmetros fornecidos pelo usuário e obter melhores valores que possibilitem bons desempenhos das estratégias da evolução diferencial.

Para realizar esta análise, foram utilizados inicialmente os parâmetros recomendados por Storn (1997), sendo substituídos de acordo com a obtenção de melhores parâmetros resultantes deste estudo. A precisão estabelecida foi da ordem de 10^{-6} e o número máximo de gerações foi 200.

Cada estratégia foi executada 20 vezes, considerando o melhor valor obtido destas execuções. Este procedimento foi realizado para cada parâmetro testado, a saber:

1º teste: Variação do Fator de Perturbação dos indivíduos na população (F_p):

Neste primeiro teste faz-se a variação do fator de perturbação dos indivíduos na população, considerando valores na faixa de 0,1 a 2,0, enquanto os parâmetros fixos são: $Np=10$, $N_{ger}=200$ e $P_c=0,5$;

A Figura 5.2 mostra as curvas das dez estratégias do algoritmo de evolução diferencial considerando a influência do fator de perturbação F_p , sobre o erro absoluto dos valores ótimos da função em estudo, dada pela Eq.(5.1).

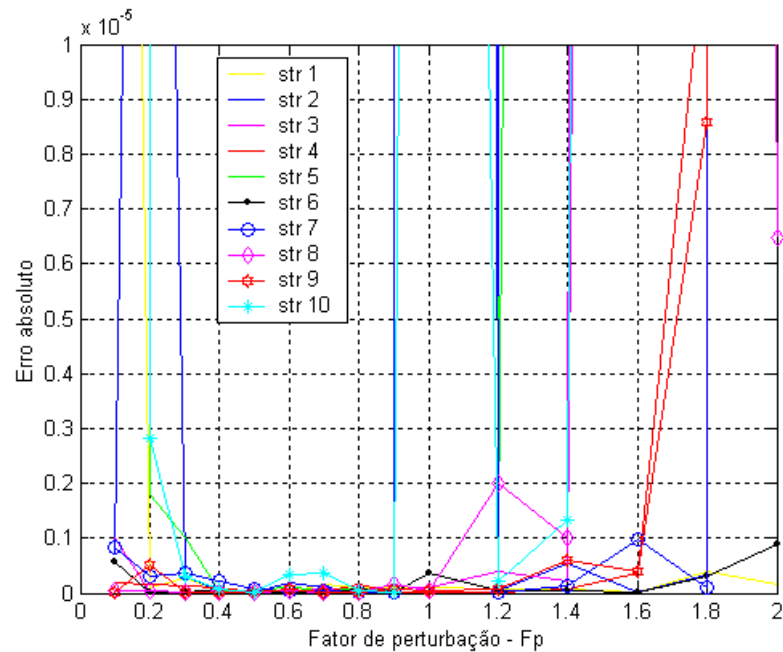


Figura 5.2 - Erro absoluto da função objetivo em relação à variação do fator de perturbação F_p (fixando: $N_p=10$, $N_{ger}=200$ e $P_c=0,5$)

Observa-se que para um erro da ordem de $0,1 \times 10^{-5}$, todas as estratégias apresentaram bons resultados para F_p na faixa de 0,4 a 0,8.

Analisando o gráfico de F_p versus número de avaliações da função, conforme Fig. 5.3, observa-se que para todas as estratégias, uma maior perturbação na população (aumentando o valor de F_p) representa a necessidade de um número maior de avaliação da função objetivo, resultando em um maior esforço computacional. Na faixa considerada melhor pela análise do gráfico anterior, com F_p variando de 0,2 a 0,8, a maioria das estratégias apresentam um número de avaliações da função razoável (inferior a 1500 avaliações, exceções feitas para as estratégias 3, 5, 8 e 10).

2º teste: Variação da Probabilidade de Cruzamento (P_c):

Para verificar a variação da probabilidade de cruzamento dos indivíduos, foram considerados valores pré-estabelecidos na faixa de 0 a 1, adotando-se os parâmetros fixos: $N_p=10$; $N_{ger}=200$ e $F_p=0,8$;

Observa-se na Fig. 5.4, que estabelecendo um erro da ordem de 10^{-6} , para a taxa de cruzamento na faixa de 0,4 a 0,8, todas as estratégias atendem o erro estabelecido, não sendo recomendado trabalhar com P_c abaixo de 0,3.

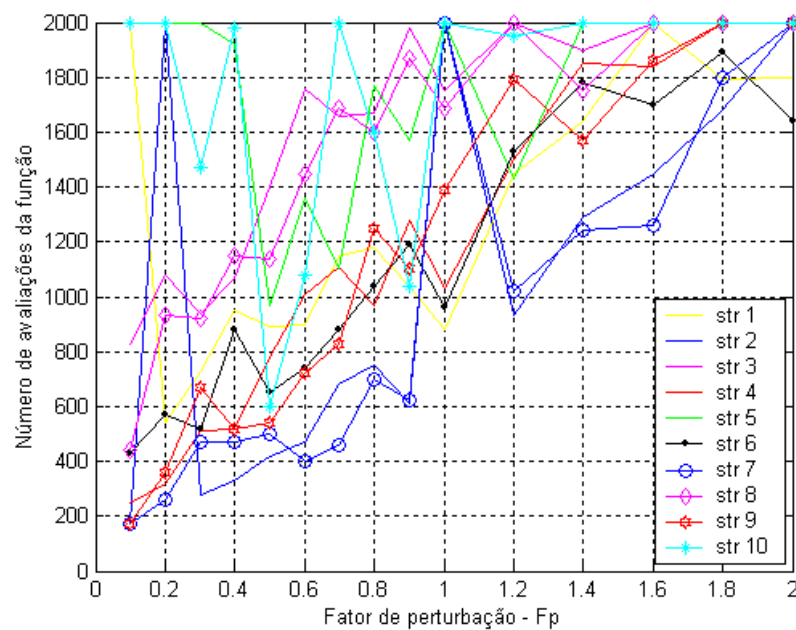


Figura 5.3 - Número de avaliação da função objetivo em relação à variação do fator de perturbação F_p (fixando $N_p=10$, $N_{ger}=200$ e $P_c=0,5$)

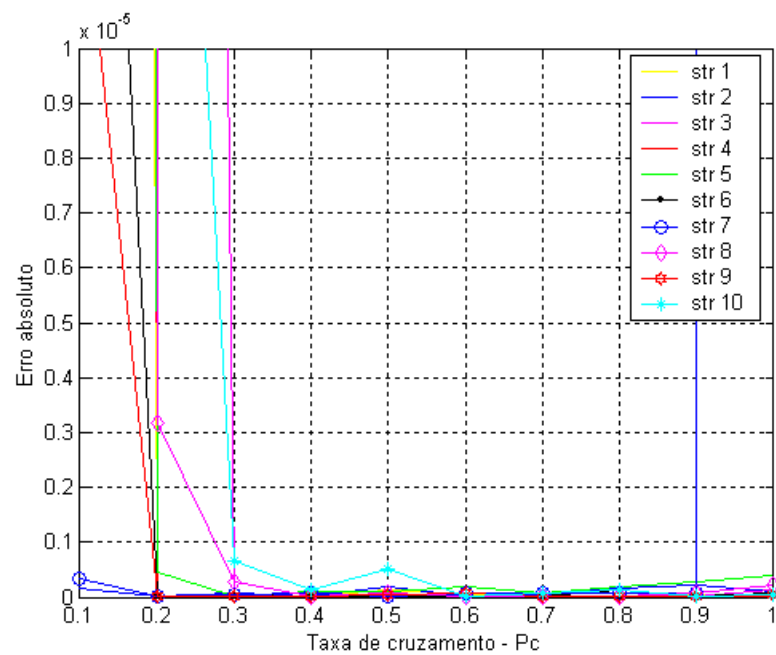


Figura 5.4 - Erro absoluto da função objetivo em relação à variação da taxa de cruzamento P_c (fixando: $N_p=10$, $N_{ger}=200$ e $F_p=0,8$)

A Figura 5.5 mostra que, para a maioria das estratégias, o número de avaliações da função decai com o aumento da taxa de cruzamento. Acima de 0,9 este comportamento é instável.

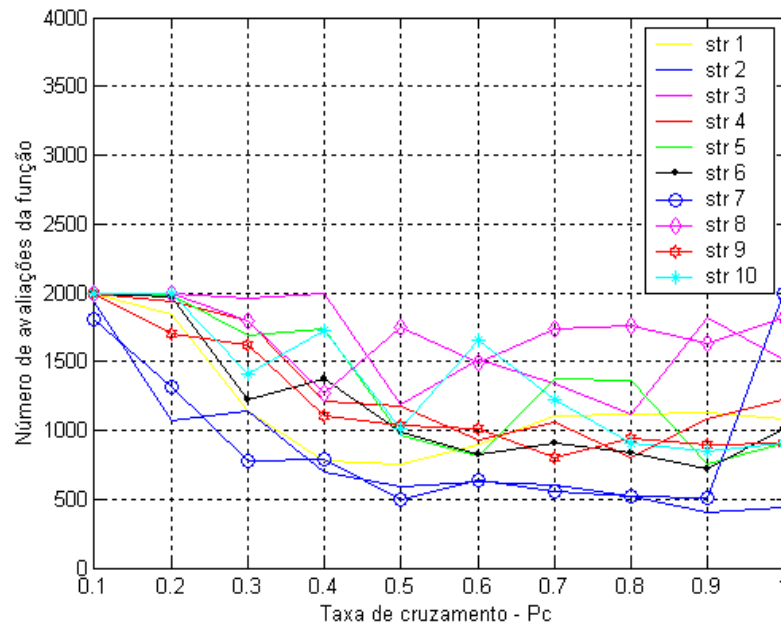


Figura 5.5 - Número de avaliação da função objetivo em relação à variação da taxa de cruzamento P_c (fixando: $N_p=10$, $N_{ger}=200$ e $F_p=0,8$)

Comparando as Figs. 5.4 e 5.5, verifica-se que a taxa de cruzamento entre 0,4 e 0,8, além de apresentar menor erro absoluto apresenta uma menor variação no número de avaliações da função. Abaixo de 0,3 e acima de 0,9, o número de avaliações da função cresceu para algumas estratégias, por exemplo, verifique o comportamento da estratégia 7.

3º teste: Variação do número de indivíduos da população (N_p):

Este teste faz a variação do número de indivíduos da população, cujo objetivo é a verificação da influência deste parâmetro no desempenho do algoritmo. Foram estabelecidas populações com 5, 10, 15, 20, 25, 30, 40, 50, 70 e 90 indivíduos. Os parâmetros fixos foram: $F_p=0,8$; $N_{ger}=200$ e $P_c=0,5$;

Observando a Fig. 5.6, verifica-se que erros da ordem de $0,5 \times 10^{-6}$ são atingidos por todas as estratégias independentemente do número de indivíduos da população. Apenas a estratégia 10 apresentou uma ligeira dificuldade com N_p inferior a 15.

Observa-se claramente na Fig. 5.7 que ao aumentar o número de indivíduos da população, o número de avaliações da função aumenta significativamente.

Comparando a Fig. 5.6 com a Fig. 5.7, comprova-se que o algoritmo de evolução diferencial é indicado a ser aplicado com pequeno número de indivíduos na população, sendo esta sua principal característica. Quando comparado com os algoritmos genéticos,

por exemplo, a vantagem representa a diminuição do tempo computacional, conforme será apresentado nas simulações numéricas.

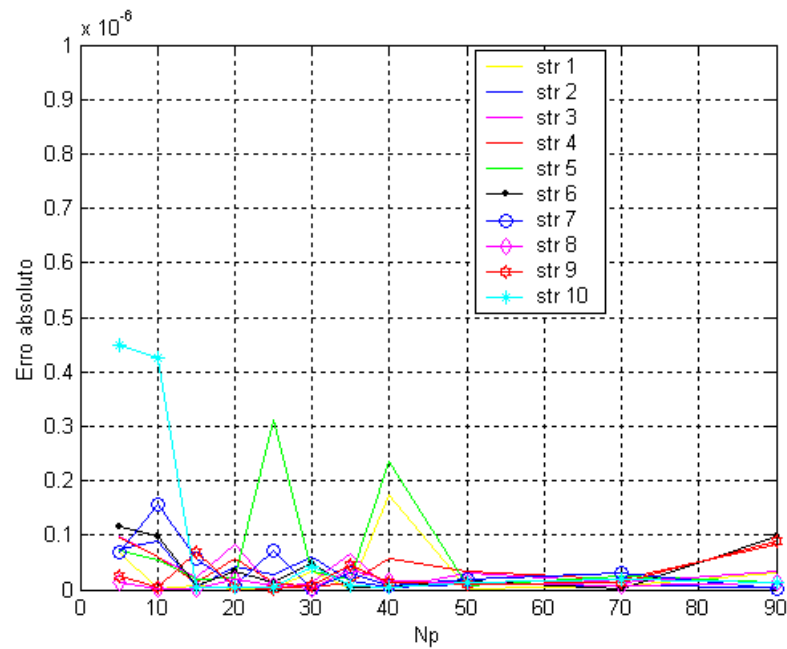


Figura 5.6 - Erro absoluto da função objetivo em relação à variação do número de indivíduos da população N_p (fixando: $F_p=0,8$, $N_{ger}=200$ e $P_c=0,5$)

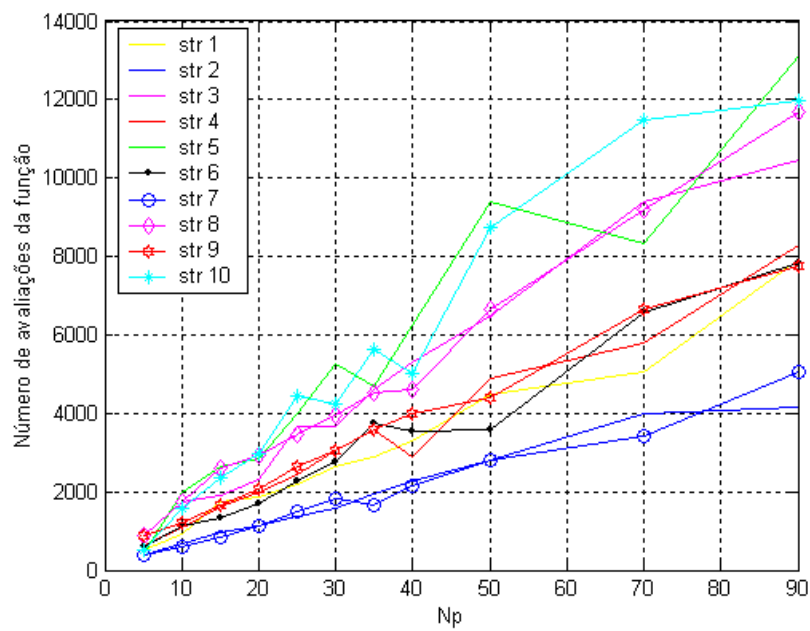


Figura 5.7 - Número de avaliação da função objetivo em relação à variação do número de indivíduos da população N_p (fixando $F_p=0,8$, $N_{ger}=200$ e $P_c=0,5$)

4º teste: Variação do número máximo de gerações (N_{ger}):

Este último teste consiste em variar o número máximo de gerações considerando 50, 100, 200 e 300 gerações. Os parâmetros fixos utilizados foram: $N_p=10$, $P_c=0,6$ e $F_p=0,8$.

A Figura 5.8 mostra claramente que ao aumentar o número de gerações de 50 para 100 houve uma queda do erro para a maioria das estratégias. Acima de 100 gerações houve uma estabilização.

Na Figura 5.9, o aumento do número de gerações significou um aumento no número de avaliações da função, mas não tão significativo como aconteceu na Fig. 5.7, quando elevou-se o número de indivíduos da população.

Comparando a Fig. 5.8 com a Fig. 5.9, verifica-se que acima de 100 gerações o erro é extremamente pequeno, não alterando-se com a variação do número de gerações. Assim não justifica aumentar o esforço computacional com a elevação do número de avaliações da função objetivo, uma vez que não causará melhoras no resultado.

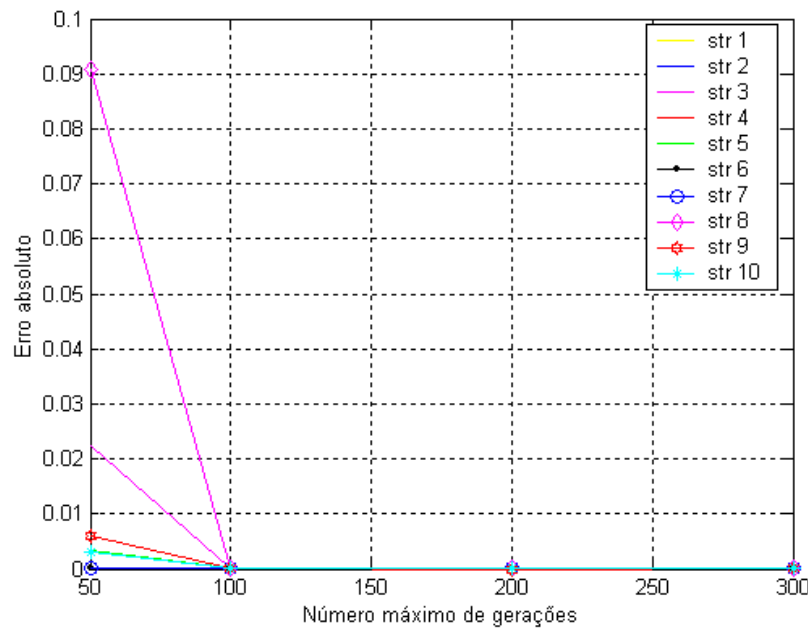


Figura 5.8 - Erro absoluto da função objetivo em relação à variação do número máximo de gerações N_{ger} (fixando: $N_p=10$, $P_c=0,6$ e $F_p=0,8$)

Conclusões:

Conclui-se, portanto, que os melhores parâmetros para a evolução diferencial obtidos nesta análise foram: valores baixos para o número de indivíduos da população; número máximo de gerações em torno de 200; taxa de perturbação dos indivíduos $F_p \in [0,4 \text{ a } 0,8]$ e probabilidade de cruzamento $P_c \in [0,4 \text{ a } 0,8]$. Vale ressaltar que foram testadas outras funções e repetiu-se estes mesmos parâmetros da Evolução Diferencial.

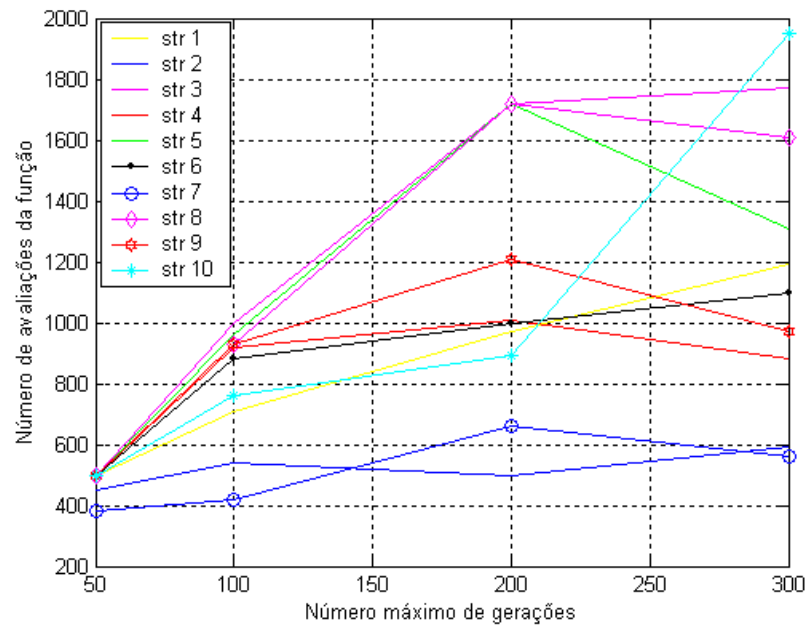


Figura 5.9 - Número de avaliação da função objetivo em relação à variação do número máximo de gerações N_{ger} (fixando: $N_p=10$, $P_c=0,6$ e $F_p=0,8$)

Os resultados da otimização utilizando os melhores parâmetros obtidos na análise podem ser observados na Tab. 5.1, verificando-se que todas as técnicas encontraram o mínimo global. Todas as estratégias da evolução diferencial encontraram resultados tão bons quanto os algoritmos genéticos e simulated annealing.

Tabela 5.1 - Resultados da otimização para a função de Rastrigin, Eq.(5.1)

	$f(X^*)$	x_1	x_2
ED	1 $5,46 \times 10^{-9}$	$4,01 \times 10^{-6}$	$-3,38 \times 10^{-6}$
	2 $3,49 \times 10^{-8}$	$-6,43 \times 10^{-6}$	$-1,16 \times 10^{-5}$
	3 $3,09 \times 10^{-9}$	$2,75 \times 10^{-6}$	$-2,83 \times 10^{-6}$
	4 $3,77 \times 10^{-8}$	$8,50 \times 10^{-7}$	$1,38 \times 10^{-5}$
	5 $3,08 \times 10^{-7}$	$-3,41 \times 10^{-5}$	$-1,98 \times 10^{-5}$
	6 $1,22 \times 10^{-9}$	$-2,08 \times 10^{-6}$	$-1,35 \times 10^{-6}$
	7 $2,42 \times 10^{-8}$	$-1,09 \times 10^{-5}$	$1,75 \times 10^{-6}$
	8 $3,52 \times 10^{-8}$	$1,29 \times 10^{-6}$	$1,326 \times 10^{-5}$
	9 $1,04 \times 10^{-7}$	$1,87 \times 10^{-5}$	$1,32 \times 10^{-5}$
	10 $1,78 \times 10^{-8}$	$7,31 \times 10^{-6}$	$6,01 \times 10^{-6}$
AGs	$2,10 \times 10^{-8}$	$7,78 \times 10^{-6}$	$6,73 \times 10^{-6}$
SA	$3,27 \times 10^{-8}$	$1,08 \times 10^{-5}$	$6,93 \times 10^{-6}$

5.2. Funções Matemáticas sem restrições

Nesta seção, o objetivo é a minimização de algumas funções matemáticas sem restrições, amplamente encontradas na literatura. Os parâmetros adotados para a Evolução Diferencial, baseando-se na análise apresentada na seção 5.1 foram: $Np=15$; $Nger=200$; $F_p=0,8$ e $P_c=0,6$.

