MARÍA ALEJANDRA GUZMÁN PARDO

Técnicas de otimização baseadas em quimiotaxia de bactérias

Tese apresentada à Escola de Engenharia de São Carlos da Universidade de São Paulo para a obtenção do título de Doutora em Engenharia Mecânica.

Área de concentração: Projeto Mecânico
Orientador: Prof. Dr. Jonas de Carvalho

São Carlos
2009

FOLHA DE APROVAÇÃO

María Alejandra Guzmán Pardo
Técnicas de otimização baseadas em quimiotaxia de bactérias

Tese apresentada à Escola de Engenharia de São
Carlos da Universidade de São Paulo para a
obtenção do título de Doutora.
Área de Concentração: Projeto Mecânico

Aprovado em:

Banca Examinadora

Prof. Dr. _____

Instituição _____ Assinatura _____

Prof. Dr. _____

Instituição _____ Assinatura _____

Prof. Dr. _____

Instituição _____ Assinatura _____

Prof. Dr. _____

Instituição _____ Assinatura _____

Prof. Dr. _____

Instituição _____ Assinatura _____

AGRADECIMENTOS

Desejo expressar minha sincera e profunda gratidão para as seguintes pessoas e instituições que contribuíram para a realização deste trabalho:

- Ao Professor Dr. Jonas de Carvalho pela sua valiosa orientação e colaboração permanentes.
- Ao Professor Dr. Alberto Delgado pela sua dedicação e orientação e por me oferecer uma amizade duradoura.
- Ao Professor Dr. Rodrigo Nicoletti pela sua colaboração na implementação do modelo de elementos finitos para o cálculo das velocidades críticas dos eixos.
- Aos professores, funcionários e amigos da Escola de Engenharia de São Carlos pelo apoio oferecido e convívio gratificante.
- À minha família: meus pais Gilma e Reinaldo, minhas irmãs Liliana e Tania, meus primos Lénin e Jenny, meus sobrinhos Lucía, Luis Alejandro e Luciano, e meus cunhados Luis Alberto e Giovanni. Todos eles constituíram o apoio afetivo fundamental para atingir esta meta.
- Aos meus amigos Claudia, Mauricio e César pela companhia, solidariedade e carinho.
- Ao Juan Carlos: as nossas lembranças sempre bastarão para iluminar toda a névoa que me envolva.
- À CAPES e ao CNPq pelo suporte financeiro dispensado para a execução desta pesquisa.
- À Universidad Nacional de Colombia por o suporte financeiro e por me oferecer esta magnífica oportunidade de crescimento acadêmico e pessoal.

VENTANA SOBRE LA UTOPIA

Ella está en el horizonte.

Me acerco dos pasos, ella se aleja dos pasos.

Camino diez pasos y ella se corre diez pasos más allá.

Por mucho que yo camine, nunca la alcanzaré.

¿Para qué sirve la utopía?

Para eso sirve: para caminar.

Eduardo Galeano

RESUMO

GUZMÁN, M. A. **Técnicas de otimização baseadas em quimiotaxia de bactérias**. 2009. 186 p. Tese (Doutorado) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2009.

Em sentido geral, a quimiotaxia é o movimento dirigido que desenvolvem alguns seres vivos em resposta aos gradientes químicos presentes no seu ambiente. Uma bactéria é um organismo unicelular que usa a quimiotaxia como mecanismo de mobilização para encontrar os nutrientes de que precisa para sobreviver e para escapar de ambientes nocivos. Evoluída durante milhões de anos pela natureza, a quimiotaxia de bactérias é um processo altamente otimizado de busca e exploração em espaços desconhecidos. Graças aos avanços no campo da computação, as estratégias quimiotáticas das bactérias e sua excelente capacidade de busca podem ser modeladas, simuladas e emuladas para desenvolver métodos de otimização inspirados na natureza que sejam uma alternativa aos métodos já existentes. Neste trabalho, desenvolvem-se dois algoritmos baseados em estratégias quimiotáticas de bactérias: o BCBTOA (*Bacterial Chemotaxis Based Topology Optimization Algorithm*) e o BCMOA (*Bacterial Chemotaxis Multiobjective Optimization Algorithm*) os quais são um algoritmo de otimização topológica e um algoritmo de otimização multi-objetivo, respectivamente. O desempenho dos algoritmos é avaliado mediante a sua aplicação à solução de diversos problemas de prova e os resultados são comparados com os de outros algoritmos atualmente relevantes. O algoritmo de otimização multi-objetivo desenvolvido, também foi aplicado na solução de três problemas de otimização de projeto mecânico de eixos. Os resultados obtidos e os análise comparativos feitos, permitem concluir que os algoritmos desenvolvidos são altamente competitivos e demonstram o potencial do processo de quimiotaxia de bactérias como fonte de inspiração de algoritmos de otimização distribuída, contribuindo assim, a dar resposta à constante demanda por técnicas de otimização mais eficazes e robustas.

Palavras-chave: quimiotaxia de bactérias; otimização topológica; otimização multi-objetivo.

ABSTRACT

GUZMÁN, M. A. **Optimization techniques based on bacterial chemotaxis**. 2009. 186 p. Thesis (Doctoral) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2009.

In general, chemotaxis is the biased movement developed by certain living organisms as a response to chemical gradients present in their environment. A bacterium is a unicellular organism that uses chemotaxis as a mechanism for mobilization that allows it to find nutrients needed to survive and to escape from harmful environments. Millions of years of natural evolution became bacterial chemotaxis a highly optimized process in searching and exploration of unknown spaces. Thanks to advances in the computing field, bacterial chemotactical strategies and its excellent ability in searching can be modeled, simulated and emulated developing bio-inspired optimization methods as alternatives to classical methods. Two algorithms based on bacterial chemotactical strategies were designed, developed and implemented in this work: i) the topology optimization algorithm, BCBTOA (Bacterial Chemotaxis Based Topology Optimization Algorithm) and ii) the multi-objective optimization algorithm, BCMOA (Bacterial Chemotaxis Multiobjective Optimization Algorithm). Algorithms performances were evaluated by their applications in the solution of benchmark problems and the results obtained were compared with other algorithms also relevant today. The BCMOA developed here was also applied in the solution of three mechanical design problems. The results obtained as well as the comparative analysis conducted lead to conclude that the algorithms developed were competitive. This also demonstrates the potential of bacterial chemotaxis as a process in which distributed optimization techniques can be inspired.

Keywords: bacterial chemotaxis; topology optimization; multi-objective optimization.

LISTA DE ILUSTRAÇÕES

Figura 1. Padrões formados por colônias de bactérias.....	23
Figura 2. Estrutura bidimensional otimizada.....	23
Figura 3. Otimização Estrutural.....	30
Figura 4. Tabuleiro de xadrez em OT.....	36
Figura 5. Idealização do meio contínuo.....	37
Figura 6. Diagrama de corpo livre elemento interior.....	38
Figura 7. Deformação simples no plano.....	39
Figura 8. Frente Ótima de Pareto.....	47
Figura 9. Metas da otimização multi-objetivo.....	48
Figura 10. Procedimento do NSGA-II.....	56
Figura 11. Natação e giros de bactérias.....	67
Figura 12. Movimento quimiotático de bactérias.....	68
Figura 13. Sinal bactéria – bactéria.....	75
Figura 14. Bactérias aplicando estratégia quimiotática em BCBTOA.....	78
Figura 15. Discretização do domínio de projeto em BCBTOA.....	79
Figura 16. Avaliação da função objetivo em BCBTOA.....	80
Figura 17. Esvaziamento dos elementos ineficientes em BCBTOA.....	81
Figura 18. Aderência de bactérias em BCBTOA.....	82
Figura 19. Viga do POT 1.....	84
Figura 20. Topologia final do POT 1 com BCBTOA.....	85
Figura 21. Topologia final do POT 1 com SIMP.....	85
Figura 22. Energia interna de deformação do POT 1 com BCBTOA.....	86
Figura 23. Energia interna de deformação do POT 1 com SIMP.....	86
Figura 24. Viga do POT 2.....	87
Figura 25. Topologia final do POT 2 com BCBTOA.....	88
Figura 26. Topologia final do POT 2 com SIMP.....	88
Figura 27. Energia interna de deformação do POT 2 com BCBTOA.....	89
Figura 28. Energia interna de deformação do POT 2 com SIMP.....	89
Figura 29. Viga do POT 3.....	90
Figura 30. Topologia final do POT 3 com BCBTOA.....	91

Figura 31. Topologia final do POT 3 com SIMP.....	91
Figura 32. Energia interna de deformação do POT 3 com BCBTOA.....	92
Figura 33. Energia interna de deformação do POT 3 com SIMP.....	92
Figura 34. Viga do POT 4.....	93
Figura 35. Topologia final do POT 4 com BCBTOA.....	94
Figura 36. Topologia final do POT 4 com SIMP.....	94
Figura 37. Energia interna de deformação do POT 4 com BCBTOA.....	95
Figura 38. Energia interna de deformação do POT 4 com SIMP.....	95
Figura 39. Viga do POT 5.....	96
Figura 40. Etapas da otimização topológica com BCBTOA.....	97
Figura 41. Topologia final do POT 5 com BCBTOA.....	98
Figura 42. Topologia final do POT 5 com SIMP.....	98
Figura 43. Energia interna de deformação do POT 5 com BCBTOA.....	99
Figura 44. Energia interna de deformação do POT 5 com SIMP.....	99
Figura 45. Estratégia quimiotáctica para bactérias fortes em BCMOA.....	105
Figura 46. Estratégia quimiotáctica para bactérias fracas em BCMOA.....	106
Figura 47. Método da Parede Absorvente.....	110
Figura 48. Soluções encontradas para o problema SCH com BCMOA.....	116
Figura 49. Soluções encontradas para o problema FON com BCMOA.....	116
Figura 50. Soluções encontradas para o problema POL com BCMOA.....	117
Figura 51. Soluções encontradas para o problema KUR com BCMOA.....	117
Figura 52. Soluções encontradas para o problema ZDT1 com BCMOA.....	118
Figura 53. Soluções encontradas para o problema ZDT2 com BCMOA.....	118
Figura 54. Soluções encontradas para o problema ZDT3 com BCMOA.....	119
Figura 55. Soluções encontradas para o problema ZDT4 com BCMOA.....	119
Figura 56. Iterações de ZDT1 com BCMOA.....	120
Figura 57. Passos quimiotácticos em SCH.....	121
Figura 58. Passos quimiotácticos em FON.....	121
Figura 59. Passos quimiotácticos em ZDT1.....	121
Figura 60. Passos quimiotácticos em ZDT2.....	122
Figura 61. Passos quimiotácticos em ZDT3.....	122
Figura 62. Passos quimiotácticos em ZDT4.....	122
Figura 63. Soluções encontradas para o problema CONSTR com BCMOA.....	123