5.2.1 Exemplo 1

Trata-se da função de Rosenbrok, com duas variáveis de projetos, dada pela Eq.(5.2), compreendida no espaço de busca $x_i \in [-2.048, 2.048]$.

$$f(x_1, x_2) = 100(x_1^2 - x_2)^2 + (1 - x_1)^2 \quad (5.2)$$

A Tabela 5.2 mostra os valores ótimos das variáveis de projeto e da função objetivo, utilizando as 10 estratégias da evolução diferencial, algoritmos genéticos e recozimento simulado.

Tabela 5.2 – Resultados ótimos para a função de Rosenbrok, Eq.(5.2)

	$f(X^*)$	x_1	x_2
1	$1,34 \times 10^{-8}$	0,9999	0,9998
2	$1,39 \times 10^{-8}$	0,9999	0,9999
3	$1,75 \times 10^{-7}$	1,0002	1,0005
4	$2,07 \times 10^{-8}$	0,9999	0,9997
5	$3,19 \times 10^{-8}$	1,0002	1,0003
ED 6	$3,24 \times 10^{-9}$	1,0000	1,0000
7	$3,01 \times 10^{-9}$	0,9999	0,9999
8	$3,19 \times 10^{-8}$	0,9998	0,9997
9	$4,95 \times 10^{-8}$	1,0002	1,0004
10	$7,52 \times 10^{-10}$	1,0000	0,9999
AGs	$1,00 \times 10^{-4}$	0,9926	0,9852
SA	$3,32 \times 10^{-9}$	0,9999	0,9999

A solução analítica desta função é $f(X^*)=0$ e $X^*=(1,1)^T$. Observando a Tab. 5.2, verifica-se que todas as técnicas estudadas mostraram eficiência na busca da solução ótima. As estratégias da evolução diferencial apresentaram melhores resultados que os AGs.

5.2.2 Exemplo 2

Esta é uma função não linear, dada na Eq.(5.3), cuja solução analítica é $f(X^*)=0$ e $X^*=(\sqrt{10}, 10)^T$. Na Tab. 5.3, observa-se que foram obtidos mínimos negativos e positivos, que são soluções simétricas, conforme Fig. 5.10.

$$f(x_1, x_2) = \left[(x_1^2 - 10)^2 + (x_2 - 10)^2 \right]^{\frac{1}{10}}, \quad x_i \in [-4, 15] \quad (5.3)$$

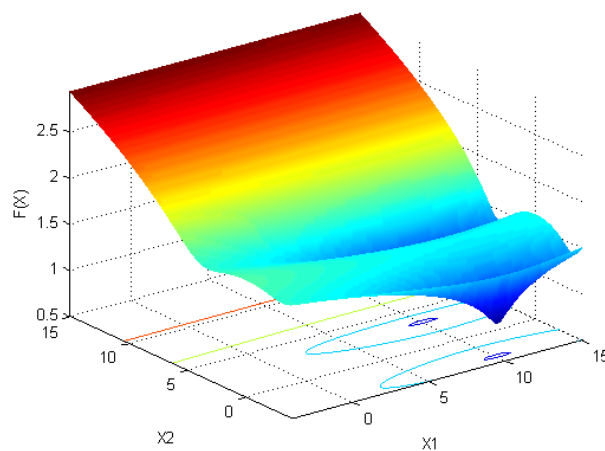


Figura 5.10 - Representação gráfica da função dada pela Eq.(5.3)

Observando a Tab. 5.3, verifica-se que todas as estratégias da evolução diferencial apresentam soluções com melhores precisões que os AGs e SA.

Tabela 5.3 - Resultados da otimização da função dada pela Eq.(5.3)

	$f(X^*)$	x_1	x_2
ED	1 $1,10 \times 10^{-3}$	-3,1623	10,0000
	2 $1,10 \times 10^{-3}$	-3,1623	10,0000
	3 $1,80 \times 10^{-2}$	3,1623	10,0000
	4 $5,40 \times 10^{-3}$	3,1623	10,0000
	5 $1,10 \times 10^{-3}$	3,1623	10,0000
	6 $2,40 \times 10^{-3}$	-3,1623	10,0000
	7 $1,10 \times 10^{-3}$	-3,1623	10,0000
	8 $2,15 \times 10^{-2}$	-3,1623	10,0000
	9 $5,40 \times 10^{-3}$	3,1623	10,0000
	10 $1,10 \times 10^{-3}$	-3,1623	10,0000
AGs	$9,55 \times 10^{-2}$	3,1623	10,0000
SA	$1,29 \times 10^{-1}$	3,1623	10,0000

5.2.3 Exemplo 3

Neste exemplo, considera-se a função de Rastrigin com $n=5$, com o objetivo de verificar o desempenho/ eficácia da ED, uma vez que aumentando o número de variáveis de projeto também aumenta a dificuldade dos métodos de otimização.

$$f(x_1, \dots, x_n) = 10n + \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i)], x_i \in [-10, 10] \quad (5.4)$$

Considerando $n=5$, tem-se 10^5 mínimos locais. Naturalmente, torna-se difícil encontrar o mínimo global, cuja solução analítica é $f(X^*)=0$ e $X^*=(0, 0, 0, 0, 0)^T$.

Tabela 5.4 - Resultados da otimização para a função de Rastrigin, dada pela Eq.(5.4)

	$f(X^*)$	x_1	x_2	x_3	x_4	x_5
ED	1	$6,32 \times 10^{-4}$	$-1,60 \times 10^{-3}$	$-6,00 \times 10^{-4}$	$1,00 \times 10^{-4}$	$-4,00 \times 10^{-4}$
	2	$6,91 \times 10^{-7}$	$-4,55 \times 10^{-5}$	$1,91 \times 10^{-5}$	$-3,05 \times 10^{-6}$	$1,58 \times 10^{-5}$
	3	$4,01 \times 10^{-1}$	$-1,39 \times 10^{-2}$	$9,90 \times 10^{-3}$	$4,80 \times 10^{-3}$	$2,13 \times 10^{-2}$
	4	$5,96 \times 10^{-4}$	$-4,00 \times 10^{-4}$	$7,00 \times 10^{-4}$	$1,40 \times 10^{-3}$	$-1,00 \times 10^{-4}$
	5	$1,58 \times 10^{-4}$	$-2,07 \times 10^{-5}$	$-6,34 \times 10^{-4}$	$6,14 \times 10^{-4}$	$-9,44 \times 10^{-5}$
	6	$4,12 \times 10^{-5}$	$-3,33 \times 10^{-4}$	$-1,82 \times 10^{-4}$	$1,13 \times 10^{-4}$	$-4,48 \times 10^{-5}$
	7	$7,98 \times 10^{-7}$	$-3,91 \times 10^{-5}$	$9,38 \times 10^{-6}$	$3,15 \times 10^{-5}$	$3,71 \times 10^{-5}$
	8	$3,44 \times 10^{-1}$	$7,00 \times 10^{-3}$	$-3,37 \times 10^{-2}$	$-1,07 \times 10^{-2}$	$2,00 \times 10^{-4}$
	9	$3,56 \times 10^{-4}$	$3,12 \times 10^{-4}$	$4,06 \times 10^{-4}$	$-9,49 \times 10^{-4}$	$-2,21 \times 10^{-4}$
	10	$1,22 \times 10^{-6}$	$3,10 \times 10^{-5}$	$-5,87 \times 10^{-5}$	$-7,85 \times 10^{-6}$	$-3,44 \times 10^{-6}$
AGs	$1,22 \times 10^{-4}$	$-3,91 \times 10^{-4}$	$-8,10 \times 10^{-6}$	$1,22 \times 10^{-4}$	$3,08 \times 10^{-4}$	$-5,92 \times 10^{-4}$

Observando a Tab. 5.4, verifica-se que as estratégias 3 e 8 apresentaram resultados piores que AGs. No entanto, as demais estratégias obtiveram melhores resultados ou equivalentes aos AGs. Isso mostra a eficiência das estratégias da evolução diferencial. O Recozimento Simulado não foi eficiente na otimização desta função, encontrando apenas mínimos locais.

5.3. Funções Matemáticas com Restrições

Os algoritmos de otimização natural foram inicialmente desenvolvidos para resolver problemas irrestritos. Assim, nesta pesquisa, para solucionar problemas com a presença de restrições, funções de penalidades são impostas à função objetivo originando uma nova função irrestrita denominada função pseudo objetivo (VANDERPLAATS, 1999), conforme definido nas Eqs.(2.8) e (2.9).

Nesta seção, deseja-se minimizar duas funções com restrições e comparar os resultados entre as técnicas descritas anteriormente. Os parâmetros utilizados foram os mesmos descritos anteriormente. Considerando a dificuldade dos algoritmos evolutivos obedecerem às restrições, adotou-se um fator de penalidade alto para garantir a eficiência do método. Assim, para todos os casos, utilizou-se o fator de penalidade $k_p = 1000$, para a aplicação da Eq.(2.8).

5.3.1 Exemplo 4

Seja o problema de otimização definido por:

$$\text{Min } f(x_1, x_2) = x_1 \sin(4x_1) + 1,1x_2 \sin(2x_2), \quad x_i \in [0, 10] \quad (5.5)$$

$$\text{Sujeito a: } \begin{cases} g_1(X) = x_1 + x_2 - 16 \leq 0 \\ g_2(X) = x_1 - x_2 \leq 0 \end{cases} \quad (5.6)$$

Usando as Eqs.(2.8) e (2.9), esta função pode ser reescrita como um problema irrestrito:

$$\text{Min } \phi(X) = f(X) + k_p P(X)$$

$$\text{onde } P(X) = \{\max[0, g_1(X)]\}^2 + \{\max[0, g_2(X)]\}^2 \quad (5.7)$$

Tabela 5.5 - Resultados da otimização restrita para o problema definido nas Eqs.(5.5) e (5.6)

	$f(X^*)$	x_1	x_2	g_1	g_2	
ED	1	-16,7676	7,4340	8,5818	0,0159	-1,1478
	2	-16,7439	7,4562	8,5580	0,0141	-1,1018
	3	-16,7181	7,4390	8,5627	0,0018	-1,1237
	4	-16,3701	7,4736	8,4870	-0,0394	-1,0134
	5	-16,6373	7,4024	8,6033	0,0058	-1,2009
	6	-16,5282	7,4702	8,5123	-0,0175	-1,0421
	7	-16,6697	7,4443	8,5472	-0,0085	-1,1029
	8	-16,5801	7,4075	8,5719	-0,0206	-1,1645
	9	-16,5514	7,4159	8,5506	-0,0336	-1,1347
	10	-16,5496	7,3962	8,5899	-0,0139	-1,1937
AGs	-16,7189	7,4328	8,5694	0,0022	-1,1365	
SA	-16,7189	7,4369	8,5650	0,0036	-1,1280	

Observando a Tab. 5.5, verifica-se que todas as estratégias da evolução diferencial foram eficientes na minimização, sendo que as estratégias 1 e 2 apresentam resultados melhores que as outras técnicas. Observe que quando as restrições não são obedecidas, elas estão dentro de uma faixa de tolerância aceita pelos programas.

5.3.2 Exemplo 5

Considere o problema de otimização, onde tanto a função quanto as restrições são não-lineares, dado por:

$$\text{Min } f(x_1, \dots, x_4) = x_1^2 - 5x_1 + x_2^2 - 5x_2 + 2x_3^2 - 21x_3 + x_4^2 + 7x_4 + 50 \quad (5.8)$$

$$\text{Sujeito a: } g_1(X) = x_1^2 - x_1 + 2x_2^2 + x_3^2 - 2x_4^2 - x_4 - 10 \leq 0$$

$$g_2(X) = x_1^2 + x_1 + x_2^2 - x_2 + x_3^2 + x_3 + x_4^2 - x_4 - 8 \leq 0 \quad (5.9)$$

$$g_3(X) = 2x_1^2 + 2x_1 + x_2^2 - x_2 + x_3^2 - x_4 - 5 \leq 0, \quad x_i \in [-10, 10]$$

Este exemplo, cuja solução teórica é $X = (0, 1, 2, -1)^T$, $f(X^*) = 6$, $g_1 = -5$, $g_2 = 0$ e $g_3 = 0$, pode ser reescrito como:

$$\text{Min } \phi(X) = f(X) + k_p P(X)$$

$$\text{onde } P(X) = \{\max[0, g_1(X)]\}^2 + \{\max[0, g_2(X)]\}^2 + \{\max[0, g_3(X)]\}^2 \quad (5.10)$$

Tabela 5.6 - Resultados da otimização restrita para o problema definido nas Eqs.(5.8) e (5.9)

	$f(X^*)$	x_1	x_2	x_3	x_4	g_1	g_2	g_3	
ED	1	5,9979	-0,0034	1,0085	2,0003	-0,9978	-4,9544	$4,29 \times 10^{-4}$	$9,99 \times 10^{-4}$
	2	5,9975	0,0003	0,9997	2,0002	-0,9998	-4,9998	$4,97 \times 10^{-4}$	$9,99 \times 10^{-4}$
	3	6,0044	0,0225	0,9893	1,9875	-1,0158	-5,1623	$-2,40 \times 10^{-3}$	$1,30 \times 10^{-3}$
	4	6,0022	-0,0194	1,0112	2,0108	-0,9842	-4,8448	$-3,83 \times 10^{-4}$	$9,96 \times 10^{-4}$
	5	5,9975	0,0001	1,0004	2,0002	-0,9997	-4,9970	$5,13 \times 10^{-4}$	$9,89 \times 10^{-4}$
	6	5,9982	0,0030	0,9908	2,0007	-1,0012	-5,0406	$7,66 \times 10^{-4}$	$7,16 \times 10^{-4}$
	7	5,9975	0,0002	1,0002	2,0002	-0,9997	-4,9978	$4,97 \times 10^{-4}$	$1,00 \times 10^{-3}$
	8	6,0071	-0,0116	1,0328	1,9984	-0,9950	-4,8465	$-7,94 \times 10^{-4}$	$-5,89 \times 10^{-4}$
	9	5,9936	0,0012	1,0077	1,9981	-1,0006	-4,9798	$8,96 \times 10^{-4}$	$2,90 \times 10^{-3}$
	10	5,9975	0,0002	1,0001	2,0002	-0,9997	-4,9982	$5,01 \times 10^{-4}$	$1,00 \times 10^{-3}$
AGs	6,0416	-0,0469	0,9431	2,0478	-0,9510	-4,8366	$-1,80 \times 10^{-3}$	$1,30 \times 10^{-3}$	
SA	5,9832	0,0003	0,9998	2,0002	-0,9997	-4,9962	$5,10 \times 10^{-3}$	$5,90 \times 10^{-3}$	

Observando a Tab. 5.6, verifica-se que todas as estratégias da Evolução Diferencial minimizaram a função obedecendo as restrições. A eficiência do método pode ainda ser comprovada ao se verificar que todas as estratégias obtiveram valores inferiores ao resultado encontrado pelos AGs.

5.4. Problema de Engenharia: Despacho Econômico e Ambiental (DEA)

O objetivo do problema do despacho econômico e ambiental de geração de energia elétrica é selecionar as potências geradoras assim como encontrar a demanda de carga com custo mínimo e emissão mínima de poluentes na atmosfera, enquanto satisfaz todas as unidades e restrições do sistema. Assim, o objetivo é minimizar duas funções objetivo conflitantes: o custo do combustível e a emissão, satisfazendo as restrições de igualdade e laterais. O vetor das potências reais dos geradores é representado por $X = (x_1, x_2, \dots, x_n)^T$.

As curvas do custo $F_c(X)$ do gerador são representadas por funções quadráticas. Assim, o custo total (\$/h) do combustível pode ser expresso conforme a Eq.(5.11).

$$F_c(X) = \sum_{i=1}^N a_i + b_i x_i + c_i x_i^2 \quad (5.11)$$

onde N é o número de geradores, a_i , b_i e c_i , são os coeficientes do custo do i -ésimo gerador e x_i é a potência real para cada gerador.

A emissão total $F_e(X)$ dos poluentes atmosféricos (em ton/h), tais como o óxido de enxofre SO_x e o óxido de nitrogênio NO_x causados por unidades térmicas que consomem combustíveis fósseis pode ser expresso através da Eq.(5.12).

$$F_e(X) = \sum_{i=1}^N 10^{-2} (\alpha_i + \beta_i x_i + \gamma_i x_i^2) + \xi_i e^{(\lambda_i x_i)} \quad (5.12)$$

onde α_i , β_i , γ_i , ξ_i e λ_i são os coeficientes característicos de emissão do i -ésimo gerador.

Para uma operação estável, a potência real de cada gerador é restrita pelos limites inferiores x_i^{inf} e superiores x_i^{sup} , $i=1, \dots, n$.

A potência total gerada deve cobrir a demanda total P_D e a potência real dissipada na linha de transmissão P_{loss} , conforme a Eq.(5.13).

$$\sum_{i=1}^N x_i = P_D + P_{loss} \quad (5.13)$$

Seja $N=6$ geradores, os coeficientes do custo e os coeficientes característicos da emissão dados na Tab. 5.7, de acordo com Abido (2003) e a restrição de igualdade dada na Eq.(5.13) escrita como duas restrições de desigualdade, conforme Eq.(5.15).

Tabela 5.7 - Coeficientes do custo e da emissão de 6 geradores

		G_1	G_2	G_3	G_4	G_5	G_6
Custo	a	10	10	20	10	20	10
	b	200	150	180	100	180	150
	c	100	120	40	60	40	100
Emissão	α	4,091	2,543	4,258	5,426	4,258	6,131
	β	-5,554	-6,047	-5,094	-3,550	-5,094	-5,555
	γ	6,490	5,638	4,586	3,380	4,586	5,151
	ξ	2×10^{-4}	5×10^{-4}	1×10^{-6}	2×10^{-3}	1×10^{-6}	1×10^{-5}
	λ	2,857	3,333	8,000	2,000	8,000	6,667

O problema pode ser matematicamente formulado como um problema de otimização multi-objetivo irrestrito, conforme as Eqs.(2.2) a (2.9), como segue:

$$\text{Min} \phi(X) = f(X) + k_p P(X) \quad (5.14)$$

$$\text{Sujeito a} \begin{cases} g_1(X) = \sum_{i=1}^6 x_i - 290 \leq 0 & (\text{MW}) \\ g_2(X) = 280 - \sum_{i=1}^6 x_i \leq 0 & (\text{MW}) \\ 10 \leq x_i \leq 120, \quad i = 1, \dots, 6 & (\text{MW}) \end{cases} \quad (5.15)$$

$$\text{Onde} \begin{cases} f(X) = w_1 \frac{F_c(X)}{F_c^0} + w_2 \frac{F_e(X)}{F_e^0} \\ P(X) = \left\{ \max[0, g_1(X)] \right\}^2 + \left\{ \max[0, g_2(X)] \right\}^2 \end{cases} \quad (5.16)$$

A solução ideal foi calculada utilizando a Eq.(2.7). O melhor resultado obtido usando todas as estratégias da evolução diferencial foi $F_c^0 = 550,1248$ \$/h e $F_e^0 = 0,1952$ ton/h. Os parâmetros utilizados pela ED na otimização foram: $Np=15$; $Nger=300$; $F_p=0,8$ e $P_c=0,6$. Os parâmetros para AGs e SA foram os mesmos descritos anteriormente. Em todas as técnicas, o fator de penalidade adotado foi $k_p = 1000$.

Como este é um problema multi-objetivo, a melhor solução depende do interesse do projetista na solução compromisso. Para o caso apresentado, $w_1 = w_2 = 0,5$, as estratégias 1, 2, 5, 6, 7 e 10 podem representar uma boa solução caso se queira um compromisso razoável entre o custo do combustível e a emissão dos poluentes. Observando a Tab. 5.8,

verifica-se ainda que a estratégia 3 apresenta melhor custo e pior emissão. Vale registrar que as restrições foram obedecidas em todas as estratégias.

Tabela 5.8 - Resultados da otimização do problema DEA, com potências em MW

	ϕ	F_c (\$/h)	F_e (ton/h)	x_1	x_2	x_3	x_4	x_5	x_6
1	1,0639	608,9130	0,1993	23,51	40,28	57,83	57,98	49,76	50,55
2	1,0616	609,2373	0,1983	31,12	39,10	53,27	59,02	52,88	44,59
3	1,1130	560,0923	0,2358	33,08	59,38	43,12	45,92	54,05	69,08
4	1,0959	623,9211	0,2065	39,08	19,66	62,32	73,05	39,79	52,22
5	1,0616	609,3281	0,1982	30,45	39,74	54,11	58,50	52,72	44,46
ED 6	1,0644	608,4453	0,1997	30,75	35,57	57,45	62,62	45,37	48,50
7	1,0616	609,3180	0,1982	29,96	39,52	53,06	58,45	53,58	45,42
8	1,0958	622,2064	0,2070	56,53	36,61	28,29	71,89	51,29	35,71
9	1,0792	612,6310	0,2040	21,28	33,42	67,97	59,84	65,68	34,84
10	1,0616	609,2606	0,1983	30,29	39,46	53,66	58,80	52,33	45,45
AGs	1,1719	674,8618	0,2181	24,79	16,52	99,11	26,86	49,66	73,05
SA	1,0621	609,9034	0,1982	30,26	39,71	53,43	58,33	53,27	44,99

A Fig. 5.11 apresenta o conjunto de soluções ótimas de Pareto, utilizando a estratégia 1 da Evolução Diferencial, variando as prioridades das funções.

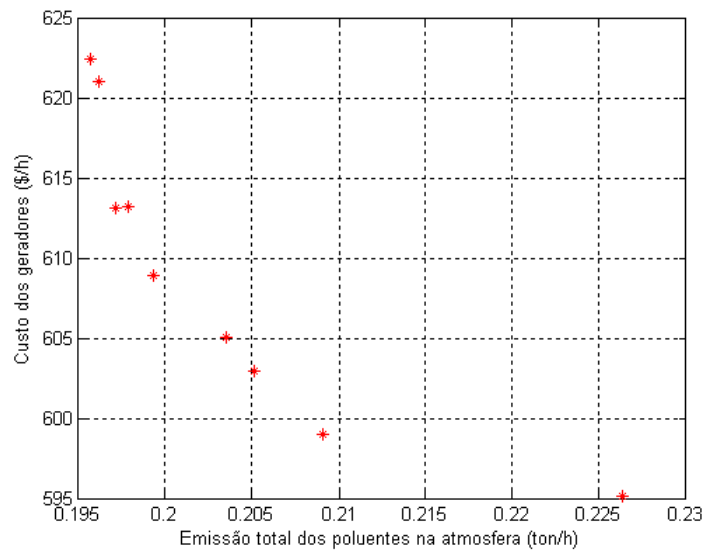


Figura 5.11 - Solução de Pareto para o problema DEA, usando a estratégia 1 da Evolução Diferencial

CAPÍTULO VI

Evolução Diferencial aplicada a sistemas robóticos

O objetivo neste capítulo é solucionar problemas de otimização com aplicações em robótica industrial utilizando as técnicas de otimização natural Algoritmos Genéticos e Evolução Diferencial. Além disso, pretende-se comparar os resultados obtidos e verificar a potencialidade da técnica Evolução Diferencial. A primeira aplicação visa a maximização do espaço de trabalho de manipuladores seriais com três juntas rotacionais (3R), conforme formulação proposta por Bergamaschi (2004). O segundo problema trata da otimização de trajetórias de um manipulador paralelo denominado CaPaMan (Cassino Parallel manipulator), utilizando a modelagem desenvolvida em Oliveira (2005b). Nos dois trabalhos utilizados foram obtidos resultados aplicando-se Algoritmos Genéticos. Assim, por se tratar de problemas bastante complexos, será possível testar a eficiência da Evolução Diferencial através da comparação das soluções ótimas.

6.1. Otimização do Espaço de Trabalho de Manipuladores 3R

Uma característica fundamental a ser considerada no projeto dimensional de robôs manipuladores é o seu volume do espaço de trabalho.

Neste trabalho, deseja-se estudar os robôs manipuladores com três juntas rotacionais (3R), cujo esquema cinemático está representado na Fig. 6.1. O efetuador é montado no final da terceira junta.

A capacidade de um robô desenvolver uma determinada tarefa depende da sua arquitetura e da dimensão de suas ligações (links), assim como da posição por ele assumida no ambiente de trabalho. Estas características devem ser consideradas no projeto dos manipuladores robóticos, ou melhor, na definição de sua geometria. No caso de manipuladores com juntas puramente rotacionais, os parâmetros de projeto são a_1 , a_2 , a_3 , d_2 , d_3 , α_1 e α_2 ; os termos θ_1 , θ_2 e θ_3 representam as variáveis cinemáticas (representados na Fig. 6.1).

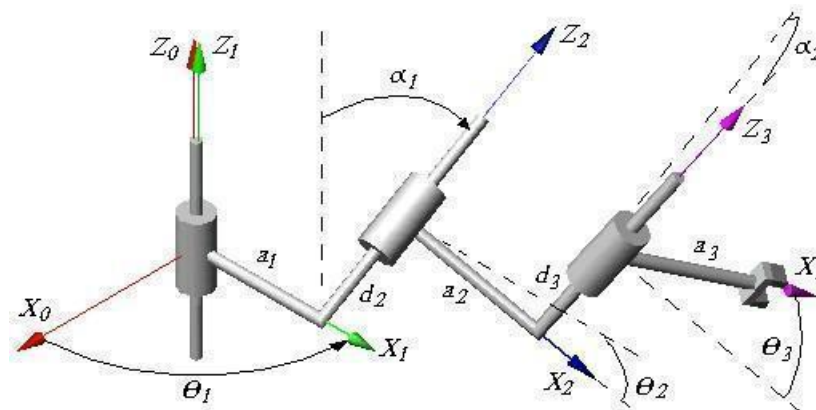


Figura 6.1 - Esquema cinemático de um robô manipulador 3R

O cálculo exato do espaço de trabalho é muito importante, pois influi no projeto dimensional, no posicionamento do manipulador no ambiente de trabalho e na sua destreza para executar tarefas.

Várias investigações têm focado as propriedades do espaço de trabalho de cadeias robóticas abertas com o propósito de enfatizar sua geometria e características cinemáticas e de criar algoritmos analíticos e procedimentos para o seu projeto. Ceccarelli (1996) apresentou uma formulação algébrica para determinar o espaço de trabalho de manipuladores de revolução. Lanni et al. (2002) investigaram e resolveram o projeto de manipuladores na forma de um problema de otimização que leva em consideração as características do seu espaço de trabalho. Eles aplicaram duas técnicas numéricas diferentes, a primeira usando a programação quadrática sequencial (SQP) e a segunda envolvendo uma técnica de busca aleatória (simulated annealing).

Algumas pesquisas têm sido realizadas visando determinar a fronteira do espaço de trabalho e detectar a presença de vazios e singularidades no espaço de trabalho. Abdel-Malek et al. (2000) propuseram uma formulação genérica para determinar vazios no espaço de trabalho de manipuladores seriais. Saramago et al. (2002) propuseram uma forma de caracterizar a fronteira do espaço de trabalho, formulando uma condição analítica geral para deduzir a existência de pontos de cúspides nas fronteiras interna e externa do espaço de trabalho. Bergamaschi et al. (2006) desenvolveram um método de determinação da fronteira do espaço de trabalho de manipuladores 3R estudando a envoltória de uma família de curvas presente neste espaço de trabalho. Ainda neste trabalho, eles aplicaram técnicas sequenciais para a solução do problema de otimização que visava a maximização do volume do espaço de trabalho sujeito a certas restrições. Este mesmo problema de otimização foi investigado por Bergamaschi et al. (2005) aplicando algoritmos genéticos.

Ceccarelli e Lanni (2004) apresentaram uma adequada formulação para o espaço de trabalho que pode ser usada no projeto de manipuladores, a qual tem sido formulada como um problema de otimização multi-objetivo usando o volume do espaço de trabalho e as dimensões do robô como funções objetivo. A otimização visa a maximização do volume e a minimização das dimensões.

Ceccarelli et al. (2005) deduziram um procedimento de projeto comum para manipuladores seriais e paralelos que considera critérios de projeto comuns, mas avaliações numéricas específicas. O problema de otimização multi-objetivo proposto por eles formula um procedimento de projeto único que leva em consideração adequados algoritmos gerais para volume do espaço de trabalho, matrizes jacobianas e deslocamentos complacentes.

Neste trabalho o problema de otimização é definido tal que a função objetivo é o volume do espaço de trabalho, sujeita a restrições que controlam o espaço de trabalho. Foram incluídas restrições laterais para obter as dimensões do manipulador dentro de valores práticos.

A determinação do volume do espaço de trabalho aqui apresentada implica no conhecimento da área de sua seção transversal em um plano radial, a qual é aqui obtida por aproximação através de sua discretização em uma malha retangular.

O problema de otimização é investigado usando as técnicas apresentadas nos capítulos III e IV. Uma aplicação numérica é apresentada, a fim de comparar a eficiência da metodologia empregada.

6.1.1. Cinemática de robôs manipuladores utilizando o método de Denavit-Hartenberg

Um dos métodos mais usados para descrever geometricamente um robô é utilizar a notação de Denavit e Hartenberg, cujo esquema é exibido na Fig. 6.2. Esta notação basicamente consiste em construir a matriz de transformação homogênea, T_{i-1}^i , que representa o sistema $X_i Y_i Z_i$, associado a i -ésima ligação do robô, em relação ao sistema $X_{i-1} Y_{i-1} Z_{i-1}$, associado a $(i-1)$ -ésima ligação, para i variando de 1 a m , onde m é o grau de liberdade do robô.

Pela Figura 6.2, pode-se observar que a representação do sistema $X_i Y_i Z_i$ em relação ao sistema $X_{i-1} Y_{i-1} Z_{i-1}$ percorre quatro etapas bem definidas, representadas pelos quatro parâmetros cinemáticos α_{i-1} , a_{i-1} , d_i e θ_i que também são conhecidos por parâmetros de Denavit-Hartenberg. Duas destas etapas são obtidas por rotação, sendo uma de um ângulo α_{i-1} em torno do eixo X_{i-1} e a outra de um ângulo θ_i em torno do eixo Z_i , e duas por translação, sendo uma de a_{i-1} unidades ao longo do eixo X_{i-1} e a outra de d_i unidades ao longo do eixo Z_i . De acordo com a ordem em que se efetua estas etapas, obtém-se a matriz

de transformação homogênea genérica de um sistema de referência em relação ao sistema precedente. Neste trabalho, a ordem destas etapas e, conseqüentemente, a matriz de transformação homogênea genérica é dada pela Eq.(6.1).

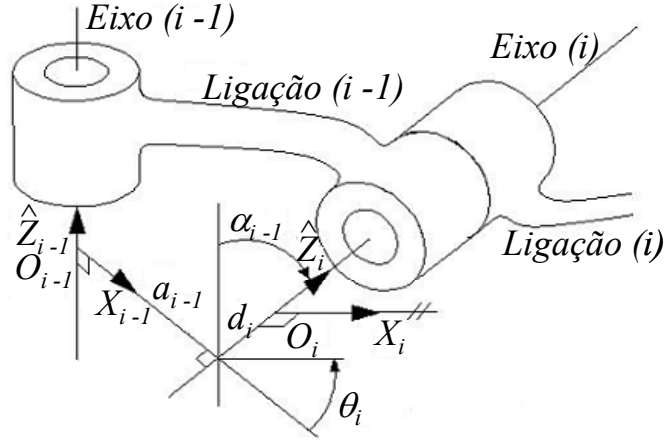


Figura 6.2 - Notação de Denavit e Hartenberg para manipuladores

$$T_{i-1}^i = Rot(\alpha_i, X_{i-1}) Trans(a_{i-1}, 0, 0) Trans(0, 0, d_i) Rot(\theta_i, Z_i) \quad (6.1)$$

$$= \begin{bmatrix} C\theta_i & -S\theta_i & 0 & a_{i-1} \\ S\theta_i C\alpha_{i-1} & C\theta_i C\alpha_{i-1} & S\alpha_{i-1} & d_i S\alpha_{i-1} \\ -S\theta_i S\alpha_{i-1} & -C\theta_i S\alpha_{i-1} & C\alpha_{i-1} & d_i C\alpha_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

A matriz de transformação homogênea que representa as coordenadas do efetuador em relação à base, T_0^m , pode ser obtida através da seguinte expressão (SARAMAGO & STEFFEN, 1999a):

$$T_0^m = T_0^1 \cdot T_1^2 \cdot T_2^3 \cdot \dots \cdot T_{m-1}^m \quad (6.2)$$

Conforme mostrado na Fig. 6.1, a cinemática do robô manipulador 3R é feita com o auxílio de quatro sistemas de referência. Nestes manipuladores, o sistema $X_1Y_1Z_1$ difere do sistema $X_0Y_0Z_0$, acoplado à base, apenas por uma rotação θ_1 em torno do eixo Z_1 . Então, a matriz de transformação homogênea do sistema $X_1Y_1Z_1$ em relação à base é determinada da seguinte forma:

$$T_0^1 = Rot(\theta_1, Z_1) = \begin{bmatrix} C\theta_1 & -S\theta_1 & 0 & 0 \\ S\theta_1 & C\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6.3)$$

Na representação do sistema $X_2Y_2Z_2$ em relação ao sistema $X_1Y_1Z_1$ e do sistema $X_3Y_3Z_3$ em relação ao sistema $X_2Y_2Z_2$, os parâmetros de Denavit-Hartenberg podem ser todos diferentes de zero. Portanto, as respectivas matrizes de transformação homogênea são dadas por:

$$T_1^2 = \begin{bmatrix} C\theta_2 & -S\theta_2 & 0 & a_1 \\ S\theta_2 C\alpha_1 & C\theta_2 C\alpha_1 & S\alpha_1 & d_2 S\alpha_1 \\ -S\theta_2 S\alpha_1 & -C\theta_2 S\alpha_1 & C\alpha_1 & d_2 C\alpha_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6.4)$$

$$T_2^3 = \begin{bmatrix} C\theta_3 & -S\theta_3 & 0 & a_2 \\ S\theta_3 C\alpha_2 & C\theta_3 C\alpha_2 & S\alpha_2 & d_3 S\alpha_2 \\ -S\theta_3 S\alpha_2 & -C\theta_3 S\alpha_2 & C\alpha_2 & d_3 C\alpha_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6.5)$$

Considere agora, um ponto H no sistema $X_3Y_3Z_3$, que pode ser escolhido como o centro do efetuador. Por se tratar do sistema $X_3Y_3Z_3$, este ponto será denotado por H_3 e representado por:

$$H_3 = \begin{bmatrix} a_3 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (6.6)$$

Note que a_3 deve ser diferente de zero. Caso contrário, H_3 seria a origem do sistema $X_3Y_3Z_3$, não sofrendo assim nenhum movimento decorrente da terceira junta, o que não é de interesse prático.

Utilizando a Eq.(6.2), a representação vetorial de H_3 em relação à base, denotada por H_0 , é obtida de acordo com a Eq.(6.7).

$$H_0 = T_0^1 T_1^2 T_2^3 H_3 = T_0^1 H_1 \quad (6.7)$$

Expandindo a Eq.(6.7), pode-se obter:

$$H_2 = \begin{bmatrix} H_2^x \\ H_2^y \\ H_2^z \\ 1 \end{bmatrix} = T_2^3 H_3 = \begin{bmatrix} a_3 C\theta_3 + a_2 \\ a_3 S\theta_3 C\alpha_2 + d_3 S\alpha_2 \\ -a_3 S\theta_3 S\alpha_2 + d_3 C\alpha_2 \\ 1 \end{bmatrix} \quad (6.8)$$

$$H_1 = \begin{bmatrix} H_1^x \\ H_1^y \\ H_1^z \\ 1 \end{bmatrix} = T_1^2 H_2 = \begin{bmatrix} H_2^x C\theta_2 - H_2^y S\theta_2 + a_1 \\ H_2^x S\theta_2 C\alpha_2 + H_2^y C\theta_2 C\alpha_1 + H_2^z S\alpha_1 + d_2 S\alpha_1 \\ -H_2^x S\theta_2 S\alpha_1 - H_2^y C\theta_2 S\alpha_1 + H_2^z C\alpha_1 + d_2 C\alpha_1 \\ 1 \end{bmatrix} \quad (6.9)$$

e, por fim

$$H_0 = T_0^1 H_1 = \begin{bmatrix} H_1^x C\theta_1 - H_1^y S\theta_1 \\ H_1^x S\theta_1 + H_1^y C\theta_1 \\ H_1^z \\ 1 \end{bmatrix} \quad (6.10)$$

6.1.2. Espaço de Trabalho de Manipuladores 3R

O espaço de trabalho $W(H)$ de um ponto H situado na extremidade do robô manipulador é o conjunto de todos os pontos que H ocupa quando as variáveis de junta são variadas em todo os seus intervalos de definição (GUPTA & ROTH, 1982) e é representado na Fig. 6.3 (a).

O procedimento mais imediato para investigar o espaço de trabalho é variar os ângulos θ_1 , θ_2 e θ_3 sobre seus intervalos de definição e estimar as coordenadas do ponto H com relação à base do manipulador. Desta maneira, obtém-se a posição do órgão terminal, tendo como resultado a representação vetorial dada por H_0 na Eq.(6.10).

A Fig. 6.3 (a) exhibe o espaço de trabalho seccionado de um determinado manipulador 3R. Como pode ser visto, o espaço de trabalho deste tipo de robô é um sólido de revolução, tendo Z como o seu eixo de revolução. Desta forma, é natural imaginar que o espaço de trabalho é o resultado da rotação, em torno do eixo Z , de uma seção radial plana que funcione como uma seção geratriz. A Fig. 6.3 (b) esboça a seção radial relativa ao espaço de trabalho exibido na Fig. 6.3 (a).

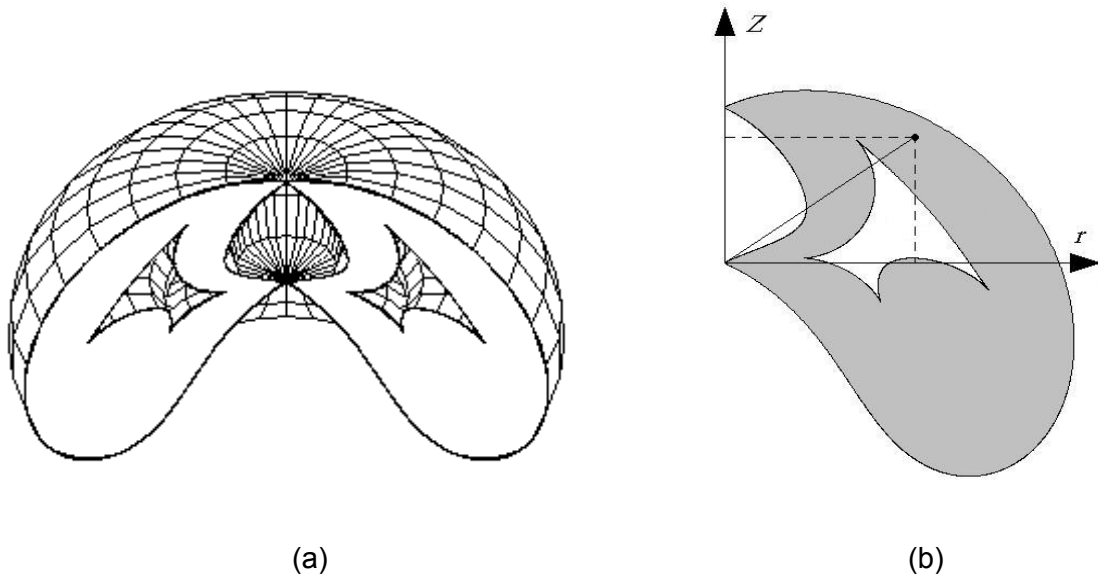


Figura 6.3 - (a) Espaço de trabalho de um robô 3R; (b) seção radial plana

Então, o espaço de trabalho de robôs com estrutura 3R pode ser obtido por intermédio da extensão radial r e da extensão axial z com relação à base (CECCARELLI, 1996; CECARELLI & LANNI, 1999). Para esta representação, r é a distância de um ponto genérico do espaço de trabalho ao eixo Z_1 e z é a distância desse mesmo ponto ao plano X_1Y_1 (Fig. 6.1). Assim, usando a Eq.(6.10), as equações paramétricas do lugar geométrico descrito pelo ponto H sobre um plano radial são:

$$\begin{cases} r^2 = (H_1^x)^2 + (H_1^y)^2 \\ z = H_1^z \end{cases} \quad (6.11)$$

Para exemplificar, considere uma configuração particular, cujos parâmetros de projeto são: $a_1=3,0$; $a_2=1,0$; $a_3=0,5$; $d_2=1,0$; $d_3=1,0$; $\alpha_1=60^\circ$ e $\alpha_2=60^\circ$. Utilizando a Eq.(6.11), obtém-se o conjunto dos pontos (r, z) , que formam a família de curvas que está contida na seção radial do espaço de trabalho deste robô, conforme mostrado na Fig. 6.4.

O volume do espaço de trabalho, V , é o volume do sólido de revolução obtido pela rotação da seção radial em torno do eixo Z_1 . Assim, usando o Teorema de Pappus-Guldin [BEER & JOHNSTON, 1991], em acordo com o esquema mostrado na Fig. 6.5, o volume é dado através da equação:

$$V = 2 \pi r_g A, \quad (6.12)$$

onde A é a área da seção radial plana que é coberta pela família de curvas.