Figura 64. Soluções encontradas para o problema SRN com BCMOA.....	124
Figura 65. Soluções encontradas para o problema TNK com BCMOA.....	124
Figura 66. Métrica de diversidade.....	127
Figura 67. Métrica de convergência.....	127
Figura 68. . Soluções encontradas para o problema POL com BCMOA e NSGA-II..	132
Figura 69. Soluções encontradas para o problema KUR com BCMOA e NSGA-II...	132
Figura 70. Soluções encontradas para o problema ZDT4 com BCMOA, NSGA-II e NSPSO.....	133
Figura 71. Desenho esquemático do PPM 1.....	138
Figura 72. Soluções ótimas para o PPM 1.....	140
Figura 73. Desenho esquemático dos eixos dos PPM 2 e PPM 3.....	143
74. Desenho esquemático comparativo de geometrias de eixos do PPM 2.....	146
Figura 75. Soluções ótimas para o PPM 2.....	147
Figura 76. Desenho esquemático comparativo de geometrias de eixos do PPM 3.....	150
Figura 77. Soluções ótimas para o PPM 3.....	151
Figura 78. Desenho esquemático do eixo do PPM 4.....	152
Figura 79. Desenho esquemático comparativo de geometrias de eixos do PPM 4.....	155
Figura 80. Soluções ótimas para o PPM 4.....	156
Figura 81. Soluções ótimas para o PPM 4: Peso - Inverso do Fator de Segurança.....	156

LISTA DE TABELAS

Tabela 1 -	Valores dos parâmetros de BCBTOA no POT 1.....	85
Tabela 2 -	Valores dos parâmetros de SIMP no POT 1.....	85
Tabela 3 -	Valores dos parâmetros de BCBTOA no POT 2.....	87
Tabela 4 -	Valores dos parâmetros de SIMP no POT 2.....	87
Tabela 5 -	Valores dos parâmetros de BCBTOA no POT 3.....	90
Tabela 6 -	Valores dos parâmetros de SIMP no POT 3.....	90
Tabela 7 -	Valores dos parâmetros de BCBTOA no POT 4.....	93
Tabela 8 -	Valores dos parâmetros de SIMP no POT 4.....	93
Tabela 9 -	Valores dos parâmetros de BCBTOA no POT 5.....	96
Tabela 10 -	Valores dos parâmetros de SIMP no POT 5.....	96
Tabela 11 -	Problemas de validação sem restrições – continua.....	114
Tabela 11 -	Problemas de validação sem restrições – conclusão.....	115
Tabela 12 -	Problemas de validação com restrições.....	115
Tabela 13 -	Resultados da métrica de diversidade.....	130
Tabela 14 -	Resultados da métrica de convergência.....	130
Tabela 15 -	Resultados da métrica de proximidade.....	131
Tabela 16 -	Resultados dos tempos de execução.....	134
Tabela 17 -	Parâmetros fixos do PPM 2.....	143
Tabela 18 -	Variáveis do PPM 2.....	143
Tabela 19 -	Exemplos de soluções para o PPM 2.....	146
Tabela 20 -	Parâmetros fixos do PPM 3.....	148
Tabela 21 -	Variáveis do PPM 3.....	148
Tabela 22 -	Exemplos de soluções para o PPM 3.....	150
Tabela 23 -	Parâmetros fixos do PPM 4.....	153
Tabela 24 -	Variáveis do PPM 4.....	153
Tabela 25 -	Exemplos de soluções para o PPM 4.....	155

LISTA DE ABREVIATURAS E SIGLAS

AE	Algoritmos Evolucionários
AEOM	Algoritmos Evolucionários de Otimização Multi-objetivo
AG	Algoritmos Genéticos
AOMO	Algoritmos de Otimização Multi-objetivo
BESO	<i>Bi-directional Evolutionary Structural Optimization</i>
BCBTOA	<i>Bacterial Chemotaxis Based Topology Optimization Algorithm</i>
BCMOA	<i>Bacterial Chemotaxis Multiobjective Optimization Algorithm</i>
ESO	<i>Evolutionary Structural Optimization</i>
FOP	Frente Ótima de Pareto
HBTO	<i>Homogeneization Based Topology Optimization</i>
MOGA	<i>Multi-objective Genetic Algorithm</i>
MOPSO	<i>Multiobjective Particle Swarm Optimization</i>
NSGA	<i>Non-dominated Sorting Genetic Algorithm</i>
NSGA-II	<i>Elitist Non-dominated Sorting Genetic Algorithm</i>
NSPSO	<i>Non-dominated Sorting Particle Swarm Optimizer</i>
SIMP	<i>Solid Isotropic Microstructure with Penalization</i>
SPEA	<i>Strength Pareto Evolutionary Algorithm</i>
OE	Otimização Estrutural
OT	Otimização Topológica
POMO	Problema de Otimização Multi-objetivo
POT	Problema de Otimização Topológica
PPM	Problema de Projeto Mecânico
PSO	<i>Particle Swarm Optimizer</i>
TV-MOPSO	<i>Time Variant Multi-Objective Particle Swarm Optimization</i>
VEGA	<i>Vector Evaluated Genetic Algorithm</i>

LISTA DE SIMBOLOS

$\rho(x)$	Densidade fictícia do material para o método SIMP
$E_i^0(x)$	Tensor de rigidez do material isotrópico de referência
$E_i(x_i)$	Tensor de rigidez fictício do material no método SIMP
p'	Potencia de penalização no método SIMP
Ω	Domínio de referência
V	Volume fixo máximo admissível
ν^0	Coefficiente de Poisson do material isotrópico
σ_e	Tensão de Von Misses
RR	Razão de rejeição do método BESO
IR	Razão de inclusão do método BESO
SS	Número do estado estável no método BESO
ON	Número oscilatório no método BESO
PI	Índice de desempenho no método BESO
b_x, b_y, b_z	Forças externas distribuídas por unidade de volume nas direções x, y e z
p_x, p_y, p_z	Forças externas distribuídas por unidade de superfície nas direções x, y e z
σ_x, σ_y	Tensões normais nas direções x e y
τ_{xy}, τ_{yx}	Tensões cisalhantes
Γ	Superfície de contorno
$u(x, y)$	Deslocamento de um ponto na direção do eixo x
$v(x, y)$	Deslocamento de um ponto na direção do eixo y
γ_{xy}	Distorção angular
$\underline{\varepsilon}$	Vetor de deformações significativas para tensão e deformação planas
$\underline{T}$	Vetor de componentes das tensões
$\underline{\underline{D}}$	Tensor constitutivo de flexibilidade
$C(\underline{x})$	Energia interna de deformação total
$\underline{d}$	Vetor de deslocamentos
$\underline{p}$	Vetor das forças externas

$\underline{K}$	Matriz de rigidez
W_i	Peso de um elemento finito
W^*	Peso total desejado
$f_m(x)$	m funções objetivo de um POMO
g_j	J desigualdades de um POMO
h_k	K igualdades de um POMO
x_i^L, x_i^U	Valores mínimo e máximo aceitáveis para uma variável x_i
S^*	Espaço de busca em um POMO
σ_{share}	Parâmetro do algoritmo NSGA
w	Peso de inércia em PSO
c_1, c_2	Constantes de aceleração cognitiva e social respectivamente em PSO
$\rightarrow x_i$	Vetor da posição atual em PSO
$\rightarrow p_i$	Vetor da melhor posição histórica em PSO
$\rightarrow v_i$	Vetor da velocidade em PSO
$pbest_i$	Melhor posição histórica encontrada por cada indivíduo em PSO
$\rightarrow p_g$	Melhor posição histórica encontrada por todos os membros do enxame em PSO
$q(r)$	Sinal enviado pelas bactérias de uma colônia
R	Máxima distancia influenciada por uma bactéria aderida
r	Distância radial medida desde uma bactéria aderida
Q	Parâmetro do ambiente em colônias de bactérias
$Nelem$	Número total de elementos finitos
$Vmax$	Volume máximo desejado para o domínio de projeto
I_{total}	Número total de iterações desejadas
f_{xy}	Valor real da função objetivo calculado no elemento finito localizado nas coordenadas (x, y) do domínio de projeto
f_{max}	Valor máximo da função objetivo para um elemento do domínio de projeto
bac_{xy}	Bactéria localizada no elemento finito de coordenadas (x, y)

$q(r)_{bac_{xy}}$	sinal enviado pela bactéria a uma distancia radial r em BCBTOA
f_{mod}	Valor da função objetivo modificada para um elemento finito
$Nelx$	Quantidade de elementos finitos na direção horizontal
$Nely$	Quantidade de elementos finitos na direção vertical
CHS_{max}	Valor máximo de passos quimiotáticos permitidos
p	Número de parâmetros de um POMO
$POFI(j)$	Conjunto de soluções não dominadas em BCMOA
$\theta_k(j)_{strong}$	Localização de uma bactéria forte no parâmetro k em BCMOA
$\theta_k(j)_{weak}$	Localização de uma bactéria fraca no parâmetro k em BCMOA
ST	Tamanho do passo Giro Curto em BCMOA
LT	Tamanho do passo Giro Longo em BCMOA
SW	Tamanho do passo Natação em BCMOA
$max(POFI)_k$	Valor máximo para o parâmetro k dentre o conjunto de soluções
$min(POFI)_k$	Valor mínimo para o parâmetro k dentre o conjunto de soluções
S	Número de bactérias
$J(\theta^i(j))$	Valor das funções objetivo numa localização para BCMOA
$\theta^i(j)$	Localização atual da bactéria i em BCMOA
$\theta^i(j)_{prev}$	Localização previa da bactéria i em BCMOA
$\Delta^i(j)$	Vetor randômico de grandeza igual a um
$r1_k$	Numero randômico que $\in [-0,01, 0,01]$
M	Número de objetivos em um POMO
d_i	Distancia euclidiana entre duas soluções ótimas consecutivas
$Prox$	Métrica de proximidade
Δ	Métrica de diversidade
d_f, d_l	Distâncias euclidianas entre as soluções extremas e as soluções de contorno na métrica diversidade
$\bar{d}$	Media das distancias euclidianas entre duas soluções na métrica diversidade
P	Carga normal
σ	Tensões normais máximas
K	Fator de concentração de tensões
A	Área de uma seção transversal

FS	Fator de segurança estático
S_y	Resistência ao escoamento
α	Ângulo de incidência da engrenagem reta
d	diâmetro
L	Comprimento da seção de um eixo
y_{xy}, y_{xz}	Deflexões laterais de um eixo nos planos xy e xz
y_{max}	Deflexão lateral máxima de um eixo
K_f, K_{fm}	Fatores de concentração de tensão de fadiga por flexão para componentes media e alternantes
K_{fs}, K_{fsm}	Fatores de concentração torcional de fadiga para componentes media e alternantes
M_m, M_a	Componentes media e alternante do momento flexor
T_m, T_a	Componentes media e alternante do torque
S_f	Resistência à fadiga
S_{ut}	Resistência ultima a tração
N_f	Coeficiente de segurança à fadiga
ω_n	Frequência natural de um eixo
ω_t	Frequência de trabalho de um eixo