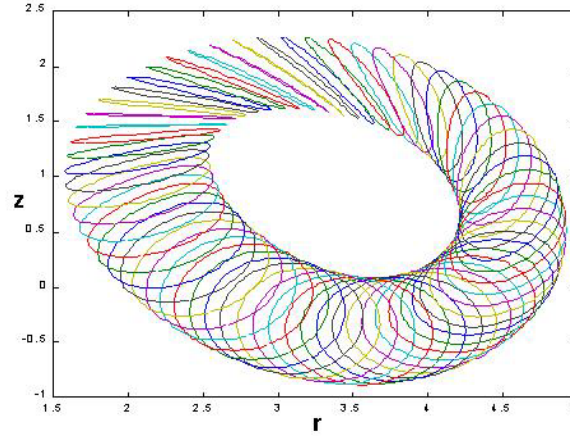


Figura 6.4 - Família de curvas contida na seção radial do espaço de trabalho de um particular manipulador 3R

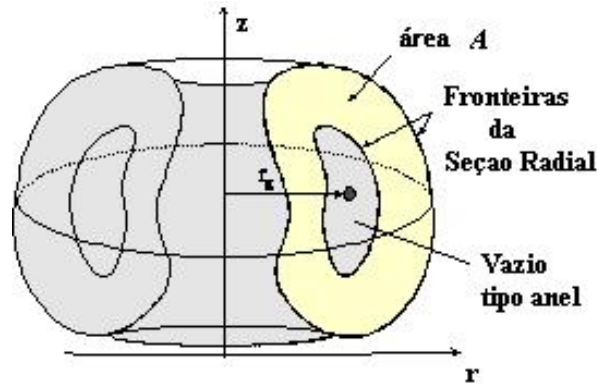


Figura 6.5 - Cálculo do volume do espaço de trabalho de manipuladores 3R

Esta pesquisa propõe uma formulação numérica para aproximar o cálculo da área da seção radial, através de sua discretização em uma malha retangular (OLIVEIRA et al., 2006a). Inicialmente, deve-se obter os valores extremos dos vetores r e z , ou seja,

$$\begin{aligned} r_{\min} &= \min \{r\} & \text{e} & & r_{\max} &= \max \{r\} \\ z_{\min} &= \min \{z\} & \text{e} & & z_{\max} &= \max \{z\} \end{aligned} \quad (6.13)$$

Adotando-se o número de sub-intervalos desejados para a discretização ao longo de r e z (n_r e n_z), pode-se calcular as dimensões das áreas elementares da malha:

$$\Delta r = \frac{r_{\max} - r_{\min}}{n_r} \quad \text{e} \quad \Delta z = \frac{z_{\max} - z_{\min}}{n_z} \quad (6.14)$$

A Eq.(6.11) permite calcular todos os pontos da família de curvas que compõem a

seção radial do espaço de trabalho. Dado um determinado ponto (r, z) , determina-se sua posição dentro da malha de discretização, através do seguinte controle de índices:

$$i = \text{int} \left(\frac{r - r_{\min}}{\Delta r} \right) + 1 \quad \text{e} \quad j = \text{int} \left(\frac{z - z_{\min}}{\Delta z} \right) + 1 \quad (6.15)$$

Conforme mostrado no esquema da Fig. 6.6, o ponto da malha que pertence ao espaço de trabalho será identificado como $P_{ij} = 1$, do contrário terá valor nulo, ou seja:

$$P_{ij} = \begin{cases} 0 & \text{se } P_{ij} \notin W(H) \\ 1 & \text{se } P_{ij} \in W(H) \end{cases} \quad (6.16)$$

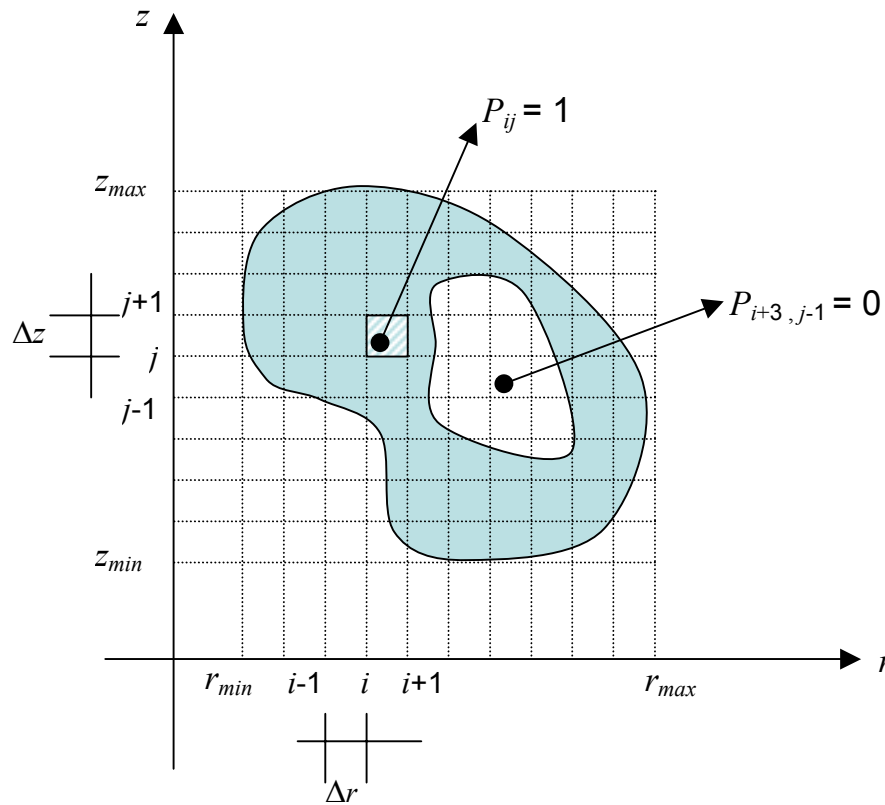


Figura 6.6 - Discretização da seção radial usando malha retangular

Desta forma, a área total será obtida pela soma de todas as áreas elementares da malha que estão contidas, totalmente ou parcialmente, na seção radial. Na Eq.(6.17), observa-se que apenas os pontos pertencentes ao espaço de trabalho contribuem para o cálculo da área:

$$A = \sum_{i=1}^{i_{\max}} \sum_{j=1}^{j_{\max}} (P_{ij} \Delta r \Delta z) \quad (6.17)$$

A coordenada do baricentro é calculada considerando a soma dos baricentros de cada área elementar, dividida pela área total, ou seja:

$$r_g = \frac{\sum_{i=1}^{i_{\max}} \sum_{j=1}^{j_{\max}} (P_{ij} \Delta r \Delta z) \left((i-1) \Delta r + \frac{\Delta r}{2} + r_{\min} \right)}{A} \quad (6.18)$$

Finalmente, conhecendo-se os valores da área e do baricentro da seção radial, dados pelas Eqs.(6.17) e (6.18), pode-se calcular o volume do espaço de trabalho do manipulador usando a Eq.(6.12).

6.1.3. Formulação do Problema Ótimo

O trabalho proposto consiste na síntese dimensional de um manipulador 3R. Assim, o problema de otimização é formulado visando a maximização do volume do espaço de trabalho, V . O problema de otimização é, então, definido como:

$$\text{Maximizar } f(X) = V$$

$$\text{sujeito a } x_i^{\inf} \leq x_i \leq x_i^{\sup} \quad (6.19)$$

O vetor das variáveis de projeto é dado por $X = (a_1, a_2, a_3, d_2, d_3, \alpha_1, \alpha_2)^T$, $x^{\inf}$ e $x^{\sup}$ representam os limites inferiores e superiores, respectivamente.

6.1.4. Simulações Numéricas

A maximização do volume do espaço de trabalho foi desenvolvida segundo o problema de otimização definido na Eq.(6.19), cujos resultados obtidos podem ser observados na Tab. 6.1. O espaço de busca foi limitado em:

$$0 \leq a_i \leq 3 \text{ [u.m.]} \quad \text{para } i = 1, 2, 3$$

$$0 \leq d_j \leq 3 \text{ [u.m.]} \quad \text{para } j = 2, 3$$

$$0^\circ \leq \alpha_k \leq 90^\circ, \quad \text{para } k = 1, 2$$

A família de curvas (por exemplo, conforme mostrado na Fig. 6.4) é obtida utilizando-se as Eqs.(6.7) a (6.11). Para a discretização da malha retangular deve-se variar os ângulos θ_1 e θ_2 . Foram adotados $n_r = n_z = 200$ subintervalos e discretização dos ângulos para gerar as curvas de $\Delta\theta_1 = \Delta\theta_2 = 0,025$ rd.

Assim, a função objetivo é definida pelo volume do espaço de trabalho que é calculado usando a Eq.(6.12), sendo a área de seção transversal obtida aplicando-se a Eq.(6.17) e o baricentro da seção radial dado pela Eq.(6.18).

O algoritmo de Evolução Diferencial foi modificado para a maximização da função objetivo e os parâmetros utilizados foram: número de indivíduos da população $Np=15$, $N_{ger}=100$, $F_p=0,8$ e $P_c=0,6$. Os Algoritmos Genéticos foram aplicados adotando-se o número de indivíduos da população $Np=200$ e $N_{ger}=100$.

Tabela 6.1 - Maximização do volume do espaço de trabalho dos robôs manipuladores 3R

ED	Volume Máximo [u.v.]	a_1 [u.m.]	a_2 [u.m.]	a_3 [u.m.]	d_2 [u.m.]	d_3 [u.m.]	α_1 [graus]	α_2 [graus]	Tempo Computacional [min]
1	3300,68	3,00	3,00	3,00	3,00	3,00	83,65	75,16	61
2	3299,84	3,00	3,00	3,00	3,00	3,00	83,57	75,08	59
3	3298,16	3,00	3,00	3,00	3,00	3,00	83,02	76,70	64
4	3298,54	3,00	3,00	3,00	3,00	3,00	84,17	77,28	59
5	3276,21	2,99	3,00	2,99	2,99	2,99	89,72	77,68	62
6	3300,18	3,00	3,00	3,00	3,00	3,00	83,13	77,31	66
7	3302,23	3,00	3,00	3,00	3,00	3,00	84,61	75,92	64
8	3298,52	3,00	3,00	3,00	2,99	3,00	83,69	75,96	66
9	3299,65	3,00	3,00	3,00	3,00	3,00	84,14	77,17	64
10	3300,45	2,99	2,99	3,00	2,99	2,99	84,61	75,92	61
AG	3298,40	2,99	2,99	3,00	2,99	2,99	83,47	75,19	124

Apesar de tratar-se de um problema complexo, todas as estratégias da Evolução Diferencial apresentaram bons resultados. Na Tab. 6.1 observa-se que o volume ótimo obtido foi semelhante em todas as técnicas. É importante observar que o tempo computacional gasto pela Evolução Diferencial é bem inferior ao tempo gasto pelos Algoritmos Genéticos, como pode ser visto na Tab. 6.1. Este resultado era esperado uma vez que a Evolução Diferencial necessita de um número pequeno de indivíduos. A seção transversal do espaço de trabalho ótimo é apresentada na Fig. 6.7, sendo sua forma

fortemente dependente essencialmente dos ângulos α_1 e α_2 , uma vez que as variáveis dimensionais a_i e d_j convergem para os limites máximos permitidos.

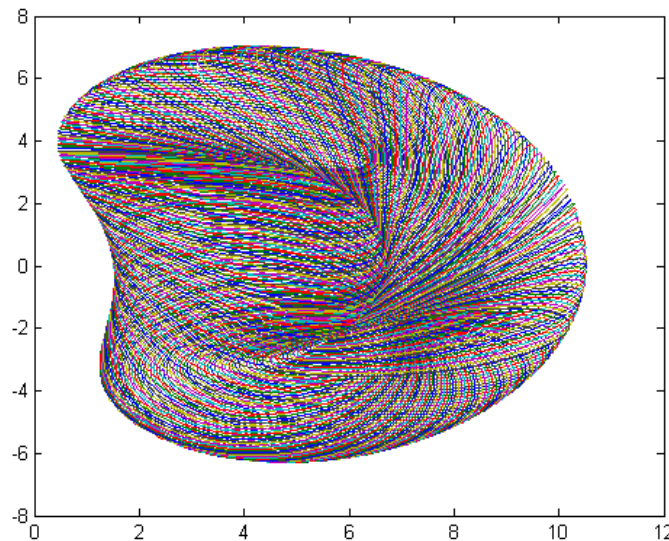


Figura 6.7 - Seção transversal do espaço de trabalho ótimo de um manipulador 3R de acordo com as técnicas utilizadas

A Evolução Diferencial exige um esforço computacional pequeno quando comparado aos Algoritmos Genéticos, mostrando ser uma técnica de grande potencial para ser aplicada a problemas complexos.

6.2. Otimização de Trajetórias do Manipulador CaPaMan

6.2.1. Descrição do CaPaMan

O objetivo desta aplicação é a obtenção de trajetórias ótimas de uma estrutura paralela denominada CaPaMan (Cassino Parallel manipulator). O problema de otimização multi-objetivo é escrito considerando a minimização da energia gasta pelos atuadores, o tempo total de percurso e a variação das acelerações (jerk). As trajetórias são modeladas utilizando-se curvas paramétricas B-splines (FOLEY et al., 1990), em que seus pontos de controle são adotados como variáveis de projeto. No processo de otimização, um código geral de análise foi acoplado ao programa de otimização. Este código de análise permite calcular os modelos: geométrico, cinemático e dinâmico, resultando no cálculo de energia para cada trajetória do manipulador. Dados os pontos inicial e final do centro da plataforma móvel, o programa de otimização gera trajetórias aleatórias que obedecem a estes pontos, as quais são submetidas ao código de análise para calcular o valor da função multi-objetivo.

Os resultados obtidos com a utilização de Evolução Diferencial são comparados com os resultantes da otimização aplicando algoritmos genéticos.

O CaPaMan é um manipulador paralelo simétrico com três graus de liberdade, composto de uma plataforma fixa, PF, e uma móvel, PM, que são conectadas por três pernas. Cada uma das pernas é presa à PF através de um mecanismo articulado de quatro barras, os quais se mantêm sempre na vertical em relação à PF e possuem juntas rotacionais. Os centros das bases destes mecanismos estão dispostos nos vértices de um triângulo equilátero na PF, de modo que os planos que os contém formam entre si ângulos de 120° , atribuindo desta forma propriedades de simetria ao manipulador. Conforme pode ser observado na Fig. 6.8, para $i=1,2,3$, as barras de ligações (conectoras) entre os mecanismos de quatro barras h_i e a PM contém duas juntas; uma esférica conectando a extremidade superior da barra à PM nos respectivos pontos H_i e a outra prismática, a qual é fixada no ponto médio e perpendicular à biela do mecanismo de quatro barras. S_i é a variável deslocamento da junta prismática; δ_i ângulos de rotações estruturais entre OX e OX_i bem como entre PX_P e PH_i ; α_i variáveis cinemáticas: ângulos de entrada formados entre as bases e manivelas de entrada dos mecanismos de acionamento (pernas).

Considere conhecidos todos os parâmetros dimensionais do CaPaMan, que conforme as Figs. 6.8 (a) e (b) são: a_i é o comprimento da base do mecanismo de acionamento (perna); m_i é o comprimento da manivela de entrada; c_i é comprimento da biela; d_i é comprimento da manivela de saída; h_i comprimento da barra conectora; r_P raio da PM dado pela distância do centro P da PM às juntas esféricas H_i ; r_B raio da PF dado pela distância do seu centro O aos pontos médios O_i das bases a_i .

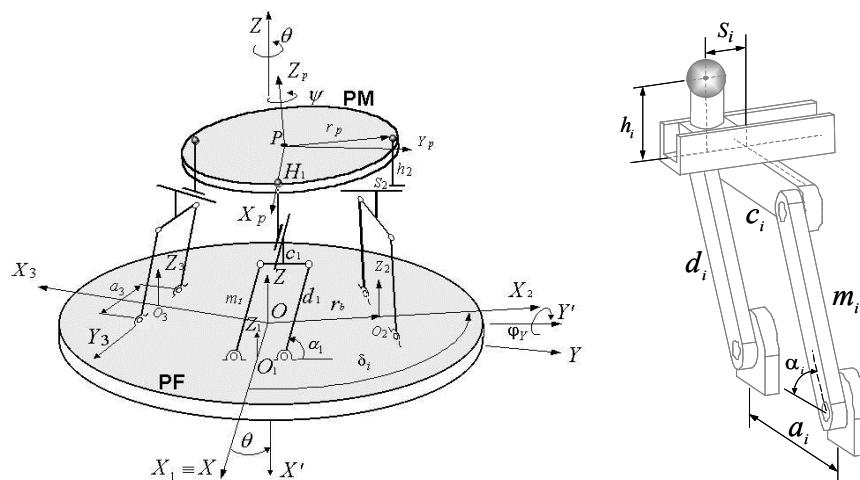


Figura 6.8 - (a) Arquitetura do CaPaMan; (b) Mecanismo de acionamento (perna i)

6.2.2. Formulação da Trajetória

Como as variáveis cinemáticas α_i são os ângulos de entrada dos mecanismos de acionamento, cada trajetória $\alpha_i(t)$ será modelada por uma B-spline uniforme cúbica, utilizando funções de concordância, sendo escrita como:

$$\alpha_i(t) = \sum_{k=0}^{n_p} p_k^i B_{k,d}^i(t) \quad t_o \leq t \leq t_f \quad n_p \geq 3, \quad i=1,2,3 \quad (6.20)$$

sendo que p_k^i são n_p+1 pontos de controle correspondente de cada trajetória $\alpha_i(t)$, e $B_{k,d}$ são polinômios definidos pelas fórmulas de recorrência de Cox de Boor, (FOLEY et al.,1990), para a spline cúbica ($d=3$) e tem-se:

$$B_{k,l}(t) = \begin{cases} 1 & \text{se } t_k \leq t \leq t_{k+1} \\ 0 & \text{fora} \end{cases} \quad (6.21)$$

$$B_{k,d}(t) = \frac{t - t_k}{t_{k+d-1} - t_k} B_{k,d-1}(t) + \frac{t_{k+d} - t}{t_{k+d} - t_{k+1}} B_{k+1,d-1}(t) \quad (6.22)$$

Cada função de concordância é definida sobre d subintervalos do intervalo total. O conjunto dos pontos extremos dos subintervalos t_j é chamado vetor-nó.

Visto que $\alpha_i(t)$ é constituída de polinômios, a sua j -ésima derivada em relação a t pode ser obtida diretamente como:

$$\frac{d^j \alpha_i(t)}{dt^j} = \sum_{k=0}^{n_p} p_k^i \frac{d^j B_{k,d}^i}{dt^j} \quad (6.23)$$

Assim, as derivadas temporais de primeira e segunda ordem, são dadas por:

$$\dot{\alpha}_i(t) = \sum_{k=0}^{n_p} p_k^i \dot{B}_{k,d}^i(t) \quad i=1,2,3 \quad (6.24)$$

e

$$\ddot{\alpha}_i(t) = \sum_{k=0}^{n_p} p_k^i \ddot{B}_{k,d}^i(t) \quad i=1,2,3 \quad (6.25)$$

6.2.3. Posição das juntas esféricas e suas derivadas temporais

Como os planos que contém os mecanismos de quatro barras formam entre si ângulos de 120° , considere um sistema cartesiano de referência $O_i X_i Y_i Z_i$ para $i=1,2,3$ tomado de maneira que O_i coincida com o centro da base a_i do mecanismo de quatro barras. O eixo X_i é perpendicular ao plano do mecanismo de quatro barras, o eixo Y_i coincide com a base do mecanismo e Z_i é tomado de modo que o sistema de referência $O_i X_i Y_i Z_i$ seja cartesiano. Desse modo cada eixo X_i é girado de 120° em relação ao X_i imediatamente anterior. Conforme pode ser observado na Fig. 6.9, os vetores v_i são tomados de modo que $|v_1| = |v_2| = |v_3| = r_B$ (raio da PF). Considerando $m_i = d_i$ e $\delta_1 = 0^\circ$; $\delta_2 = 120^\circ$ e $\delta_3 = 240^\circ$ pode-se escrever:

$$\begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} = \begin{pmatrix} S_i \\ m_i \cos \alpha_i \\ m_i \sin \alpha_i + h_i \end{pmatrix} \quad (6.26)$$

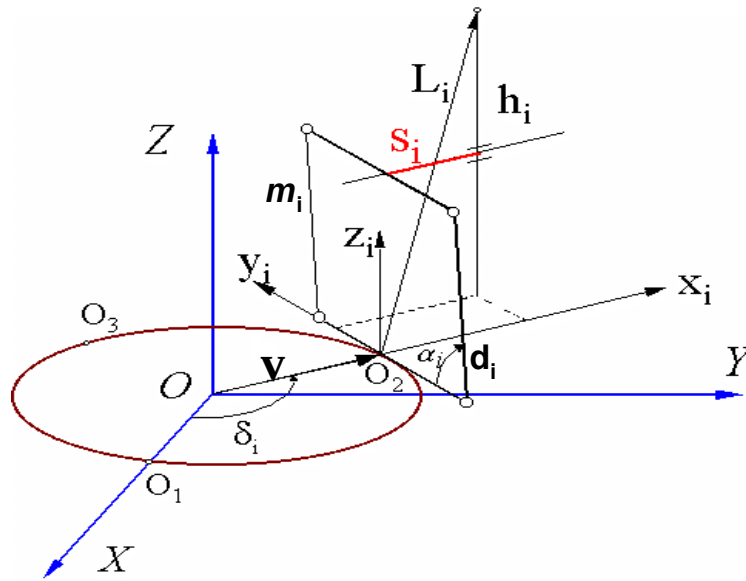


Figura 6.9 - Parâmetros associados à perna i

Assim, as expressões da primeira e segunda derivadas das coordenadas y_i e z_i , usando as Eqs. (6.26), são dadas respectivamente por:

$$\dot{y}_i = -m_i \dot{\alpha}_i \sin \alpha_i \quad (6.27)$$

$$\dot{z}_i = m_i \dot{\alpha}_i \cos \alpha_i \quad (6.28)$$

$$\ddot{y}_i = -m_i \ddot{\alpha}_i \sin \alpha_i - m_i \dot{\alpha}_i^2 \cos \alpha_i \quad (6.29)$$

$$\ddot{z}_i = m_i \ddot{\alpha}_i \cos \alpha_i - m_i \dot{\alpha}_i^2 \sin \alpha_i \quad (6.30)$$

Cabe ressaltar que devido a forma geométrica do manipulador não é necessário calcular as derivadas em relação à coordenada x .

6.2.4. Determinação dos ângulos de Euler

Para descrever o comportamento cinemático e dinâmico do CaPaMan, consideram-se dois sistemas de coordenadas. Um sistema inercial $OXYZ$ é fixado à PF, cuja origem O é o centro da PF e o outro $PX_PY_PZ_P$ é fixado à PM sendo P o centro da PM. O sistema móvel $PX_PY_PZ_P$ é fixado de modo que o eixo X_P seja coincidente com a linha unindo os pontos P e H_I , o eixo Z_P é perpendicular à PM e o eixo Y_P é colocado sobre a PM de maneira que o referencial móvel seja ortogonal.

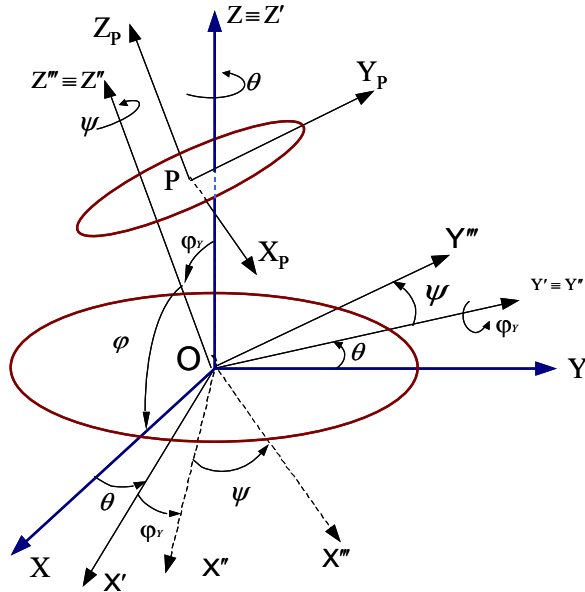


Figura 6.10 - Orientação da plataforma móvel em função dos ângulos de rotação θ , φ e ψ

A orientação do sistema móvel $PX_PY_PZ_P$ fixo à PM é descrita em relação ao sistema inercial $PXYZ$ fixo à PF, pelos ângulos de Euler θ , φ e ψ , conforme representado na Fig. 6.10. Esta figura mostra a plataforma móvel após três rotações, de acordo com a regra da mão direita, na seguinte seqüência: a primeira rotação, de um ângulo θ , é realizada em torno do eixo Z ; a segunda rotação φ_y é feita em torno do eixo Y' ; a terceira rotação ψ , é

realizada em torno do eixo Z'' , obtendo-se finalmente o sistema $OX'''Y'''Z'''$. Fazendo-se a translação do sistema $OX'''Y'''Z'''$ com centro em O , para o centro da plataforma móvel, no ponto P , obtém-se o sistema $PX_pY_pZ_p$.

Oliveira (2005b) mostrou que os ângulos de orientação θ , φ e ψ podem ser obtidos em função das coordenadas dos pontos H_1 , H_2 e H_3 , correspondentes à posição das juntas esféricas como:

$$\begin{cases} \psi = \operatorname{tg}^{-1} \left(\sqrt{3} \frac{z_3 - z_2}{2z_1 - z_2 - z_3} \right) \\ \varphi = \cos^{-1} \left(\pm \frac{2}{3r_p} \sqrt{z_1^2 + z_2^2 + z_3^2 - z_1z_2 - z_1z_3 - z_2z_3} \right) \\ \theta = \sin^{-1} \left[\frac{2(y_1 + y_2 + y_3)}{3r_p(1 + \sin \varphi)} \right] - \psi \end{cases} \quad (6.31)$$

As coordenadas y_i e z_i necessárias na equação anterior são dadas pela Eq. (6.26). A ambigüidade existente na Eq.(6.31) que define o ângulo φ devido ao sinal $\pm$, pode ser resolvida considerando a geometria e o modelo do CaPaMan, se $z \geq z_l$ usa-se o sinal (+) em φ , caso contrário usa-se (-). Observe que z é a terceira componente do ponto P (centro da plataforma móvel).

6.2.5. Determinação das derivadas temporais dos ângulos de Euler

As velocidades e acelerações angulares dependem das derivadas dos ângulos de Euler θ , φ e ψ . Para simplificar as expressões das derivadas, foi utilizada a notação V_i que são apresentadas no Anexo I. Assim, derivando a Eq.(6.31) resulta:

$$\dot{\psi} = \frac{\sqrt{3}}{2V_1} \left[(z_2 - z_3)\dot{z}_1 - (z_1 - z_3)\dot{z}_2 + (z_1 - z_2)\dot{z}_3 \right] \quad (6.32)$$

$$\dot{\varphi} = \pm \frac{1}{\sqrt{(9r_p^2 - 4V_1)V_1}} \left[(2z_1 - z_2 - z_3)\dot{z}_1 + (-z_1 + 2z_2 - z_3)\dot{z}_2 + (-z_1 - z_2 + 2z_3)\dot{z}_3 \right] \quad (6.33)$$

$$\dot{\theta} = \frac{2}{(1 + \sin \varphi)\sqrt{V_2}} \left[(1 + \sin \varphi)(\dot{y}_1 + \dot{y}_2 + \dot{y}_3) - (y_1 + y_2 + y_3)\dot{\varphi} \cos \varphi \right] - \dot{\psi} \quad (6.34)$$

As derivadas de segunda ordem são apresentadas pelas Eqs.(6.36) a (6.37).

$$\ddot{\psi} = \frac{\sqrt{3}}{2V_1} \left[(z_2 - z_3)\ddot{z}_1 - (z_1 - z_3)\ddot{z}_2 + (z_1 - z_2)\ddot{z}_3 \right] +$$

$$\frac{\sqrt{3}}{2V_1^2} \left[- (z_2 - z_3)(2z_1 - z_2 - z_3)\dot{z}_1^2 + \right.$$

$$\left. + (z_1 - z_3)(-z_1 + 2z_2 - z_3)\dot{z}_2^2 - (z_1 - z_2)(-z_1 - z_2 + 2z_3)\dot{z}_3^2 \right] +$$

$$+ \frac{\sqrt{3}}{V_1^2} \left[- (z_1 - z_2)(-z_1 - z_2 + 2z_3)\dot{z}_1\dot{z}_2 + (z_1 - z_3)(-z_1 + 2z_2 - z_3)\dot{z}_1\dot{z}_3 - \right.$$

$$\left. - (z_2 - z_3)(2z_1 - z_2 - z_3)\dot{z}_2\dot{z}_3 \right]$$
(6.35)

$$\ddot{\varphi} = \pm \left\{ \frac{1}{\sqrt{(9r_p^2 - 4V_1)V_1}} \left[(2z_1 - z_2 - z_3)\ddot{z}_1 + (-z_1 + 2z_2 - z_3)\ddot{z}_2 + (-z_1 - z_2 + 2z_3)\ddot{z}_3 \right] + \right.$$

$$+ \frac{1}{2\sqrt{[(9r_p^2 - 4V_1)V_1]^3}} \left(V_3\dot{z}_1^2 + V_4\dot{z}_2^2 + V_5\dot{z}_3^2 \right) +$$

$$\left. \frac{1}{\sqrt{[(9r_p^2 - 4V_1)V_1]^3}} \left[(V_5 - 27r_p^2 V_1)\dot{z}_1\dot{z}_2 + (V_4 - 27r_p^2 V_1)\dot{z}_1\dot{z}_3 + (V_3 - 27r_p^2 V_1)\dot{z}_2\dot{z}_3 \right] \right\}$$
(6.36)

$$\ddot{\theta} = \frac{2}{\sqrt{V_2}} \left(\ddot{y}_1 + \ddot{y}_2 + \ddot{y}_3 \right) - \frac{2(y_1 + y_2 + y_3)\cos\varphi}{(1 + \sin\varphi)\sqrt{V_2}} \left(\ddot{\varphi} \right) + \frac{8(y_1 + y_2 + y_3)}{\sqrt{V_2^3}} \left(\dot{y}_1^2 + \dot{y}_2^2 + \dot{y}_3^2 \right) +$$

$$+ \frac{(y_1 + y_2 + y_3)[18r_p^2(1 + \sin\varphi)\cos^2\varphi + 2V_2]}{(1 + \sin\varphi)\sqrt{V_2^3}} \dot{\varphi}^2 + \frac{16(y_1 + y_2 + y_3)}{\sqrt{V_2^3}} \left(\dot{y}_1\dot{y}_2 + \dot{y}_1\dot{y}_3 + \dot{y}_2\dot{y}_3 \right) -$$

$$- \frac{36r_p^2(1 + \sin\varphi)^2\cos\varphi}{(1 + \sin\varphi)\sqrt{V_2^3}} \left(\dot{y}_1^2 + \dot{y}_2^2 + \dot{y}_3^2 \right) \dot{\varphi} - \ddot{\psi}$$
(6.37)

6.2.6. Matriz de transformação

Fazendo $R = [R(\theta, z)][R(\varphi_y, y)][R(\psi, z)]$ e efetuando o produto, obtém-se a matriz de transformação em função dos ângulos de Euler. Além disso, pode-se observar na Fig. 6.10 que $\varphi_y = \frac{\pi}{2} - \varphi$. Após efetuar o produto das matrizes de transformação, obtém-se:

$$R = \begin{pmatrix} \cos\theta \sin\varphi \cos\psi - \sin\theta \sin\psi & -\cos\theta \sin\varphi \sin\psi - \sin\theta \cos\psi & \cos\theta \cos\varphi \\ \sin\theta \sin\varphi \cos\psi + \cos\theta \sin\psi & -\sin\theta \sin\varphi \sin\psi + \cos\theta \cos\psi & \sin\theta \cos\varphi \\ -\cos\varphi \cos\psi & \cos\varphi \sin\psi & \sin\varphi \end{pmatrix}$$
(6.38)

6.2.7. Coordenadas do ponto central da plataforma móvel

Na Figura 6.10, considere X_p o vetor das coordenadas do ponto P em relação ao sistema $OX'''Y'''Z'''$ e X o vetor das coordenadas de P em relação ao sistema $OXYZ$. Este vetor pode ser escrito como:

$$X = [R(\theta, z)][R(\varphi_y, y)][R(\psi, z)]X^p \quad (6.39)$$

Usando as relações geométricas e a matriz dada pela Eq.(6.38), Oliveira (2005b) demonstrou que:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} r_B - r_P R_{11} + x_1 \\ -r_P R_{21} + y_1 \\ -r_P R_{31} + z_1 \end{pmatrix} \quad (6.40)$$

Após manipulações matemáticas, as coordenadas do ponto central da plataforma móvel em relação ao sistema inercial podem ser escritas em função das coordenadas das juntas esféricas H_1 , H_2 e H_3 e também dos ângulos de rotação θ , φ e ψ :

$$\begin{cases} x = \frac{y_3 - y_2}{\sqrt{3}} - \frac{r_P}{2}(1 - \sin \varphi) \cos(\psi - \theta) \\ y = y_1 - r_P(\cos \theta \sin \psi + \sin \theta \sin \varphi \cos \psi) \\ z = r_P \cos \varphi \cos \psi + z_1 \end{cases} \quad (6.41)$$

Portanto a plataforma móvel PM tem posição dada pelas Eqs.(6.41) e orientação definida pelas Eqs.(6.31).

6.2.8. Derivadas temporais das coordenadas do ponto central

As componentes de velocidade e aceleração do ponto P são obtidas usando as Eqs.(6.31) e (6.41) respectivamente. Assim, as derivadas de primeira e segunda ordem são dadas pelas equações matriciais Eq.(6.42) e Eq.(6.43), respectivamente.

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} = \begin{pmatrix} 0 & -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \dot{y}_1 \\ \dot{y}_2 \\ \dot{y}_3 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{pmatrix} \begin{pmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \end{pmatrix} +$$

$$+ r_p \begin{pmatrix} -\frac{(1 - \sin \varphi) \sin(\psi - \theta)}{2} & \frac{\cos \varphi \cos(\psi - \theta)}{2} & \frac{(1 - \sin \varphi) \sin(\psi - \theta)}{2} \\ -(\sin \varphi \cos \theta \cos \psi - \sin \theta \sin \psi) & -\cos \varphi \sin \theta \cos \psi & \sin \varphi \sin \theta \sin \psi - \cos \theta \cos \psi \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \dot{\theta} \\ \dot{\varphi} \\ \dot{\psi} \end{pmatrix} \quad (6.42)$$

$$\begin{pmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{pmatrix} = \begin{pmatrix} 0 & -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \ddot{y}_1 \\ \ddot{y}_2 \\ \ddot{y}_3 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{pmatrix} \begin{pmatrix} \ddot{z}_1 \\ \ddot{z}_2 \\ \ddot{z}_3 \end{pmatrix} +$$

$$r_p \begin{pmatrix} -\frac{(1 - \sin \varphi) \sin(\psi - \theta)}{2} & \frac{\cos \varphi \cos(\psi - \theta)}{2} & \frac{(1 - \sin \varphi) \sin(\psi - \theta)}{2} \\ -(\sin \varphi \cos \theta \cos \psi - \sin \theta \sin \psi) & -\cos \varphi \sin \theta \cos \psi & \sin \varphi \sin \theta \sin \psi - \cos \theta \cos \psi \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \ddot{\theta} \\ \ddot{\varphi} \\ \ddot{\psi} \end{pmatrix} +$$

$$r_p \begin{pmatrix} \frac{(1 - \sin \varphi) \cos(\psi - \theta)}{2} & -\frac{\sin \varphi \cos(\psi - \theta)}{2} & \frac{(1 - \sin \varphi) \cos(\psi - \theta)}{2} \\ \sin \varphi \sin \theta \cos \psi - \cos \theta \sin \psi & \sin \varphi \sin \theta \cos \psi & \sin \varphi \sin \theta \cos \psi - \cos \theta \sin \psi \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \dot{\theta}^2 \\ \dot{\varphi}^2 \\ \dot{\psi}^2 \end{pmatrix} +$$

$$r_p \begin{pmatrix} \cos \varphi \sin(\psi - \theta) & -(1 - \sin \varphi) \cos(\psi - \theta) & -\cos \varphi \sin(\psi - \theta) \\ -2 \cos \varphi \cos \theta \cos \psi & 2(\sin \varphi \cos \theta \sin \psi + \sin \theta \cos \psi) & 2 \cos \varphi \sin \theta \sin \psi \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \dot{\theta} \dot{\varphi} \\ \dot{\theta} \dot{\psi} \\ \dot{\varphi} \dot{\psi} \end{pmatrix} \quad (6.43)$$

sendo que as expressões da primeira e segunda derivadas das coordenadas y_i e z_i , são obtidas através das Eqs.(6.27) a (6.31).

6.2.9. Velocidades e acelerações angulares

As componentes do vetor velocidade angular ω , no referencial inercial, podem ser escritas em termos dos ângulos de Euler e suas derivadas temporais, da seguinte forma:

$$\begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} = \begin{pmatrix} -\cos \varphi \cos \psi & \sin \psi & 0 \\ \cos \varphi \sin \psi & \cos \psi & 0 \\ \sin \varphi & 0 & 1 \end{pmatrix} \begin{pmatrix} \dot{\theta} \\ \dot{\varphi} \\ \dot{\psi} \end{pmatrix} \quad (6.44)$$

Derivando-se as Eqs.(6.44), as componentes do vetor aceleração $\dot{\omega}$, podem ser escritas como:

$$\begin{pmatrix} \dot{\omega}_x \\ \dot{\omega}_y \\ \dot{\omega}_z \end{pmatrix} = \begin{pmatrix} \cos\varphi \cos\psi & \sin\psi & 0 \\ \cos\varphi \sin\psi & \cos\psi & 0 \\ \sin\varphi & 0 & 1 \end{pmatrix} \begin{pmatrix} \ddot{\theta} \\ \ddot{\varphi} \\ \ddot{\psi} \end{pmatrix} + \begin{pmatrix} \sin\varphi \cos\psi & \cos\varphi \sin\psi & \cos\psi \\ -\sin\varphi \sin\psi & \cos\varphi \cos\psi & -\sin\psi \\ \cos\varphi & 0 & 0 \end{pmatrix} \begin{pmatrix} \dot{\theta} \\ \dot{\varphi} \\ \dot{\psi} \end{pmatrix} \quad (6.45)$$

Onde as derivadas temporais de primeira e segunda ordem dos ângulos de Euler são dadas pelas Eqs.(6.32) a (6.37).

Conhecendo-se as velocidades e acelerações lineares e angulares, torna-se possível calcular as forças e torques atuantes na estrutura, permitindo o desenvolvimento do modelo dinâmico.

6.2.10. Dinâmica da plataforma móvel

As equações de Newton-Euler para o CaPaMan podem ser formuladas considerando todos os componentes como corpos rígidos. A orientação, a posição, a velocidade e a aceleração da plataforma móvel são dadas, respectivamente, pelas Eqs.(6.31), (6.41) e (6.42) a (6.45).

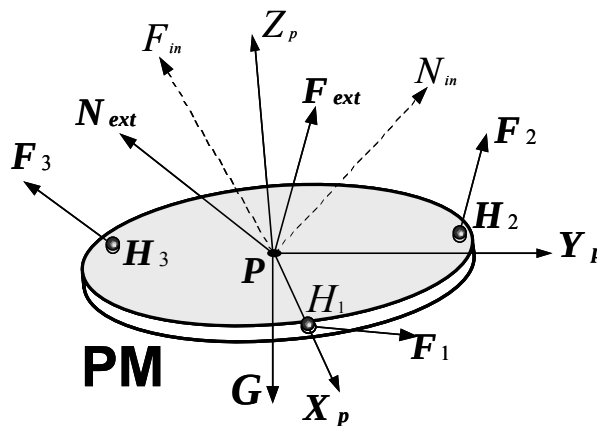


Figura 6.11 - Forças atuantes na PM

Nesta formulação, assume-se que os três mecanismos de quatro barras são paralelogramos articulados com $a_i = c_i$ e $m_i = d_i$. Considera-se também $r_B = r_P$. Deste modo,

conforme representado na Fig. 6.11, o equilíbrio dinâmico para a PM é representado pelas equações de Newton-Euler na forma:

$$F + F_{ext} + G = F_{in} \quad (6.46)$$

$$N + N_{ext} = N_{in} \quad (6.47)$$

Sendo F_{ext} a força externa, N_{ext} o torque externo, G o peso da plataforma móvel, F a soma das forças de reação F_i ($i=1,2,3$) nos pontos H_i da PM e N o torque resultante devido as forças F_i , em relação ao sistema de referência $OXYZ$ fixo na base PF.

Além disso, deve-se considerar as expressões:

$$F_{in} = M a_p ; \quad N_{in} = I \dot{\omega} + \omega \times I \omega \quad (6.48)$$

$$F = \sum_{i=1}^3 F_i ; \quad N = \sum_{i=1}^3 (r_p \times R u_{pi}) \times F_i \quad (6.49)$$

onde M é a massa da PM; a_p é a aceleração do ponto central P; $\dot{\omega}$ e ω são a aceleração angular e velocidade angular, respectivamente e I é a matriz de inércia da plataforma móvel em relação ao sistema inercial. A matriz de inércia I pode ser determinada por:

$$I = R I_c R^T \quad (6.50)$$

sendo R a matriz de transformação, dada pela Eq.(6.38), R^T a sua transposta e I_c a matriz de inércia do centro de massa da PM com respeito ao sistema não inercial.

Nesta análise são desprezados os atritos nas juntas do CaPaMan. As componentes das forças atuantes nas articulações H_i , devido às barras conectoras h_i podem ser consideradas como pertencentes a um plano paralelo aos respectivos paralelogramos articulados e mais uma componente na direção ortogonal. Assim, caso a componente ortogonal exista, ela serve para reposicionar a plataforma móvel em uma posição de equilíbrio. Então, as forças nas articulações H_i possuem somente as componentes F_{iy} e F_{iz} para $i=1,2,3$, como esquematizado na Fig. 6.12. Deve-se observar que as componentes de

torque são produzidas pela ação, dos atuadores e que as componentes de forças são resultantes do equilíbrio das forças nos corpos.