SUMÁRIO

1.	INTRODUÇÃO.....	21
1.1	CONTEXTO E MOTIVAÇÃO.....	21
1.2	OBJETIVOS DA TESE.....	25
1.3	ORGANIZAÇÃO DA TESE.....	25
2.	OTIMIZAÇÃO TOPOLÓGICA.....	27
2.1	OTIMIZAÇÃO ESTRUTURAL.....	27
2.2	OTIMIZAÇÃO TOPOLÓGICA ESRUTURAL.....	30
2.2.1	Método SIMP (<i>Solid Isotropic Microstructures with Penalization</i>).....	31
2.2.2	Método ESO (<i>Evolutionary Structural Optimization</i>).....	32
2.2.3	Método BESO (<i>Bi-directional ESO</i>).....	34
2.3	DIFICULDADES DA OTIMIZAÇÃO TOPOLÓGICA.....	36
2.4	FORMULAÇÃO DO PROBLEMA DE OTIMIZAÇÃO TOPOLÓGICA..	36
3.	OTIMIZAÇÃO MULTI-OBJETIVO.....	43
3.1	FORMULAÇÃO DO PROBLEMA DE OTIMIZAÇÃO MULTI-OBJETIVO.....	43
3.2	SOLUÇÕES ÓTIMAS DE PARETO.....	45
3.3	MÉTODOS CONVENCIONAIS DE OTIMIZAÇÃO MULTI-OBJETIVO.....	48
3.3.1	Programação linear.....	49
3.3.2	Somatório dos pesos.....	49
3.3.3	Programação de metas.....	49
3.3.4	Dificuldades dos métodos convencionais de otimização multi-objetivo.....	50
3.4	OTIMIZAÇÃO MULTI-OBJETIVO INSPIRADA NA NATUREZA.....	50
3.4.1	Algoritmos evolucionários.....	51
3.4.2	Algoritmos Genéticos (AG).....	52
3.4.3	Otimização multi-objetivo com Algoritmos Genéticos.....	53
3.4.3.1	<i>Elitist Non-dominated Sorting Genetic Algorithm: NSGA-II</i>	55
3.4.4	Inteligência de enxame.....	57

3.4.5	Otimização por enxame de partículas – PSO (<i>Particle Swarm Optimizer</i>)..	57
3.4.5.1	NSPSO (<i>Non-dominated Sorting Particle Swarm Optimizer</i>).....	60
4.	QUIMIOTAXIA DE BACTÉRIAS.....	63
4.1	O QUE É A QUIMIOTAXIA?.....	63
4.2	AS BACTÉRIAS.....	64
4.3	O MECANISMO DA QUIMIOTAXIA DE BACTÉRIAS.....	65
4.4	APLICAÇÕES EM ENGENHARIA DA ESTRATÉGIA QUIMIOTÁCTICA.....	69
4.4.1	Aplicações da estratégia quimiotáctica em otimização.....	70
4.5	COMUNICAÇÃO ENTRE BACTÉRIAS.....	73
4.5.1	Modelo baseado em quimiotaxia e otimização de recursos para comunicação bactéria – bactéria.....	74
5.	OTIMIZAÇÃO TOPOLÓGICA BASEADA EM QUIMIOTAXIA DE BACTÉRIAS.....	77
5.1	ALGORITMO DE OTIMIZAÇÃO TOPOLÓGICA BASEADA EM QUIMIOTAXIA DE BACTERIAS - BCBTOA	77
5.2	VALIDAÇÕES DESENVOLVIDAS.....	84
5.2.1	Problema de Otimização Topológica (POT) 1 - Viga simplesmente apoiada.....	84
5.2.2	POT 2 - Viga engastada em extremo direito.....	86
5.2.3	POT 3 - Viga engastada com orifício circular.....	89
5.2.4	POT 4 - Viga engastada com carga central.....	92
5.2.5	POT 5 - Viga simplesmente apoiada com carga central.....	95
5.3	ANÁLISE DOS RESULTADOS.....	99
6.	OTIMIZAÇÃO MULTI-OBJETIVO BASEADA EM QUIMIOTAXIA DE BACTÉRIAS.....	103
6.1	ALGORITMO DE OTIMIZAÇÃO MULTI-OBJETIVO BASEADO EM QUIMIOTAXIA DE BACTÉRIAS - BCMOA.....	103
7.	VALIDAÇÕES DE DESEMPENHO DO ALGORITMO BCMOA.....	113
7.1	PROBLEMAS DE VALIDAÇÃO.....	113
7.2	VALIDAÇÕES APLICADAS AO BCMOA.....	115
7.2.1	Análise dos resultados das validações aplicadas ao BCMOA.....	125
7.3	MÉTRICAS COMPARATIVAS DO DESEMPENHO.....	125

7.3.1	Diversidade Δ	126
7.3.2	Convergência.....	126
7.3.3	Proximidade.....	128
7.3.4	Quantidade de soluções ótimas.....	128
7.3.5	Tempos de execução.....	128
7.4	APLICAÇÃO DAS MÉTRICAS COMPARATIVAS.....	129
7.4.1	Métrica diversidade.....	130
7.4.2	Métrica convergência.....	130
7.4.3	Métrica proximidade.....	131
7.4.4	Quantidade de soluções ótimas.....	133
7.4.5	Tempos de execução.....	133
7.4.6	Análise dos resultados das métricas comparativas.....	134
8.	APLICACÕES DO ALGORITMO BCMOA EM PROJETO MECÂNICO.....	137
8.1	FORMULAÇÃO DE PROBLEMAS DE PROJETO MECÂNICO (PPM) COMO PROBLEMAS DE OTIMIZAÇÃO MULTI-OBJETIVO.....	137
8.2	OTIMIZAÇÃO DIMENSIONAL DE EIXOS.....	141
8.2.1	PPM 2 - Eixo submetido a torque variável.....	142
8.2.1.1	Definição do PPM 2 como um POMO.....	144
8.2.1.2	Soluções obtidas com BCMOA para o PPM 2.....	145
8.2.2	PPM 3 - Eixo com torques constantes, variando o ponto de apoio e considerando a velocidade crítica.....	147
8.2.2.1	Definição do PPM 3 como um POMO.....	148
8.2.2.2	Soluções obtidas com BCMOA para o PPM 3.....	149
8.2.3	PPM 4 - Eixo com torque constante, considerando a velocidade critica calculada por elementos finitos.....	151
8.2.3.1	Definição do PPM 4 como um POMO.....	153
8.2.3.2	Soluções obtidas com BCMOA para o PPM 4.....	154
9.	CONCLUSÕES.....	157
9.1	CONTRIBUIÇÕES.....	157
9.2	RESULTADOS.....	158
9.3	EXPERIENCIAS ADQUIRIDAS.....	160
9.4	TRABALHOS FUTUROS.....	160

REFERÊNCIAS.....	163
APÊNDICE A. - Código fonte em Matlab dos algoritmos BCBTOA e	175
BCMOA.....	
APÊNDICE B - Resumos dos artigos apresentados em congressos	181
internacionais.....	

CAPÍTULO I

1. INTRODUÇÃO

Neste capítulo é realizada uma introdução sobre o trabalho desenvolvido, sendo apresentados o contexto e a motivação para a escolha do tema abordado, os objetivos e a organização do trabalho.

1.1 CONTEXTO E MOTIVAÇÃO

Projetar um produto de engenharia não é uma tarefa fácil. Num mercado globalizado como o atual, as demandas e as necessidades dos clientes são cada vez mais refinadas e mudam com extrema rapidez. A velocidade das mudanças junto com a altíssima concorrência pelos mercados fazem com que os resultados esperados sejam produtos inovadores de ótima qualidade, que podem ser ofertados no mercado a preços baixos. Como consequência, o desenvolvimento de um produto é um processo de complexidade crescente, o qual demanda melhoras permanentes que conduzam ao uso eficiente dos recursos. A relevância da otimização do projeto de um produto justifica a busca permanente de técnicas de otimização mais eficazes e robustas.

Nesse sentido, a teoria de otimização matemática é uma área do conhecimento sempre em demanda. É constituída por um conjunto de resultados, métodos numéricos e técnicas, enfocados a encontrar o melhor candidato dentre uma coleção de alternativas, sem ter que avaliar explicitamente todas elas (VICTORIA, 2006). Graças aos avanços significativos que apresentou nas últimas duas décadas, a otimização matemática tem encontrado inúmeras

aplicações diretas e indiretas em campos tão diversos, tais como: a engenharia civil, o projeto mecânico, as indústrias automobilística, aeronáutica, química, eletrônica, entre outras.

A emulação da natureza tem inspirado cientistas em diversos campos da ciência no decorrer da história da humanidade. Recentemente, devido aos avanços em computação e ao auge da implementação de idéias extraídas dos comportamentos dos seres vivos e dos processos naturais em geral, as técnicas de computação inspiradas na natureza têm ganhado adeptos continuamente, motivados principalmente por duas causas (DE CASTRO; VON ZUBEN, 2004):

- os métodos tradicionais demonstram serem incapazes de tratar adequadamente problemas complexos caracterizados pela manipulação de uma grande quantidade de variáveis e pela ausência de modelos matemáticos completos;
- para a maioria dos problemas complexos, com os quais está se lidando atualmente, existe uma versão similar na natureza.

As técnicas de otimização matemática apoiadas por computadores não foram excluídas desta tendência, principalmente devido à observação de que os processos naturais são freqüentemente auto-otimizados. É desta forma que têm sido propostos algoritmos de otimização, baseados em fenômenos naturais, tais como: i) colônias de formigas (KONGA; TIANA; KAO, 2008); ii) bactérias (PASSINO, 2002); iii) sistemas imunes (DE CASTRO; TIMMIS, 2002); iv) recozimento de materiais (PEDAMALLU; OZDAMAR, 2008); v) redes neurais (EFFATI; JAFARZADEH, 2007); vi) transmissão de informação genética (LIU, X. et al., 2008), entre outros.

No ano 2005, Thar e Kühl (2005) publicaram um artigo no qual apresentaram resultados satisfatórios obtidos com a implementação de um modelo computacional para a simulação de padrões biológicos observados naturalmente nas colônias de bactérias marinhas do tipo *Thiovulum majus*. O modelo proposto consiste em equações muito simples baseadas exclusivamente na quimiotaxia e na otimização de recursos desenvolvidos por esse tipo de bactérias, as quais em condições naturais formam padrões geométricos muito particulares, como os que se ilustram na figura 1.

Figura 1. Padrões formados por colônias de bactérias
Fonte – (THAR; KUHL, 2005)

O aspecto físico desses padrões de alguma forma tem similitudes com os resultados esperados de um processo de otimização topológica, no qual se pretende encontrar um domínio de projeto constituído por áreas sólidas (preenchidas de material) e por áreas vazias (orifícios) como é ilustrado na figura 2.

Figura 2. Estrutura bidimensional otimizada

Devido a reflexões realizadas acerca dessas aparentes semelhanças, surgiu a idéia de adaptar o modelo de quimiotaxia e de formação de padrões desse tipo de bactérias no desenvolvimento de um método heurístico de otimização topológica de estruturas contínuas.