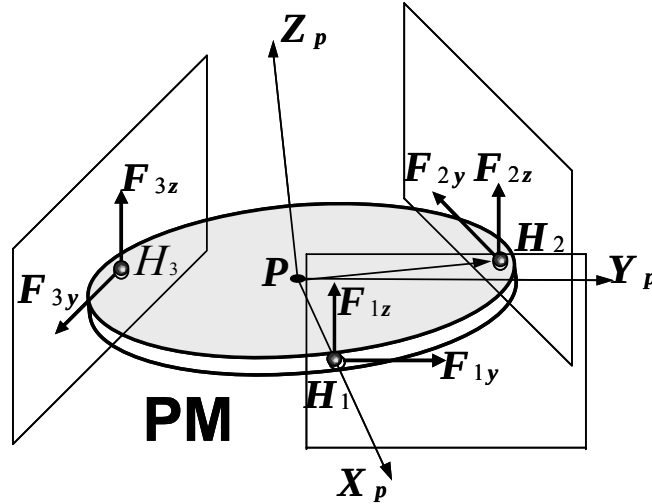


Figura 6.12 - Forças atuantes nos pontos de articulação H_i

Resolvendo as Eqs.(6.49) obtém-se as componentes da força resultante F e do torque N em relação ao sistema inercial, dados por:

$$\begin{pmatrix} F_x \\ F_y \\ F_z \\ N_x \\ N_y \\ N_z \end{pmatrix} = \begin{pmatrix} -\frac{\sqrt{3}}{2}F_{2y} + \frac{\sqrt{3}}{2}F_{3y} \\ F_{1y} - \frac{1}{2}F_{2y} - \frac{1}{2}F_{3y} \\ F_{1z} + \frac{1}{2}F_{2z} + \frac{1}{2}F_{3z} \\ -u_{1z}F_{1y} + \frac{1}{2}u_{2z}F_{2y} + \frac{1}{2}u_{3z}F_{3y} + u_{1y}F_{1z} + u_{2y}F_{2z} + u_{3y}F_{3z} \\ -\frac{\sqrt{3}}{2}u_{2z}F_{2y} + \frac{\sqrt{3}}{2}u_{3z}F_{3y} - u_{1x}F_{1z} - u_{2x}F_{2z} - u_{3x}F_{3z} \\ \frac{1}{2}(\sqrt{3}u_{2y} - u_{2x})F_{2y} - \frac{1}{2}(\sqrt{3}u_{3y} + u_{3x})F_{3y} + u_{1x}F_{1y} \end{pmatrix} \quad (6.51)$$

onde

$$\begin{pmatrix} u_{ix} \\ u_{iy} \\ u_{iz} \end{pmatrix} = r_p R \begin{pmatrix} \cos \delta_i \\ \sin \delta_i \\ 0 \end{pmatrix}, \quad i=1,2,3 \quad (6.52)$$

As Eqs.(6.46) a (6.51) podem ser resolvidas para obter as componentes de força F_{iy} e torques N_{iz} expressas pela equação matricial:

$$\begin{pmatrix} F_{1y} \\ F_{2y} \\ F_{3y} \\ N_{1z} \\ N_{2z} \\ N_{3z} \end{pmatrix} = A \times \begin{pmatrix} F_x \\ F_y \\ F_z \end{pmatrix} + \begin{pmatrix} 0 & 0 & \frac{2}{2u_{1x} - u_{2x} + \sqrt{3}u_{2y} + u_{3x} + \sqrt{3}u_{3y}} \\ 0 & 0 & \frac{2}{2u_{1x} - u_{2x} + \sqrt{3}u_{2y} + u_{3x} + \sqrt{3}u_{3y}} \\ 0 & 0 & \frac{2}{2u_{1x} - u_{2x} + \sqrt{3}u_{2y} + u_{3x} + \sqrt{3}u_{3y}} \\ \frac{3V_{10}}{V_6} & \frac{3V_{11}}{V_6} & \frac{3V_{12}}{V_6} \\ -\frac{3V_{16}}{V_6} & -\frac{V_{17}}{V_6} & -\frac{3V_{18}}{V_6} \\ \frac{3V_{22}}{V_6} & \frac{3V_{23}}{V_6} & \frac{3V_{24}}{V_6} \end{pmatrix} \begin{pmatrix} N_x \\ N_y \\ N_z \end{pmatrix} \quad (6.53)$$

onde a matriz A é dada na Eq.(6.54):

$$A = \begin{pmatrix} \frac{\sqrt{3}(-u_{2x} + \sqrt{3}u_{2y} + u_{3x} + \sqrt{3}u_{3y})}{3(2u_{1x} - u_{2x} + \sqrt{3}u_{2y} + u_{3x} + \sqrt{3}u_{3y})} & \frac{\sqrt{3}(-u_{2x} + \sqrt{3}u_{2y} - u_{3x} - \sqrt{3}u_{3y})}{2u_{1x} - u_{2x} + \sqrt{3}u_{2y} + u_{3x} + \sqrt{3}u_{3y}} & 0 \\ \frac{2\sqrt{3}(-u_{1x} + u_{3x} + \sqrt{3}u_{3y})}{3(2u_{1x} - u_{2x} + \sqrt{3}u_{2y} + u_{3x} + \sqrt{3}u_{3y})} & \frac{-2u_{1x}}{2u_{1x} - u_{2x} + \sqrt{3}u_{2y} + u_{3x} + \sqrt{3}u_{3y}} & 0 \\ \frac{2\sqrt{3}(u_{1x} - u_{2x} + \sqrt{3}u_{2y})}{3(2u_{1x} - u_{2x} + \sqrt{3}u_{2y} + u_{3x} + \sqrt{3}u_{3y})} & \frac{-2u_{1x}}{2u_{1x} - u_{2x} + \sqrt{3}u_{2y} + u_{3x} + \sqrt{3}u_{3y}} & 0 \\ \frac{V_7}{V_6} & \frac{3V_8}{V_6} & \frac{3V_9}{V_6} \\ -\frac{V_{13}}{V_6} & -\frac{3V_{14}}{V_6} & -\frac{3V_{15}}{V_6} \\ \frac{V_{19}}{V_6} & \frac{3V_{20}}{V_6} & \frac{3V_{21}}{V_6} \end{pmatrix} \quad (6.54)$$

As expressões V_i utilizadas para simplificar as equações, foram desenvolvidas por Carvalho e Ceccarelli (1999a) e são apresentadas no Anexo I.

Observando as Figs. 6.8 e 6.13, os torques τ_{Pi} $i=1,2,3$, de entrada na manivela de cada mecanismo articulado de quatro barras, devido a inércia da plataforma móvel, podem

ser obtidos do equilíbrio dinâmico das pernas, segundo Carvalho e Ceccarelli (1999a), através das seguintes expressões:

$$\tau_{Pi} = \frac{F_{iz} b_i \sin(2\alpha_i)}{2 \sin \alpha_i} - F_{iy} b_i \left(\frac{h_i}{c_i \tan \alpha_i} + 1 \right) \left(1 - \frac{h_i}{h_i \cos \alpha_i + c_i \sin \alpha_i} \right) \sin \alpha_i, \quad i=1,2,3 \quad (6.55)$$

A posição e orientação do ponto central da plataforma móvel, assim como sua velocidade e aceleração (lineares e angulares), dependem do ângulo de entrada, dado como uma trajetória no tempo $\alpha_i(t)$. Assim, conhecida a trajetória da plataforma móvel pode-se computar os torques de entrada nas manivelas dos mecanismos, sendo que as Eqs.(6.46) a (6.52) são usadas para calcular as forças que atuam nas juntas esféricas e finalmente simular a dinâmica da plataforma móvel do CaPaMan usando as Eqs.(6.55).

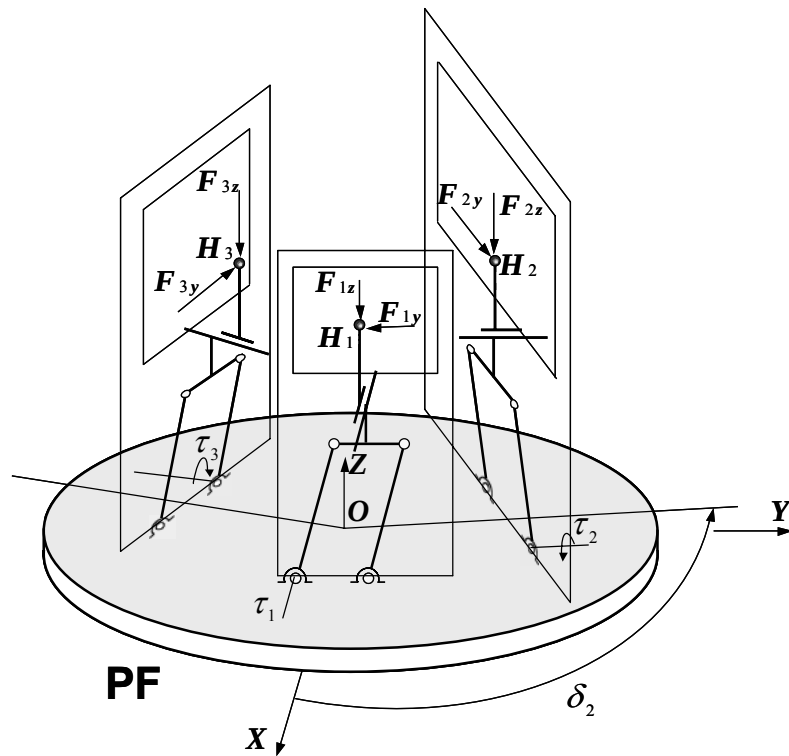


Figura 6.13 - Forças atuantes nas juntas esféricas

6.2.11. Dinâmica dos paralelogramos articulados

Para a análise dinâmica considerando-se a inércia dos paralelogramos articulados, assume-se que as acelerações lineares dos centros de massa e as acelerações angulares de cada segmento foram determinadas a partir da análise cinemática dos paralelogramos

articulados; os efeitos gravitacionais dos segmentos m_i e d_i são desprezados, devido a sua massa ser muito pequena em comparação com a massa da plataforma móvel; os centros dos segmentos são coincidentes com os centros das figuras. Usando a análise cinetostática de mecanismos, o equilíbrio dinâmico na presença das três forças de inércia F_{inbi} , F_{inci} e F_{indi} , cujos pontos de aplicação são determinados pelos offsets de inércia e_{bi} , e_{ci} e e_{di} do centro de massa dos segmentos m_i , c_i e d_i respectivamente, como mostrado na Fig. 6.14, são dados por:

$$\vec{F}_{inbi} = -m_{bi} \vec{a}_{Gbi}; \quad \vec{F}_{inci} = -m_{ci} \vec{a}_{Gci}; \quad \vec{F}_{indi} = -m_{di} \vec{a}_{Gdi} \quad (6.56)$$

$$e_{bi} = \frac{I_{Gbi} \dot{\omega}_{bi}}{F_{inbi}} = \frac{I_{Gbi} \ddot{\alpha}_i}{F_{inbi}}; \quad e_{ci} = \frac{I_{Gci} \dot{\omega}_{ci}}{F_{inci}} = 0; \quad e_{di} = \frac{I_{Gdi} \dot{\omega}_{di}}{F_{indi}} = \frac{I_{Gdi} \ddot{\alpha}_i}{F_{indi}} \quad (6.57)$$

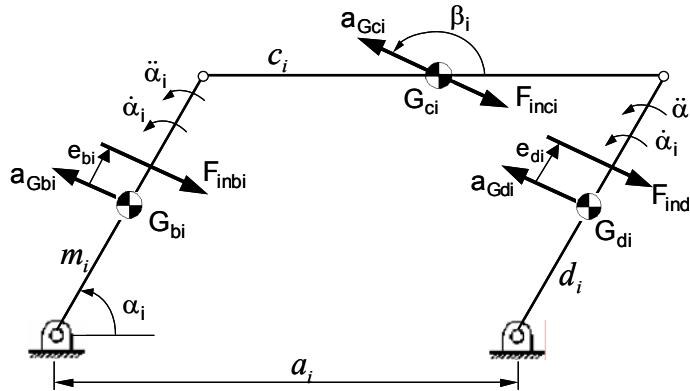


Figura 6.14 - Forças nos paralelogramos articulados i ($i=1,2,3$)

Usando o princípio da superposição, o efeito das forças de inércia em cada segmento pode ser tratado separadamente e então superposto para determinar o efeito combinado. Com exemplo, a Fig. 6.15 mostra o diagrama de corpo livre devido à força de inércia F_{indi} .

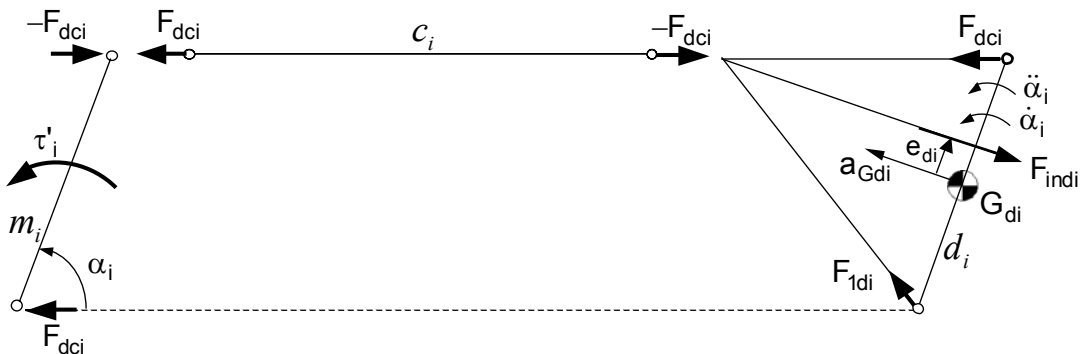


Figura 6.15 - Diagrama de corpo livre para análise do efeito da força de inércia no segmento d_i

Análise de forças semelhantes são feitas para cada segmento. O torque total de entrada, τ_{Mi} , segundo Carvalho e Ceccarelli (1999a) é obtido do efeito de inércia do movimento dos três segmentos e do efeito gravitacional do segmento c_i . Assim, o torque de entrada τ_{Mi} pode ser escrito como:

$$\tau_{Mi} = 2l_{bi} F_{inbi} \sin(\alpha_i - \beta_i + \pi) + F_{23i} b \sin(\alpha_i + \pi - \gamma_i) + b \left[m_{bi} \cos \alpha_i + \frac{(m_{ci} + m_h) \sin 2\alpha_i}{2 \sin \alpha_i} \right] g \quad (6.58)$$

onde

$$l_{bi} = \frac{b}{2} + \frac{I_{Gbi} \ddot{\alpha}_i}{F_{inbi}} \frac{1}{\sin(\alpha_i - \beta_i + \pi)} \quad (6.59)$$

$$F_{23i} = \left| \sqrt{\left\{ F_{inci} \left[\cos(\beta_i + \pi) + \frac{\sin(\pi - \beta_i)}{2 \tan \alpha_i} \right] \right\}^2 + \left[\frac{F_{inci} \sin(\beta_i + \pi)}{2} \right]^2} \right| \quad (6.60)$$

$$\gamma_i = \tan^{-1} \left\{ \frac{\left[\frac{F_{inci} \sin(\beta_i + \pi)}{2} \right]}{F_{inci} \left[\cos(\beta_i + \pi) + \frac{\sin(\pi - \beta_i)}{2 \tan \alpha_i} \right]} \right\} \quad (6.61)$$

Nas quais os ângulos β_i definem a direção da aceleração do centro de massa do segmento i com relação ao eixo horizontal, assumido positivo no sentido anti-horário. Semelhantemente, γ_i define a direção do vetor força de reação atuando na base do segmento d_i .

6.2.12. O Torque total de entrada

Visto que as equações deduzidas são algébricas e lineares nas forças de inércia, o princípio da superposição pode ser aplicado. Desse modo, o efeito dinâmico da plataforma móvel pode ser superposto ao efeito dinâmico do paralelogramo articulado.

O torque total τ_i na manivela de entrada de cada paralelogramo articulado pode ser obtido somando os torques τ_{pi} e τ_{Mi} que são obtidos pela análise dinâmica da plataforma móvel e dos paralelogramos articulados, dados pelas Eqs.(6.55) e (6.58), respectivamente.

$$\tau_i = \tau_{Pi} + \tau_{Mi}, \quad i=1,2,3. \quad (6.62)$$

6.4.13. Formulação do Problema ótimo

Seja o problema de otimização da trajetória de uma estrutura paralela, considerando três critérios conflitantes entre si, escrito como:

$$\text{Minimizar } f = w_1 \frac{E}{E_0} + w_2 \frac{Tt}{T_0} + w_3 \frac{J}{J_0} \quad (6.63)$$

$$\text{sujeito a } \alpha_i^{\inf} \leq \alpha_i(t) \leq \alpha_i^{\sup}, \quad i=1,2,3 \quad (6.64)$$

$$Tt^{\inf} \leq Tt \leq Tt^{\sup} \quad (6.65)$$

$$\tau_i^{\inf} \leq \tau_i \leq \tau_i^{\sup} \quad (6.66)$$

Sendo a energia total do manipulador dada por:

$$E = \int_0^{Tt} \sum_{i=1}^3 [\tau_i(t) \dot{\alpha}_i(t)] dt \quad (6.67)$$

Na Eq.(6.63), w_1 , w_2 e w_3 são os parâmetros de ponderação da função multi-objetivo, E é a energia total do manipulador, E_0 é a energia inicial (da primeira trajetória gerada), J e J_0 são a variação de aceleração (jerk) atual e inicial respectivamente, Tt é o tempo total de percurso e T_0 é o tempo total para a trajetória inicial.

As restrições laterais estão formuladas nas Eqs.(6.64) a (6.66), sendo representadas pelos limites inferiores e superiores dos ângulos de cada manivela de entrada ($\alpha_i^{\inf}$, $\alpha_i^{\sup}$), pelos limites do tempo total de percurso ($Tt^{\inf}$ e $Tt^{\sup}$) e também pelos limites de torque ($\tau_i^{\inf}$, $\tau_i^{\sup}$), para $i=1,2,3$; τ_i é o torque na i -ésima manivela de entrada; α_i é a i -ésima variável de junta.

A energia total é calculada na Eq.(6.67), onde t é o tempo, variável no intervalo $[0, Tt]$ para uma trajetória entre P_0 e P_m ; $\dot{\alpha}_i$ é a primeira derivada temporal. A formulação proposta exige o cálculo do modelo cinemático e dinâmico do CaPaMan. Para determinar as trajetórias pode-se usar os pontos iniciais e finais, P_0 e P_m , no sistema Cartesiano de coordenadas. Estes pontos devem ser transformados em coordenadas de junta resolvendo-se a cinemática inversa. Polinômios cúbicos geralmente são escolhidos para descrever as trajetórias, visto que controle Derivativo Proporcional é geralmente usado nos atuadores dos manipuladores na robótica industrial.

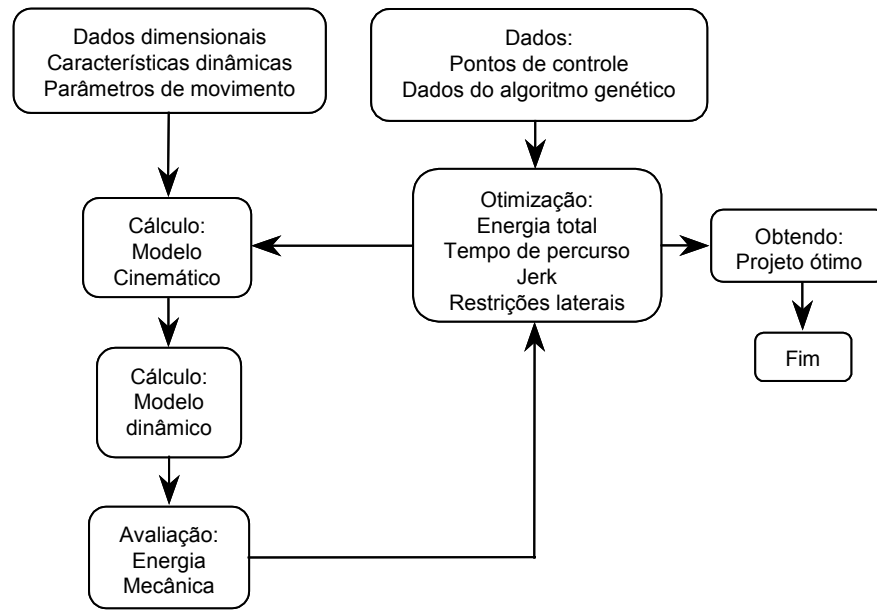


Figura 6.16 - Fluxograma do procedimento numérico para otimização (OLIVEIRA, 2005b)

A metodologia apresentada foi utilizada para a solução do problema de otimização multi-objetivo definido pelas Eqs.(6.63) a (6.67). No processo de otimização, um código geral de análise, desenvolvido por Saramago et al. (2005) foi acoplado ao programa de otimização, conforme mostrado no fluxograma da Fig. 6.16. Este código de análise permite calcular a trajetória do manipulador modelada por splines conforme a Eq.(6.20), o modelo cinemático segundo as Eqs.(6.27) a (6.45), o modelo dinâmico dado pelas Eqs.(6.46) a (6.62); e a energia usando a Eq.(6.67).

A plataforma móvel tem massa $M=2,912$ Kg, os segmentos h , b e c tem massas respectivamente iguais a $m_h=0,1$ Kg, $m_b=0,103$ Kg e $m_c=0,547$ Kg. Os dados dimensionais do CaPaMan estão relacionados na Tab. 6.2. O robô está inicialmente em repouso e fica completamente parado ao final da trajetória, ou seja, $\dot{\alpha}_i(0)=\dot{\alpha}_i(T_i)=0$, $i=1,2,3$.

Tabela 6.2 - Parâmetros dimensionais do protótipo CaPaMan

$a_i = c_i$ [mm]	$m_i = d_i$ [mm]	h_i [mm]	$r_p = r_B$ [mm]	s_i [mm]
200	80	116	109,5	-50; 50.

As trajetórias iniciais são obtidas aleatoriamente, impondo a condição de que sejam executadas no tempo inicial $T_0 = 0,3$ s e obedecidas as restrições laterais dadas por:

$$60 \leq \alpha_1(t) \leq 90 \text{ [graus]}$$

$$50 \leq \alpha_2(t) \leq 120 \text{ [graus]}$$

$$80 \leq \alpha_3(t) \leq 100 \text{ [graus]} \quad (6.68)$$

$$0,1 \leq Tt \leq 0,5 \text{ [s]}$$

$$0,1 \leq \tau_i \leq 0,5 \text{ [Nm]}$$

6.2.14. Simulações Numéricas

A simulação do movimento é realizada adotando os seguintes coeficientes de ponderação: $w_1 = w_2 = 0,3$ e $w_3 = 0,4$.

Os parâmetros usados pela ED foram: $Np=15$, $Nger=200$, $F_p=0,8$ e $Pc=0,6$. Os Algoritmos Genéticos utilizaram $Np=200$ e $Nger=300$.

Os valores iniciais e ótimos são apresentados na Tab. 6.3, mostrando que há uma melhora do índice de performance utilizando algoritmos evolutivos. Além disso, os valores ótimos da energia e do jerk foram fortemente reduzidos, apesar do tempo total de percurso ter um pequeno acréscimo.

Os resultados obtidos nesta simulação foram apresentados na III European Conference on Computational Mechanics Solids, Structures and Coupled Problems in Engineering, em 2006 (OLIVEIRA et al., 2006b).