Também no mesmo ano, Amos, Hodgson e Gibbons (2005) publicaram um artigo no qual expuseram as potencialidades que encontravam na exploração da quimiotaxia de bactérias como processo de otimização. Destacaram o fato de que os algoritmos existentes até o momento careciam da consideração sobre interações e comunicações entre as bactérias. Sabendo que é, precisamente, esse comportamento coletivo que lhes permite desenvolver padrões biologicamente vantajosos, os autores arriscaram-se a sugerir uma possível aplicação dessa interação entre indivíduos no desenvolvimento de métodos de otimização distribuída.

Embora a aplicação da quimiotaxia de bactérias como processo de otimização não seja uma proposta nova (BREMERMANN, 1974), essa aproximação não tem sido aplicada na área de otimização multi-objetivo. Para esse tipo de otimização, uma vantagem fundamental dos algoritmos inspirados na natureza (freqüentemente baseados em populações) sobre os algoritmos clássicos, é precisamente a capacidade de trabalhar de forma distribuída a fim de encontrar simultaneamente várias soluções ótimas em uma única rodada.

Da possível correlação, ainda inexplorada entre estas áreas, surgiu a idéia de desenvolver uma aplicação da quimiotaxia e da formação de padrões em colônias de bactérias na solução de problemas de otimização multi-objetivo. Propõe-se aproveitar as relações diretas ou indiretas que podem se projetar entre agentes relativamente simples junto com a flexibilidade e robustez que caracterizam os algoritmos baseados em comportamentos sociais de indivíduos.

Como se pode extrair das considerações expostas, as idéias contêm um forte componente intuitivo, característica constante em muitos dos métodos bio-inspirados. Desse modo, objetiva-se nesta pesquisa a verificação das potencialidades em relação à adaptação da quimiotaxia de bactérias como processo de otimização, tanto na área de otimização topológica, quanto na área de otimização multi-objetivo.

Considera-se necessário esclarecer que para o desenvolvimento de algoritmos de otimização baseados na natureza, não é essencial construir modelos plausíveis biologicamente. O inovador e prático nesses casos é extrair, precisamente, idéias da natureza para o desenvolvimento de procedimentos que, passo a passo, resolvam problemas computacionais complexos.

1.2 OBJETIVOS DA TESE

O objetivo principal da tese é o desenvolvimento de novos métodos de otimização inspirados na natureza que apliquem os princípios da quimiotaxia de bactérias e explorem a capacidade de busca distribuída desenvolvida pelas colônias de bactérias.

No trabalho desenvolvido nesta tese serão considerados os seguintes objetivos parciais:

- projetar, desenvolver e implementar um algoritmo de otimização topológica para estruturas bidimensionais contínuas, baseado nos processos de quimiotaxia de bactérias e comunicação entre os indivíduos de uma colônia;
- aplicar o algoritmo de otimização topológica à otimização de estruturas bidimensionais contínuas;
- projetar, desenvolver e implementar um algoritmo de otimização multi-objetivo baseado no processo de quimiotaxia de bactérias;
- aplicar o algoritmo de otimização multi-objetivo na solução de problemas de otimização de validação;
- analisar e comparar os resultados obtidos com o algoritmo de otimização multi-objetivo desenvolvido, com a bibliografia mais representativa, mediante a aplicação de métricas de desempenho;
- aplicar o algoritmo de otimização multi-objetivo desenvolvido na solução de problemas práticos de projeto mecânico.

1.3 ORGANIZAÇÃO DA TESE

A tese está estruturada em nove capítulos, incluindo esse primeiro capítulo introdutório, e dois apêndices.

No Capítulo 1, são apresentados o contexto geral que envolve o tema abordado, as motivações que levaram a escolha do tema e os objetivos da tese.

No Capítulo 2, são elucidadas as definições básicas da otimização topológica, são expostas suas bases teóricas e, brevemente, são explicados os algoritmos existentes mais representativos.

No Capítulo 3, são explicados os princípios e conceitos básicos da otimização multi-objetivo. É exposta a formulação geral desse tipo de problema e a revisão bibliográfica dos algoritmos existentes mais representativos.

O Capítulo 4 é dedicado à quimiotaxia de bactérias. Explica-se detalhadamente a estratégia quimiotáctica e é apresentada uma revisão bibliográfica das aplicações em engenharia e, particularmente, no campo da otimização.

O Capítulo 5 apresenta o algoritmo de otimização topológica desenvolvido. Nesse capítulo encontram-se os resultados das validações desenvolvidas e a análise dos mesmos também.

No Capítulo 6, apresenta-se o algoritmo de otimização multi-objetivo desenvolvido. Explicam-se, detalhadamente, os passos do algoritmo.

No Capítulo 7, mostram-se as validações desenvolvidas para o algoritmo de otimização multi-objetivo proposto; detalham-se as métricas de desempenho selecionadas e se analisam os resultados obtidos, quando comparados com outros algoritmos.

No Capítulo 8 apresentam-se as quatro aplicações feitas do algoritmo de otimização multi-objetivo desenvolvido na solução de problemas de projeto mecânico.

O Capítulo 9 contém as conclusões deste trabalho e as sugestões para trabalhos futuros.

O Apêndice A contém o código fonte em Matlab dos laços principais dos dois algoritmos desenvolvidos neste trabalho.

O Apêndice B compreende os resumos dos artigos apresentados nos congressos internacionais: 8º Congresso Ibero-americano de Engenharia Mecânica, *International Conference on Engineering Optimization* e *8º World Congress on Computational*.

CAPÍTULO 2

2. OTIMIZAÇÃO TOPOLÓGICA

Neste capítulo explicam-se os conceitos básicos da otimização estrutural e suas divisões em otimização dimensional, de forma e topológica. A otimização topológica é detalhada e são explicados três algoritmos deste tipo. Na última parte do capítulo são expostos os conceitos da teoria da elasticidade bidimensional necessários para a compreensão total da formulação de um problema de otimização topológica.

2.1 OTIMIZAÇÃO ESTRUTURAL

A otimização estrutural (OE) faz parte integral do processo de conceitualização do projeto de produto, estando intimamente relacionada ao uso eficiente dos materiais, do qual se derivam processos de produção mais baratos e empresas mais competitivas.

Em termos gerais, a otimização estrutural refere-se ao projeto e análise de estruturas que tenham um desempenho ótimo, ou seja, aquelas que para um material e umas condições de carga determinadas, apresentem as propriedades mecânicas desejadas com a menor quantidade possível de masa.

A seguir são esclarecidos alguns conceitos básicos no contexto da OE:

- **Variáveis de decisão ou variáveis de projeto:** são magnitudes que influem no processo de OE, cujos valores podem mudar na medida em que avança o processo. Propriedades

como a área ou o momento de inércia de uma seção transversal, a densidade do material, o módulo de elasticidade, dimensões como a altura, largura, espessura, são exemplos típicos de variáveis de projeto.

- **Função objetivo ou de avaliação:** estabelece o critério a partir do qual se avaliará se um projeto é melhor do que outro. Geralmente é uma função escalar que depende das variáveis de projeto. Uma solução ótima é aquela que leva esta função a um extremo, ou seja, aquela que minimiza ou maximiza a função objetivo, de acordo com o caso desejado.
- **Domínio de projeto:** é um espaço de projeto com uma forma fixa, que é limitado pelos pontos de aplicação das cargas e pelos pontos de apoio da estrutura. O projeto ótimo da estrutura estará contido neste domínio fixo inicial.
- **Parâmetros do problema:** são magnitudes que permanecem fixas durante o processo de otimização. Geralmente respondem a condições impostas ao problema desde o exterior.
- **Restrições:** São condições que devem se cumprir obrigatoriamente para que uma solução possa ser considerada viável. Devido ao fato de serem uma função das variáveis de projeto, definem os limites dos valores das variáveis, delimitando assim o espaço de busca.

As diferentes aplicações e alcances dos problemas de otimização estrutural podem se classificar em três categorias, de acordo com o tipo de variáveis usadas (KICINGER; ARCISZEWSKI; DE JONG, 2005):

- **Otimização Dimensional (OD) - (*Sizing Optimization*):** esta categoria é aplicada quando são dadas a forma e a topologia do elemento que se quer otimizar. As variáveis de projeto são algumas dimensões modificáveis. É o caso quando se procuram as dimensões ótimas de uma seção transversal, os diâmetros ou espessuras ótimos e, em geral, qualquer mudança de dimensões que represente uma otimização do desempenho do elemento.
- **Otimização de Forma (OF) - (*Shape Optimization*):** procura-se a forma ou contorno ótimo da peça, tendo-se a topologia fixa. Alguns pontos de controle sobre o contorno são

modificáveis. As variáveis do processo são os parâmetros diretamente ligados à geometria. As formas ótimas encontradas não são necessariamente as melhores possíveis.

- **Otimização Topológica (OT) - (*Topology Optimization - Layout Optimization*):** é a mais complexa das categorias. É um método assistido por computador, no qual se procura uma distribuição ótima do material dentro de um volume finito, chamado domínio de projeto. O objetivo é determinar o número, a localização e a forma dos vazios que conterá a topologia de um elemento. Em cada ponto, o material pode variar desde vazio (não há material, formam-se orifícios) até existência total de material (sólido). Dependendo do método de otimização aplicado, estados intermediários do material entre vazio e sólido são aceitáveis.

Por sua vez, considerando o campo de aplicação, a OT pode se dividir em dois tipos:

- **Otimização topológica de estruturas discretas:** o problema consiste em determinar o número, a posição e a inter-conectividade entre os membros da estrutura.
- **Otimização topológica de estruturas contínuas:** o domínio de projeto é discretizado em pequenos elementos, formando uma malha, onde cada elemento pode conter material ou não. Os elementos sem material representam vazios. A forma externa, os limites internos, a quantidade e a posição dos vazios dentro da estrutura são otimizados simultaneamente.

Enquanto que a otimização dimensional é aplicável a treliças, a otimização de forma e a topológica aplicam-se em meios contínuos. Na figura 3 apresenta-se um desenho que ilustra as três categorias mencionadas.

Por ser o tema de interesse neste trabalho, a Otimização Topológica será detalhada.

Figura 3. Otimização Estrutural

2.2 OTIMIZAÇÃO TOPOLÓGICA ESTRUTURAL

Os primeiros estabelecimentos dos princípios da OT são atribuídos ao trabalho com armações desenvolvido por Michell (1904) há mais de um século. Michell desenvolveu um método de otimização de estruturas compostas por barras delgadas com o objetivo de minimizar o peso delas. Quase sete décadas mais tarde, os princípios teóricos da OT foram formulados por Prager e Rozvany (1977), e desde então já são incontáveis as publicações científicas que tratam o tema. Devido à complexidade dos problemas que podem envolver milhares de variáveis, atualmente as técnicas de OT estão baseadas no modelo de elementos finitos com a ajuda de computadores. O trabalho pioneiro neste sentido foi desenvolvido por Bendsøe e Kikuchi, os quais em 1988 propuseram o método da homogeneização ou HBTO (*Homogenization Based Topology Optimization*) (BENDSØE; KIKUCHI, 1988). No ano de 1993, foram apresentados dois métodos que diferem no seu enfoque, mas que conservam sua relevância até hoje. São eles: o método SIMP (*Solid Isotropic Microstructure with Penalization*) (BENDSØE; DIAZ; KIKUCHI, 1993) e o método ESO (*Evolutionary Structural Optimization*) (XIE; STEVEN, 1993). A diferença profunda entre estes métodos consiste em que o primeiro método aplica um modelo de material, enquanto que o segundo não. O modelo de material é uma equação que define a mistura em micro escala de dois ou mais materiais, em que um dos materiais pode ser “vazio” (ausência de material), permitindo que existam estágios intermediários ao passar da condição de vazio à condição de sólido em cada ponto do domínio (DE LIMA, 2002).

Mais recentemente têm surgido vários métodos de OT que aplicam procedimentos bio-inspirados como os GA (*Genetic Algorithms*) (SANDGREN; JENSEN; WELTON, 1990), o método de recozido de materiais (*Simulated Annealing*) (SHIM; MANOOCH, 1997), e o método HCA (*Hybrid Cellular Automata*) (TOVAR, 2005).

Dos diversos métodos de OT existentes, serão tratados com mais detalhe o método SIMP, por ser considerado o método mais popular de OT na atualidade, e os métodos ESO e BESO por apresentarem uma abordagem do problema de OT que de alguma forma se assemelha com o algoritmo desenvolvido neste trabalho.

2.2.1 Método SIMP (*Solid Isotropic Microstructures with Penalization*)

É o método de otimização topológica mais popular entre aqueles que aplicam o conceito de distribuição do material. Foi proposto em (BENDSØE; DIAZ; KIKUCHI, 1993), sendo que o objetivo do método é determinar a quantidade de material de cada um dos elementos finitos, no qual é dividido o volume que vai ser ocupado pelo corpo, de forma que a energia de deformação da estrutura seja mínima. O modelo baseia-se num material cuja densidade “fictícia” $\rho(x)$ pode adotar valores desde zero até um, incluindo todos os valores intermediários. Assim $\rho(x)=1$ significa material sólido e $\rho(x)=0$ significa que não há material. O modelo representa-se com a relação heurística expressada pelas eq.(1) e eq.(2):

$$E_i(x_i) = \rho(x_i)^{p'} E_i^0 \quad p' > 1, \quad (1)$$

$$\int_{\Omega} \rho(x) d\Omega \leq V ; \quad 0 \leq \rho(x) \leq 1, \quad x \in \Omega \quad (2)$$

Nas quais a densidade $\rho(x)$ é a variável de projeto, $E_i^0(x)$ representa o tensor de rigidez do material isotrópico de referência, Ω é o domínio de referência (pode ser 2D ou 3D), e V é um volume fixo máximo admissível.

Devido à necessidade de um projeto final que consista inteiramente de regiões sólidas e vazias, todos os estados intermediários de densidade artificial são penalizados através do uso

da potência $p' > 1$, o qual garante que a rigidez obtida para as densidades intermediárias seja pequena quando comparada com o volume. O modelo implementado por SIMP pode ser considerado um modelo de material se p' satisfizer as eq.(3) e eq.(4):

$$p' \geq \max \left\{ \frac{2}{1-\nu^0}, \frac{4}{1+\nu^0} \right\} \quad (\text{em 2D}) \quad (3)$$

$$p' \geq \max \left\{ 15 \left(\frac{1-\nu^0}{7-5\nu^0} \right), \frac{3}{2} \left(\frac{1-\nu^0}{1-2\nu^0} \right) \right\} \quad (\text{em 3D}) \quad (4)$$

Nas quais ν^0 é o coeficiente de Poisson do material isotrópico determinado. Geralmente se usa uma filtragem de imagem ao final do processo de otimização para facilitar a interpretação dos resultados.

Vantagens (ROZVANY, 2001):

- eficiência computacional;
- é robusto já que pode ser usado para qualquer combinação de restrições;
- a penalização pode ser ajustada livremente;
- é conceitualmente simples, pois não envolve matemáticas complexas;
- não exige homogeneização da estrutura.

Desvantagens:

- os projetos obtidos mostram dependência do valor de p' e da malha de elementos finitos usada;
- não converge necessariamente a uma solução ótima.

2.2.2 Método ESO (*Evolutionary Structural Optimization*)

O ESO é um método heurístico de otimização que imita o crescimento natural de estruturas biológicas, o qual foi proposto em (XIE; STEVEN, 1993). Baseia-se na simples idéia de remover paulatinamente os elementos estruturais ineficientes, de forma que a estrutura evolua até uma topologia ótima, ou seja, até uma estrutura completamente carregada ao limite. Pode-

se aplicar para diferentes funções objetivo, mas sempre usando o método dos elementos finitos para a avaliação do desempenho das estruturas.

Embora, neste caso, a palavra evolucionaria não tenha o mesmo significado do que para os algoritmos evolucionários, a ESO inspira-se também em processos naturais. Exposta por passos, a ESO pode ser resumida assim:

1. Determinar o modelo da distribuição de tensões da estrutura pelo método dos elementos finitos.
2. Usar um critério de rejeição (CR), geralmente a tensão de Von Misses, para avaliar cada elemento.
3. Estabelecer uma razão de rejeição (RR), de acordo com a qual, se um elemento for menor desta RR, será eliminado.
4. Eliminar os elementos ineficientes.
5. Repetir os passos 2 ao 4 até que nenhum elemento deva ser eliminado.
6. Determinar uma razão de evolução e somar a RR.
7. Repetir os passos 1 ao 6 até que seja encontrada a topologia ótima procurada.

Inicialmente a ESO foi formulada como uma OT baseada em tensões, porém, desde a sua aparição, a formulação inicial tem sido modificada de diversas formas e aplicada na solução de novos problemas, principalmente com a variação da função objetivo. Por exemplo, em (DAS; JONES; XIE, 2005) a função objetivo é o fator de intensidade de tensões; em (DAS; JONES; CHANDRA, 2007) a função objetivo são as tensões residuais que se aplicam na otimização de forma num projeto baseado em fadiga; em (HAIBA et al., 2005) a função objetivo é vida à fadiga; em (YANG; XIE; STEVEN, 2005) desenvolve-se um algoritmo bi-direcional e aplica-se na otimização de estruturas contínuas com restrições globais de rigidez em projetos dependentes de cargas.

Embora a ESO se configurasse como um método puramente intuitivo, em (TANSKANEN, 2002) encontra-se uma análise sustentada das bases teóricas desta abordagem. Algumas vantagens e desvantagens desse método são (TANSKANEN, 2002), (ESCHENAUER; OLHOFF, 2001):

Vantagens:

- é um método muito simples de aplicar e de programar com o auxílio de software de elementos finitos;
- usa um único modelo de malha de elementos finitos durante todo o processo de otimização;
- é um método não baseado em gradiente, sendo uma vantagem devido a que o cálculo do gradiente é computacionalmente caro e difícil quando a função objetivo tem descontinuidades.

Desvantagens:

- o método requer que as modificações feitas à estrutura em cada ciclo sejam muito pequenas, e o custo computacional pode rapidamente tornar-se um problema;
- os elementos removidos da estrutura não podem ser adicionados novamente;
- não podem ser adicionadas diferentes restrições, como restrições gerais de tensões ou restrições de deslocamento;
- não apresenta bases teóricas sólidas, por isso o problema de minimização de ESO ainda se encontra sem resolução.

2.2.3 Método BESO (*Bi-directional ESO*)

Com a finalidade de superar os problemas gerados pela impossibilidade de que os elementos removidos possam voltar de novo a fazer parte da estrutura otimizada, Querin et al (2000) propuseram o BESO (*Bi-directional ESO*). Este novo algoritmo permite ao método buscar todas as possíveis direções devido a que, além da remoção de material para eliminar os elementos submetidos a baixas tensões, inclui também a adição de material para aliviar as tensões elevadas.

As regras do algoritmo BESO são as seguintes:

1. Determinar o máximo domínio de projeto que a estrutura pode ocupar.
2. Subdividir o domínio físico em uma densa malha de elementos finitos.

3. Definir todas as restrições cinemáticas de contorno.
4. Selecionar o número mínimo de elementos que conectam as cargas com os apoios ou especificar um domínio de projeto inicial que se encaixe dentro do domínio máximo possível.
5. Dar aos elementos não inclusos no domínio inicial um valor de propriedade igual a zero.
6. Especificar o critério de otimização.
7. Desenvolver uma análise linear de elementos finitos para a estrutura.
8. Remover os elementos das regiões submetidas a baixas tensões que satisfaçam a eq.(5) e adicionar os elementos que satisfaçam a eq.(6) às regiões sobrecarregadas:

$$\sigma_e \leq (RR)\sigma_{\max} \quad (5)$$

$$\sigma_e \geq (IR)\sigma_{\max} \quad (6)$$

nas quais σ_e é a tensão de Von Misses do elemento, ou qualquer outro critério da estrutura que se defina; RR é a razão de rejeição e IR é a razão de inclusão. $0 \leq RR \leq 1$ e $0 \leq IR \leq 1$. As duas razões dependem do número de estado estável SS e do número oscilatório ON .

9. Caso se encontre um estado no qual nenhum elemento no domínio de projeto cumpra nem a eq.(5) nem a eq.(6), atingiu-se um estado estável e o número de estado estável SS deve ser aumentado em 1 e voltar ao passo 8.
10. Quando um grupo de elementos é removido e adicionado repetidamente após várias iterações, atingiu-se um estado oscilatório. O número oscilatório ON deve ser aumentado em 1 e voltar ao passo 8.
11. Os passos de 7 ao 9 são repetidos até quando o índice de desempenho PI atinja seu valor mínimo, ou até quando os limites do critério usado forem atingidos.

Em comparação com o método ESO, o tempo necessário para otimizar uma estrutura em 3D diminui no algoritmo BESO. Além, graças à capacidade de manipular múltiplas cargas em 2D e 3D, os resultados do BESO são muito mais confiáveis (ESCHENAUER; OLHOFF, 2001).

2.3 DIFICULDADES DA OTIMIZAÇÃO TOPOLÓGICA

Em todos os métodos de OT existentes até hoje, persistem duas possíveis instabilidades numéricas (TOVAR, 2005):

- Tabuleiros de xadrez: refere-se a regiões na estrutura final nas quais se alternam elementos pretos (sólidos) e brancos (vazios). A figura 4 ilustra este conceito.

Figura 4. Tabuleiro de xadrez em OT

- Dependência da malha: Refere-se à obtenção de estruturas que diferem qualitativamente para diferentes tamanhos da malha de elementos finitos assumida para o domínio de projeto.

2.4 FORMULAÇÃO DO PROBLEMA DE OTIMIZAÇÃO TOPOLÓGICA

Neste trabalho, o algoritmo desenvolvido é aplicável à otimização topológica de estruturas contínuas. A seguir, detalham-se os conceitos básicos da teoria da elasticidade bidimensional, que esclarecem a formulação do problema de otimização resolvido. Os conceitos aqui explicados estão baseados nas referências (PROENÇA, 2006) e (VICTORIA, 2006).