Tabela 6.3 - Valores ótimos obtidos do processo de otimização ($w_1 = w_2 = 0,3$ e $w_3 = 0,4$)

	ϕ	Energia [Nm/s ²]	Jerk [rad/s ³]	Tempo do percurso [s]	Tempo computacional [s]
Valor inicial	1,00	312,83	832,50	0,30	
AG (Oliveira, 2005b)	0,71	146,24	201,40	0,46	630
1	0,68	148,26	224,65	0,46	299
2	0,64	162,53	257,87	0,44	340
3	0,63	156,19	263,62	0,44	410
4	0,64	152,99	263,14	0,44	408
5	0,70	143,91	201,36	0,48	320
6	0,67	167,52	228,73	0,46	348
7	0,64	150,28	257,85	0,44	290
8	0,61	169,02	298,53	0,42	366
9	0,63	137,98	271,87	0,43	229
10	0,65	151,38	243,43	0,45	318

A Figura 6.17 mostra que as trajetórias ótimas do ponto central da plataforma móvel obedecem aos pontos iniciais P_0 e P_m . Para facilitar a visualização da trajetória, são apresentadas também as projeções no plano cartesiano.

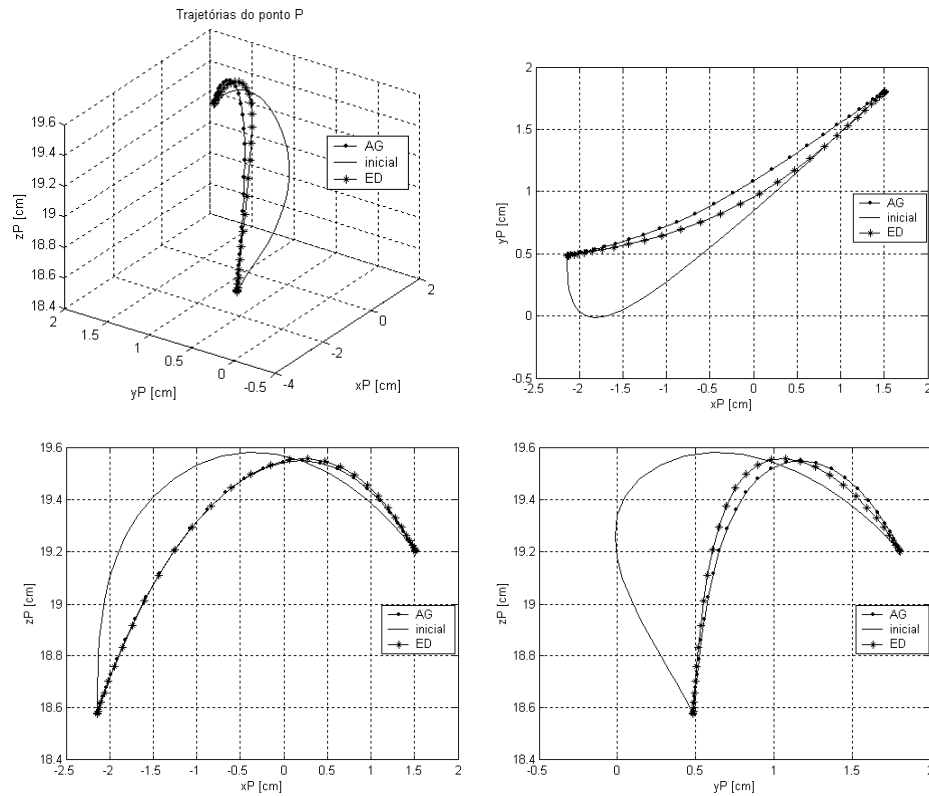


Figura 6.17 - Gráfico 3D da trajetória ótima do centro da plataforma móvel do CaPaMan e suas projeções no plano cartesiano usando AG (---) e ED (-*-)

As Figuras 6.18 a 6.22 mostram resultados iniciais e ótimos usando AG e a estratégia ED/best/2/exp da Evolução Diferencial. A Fig. 6.18 mostra os gráficos das trajetórias iniciais e ótimas para os ângulos de entrada nos mecanismos 1, 2 e 3, podendo-se constatar que o tempo total de percurso sofreu um pequeno acréscimo, mas que a variação dos ângulos de entrada é bastante suave. As curvas iniciais e ótimas das velocidades angulares das manivelas de entrada do CaPaMan são apresentadas na Fig. 6.19. As Figs. 6.20 e 6.21 mostram os gráficos das acelerações e jerks iniciais e ótimas das manivelas, respectivamente. Na Fig. 6.22 são apresentadas as curvas iniciais e ótimas do torque nos atuadores dos mecanismos 1, 2 e 3.

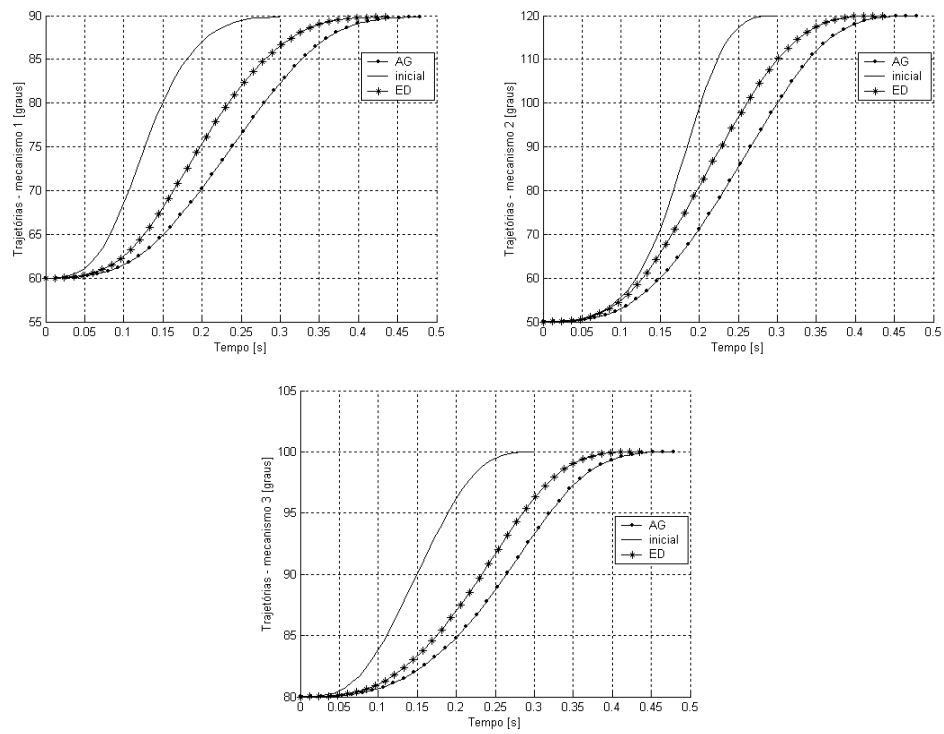


Figura 6.18 - Gráfico das trajetórias iniciais e ótimas para o ângulo de entrada em função do tempo nos mecanismos 1, 2 e 3

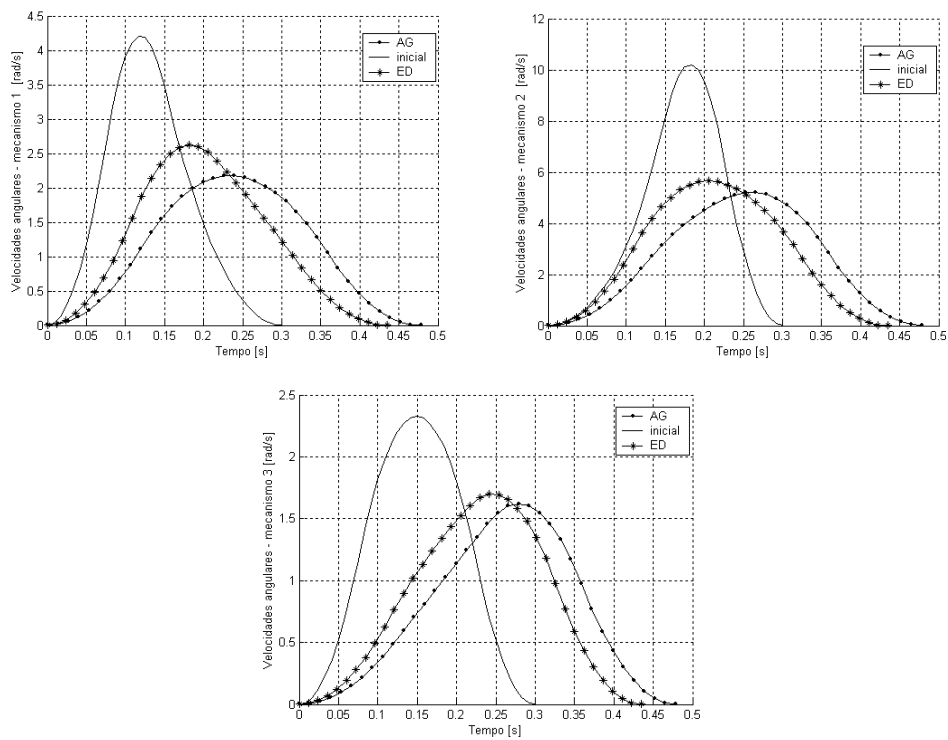


Figura 6.19 - Curvas iniciais e ótimas das velocidades angulares das manivelas de entrada nos mecanismos 1, 2 e 3

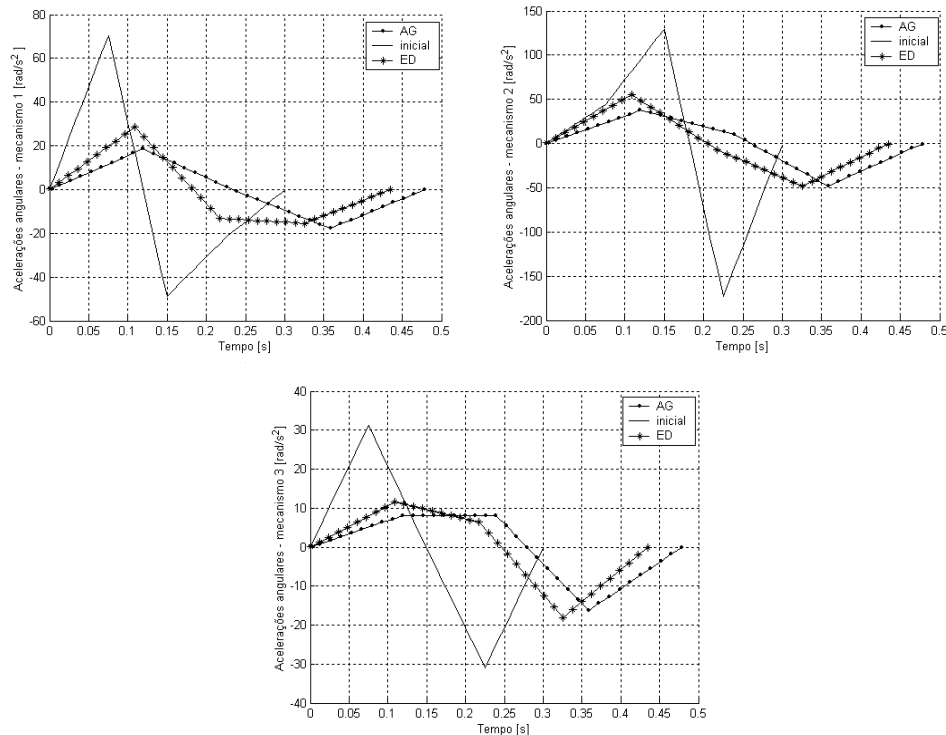


Figura 6.20 - Gráfico das acelerações iniciais e ótimas das manivelas de entrada em função do tempo para o CaPaMan

Observa-se nas Figs. 6.19 e 6.20 que ambas as técnicas de otimização obtiveram significantes reduções nas velocidades e acelerações. No caso da ED, os resultados ótimos obtiveram menor tempo de percurso das trajetórias, por esta razão, os valores das velocidades e acelerações são levemente superiores aos resultados ótimos encontrados usando AG.

Os resultados gráficos para os torques nos atuadores, mostrados na Fig. 6.22, apresentam uma comparação entre curvas iniciais e ótimas do torque. Pode-se observar que os valores ótimos foram fortemente modificados evitando variações abruptas da curva inicial.

Este comportamento suave pode ser verificado nos gráficos do jerk das manivelas de entrada, apresentados na Fig. 6.21, uma vez que grande minimização dos valores do jerk pode ser verificada para os resultados calculados por ambas as técnicas de otimização.

Além disso, a evolução diferencial foi capaz de obter uma maior redução da energia total, como mostrado na Tab. 6.3. Este fato pode ser explicado observando a Fig. 6.22, onde pode verificar-se que o torque total obtido com ED é levemente inferior ao calculado usando AG.

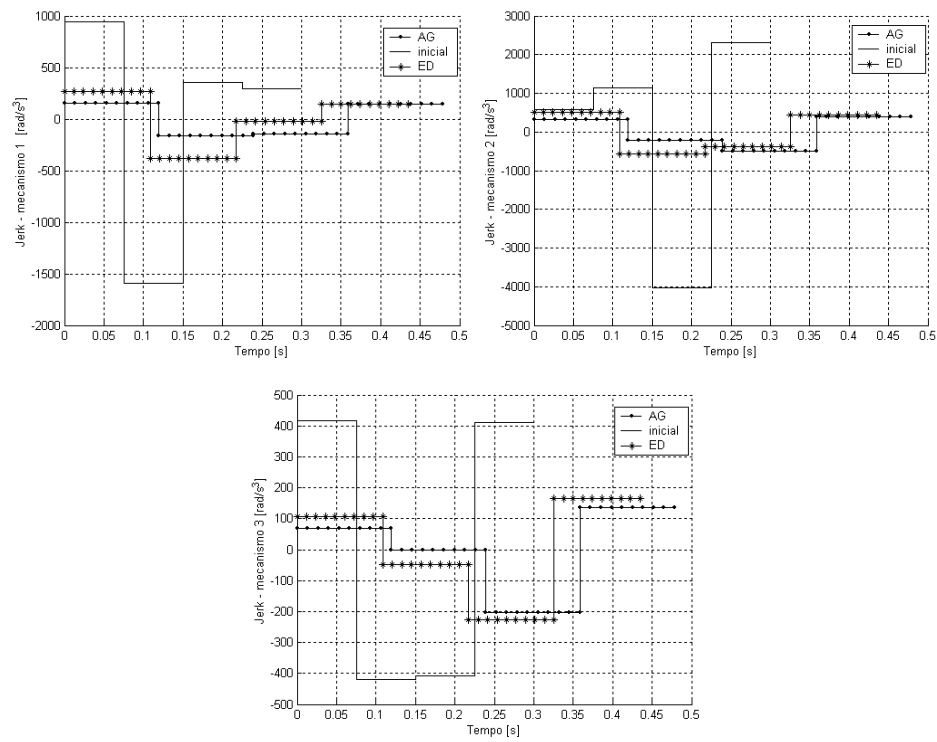


Figura 6.21 - Curva inicial e ótima do jerk das manivelas de entrada

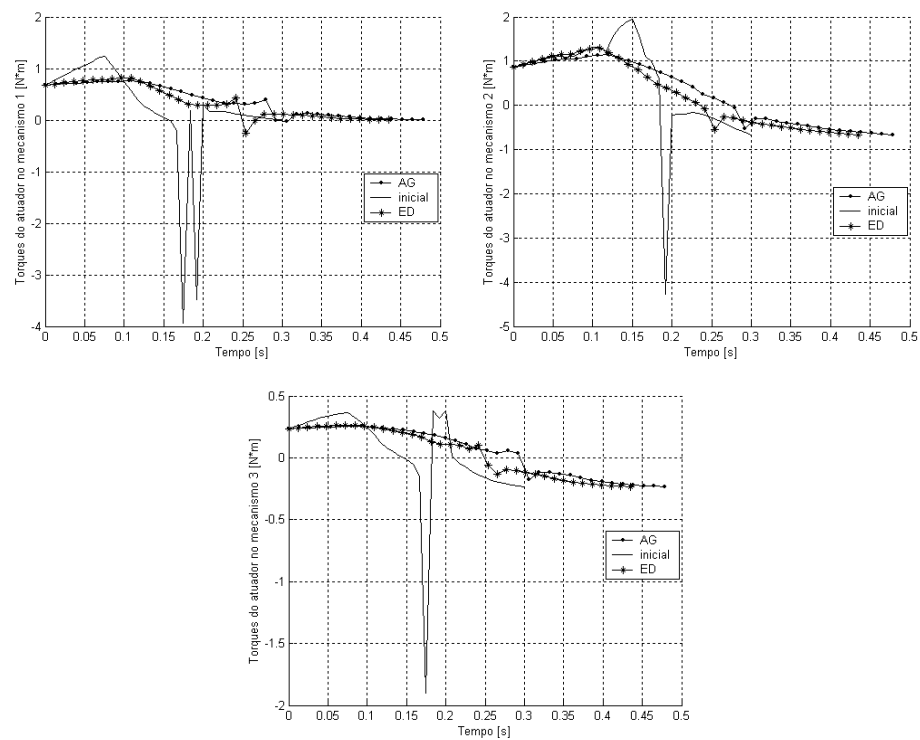


Figura 6.22 - Curvas iniciais e ótimas do torque nos atuadores

No caso de objetivos conflitantes, normalmente o conjunto de soluções é formado por mais de uma solução, o que dificulta a escolha se não houver mais informações sobre o problema. Assim, os resultados ótimos representam uma solução compromisso entre as funções que compõem a função multi-objetivo dada na Eq.(6.63).

Os resultados do procedimento ótimo demonstram a eficiência da formulação proposta, a fim de melhorar o desempenho dinâmico de um manipulador paralelo, reduzindo o consumo de energia e limitando os movimentos bruscos durante o movimento. A importância de usar B-splines cúbicas é devido ao fato de produzir curvas suaves para a aceleração e os jerks; estes fatos facilitam as operações de controle e o movimento do robô.

As duas técnicas de otimização evolucionárias que foram aplicadas apresentam boa performance. No caso estudado, a Evolução Diferencial apresentou resultados um pouco melhores que os Algoritmos Genéticos. Observando a Tab. 6.3, torna-se evidente que a principal vantagem da utilização da Evolução Diferencial é a redução do tempo computacional. Esta técnica pode representar uma ferramenta poderosa em problemas de otimização multimodal complexos ou quando muitas variáveis de projeto são consideradas. Porém verifica-se que é fortemente recomendado que o usuário teste diferentes técnicas de otimização e compare os resultados ótimos obtidos.

CAPÍTULO VII

CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho foi apresentado um estudo detalhado da Evolução Diferencial e suas estratégias, considerando sua fundamentação teórica e o algoritmo de implementação.

Considerando funções bidimensionais, foi realizado um estudo sobre a influência dos seguintes parâmetros da Evolução Diferencial: probabilidade de cruzamento (P_c), fator de perturbação dos indivíduos (F_p), número máximo de gerações (N_{ger}) e o número de indivíduos da população (N_p). Nos testes executados foram analisados os efeitos de um parâmetro fixando os demais, para isto foram considerados os valores recomendados por Storn e Price (1997). Cada parâmetro foi alterado, observando-se sua influência sobre o erro absoluto e o número de avaliações da função objetivo. Como resultado deste estudo obteve-se que a faixa de valores recomendada para F_p e P_c está compreendida entre 0,4 e 0,8, que N_{ger} deve estar em torno de 200 gerações e que pode-se trabalhar com poucos indivíduos, N_p na faixa de 15 a 30 indivíduos.

Nestes testes, evidenciou-se que o algoritmo ED apresenta grande sensibilidade em relação ao número de indivíduos da população (N_p), sendo que a escolha adequada deste parâmetro pode resultar em uma redução significativa do esforço computacional.

Obtidos os melhores parâmetros da Evolução Diferencial, funções matemáticas clássicas e alguns problemas de engenharia, considerando a presença de restrições e problemas de otimização multi-objetivos, foram testados, a fim de verificar seu comportamento e desempenho.

Os valores ótimos encontrados foram comparados com os resultados obtidos utilizando os Algoritmos Genéticos e o Recozimento Simulado. Observou-se que em todas as situações os resultados calculados com a Evolução Diferencial foram adequados, atingindo a mesma faixa de precisão das outras técnicas. Porém, deve-se destacar que a Evolução Diferencial exige um esforço computacional bem inferior quando comparada às outras metodologias.

Duas aplicações, bastante complexas, em robótica foram apresentadas. A primeira, que utiliza a formulação de Bergamaschi (2004), visa maximizar o volume do espaço de

trabalho de robôs manipuladores 3R. A segunda, baseada na formulação desenvolvida por Oliveira (2005b), propõe a otimização de trajetórias de um robô com estrutura paralela. Nas duas aplicações, a performance da Evolução Diferencial foi muito boa, obtendo valores próximos aos resultados encontrados pelos Algoritmos Genéticos, destacando uma redução significativa do tempo de processamento, praticamente metade do tempo de execução dos Algoritmos Genéticos, para a maioria das estratégias da Evolução Diferencial. Esta vantagem da Evolução Diferencial, quando comparada aos Algoritmos Genéticos, pode ser justificada pelo fato de trabalhar com uma população de poucos indivíduos, além de que seu operador mutação utiliza o princípio das diferenças vetoriais.