Considere um sólido, inserido no espaço tridimensional e idealizado como um meio contínuo de volume V e superfície de contorno Γ . Submetido tanto à ação de forças externas distribuídas por unidade de volume que atuam em pontos no interior do corpo e possuem componentes b_x , b_y e b_z , quanto às forças externas distribuídas por unidade de superfície que atuam em pontos do contorno do corpo e possuem suas componentes p_x , p_y e p_z . Assim como ilustra a figura 5.

Figura 5. Idealização do meio contínuo

A modo de simplificação, na figura 6 apresenta-se o diagrama de corpo livre de um elemento no interior de sólido no plano xy . Para esse estado de cargas, as condições de equilíbrio expressam-se de acordo com as eq.(7) e eq.(8):

$$\frac{\partial \sigma_x}{\partial x} + \frac{\partial \tau_{xy}}{\partial y} + b_x = 0 \quad (7)$$

$$\frac{\partial \tau_{yx}}{\partial x} + \frac{\partial \sigma_y}{\partial y} + b_y = 0 \quad (8)$$

Nas quais σ_x e σ_y são as tensões normais e $\tau_{xy} = \tau_{yx}$ são as tensões cisalhantes.

Figura 6. Diagrama de corpo livre elemento interior

Deslocamentos:

Para as condições dadas de tensão e deformação planas é válida a hipótese de que todas as seções transversais ao eixo z deformam-se no seu plano e de forma idêntica. Assim o campo dos deslocamentos de uma seção contida no plano xy pode ser inteiramente definido se conhecem-se os deslocamentos nas direções x e y de todos seus pontos. Essas grandezas podem ser definidas pelo vetor, eq.(9):

$$\underline{d}(x, y) = \begin{Bmatrix} u(x, y) \\ v(x, y) \end{Bmatrix} \quad (9)$$

no qual $u(x, y)$ e $v(x, y)$ são os deslocamentos de um ponto nas direções dos eixos x e y respectivamente.

Deformações:

Restringindo-se a modelagem aos regimes de pequenas deformações, na figura 7 ilustram-se três casos simples de deformações no plano.

Figura 7. Deformação simples no plano

Para estes regimes é válido considerar que os comprimentos dos segmentos AB' e AC' na figura 7 podem ser confundidos com as suas projeções nas direções dos eixos x e y , respectivamente e que a distorção angular pode ser determinada como a tangente do próprio ângulo, assim:

$$\epsilon_x \approx \frac{\Delta u_x}{\Delta x} \quad (10)$$

$$\epsilon_y \approx \frac{\Delta u_y}{\Delta y} \quad (11)$$

$$\gamma_{xy} \approx \theta_1 + \theta_2 \quad (12)$$

Avaliando o limite quando Δx e Δy tendem a zero, as relações de compatibilidade são:

$$\epsilon_x \approx \frac{\partial u}{\partial x} \quad (13)$$

$$\epsilon_y = \frac{\partial v}{\partial y} \quad (14)$$

$$\gamma_{xy} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \quad (15)$$

Finalmente o vetor de deformações significativas para tensão e deformação planas se define como:

$$\underline{\varepsilon} = [\varepsilon_x, \varepsilon_y, \gamma_{xy}]^T \quad (16)$$

Tensões:

As componentes das tensões podem se expressar como o vetor $\underline{T}$:

$$\underline{T} = [\sigma_x, \sigma_y, \tau_{xy}] \quad (17)$$

Relação constitutiva:

A equação constitutiva válida para o regime elástico-linear permite deduzir a relação entre as tensões e as deformações da qual obtém-se a seguinte relação matricial:

$$\underline{\varepsilon} = \underline{\underline{D}} \underline{T} \quad (18)$$

Na qual $\underline{\underline{D}}$ é o tensor constitutivo de flexibilidade.

Função de energia interna de deformação ou mean compliance:

A função *mean compliance* é um indicador para determinar o nível de energia da deformação interna contido em uma estrutura, podendo-se também interpretar como o inverso da rigidez.

A função de energia interna de deformação está determinada pela eq.(19):

$$C = \underline{p}^T \underline{d} \quad (19)$$

Na qual $\underline{d}$ é o vetor de deslocamentos e $\underline{p}$ é o vetor das forças externas que pode se expressar a sua vez pela eq.(20):

$$\underline{\underline{K}} \underline{d} = \underline{p} \quad (20)$$

Na qual $\underline{\underline{K}}$ é a denominada matriz de rigidez. Essa função proporciona um valor matemático que define o grau de otimalidade de uma estrutura em relação com a sua rigidez.

Formulação:

Considerando como função objetivo a energia interna de deformação, o problema de otimização topológica objeto deste trabalho pode-se formular assim (YANG; XIE; STEVEN, 2005):

Minimizar:

$$f = C(\underline{x}) = \underline{p}^T \underline{d} \quad (21)$$

Sujeito a:

$$g = W^* - \sum_{i=1}^n W_i x_i = 0 \quad (22)$$

$$x_i \in \{0,1\}$$

Na qual $C(\underline{x})$ é a energia interna de deformação total, $\underline{p}$ é o vetor de carregamentos, $\underline{d}$ é o vetor de deslocamentos, W_i é o peso de um elemento e W^* é o peso total desejado e n é o numero total de elementos no sistema.

CAPÍTULO 3

3. OTIMIZAÇÃO MULTI-OBJETIVO

Este capítulo trata dos conceitos básicos de otimização multi-objetivo, destacando entre eles, a definição da Frente Ótima de Pareto. Faz-se um resumo dos métodos de otimização multi-objetivo convencionais mais importantes, expondo suas vantagens e desvantagens quando comparados aos métodos inspirados na natureza. Detalham-se as técnicas de otimização multi-objetivo e bio-inspiradas mais importantes na atualidade e entre elas, os dois algoritmos que foram usados neste trabalho para desenvolver comparações de desempenho.

3.1 FORMULAÇÃO DO PROBLEMA DE OTIMIZAÇÃO MULTI-OBJETIVO

Na maioria dos problemas de otimização do mundo real, é necessário tomar decisões que implicam atingir mais de um objetivo. Quando esses objetivos referem-se à minimização ou maximização de funções, está se falando da otimização multi-objetivo.

A otimização multi-objetivo define-se como o problema de encontrar um vetor de variáveis de decisão que satisfaça umas restrições e otimize uma função vetor, cujos elementos representem as funções objetivo. Um problema de otimização multi-objetivo (POMO) pode ser formulado da seguinte forma (DEB, 2001):

Maximizar / minimizar:

$$f_m(x) \quad m = 1, 2, \dots, M \quad (23)$$

Restrito a:

$$g_j(x) \geq 0 \quad j = 1, 2, \dots, J \quad (24)$$

$$h_k(x) = 0 \quad k = 1, 2, \dots, K \quad (25)$$

$$x_i^L \leq x_i \leq x_i^U \quad i = 1, 2, \dots, n$$

Na eq.(23) x é o vetor das variáveis de decisão; $x = (x_1, x_2, \dots, x_n)^T$ e $f_m(x)$ são as m funções objetivo. x_i^L e x_i^U representam os valores mínimo e máximo aceitáveis respectivamente para a variável x_i . Esses valores definem a fronteira do espaço de busca. A eq.(24) representa as J desigualdades g_j e a eq.(25) representa as K igualdades h_k , todas elas conhecidas como funções de restrição.

A seguir, ampliam-se os conceitos e definições básicas da terminologia usada em otimização multi-objetivo (NOCEDAL; WRIGTH, 1999):

Variáveis de decisão: são aqueles parâmetros que se modificam durante o processo de otimização; podem ser contínuas, discretas ou inteiras. As variáveis contínuas geralmente pertencem a um conjunto de vetores com componentes reais, enquanto que as discretas são usadas em problemas nos quais a solução é um objeto entre um conjunto finito de objetos. As variáveis inteiras são aquelas que somente fazem sentido quando apresentam valores inteiros.

Função Objetivo ou de Avaliação: é cada uma das funções que se quer otimizar.

Funções de restrição: são funções de igualdade ou desigualdade que descrevem situações não admissíveis do projeto.

Espaço de busca: é o espaço ou região que inclui todas as soluções possíveis do problema de otimização. É definido pelos valores x_i^L e x_i^U e pelas funções de restrição.

Pontos Ótimos: são os pontos representados por vetores de variáveis de decisão que pertencem à Frente Ótima de Pareto e que satisfazem as restrições.

Valores ótimos: são os valores das funções objetivo num ponto ótimo.

Soluções Ótimas de Pareto: são os conjuntos de soluções ótimas para um POMO. Uma explicação mais detalhada será dada a seguir.

3.2 SOLUÇÕES ÓTIMAS DE PARETO

Um caso muito freqüente nas aplicações de otimização multi-objetivo e, particularmente, em problemas de engenharia apresenta-se quando os objetivos a atingir são contraditórios. Essa característica faz com que esses problemas não tenham uma solução única, pelo contrário, existe um conjunto de soluções, todas elas ótimas e das quais não se pode dizer que uma delas seja melhor do que as outras. Esse conjunto de soluções é conhecido como **Soluções Ótimas de Pareto**, devendo seu nome ao economista italiano Vilfredo Pareto (1848-1923).

No contexto econômico, um Ótimo Pareto é uma situação na qual ninguém pode alcançar um aumento na sua utilidade total, sem que isso implique na diminuição da utilidade de outro.

Matematicamente a otimalidade de Pareto define-se da seguinte forma (CHINCHULUUN; PARDALOS, 2007):

Ótimo de Pareto Global: Sendo S^* o espaço de busca. Um ponto $x^* \in S^*$ com $f(x^*)$ é chamado Ótimo de Pareto Global (ou eficiente, ou não dominado, ou não inferior) se, e somente se, não existe um ponto $x \in S^*$, tal que $f_i(x) \leq f_i(x^*)$ para todos os $i = 1, \dots, k$ e $f_j(x) < f_j(x^*)$ para pelo menos um dos índices $j \in \{1, 2, \dots, k\}$.

O princípio dos Ótimos de Pareto pode se aplicar a problemas de otimização multi-objetivo através do critério de dominância. Uma solução a um problema multi-objetivo faz parte do conjunto de soluções Ótimas de Pareto, caso seja uma solução **não dominada**. Para estabelecer facilmente se uma solução é dominada ou não, basta fazer a avaliação das condições de dominância, as quais se definem do seguinte modo (DEB, 1999):

Uma solução x_1' diz-se que domina a outra solução x_2' , se as duas condições seguintes se cumprem:

- a solução x_1' não é pior do que a x_2' em todos os objetivos;
- a solução x_1' é estritamente melhor do que a solução x_2' em pelo menos um objetivo.

Na otimização multi-objetivo existe um conjunto de soluções que, após a avaliação das condições de não dominância, resultam ser soluções não dominadas, e é por isso que o conceito de otimalidade refere-se a um conjunto de soluções ótimas. Na maioria dos casos, não é uma tarefa fácil encontrar expressões analíticas que representem a linha ou a curva que contenha as soluções ótimas, pelo que normalmente se calculam os pontos Pareto e seu valor nas funções objetivo.

Aplicando esses conceitos, um conjunto de soluções ótimas de Pareto é ilustrado na figura 8. Dentre as possíveis soluções de um POMO, muitas delas são soluções adequadas, mas só algumas delas pertencerão ao conjunto de soluções Ótimas de Pareto. Essas estão na fronteira identificada com a curva da figura 8; essa fronteira é conhecida como a Frente Ótima de soluções de Pareto ou Frente Ótima de Pareto (FOP). Aquelas soluções que oferecem o melhor balanceamento entre os objetivos propostos fazem parte dessa frente.

No caso da figura 8, é necessário maximizar simultaneamente as funções objetivo 1 e 2. Assim, as soluções A, B e D são soluções Ótimas de Pareto. A solução C embora cumpra com as restrições do problema, não é uma solução Ótima de Pareto. De acordo com os objetivos de otimização propostos, por exemplo, a solução C é melhor do que a solução D na função objetivo 1, mas a solução C **não é melhor do** que a solução D na função objetivo 2. Daí conclui-se que a solução D não é dominada pela solução C. Já a solução B é melhor do que a solução C nos dois objetivos, e é por isso que a solução C é dominada pela solução B.

Figura 8. Frente Ótima de Pareto

Dadas essas condições, enquanto que a principal meta da otimização mono-objetivo é encontrar o ótimo global, para a otimização multi-objetivo existem duas metas que qualquer algoritmo de otimização pretende atingir (DEB, 1999):

- dirigir a busca ao redor da região onde se encontra o ótimo de Pareto Global;
- encontrar tantas soluções ótimas de Pareto e tão diversas quanto for possível.

A figura 9 ilustra o conceito das metas a serem atingidas por um algoritmo de otimização multi-objetivo para o caso de um POMO com dois objetivos. Na figura 9, FOP é a Frente Ótima de Pareto do problema, e FOP* é a Frente Ótima obtida com o algoritmo.

Figura 9. Metas da otimização multi-objetivo

3.3 MÉTODOS CONVENCIONAIS DE OTIMIZAÇÃO MULTI-OBJETIVO

Desde os anos 50, na área de pesquisa de operações, tem se desenvolvido uma variedade de métodos para a solução de POMO, conhecidos como “clássicos” ou convencionais, baseados na lógica formal ou programação matemática.

Alguns dos métodos convencionais mais representativos são: a programação linear (*linear programming*) (DANTZIG; THAPA, 1997); o método de somatório de pesos (*Weighted Sum*); e o método de programação de metas (*Goal Programming*) (EHRGOTT, GANDIBLEUX, 2002). Eles são descritos brevemente abaixo.

3.3.1 Programação linear

É um dos métodos mais aplicados em pesquisa operacional. Refere-se à programação feita com um modelo matemático, no qual todas as funções do modelo são lineares. Esse método pode ser aplicado em uma grande variedade de problemas lineares, tal como encontrar a melhor mistura de produtos para produzir quando os recursos são limitados ou em planejamentos considerando os tempos e recursos disponíveis, dentre outras muitas aplicações.

3.3.2 Somatório dos pesos

Consiste em criar uma função objetivo única, mediante a somatória de todas as funções objetivo multiplicadas por um respectivo peso. Esses pesos são parâmetros do problema também, e a sua seleção depende da importância que o usuário lhe atribui a cada objetivo. Com exceção dos POMO com FOP não convexas, para os quais o algoritmo não consegue encontrar todas as soluções, o algoritmo precisa de várias iterações para encontrar diferentes soluções Ótimas de Pareto.

3.3.3 Programação de metas

Esta técnica pode ser considerada como uma extensão do método de programação linear. Nela se estabelece um número específico de metas para cada um dos objetivos. Às indesejadas desviações dos valores das metas, se lhes atribuí um peso de acordo com a sua importância relativa no processo de decisão. Se não existe uma solução factível que alcance as metas de todos os objetivos, então se busca uma solução que minimize o total das penalizações acumuladas.

3.3.4 Dificuldades dos métodos convencionais de otimização multi-objetivo

Os métodos convencionais têm um grande argumento a seu favor, pois geralmente apresentam teoremas que provam sua convergência para uma solução ótima. Porém, demonstram dificuldades para resolver alguns tipos específicos de POMO, por exemplo, quando a FOP é não convexa ou não conexa; e outros precisam que as funções objetivo e as restrições sejam diferenciáveis (COELLO, 2006).

Por outro lado, a maioria dos métodos clássicos resolve tratar a dificuldade de se trabalhar com vários objetivos, transformando-os num objetivo único. E como consequência, a solução ótima encontrada para esse problema convertido artificialmente num de objetivo único, tem uma forte dependência da seleção dos parâmetros feita pelo usuário (SARKER, 2002).

A sua forma de operar faz com que seja possível encontrar somente uma solução ótima por vez que pode não ser uma solução ótima global. Caso se deseje obter várias soluções ótimas, é necessário rodar o algoritmo muitas vezes, esperando-se encontrar uma solução ótima diferente a cada vez.

3.4 OTIMIZAÇÃO MULTI-OBJETIVO INSPIRADA NA NATUREZA

A computação inspirada na natureza, cujo objetivo é usar idéias extraídas da natureza para desenvolver ferramentas computacionais e implementar técnicas para a solução de problemas (DE CASTRO, 2007), faz parte do campo mais amplo da computação natural.

A solução de problemas de otimização multi-objetivo logicamente também tem sido objeto de pesquisas no desenvolvimento de técnicas bio-inspiradas que se apresentam como alternativas às técnicas convencionais. Na continuação, são detalhadas duas técnicas destacadas neste campo - os algoritmos evolucionários e a inteligência de enxame.

3.4.1 Algoritmos evolucionários

Os algoritmos evolucionários (AE) são técnicas heurísticas usadas na solução de problemas, baseadas na emulação dos mecanismos naturais de seleção natural, evolução e adaptação.

Duas características básicas que os distinguem de outras técnicas são que (SARKER, 2002):

- estão baseados em populações;
- existe comunicação e intercâmbio de informação entre os indivíduos da população, como resultado dos processos de seleção e/ou recombinação.

Graças a essas características, os AE possuem uma habilidade genérica para manipular satisfatoriamente problemas do mundo real de grande tamanho e complexidade, e isso tem contribuído para que o interesse neles tenha aumentado rapidamente nos últimos 20 anos. Particularmente os AE adaptam-se muito bem às aplicações em otimização multi-objetivo, entre outras coisas, por basear-se em populações, o que lhes permite manipular simultaneamente um conjunto de possíveis soluções, conseguindo assim encontrar várias soluções ótimas de Pareto em uma única rodada do algoritmo. Além dessa vantagem sobre os algoritmos clássicos, outras características favoráveis dos AE são que (EHRGOTT, 2002):

- são fáceis de programar e implementar;
- desempenham-se muito bem na resolução de problemas discretos e combinacionais;
- geralmente podem encontrar soluções bem próximas ao ótimo (ou no ótimo mesmo) e conseguem fugir dos mínimos locais;
- têm demonstrado serem mais robustos e menos suscetíveis à forma e à continuidade da FOP, quando comparados com os métodos convencionais.

Os AE incluem técnicas como a Programação Evolucionária, a Programação Genética e logicamente, os Algoritmos Genéticos (AG), que são o ramo mais conhecido e com mais desenvolvimentos na área de otimização multi-objetivo, pelo qual se detalham a seguir.

3.4.2 Algoritmos Genéticos (AG)

Os AG são técnicas estocásticas de otimização e busca em espaços complexos que aplicam algoritmos matemáticos baseados em processos naturais de seleção e transmissão de informação genética. Trabalham com populações de indivíduos representados por cromossomos que são submetidos a imitações dos processos de seleção natural, cruzamento, recombinação e mutação, e se aplicam à solução de problemas que envolvem um grande número de parâmetros.

Os algoritmos genéticos foram propostos inicialmente por John Holland (1975). Baseado no seu conhecimento na área de programação e influenciado pelas obras de naturalistas destacados, Holland desenvolveu as bases dos algoritmos genéticos ao propor uma técnica que, mediante sistemas artificiais, permite imitar o comportamento adaptativo dos sistemas naturais. Contudo, foi só na década de noventa, logo após a publicação do trabalho de Goldberg (1989), que os AG encontraram sua aplicação dentro da pesquisa operacional. Desde então, os campos de aplicação que encontraram os AG são muitos, em quantidade e variedade.

Devido aos seus princípios procederem de uma analogia com a biologia, Holland usou vocábulos desse campo para nomear as principais definições dos AG. São eles:

Cromossomos: são as estruturas básicas manipuladas. Representam informação relativa às variáveis do problema mediante uma cadeia de bits. No caso de otimização, seriam os vetores de variáveis de decisão, sendo que cada vetor representa uma possível solução do problema.

Gene: é a menor unidade de informação dos algoritmos genéticos. Os cromossomos são formados por genes, representando cada parâmetro de acordo com o alfabeto utilizado.

População: é o conjunto de cromossomos ou indivíduos no espaço de busca.

Geração: é o número da iteração completa que executa o AG e que gera uma nova população.

Operações Genéticas: são as operações que o AG realiza sobre cada indivíduo da população.

Seleção: é o processo de selecionar indivíduos da população com os melhores desempenhos, o que lhes oferece vantagens para serem escolhidos como pais da próxima geração de modo que seus genes “superiores” sejam transmitidos.

Cruzamento: é o mecanismo pelo qual dois indivíduos da população considerados pais, compartilham e intercambiam a sua informação genética. Trata-se de substituir alguns genes de um dos pais pelos genes do outro.

Mutação: é o processo de selecionar aleatoriamente um gene de um indivíduo e mudar seu valor.

Considerando a emulação que os AG fazem dos processos naturais de seleção natural ao transformar uma população inicial de indivíduos em uma nova geração, usando os processos de seleção, cruzamento e mutação, as operações básicas do algoritmo genético simples são (MITCHELL, 1995):

1. Começar com uma população inicial de N indivíduos que pertençam ao espaço de busca, selecionados aleatoriamente. Cada indivíduo da população representa uma possível solução ao problema proposto.
2. Avaliar os N indivíduos da população de acordo com a função objetivo.
3. Segundo sua capacidade de desempenho, selecionar os melhores indivíduos da população, os quais serão os pais da próxima geração de indivíduos.
4. Gerar N filhos mediante o cruzamento dos pais.
5. De acordo com a probabilidade de mutação, mudar alguns dos N filhos.
6. Substituir a população anterior pelos N filhos.
7. Se a condição necessária para detiver o algoritmo é atingida, deve-se terminar o algoritmo. Em outro caso, voltar ao passo 2.

3.4.3 Otimização multi-objetivo com Algoritmos Genéticos

As vantagens da aplicação dos AG em otimização multi-objetivo são particularmente aproveitadas quando se apresenta alguma das seguintes situações (MITCHELL, 1995):

- o espaço de busca é muito grande;
- não se sabe se a solução ótima encontra-se num pico ou num vale;
- uma solução de boa qualidade deve ser encontrada rapidamente;
- a função de avaliação apresenta ruído;
- o problema não é bem compreendido;
- não se conhece um método de otimização que trabalhe melhor no problema que se deseja resolver.