Em todos os exemplos considerados, todas as estratégias da Evolução Diferencial foram executadas, retornando valores muito semelhantes entre si. Estes resultados evidenciam que a escolha da estratégia a ser utilizada não é tão importante para o processo de otimização quanto poderia ser suposto inicialmente. Esta conclusão pode ser significativa quando se trabalha com problemas complexos, que exigem elevado tempo de processamento, como, por exemplo, códigos acoplados a programas de elementos finitos ou diferenças finitas. Pois, não havendo necessidade de testar todas as estratégias, uma vez que têm pouca influência sobre os resultados, reduz-se o esforço computacional.

A principal contribuição deste trabalho é apresentar a metodologia da Evolução Diferencial, mostrando sua eficiência em trabalhar com problemas complexos.

Como trabalhos futuros propõe-se:

Aprofundar a análise do efeito dos parâmetros da Evolução Diferencial utilizando funções n -dimensionais considerando $n \geq 3$;

Realizar testes cruzados, variando mais de um parâmetro simultaneamente, a fim de verificar a influência destes parâmetros sobre a performance do algoritmo;

Utilizar novas técnicas para aplicar a Evolução Diferencial a problemas restritos;

Trabalhar com a Evolução Diferencial em problemas multi-objetivos, utilizando novas metodologias para escrever este tipo de problema;

Aperfeiçoar o algoritmo computacional, propondo novos critérios de parada;

Escrever o código computacional, amigável e interativo, de forma que seja acessível para profissionais de outras áreas;

Trabalhar com otimização híbrida, associando a Evolução Diferencial com outras técnicas de otimização.

REFERÊNCIAS BIBLIOGRÁFICAS

- ABDEL-MALEK, K.; YEH H.J.; OTHMAN, S. Understanding voids in the workspace of serial robot manipulators. Proc. 23rd ASME, **Design Engineering Technical Conf.**, Baltimore, Maryland, 2000.
- ABIDO, M. A. A niched Pareto genetic algorithm for multiobjective environmental/economic dispatch. **Electrical Power and Energy Systems**, v.25, p. 97-105, 2003.
- BABU, B. V.; MUNAWAR, S. A. Optimal Design of Shell-and-Tube Heat Exchangers by Different Strategies of Differential Evolution, The Faculty Lounge, Article nº 003873, posted on website **Journal: www.PreJournal.org**, March, 03, 2001.
- BABU, B.V.; ANGIRA, R.; NILEKAR, A. Optimal Design of an Auto-Thermal Ammonia Synthesis Reactor Using Differential Evolution. **Proceedings of The Eighth World Multi-Conference on Systemics, Cybernetics and Informatics (SCI-2004)**, Orlando, Florida, USA, July 18-21, 2004.
- BABU, B.V.; PALLAVI, G. C.; Syed Mubeen, J. H. Multiobjective Differential Evolution (MODE) for Optimization of Adiabatic Styrene Reactor, **Chemical Engineering Science**, v. 60, n.17, p. 4822-4837, 2005.
- BABU, B.V.; ANGIRA, R. Modified Differential Evolution (MDE) for Optimization of Non-Linear Chemical Processes. **Computers & Chemical Engineering**, v. 30, n. 6-7, p. 989-1002, 2006.
- BEER F. P.; JOHNSTON Jr, E. R. Mecânica Vetorial para Engenheiros, 5. ed., v. 1 e 2, São Paulo: Makron, McGraw-Hill, Brasil, 1991.
- BERGAMASCHI, P.R.; SARAMAGO, S.F.P.; COELHO, L.S. Evolução Diferencial aplicada à otimização do volume do espaço de trabalho de um manipulador, **VII SBAI / II IEEE LARDS**. São Luís, 2005.
- BERGAMASCHI, P. R.; SARAMAGO S. F. P.; NOGUEIRA, A. C. Design an optimization of 3R manipulators using the workspace features. **Applied Mathematics and Computation**, v. 172, n.1, p. 439-463, 2006.
- BERGAMASCHI, P. R.; SARAMAGO S. F. P.; NOGUEIRA, A. C. An optimal design of 3R manipulators with prescribed workspace. In: INTERNATIONAL SYMPOSIUM ON MULTIBODY SYSTEMS AND MECHATRONICS, 2, 2005, Uberlândia (Brasil), **Proceedings of MUSME**. p. 140-158.

- BERGAMASCHI, P. R. **Projeto Ótimo Robôs Manipuladores 3R, Considerando as Características de seu Espaço de Trabalho**. 2004. Tese de Doutorado - Universidade Federal de Uberlândia, Uberlândia, MG, Brasil.
- BERTSIMAS, D.; TSITSIKILIS, J. N. **Introduction to Linear Optimization**. Athena Scientific, Belmont, USA, 1997.
- BRAGA, C. G. **O Uso de Algoritmos Genéticos para Aplicação em Problemas de Otimização de Sistemas Mecânicos**. 1998. Dissertação de Mestrado - Universidade Federal de Uberlândia, Uberlândia, MG, Brasil.
- BREGALDA, P. et al. **Introdução à Programação Linear**. Ed. Campus, 1988.
- CARVALHO, J.C.M; CECCARELLI, M. A Dynamic Analysis for Cassino Parallel Manipulator. In: 10th WORLD CONGRESS ON THE THEORY OF MECHANISMS, June 20-24, 1999a, Oulu, Finland, p.1202-1207.
- CECCARELLI, M.; LANNI, C. Sintesis optima de brazos manipuladores considerando las características de su espacio de trabajo, **Revista Iberoamericana de Ingenieria Mecanica**. v. 3, n. 1, p. 49-59, 1999.
- CECCARELLI, M.; LANNI, C. A multi-objective optimum design of general 3R manipulators for prescribed workspace limits. **Mechanism and Machine Theory** v. 39, p. 119-132, 2004.
- CECCARELLI, M.; CARBONE, G.; OTTAVIANO, E. An optimisation problem approach for designing both serial and parallel manipulators. In: INTERNATIONAL SYMPOSIUM ON MULTIBODY SYSTEMS AND MECHATRONICS, 2, 2005, Uberlândia (Brasil), **Proceedings of MUSME**. p. 18-32.
- CECCARELLI, M. A formulation for the workspace boundary of general N-revolute manipulators. **IFTOMM Jnl Mechanism and Machine Theory**. V.31 n.5, p.637-646, 1996.
- CHENG, S.L.; HWANG, C. Optimal Approximation of Linear Systems by a Differential Evolution Algorithm. **IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans**, v.31, n. 6, p. 698-707, Nov. 2001.
- CORANA, A.; MARCHESI, M.; MARTINI, C.; Ridella, S. Minimizing Multimodal Functions of Continuous Variables with the Simulated Annealing Algorithm. **ACM Transactions on Mathematical Software**, v. 13, n.3, p. 272-280, 1987.
- DEB, K. **Multi-objective optimization using Evolutionary Algorithms**. John Wiley & Sons, 2001.
- DEB, K. **Otimization for Engineering Design: Algorithms and Examples**. New Delhi: Prentice-Hall, 1995.

- DEJONG, K. A. **An Analysis of the Behavior of a Class of Genetic Adaptive Systems**. 1975. Ph.D. Dissertation, University of Michigan, USA.
- ESCHENAUER, H.; KOSKI, J.; OSYCHKA, A. **Multicriteria Design Optimization**. Berlin, Springer-Verlag, 1990.
- ESHELMAN, L.; SHAFFER. **Real Coded Genetic Algorithms and Interval-Schemata**. In San Mateo, CA: Morgan Kaufman, pp. 187-2002, 1993.
- FOLEY et al. **Computer Graphics: Principles and Practice**. 2.ed. Addison-Wesley Publishing Company, ISBN 0-201-12110-7, T385.C587, 006-dc20, 1990.
- FOX, R. L. **Optimization Methods for Engineering Design**, Reading, MA: Addison-Wesley, 1971.
- GOLDBERG, D. E. **Genetic Algorithms in Search, Optimization, and Machine Learning**. Reading, MA: Addison-Wesley, 1989.
- GUPTA, K. C.; ROTH, B. Design Considerations for Manipulator Workspace. ASME, **Journal of Mechanical Design**. v. 104, p. 704-711, 1982.
- HAJELA, P.; LIN, C.Y. Genetic Search Strategies in Multicriterion Optimal Design. **Structural Optimization**, v.4, pp. 99-107, 1992.
- HAUG, E. J.; ARORA, J. S. **Introduction to Optimal Design**, New York: McGraw Hill, 1989.
- HAUPT, R. L.; HAUPT, S. E. **Practical Genetic Algorithms**. Wiley-Interscience Publication, New York, 1998.
- HIMMELBLAU, D. M. **Applied Nonlinear Programming**, New York: McGraw Hill, 1972.
- HOLLAND, J.H. **Adaptation in natural and artificial systems**. MIT Press, Illinois Genetic Algorithm Laboratory. IlliGAL, University of Illinois at Urbana-Champaign, pp. 11-147, 1975
- HOUCK, C.R.; JOINEZ, J.A.; KAY, M.G. A Genetic Algorithms for Function Optimization: a Matlab Implementation. **NCSO-IE Technical Report**. University of North Caroline, USA, 1995.
- IZMAILOV, A.; SOLODOV, M. **Otimização** - Volume 1: Condições de Otimalidade, Elementos de Análise Convexa e de Dualidade. Rio de Janeiro: IMPA, 2005.
- KIRKPATRICK, S.; GELATT, C. D. JR.; VECCHI, M. P. Optimization by Simulated Annealing. **Science**, v. 220, n. 4598, p. 671-680, 1983.
- LANNI, C.; Saramago, S. F. P; CECCARELLI, M. Optimal design of 3R manipulators using classical techniques and simulated annealing. **Revista Brasileira de Ciências Mecânicas**, Brazil. v. 24 n.4, p. 293-301, 2002.
- MARTINEZ, J. M.; SANTOS, S. A. Métodos computacionais de otimização. XX COLÓQUIO BRASILEIRO DE MATEMÁTICA - IMPA, 1995, Rio de Janeiro (ISBN 85-244-0092-7, 256 p).

- MASTERS, T. **Practical Neural Network Recipes in C++**, Academic Press, 1993.
- MENDES, R. **Population Topologies and Their Influence in Particle Swarm Performance**. 2004. Doctorate Thesis - University of Minho, Portugal.
- METROPOLIS, N.; ROSENBLUTH, A. W.; ROSENBLUTH, M. N.; TELLER, A. H. Equations of state calculations by fast computing machines. **Journal of Chemical Physics**. v. 21, p. 1087-1092, 1953.
- MICHALEWICZ, Z. **Genetic Algorithms + Data Structures = Evolution Programs**. 3.ed. New York: Springer-Verlag, 1996.
- OLIVEIRA, A. C. M. **Algoritmos Evolutivos Híbridos com Detecção de Regiões Promissoras em Espaços de busca Contínuos e Discretos**. 2004. Tese de Doutorado - Instituto Nacional de Pesquisas Espaciais, São José dos Campos, SP, Brasil.
- OLIVEIRA, L. S. **Uma Contribuição ao Estudo dos Métodos de Otimização Multi-Objetivo**. 2005a. Dissertação de Mestrado - Universidade Federal de Uberlândia, Uberlândia, MG, Brasil.
- OLIVEIRA, P. J. **Otimização de Trajetórias de Robôs com Estrutura Paralela**. 2005b. Tese de Doutorado - Universidade Federal de Uberlândia, Uberlândia, MG, Brasil.
- OLIVEIRA, G. T. S.; PRADO, J. R.; SARAMAGO, S. F. P.; BERGAMASCHI, P. R. Maximização do Espaço de Trabalho de Manipuladores 3R usando Algoritmos Evolutivos. In: XXVII IBERIAN LATIN-AMERICAN CONGRESS ON COMPUTATIONL METHODS IN ENGINEERING (CILAMCE), 2006a, Belém do Pará, Brasil. **Anais do XXVII CILAMCE (CD-ROM)**. Associação Brasileira de Mecânica Computacional - ABMEC.
- OLIVEIRA, G. T. S.; SARAMAGO, S. F. P.; OLIVEIRA, P. J.; CARVALHO, J. C. M.; CECCARELLI, M. On the use of Differential Evolution in the trajectory modeling of parallel architecture robot. In: III EUROPEAN CONFERENCE ON COMPUTATIONAL MECHANICS SOLIDS, STRUCTURES AND COUPLED PROBLEMS IN ENGINEERING, **Anais III ECCM (CD-ROM)**, 1, June 5-9, 2006b, Lisboa, Portugal, p. 1-13.
- OSYCHKA, A. **Multi-criterion Optimization in Engineering**. Ellis Horwood Ltd, 1990.
- REKLAITIS, G. V.; RAVINDRAN, A.; RAGSDALL, K. M. **Engineering Optimization: Methods and Applications**. John Wiley & Sons, USA, 1983.
- SARAMAGO, S. F. P.; FARIA, E. D. Constraint Optimization Problems Using Simulated Annealing. **Ciência & Engenharia**. Brasil, v. 10, n. 1, p. 69-75, 2001.
- SARAMAGO, S. F. P.; ASSIS, E. G.; STEFFEN Jr, V. Simulated Annealing: Some Applications In Mechanical Systems Optimization. In: XX IBERIAN LATIN-AMERICAN CONGRESS ON COMPUTATIONL METHODS IN ENGINEERING (CILAMCE), 1999, São Paulo. **Anais do XX CILAMCE (CD-ROM)**. Associação Brasileira de Mecânica

Computacional - ABMEC.

- SARAMAGO, S. F. P.; OTTAVIANO, E.; CECCARELLI, M. A characterization of the workspace boundary of three-revolute manipulators. In: DESIGN ENGINEERING TECHNICAL CONFERENCES (DETC'02), 1, 2002, Montreal. **Proceedings of DETC'02 ASME**. p. 34342-34352.
- SARAMAGO, S. F. P.; STEFFEN Jr, V. Dynamic Optimization for the Trajectory Planning of Robot Manipulators in the Presence of Obstacles. **Journal of the Brazilian Society of Mechanical Sciences**, Brazil, v. XXI, n. 3, p. 371-383, 1999a.
- SARAMAGO, S. F. P.; CARBONE, G.; CECCARELLI, M.; OLIVEIRA, P. J.; CARVALHO, J. C. M. Optimum path Planning of Capaman (Cassino Parallel Manipulator) by Using Inverse Dynamics. In: INTERNATIONAL SYMPOSIUM ON MULTIBODY SYSTEMS AND MECHATRONICS, 2, 2005, Uberlândia (Brasil), **Proceedings of MUSME**, p. 332-343.
- STORN, R. System Design by Constraint Adaptation and Differential Evolution. **IEEE Transactions on Evolutionary Computation**, v. 3, n. 1, p. 22–34, 1999.
- STORN, R.; PRICE, K. Differential Evolution: a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical Report TR-95-012, **International Computer Science Institute**, Berkeley, mar, 1995.
- STORN, R.; PRICE, K. Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. **Journal of Global Optimization**, v. 11, p. 341-359, 1997.
- STORN, R.; PRICE, K. Home Page of Differential Evolution (DE) for Continuous Function Optimization. Disponível em: <<http://www.icsi.berkeley.edu/~storn/code.html#matl>>. Acesso em: 12 jul. 2006.
- VANDERPLAATS, G. N. **Numerical Optimization Techniques for Engineering Design**. 3.ed. Vanderplaats Research & Development, Inc., Colorado Springs, CO, 1999.

ANEXO I

$$V_1 = z_1^2 + z_2^2 + z_3^2 - z_1 z_2 - z_1 z_3 - z_2 z_3,$$

$$V_2 = 9r_p (1 + \sin \varphi)^2 - 4(y_1 + y_2 + y_3)^2$$

$$V_3 = 16V_1^2 + 3(z_2 - z_3)^2 (9r_p^2 - 8V_1)$$

$$V_4 = 16V_1^2 + 3(z_1 - z_3)^2 (9r_p^2 - 8V_1)$$

$$V_5 = 16V_1^2 + 3(z_1 - z_2)^2 (9r_p^2 - 8V_1)$$

$$V_6 = 3[u_{1x}(u_{2y} - u_{3y}) + u_{2x}(-u_{1y} + u_{3y}) + u_{3x}(u_{1y} - u_{2y})](2u_{1x} - u_{2x} + \sqrt{3}u_{2y} - u_{3x} - \sqrt{3}u_{3y})$$

$$V_7 = \sqrt{3}(u_{2x} - u_{3x}) \left[u_{1x}(-u_{2z} + u_{3z}) - u_{1z}(-u_{2x} + \sqrt{3}u_{2y} + u_{3x} + \sqrt{3}u_{3y}) + \right. \\ \left. u_{2z}(u_{3x} + \sqrt{3}u_{3y}) + u_{3z}(-u_{2x} + \sqrt{3}u_{2y}) \right] + \\ 3(u_{2y} - u_{3y})[u_{1x}(u_{2z} + u_{3z}) + u_{3z}(-u_{2x} + \sqrt{3}u_{2y}) - u_{2z}(u_{3x} + \sqrt{3}u_{3y})]$$

$$V_8 = \sqrt{3}u_{1x}(u_{2y} - u_{3y})(u_{2z} - u_{3z}) + \\ (u_{2x} - u_{3x})[-u_{1x}(u_{2z} + u_{3z}) + u_{1z}(u_{2x} - \sqrt{3}u_{2y} + u_{3x} + \sqrt{3}u_{3y})]$$

$$V_9 = (u_{2x}u_{3y} - u_{2y}u_{3x})(2u_{1x} - u_{2x} + \sqrt{3}u_{2y} - u_{3x} - \sqrt{3}u_{3y})$$

$$V_{10} = (-u_{2x} + u_{3x})(2u_{1x} - u_{2x} + \sqrt{3}u_{2y} - u_{3x} - \sqrt{3}u_{3y})$$

$$V_{11} = (-u_{2y} + u_{3y})(2u_{1x} - u_{2x} + \sqrt{3}u_{2y} - u_{3x} - \sqrt{3}u_{3y})$$

$$V_{12} = \sqrt{3}(-u_{2y} + u_{3y})(u_{2z} - u_{3z}) + (u_{2x} - u_{3x})(-2u_{1z} + u_{2x} + u_{3z})$$

$$V_{13} = \sqrt{3}(u_{1x} - u_{3x}) \left[\begin{aligned} &u_{1x}(u_{3z} - u_{2z}) - u_{1z}(-u_{2x} + \sqrt{3}u_{2y} + u_{3x} + \sqrt{3}u_{3y}) + \\ &u_{3z}(-u_{2x} + \sqrt{3}u_{2y}) + u_{2z}(u_{3x} + \sqrt{3}u_{3y}) \end{aligned} \right] + \\ + 3[u_{1x}(u_{2z} + u_{3z})(u_{1y} - u_{3y}) + u_{3z}(u_{1x} - u_{3y})(-u_{2x} + \sqrt{3}u_{2y}) - u_{1y}u_{2z}(u_{3x} + \sqrt{3}u_{3y}) + V_7]$$

$$V_{14} = \sqrt{3}u_{1x}(u_{1y} - u_{3y})(u_{2z} - u_{3z}) + (u_{3x} - u_{1x})[u_{1x}(u_{2z} + u_{3z}) - u_{1z}(u_{2x} - \sqrt{3}u_{2y} + u_{3x} + \sqrt{3}u_{3y})]$$

$$V_{15} = (u_{1x}u_{3y} - u_{1y}u_{3x})(2u_{1x} - u_{2x} + \sqrt{3}u_{2y} - u_{3x} - \sqrt{3}u_{3y})$$

$$V_{16} = (-u_{1x} + u_{3x})(2u_{1x} - u_{2x} + \sqrt{3}u_{2y} - u_{3x} - \sqrt{3}u_{3y})$$

$$V_{17} = (u_{3y} - u_{1y})(2u_{1x} - u_{2x} + \sqrt{3}u_{2y} - u_{3x} - \sqrt{3}u_{3y})$$

$$V_{18} = \sqrt{3}(u_{3z} - u_{2z})(u_{1y} - u_{3y}) + (u_{3x} - u_{1x})(2u_{1z} - u_{2z} - u_{3z})$$

$$V_{19} = \sqrt{3} \left\{ \begin{aligned} &(u_{1x} - u_{2x})[u_{3z} - u_{1z})(-u_{2x} + \sqrt{3}u_{2y}) + (u_{2z} - u_{1z})(u_{3x} + \sqrt{3}u_{3y})] + \\ &u_{1x}[u_{1x}(u_{3z} - u_{2z}) + u_{2x}(u_{2z} + u_{3z})] + \\ &3(u_{1y} - u_{2y})[u_{1x}(u_{2z} + u_{3z}) + u_{3z}(-u_{2x} + \sqrt{3}u_{2y}) - u_{2z}(u_{3x} + \sqrt{3}u_{3y})] \end{aligned} \right\}$$

$$V_{20} = \sqrt{3}u_{1x}(u_{1y} - u_{2y})(u_{2z} - u_{3z}) + (u_{1x} - u_{2x})[-u_{1x}(u_{2z} + u_{3z}) + u_{1z}(u_{2x} - \sqrt{3}u_{2y} + u_{3x} + \sqrt{3}u_{3y})]$$

$$V_{21} = (u_{1x}u_{2y} - u_{1y}u_{2x})(2u_{1x} - u_{2x} + \sqrt{3}u_{2y} - u_{3x} - \sqrt{3}u_{3y})$$

$$V_{22} = (-u_{1x} + u_{2x})(2u_{1x} - u_{2x} + \sqrt{3}u_{2y} - u_{3x} - \sqrt{3}u_{3y})$$

$$V_{23} = (-u_{1y} + u_{2y})(2u_{1x} - u_{2x} + \sqrt{3}u_{2y} - u_{3x} - \sqrt{3}u_{3y})$$

$$V_{24} = \sqrt{3}(-u_{2y} + u_{1y})(-u_{2z} + u_{3z}) + (u_{1x} - u_{2x})(-2u_{1z} + u_{2x} + u_{3z})$$