Porém, o AG simples tem limitações na aplicação à solução de POMO devido ao fato de que sua definição inicial e a de seus operadores de seleção, cruzamento e mutação referem-se a problemas mono-objetivo. Como consequência, aproximadamente duas décadas atrás, surgiu a necessidade de desenvolver novas técnicas que permitissem aplicar os conceitos fundamentais dos algoritmos genéticos na solução de POMO. A resposta a essa necessidade foram os chamados Algoritmos Evolucionários de Otimização Multi-objetivo (AEOM).

Dentre os AEOM destacam-se o VEGA (*Vector Evaluated Genetic Algorithm*), trabalho pioneiro na aplicação prática dos fundamentos dos AG, proposto por Schaffer (1984). Posteriormente desenvolveram-se algoritmos que aplicam o conceito de elitismo, como o SPEA (*Strength Pareto Evolutionary Algorithm*) e seu sucessor, o aperfeiçoado SPEA2, sendo que o primeiro foi proposto por Zitzler; e o segundo, em parceria com Laumanns e Thiele (2002); outros algoritmos estão baseados no conceito de dominância como o MOGA (*Multi-objective Genetic Algorithm*) de Fonseca e Fleming (1993), o NSGA (*Non-dominated Sorting Genetic Algorithm*) (DEB, 1999) e o NSGA-II (*Elitist Non-dominated Sorting Genetic Algorithm*) (DEB et al., 2002).

Devido ao seu excelente desempenho, o NSGA-II tem se convertido em uma referência quando se trata de avaliar e comparar os resultados obtidos com novos algoritmos desse tipo (PRADYUMN; DEB, 2007), (COELLO, 2006), (KUMAR; BANDYOPADHYAY; KUMAR, 2007). Esse algoritmo também foi usado neste trabalho como referência em comparação do desempenho do algoritmo projetado. Seus passos fundamentais são explicados abaixo.

3.4.3.1 Elitist Non-dominated Sorting Genetic Algorithm: NSGA-II

Foi desenvolvido por Deb e outros (2002) buscando superar algumas falhas do original NSGA, como a alta complexidade computacional, a carência de elitismo e a necessidade de especificar o parâmetro σ_{share} . O algoritmo pode ser resumido nos seguintes passos:

1. Criar uma população inicial aleatória P_o de N indivíduos. Avaliar as funções objetivo.
2. Ordenar a população, baseando-se em critérios de não dominância. O método de ordenamento por não dominância deste algoritmo constitui a primeira melhoria em comparação com o NSGA. Inicia-se com a primeira solução q , a qual é armazenada em Dp , que é o conjunto de soluções não dominadas. A partir da segunda solução até a solução N , cada solução p é comparada por dominância com cada uma das soluções que pertencem a Dp . Se uma solução p domina qualquer solução q de Dp , essa solução q é removida de Dp ; mas se a solução p for dominada por qualquer solução q de Dp , a solução p é ignorada. Se a solução p não for dominada por nenhuma solução q , essa solução p é armazenada em Dp . Assim a população de Dp aumenta só pela união de soluções não dominadas. Quando testadas todas as N soluções, as que permanecem em Dp conformam o primeiro conjunto de soluções não dominadas F_1 ou primeiro nível de não dominância. O mesmo procedimento repete-se com as soluções restantes e conforma-se o segundo conjunto de soluções não dominadas F_2 ou segundo nível de não dominância, e o procedimento continua até quando todas as soluções tenham sido classificadas dentro de um determinado conjunto de soluções não dominadas
3. Atribuir a cada solução um valor de aptidão igual ao seu nível de não dominância, para garantir o mesmo potencial reprodutivo para todas elas.
4. Aplicar os procedimentos usuais de seleção por torneio binário, cruzamento e mutação para criar uma nova população Q_o . Unir estas duas populações em um conjunto R_o de tamanho $2N$. Os quatro passos anteriores constituem o procedimento inicial, para as gerações seguintes o algoritmo trabalha com R_t e Q_t .
5. Formar uma população conjunta R_t conformada por $P_t \cup Q_t$. Ordenar esta população conjunta de acordo com critérios de não dominância e obter os conjuntos de soluções não dominadas.

6. Conformer uma nova população P_{t+1} com soluções tomadas dos conjuntos de soluções não dominadas, começando com o preenchimento de P_{t+1} pelo conjunto F_1 , logo o F_2 e assim por diante. Se o tamanho de F_1 for menor do que N , todas as soluções que pertencem a F_1 fazem parte de P_{t+1} ; e a quantidade de soluções que faltam para completar N indivíduos em P_{t+1} é escolhida dos restantes níveis de não dominância de acordo com dois critérios de ordenamento: o valor de aptidão e a distância de multidão. Por outro lado se o tamanho de F_1 for maior do que N , selecionam-se os melhores N indivíduos de F_1 de acordo com o ordenamento por distância de multidão.
7. Aplicar seleção por torneio binário, cruzamento e mutação para os indivíduos de P_{t+1} e gerar uma geração Q_{t+1} .
8. Comprovar se foi atingido o número máximo de gerações. Caso fora se deve parar; em caso contrário, voltar ao passo 5.

A figura 10 ilustra o procedimento do NSGA-II.

Figura 10. Procedimento do NSGA-II

3.4.4 Inteligência de enxame

O termo inteligência de enxame refere-se às técnicas que projetam algoritmos ou instrumentos de solução distribuída de problemas inspirados no comportamento social coletivo de colônias de insetos ou de outras sociedades de animais (BONABEAU; DORIGO; THERAULAZ, 1999).

Dentre as diferentes aproximações destacam-se a otimização com colônias de formigas (*ant colony optimization*), com fundamentos no comportamento social de insetos (DE CASTRO; VON ZUBEN, 2004); e a otimização com enxame de partículas ou PSO (*Particle Swarm Optimizer*) que se baseia na sociocognição (KENNEDY; EBERHART, 1995).

Por ser relevante no desenvolvimento deste trabalho, a seguir amplia-se a explicação da técnica PSO.

3.4.5 Otimização por enxame de partículas – PSO (*Particle Swarm Optimizer*)

Esta nova técnica foi proposta por Kennedy e Eberhart (1995). A motivação inicial desses pesquisadores foi a de explorar analogias simples de interações sociais e habilidades coletivas, diferente dos métodos evolucionários que trabalham somente com habilidades cognitivas individuais.

O PSO é uma técnica baseada em populações chamadas enxames, conformadas por partículas que são colocadas dentro do espaço de busca de um problema, no qual cada posição representa uma possível solução do problema. Neste espaço, cada partícula encarrega-se de avaliar a função objetivo na sua própria localização, e para a seguinte iteração aproveita a informação do histórico mantido sobre sua melhor localização encontrada até então, junto com um componente social que envolve a capacidade de comunicação entre as partículas do enxame. Este intercâmbio de informação se vê representado pelo conhecimento da melhor posição encontrada por todas as partículas até o momento. Após um número de iterações, o

enxame como um todo se movimenta em uma região próxima ao ótimo da função objetivo do problema.

Para cada partícula do enxame são relevantes os seguintes três vetores D – dimensionais, sendo D a dimensão do espaço de busca:

- o vetor da posição atual $\vec{x}_i$ é um conjunto de coordenadas que definem uma posição no espaço da partícula i ;
- o vetor da melhor posição histórica $pbest_i$ é um conjunto de coordenadas que definem a posição com o melhor valor de avaliação encontrado até então pela partícula i ;
- o vetor da velocidade $\vec{v}_i$ que pode ser visto como o tamanho do passo que cada partícula i .

Outra definição importante é:

- $\vec{p}_g$ é o termo usado para denominar a melhor posição histórica encontrada por todos os membros do enxame quando comparada com todas as iterações anteriores para todas as partículas.

A cada iteração k do algoritmo, cada uma das partículas i do enxame movimenta-se através do espaço de busca com uma velocidade v_i até uma nova posição x_i . A cada iteração, as velocidades e posições são ajustadas de acordo com as eq.(26) e eq.(27):

$$v_i(k+1) = wv_i(k) + c_1r_1(k)(pbest_i(k) - x_i(k)) + c_2r_2(k)(p_g(k) - x_i(k)) \quad (26)$$

$$x_i(k+1) = x_i(k) + v_i(k+1)\Delta t \quad (27)$$

Nas quais w é conhecido como o peso de inércia, c_1 e c_2 são denominadas as constantes de aceleração cognitiva e social respectivamente; r_1 e r_2 são números aleatórios uniformemente distribuídos entre 0 e 1, usados para emular o comportamento estocástico dos enxames naturais.

O algoritmo PSO original é o seguinte (POLI; KENNEDY; BLACKWELL, 2007):

1. Iniciar um arranjo das partículas da população com posições e velocidades aleatórias dentro do espaço de busca.
2. Laço.
3. Avaliar a função objetivo para cada partícula.
4. Comparar o valor da função objetivo atual com o valor de $pbest_i$. Se o valor atual da função objetivo é melhor do que o valor de $pbest_i$, então colocar o atual valor em $pbest_i$ e $\vec{p}_i$ igual ao valor atual de $\vec{x}_i$.
5. Identificar a partícula do enxame com o melhor valor de função objetivo encontrado até o momento e colocar as coordenadas desta posição em $\vec{p}_g$.
6. Ajustar a velocidade $\vec{v}_i$ e posição $\vec{x}_i$ de cada partícula de acordo com as eq.(26) e eq.(27).
7. Se o critério de conclusão é atingido, abandone o laço.
8. Fim do laço.

Assim como nos algoritmos genéticos, o que no começo só parecia uma aproximação engenhosa, rapidamente se tornou em uma técnica muito interessante com múltiplas aplicações; levando a um grupo de pesquisadores, sempre crescente, a adaptar os princípios básicos da PSO para serem aplicados à solução de diversos tipos de problemas e é claro, recentemente foram modificados para serem aplicados na solução de problemas de otimização multi-objetivo. Deste modo, por exemplo, em (LI, 2003) propõe-se o algoritmo NSPSO (*Non-dominated Sorting Particle Swarm Optimizer*); em (COELLO; TOSCANO; SALAZAR, 2004) apresenta-se o algoritmo MOPSO (*Multiobjective Particle Swarm Optimization*); e em (KUMAR; BANDYOPADHYAY; KUMAR, 2007) apresenta-se o algoritmo TV-MOPSO (*Time Variant Multi-Objective Particle Swarm Optimization*).

Na continuação se explicará o algoritmo NSPSO (*Non-dominated Sorting Particle Swarm Optimizer*), o qual foi usado como referência de comparação com os resultados desta pesquisa.

3.4.5.1 NSPSO (*Non-dominated Sorting Particle Swarm Optimizer*)

Este algoritmo foi proposto por Li (2003), estendendo os princípios básicos do PSO para fazer um melhor uso do histórico de cada partícula e aplicar processos efetivos de comparação por não dominância. O algoritmo pode ser resumido nos seguintes passos:

1. Iniciar a população de N partículas P_i em posições $\vec{x}_i$ e velocidades $\vec{v}_i$ aleatórias dentro do espaço de busca ($\vec{v}_i$ tem uma probabilidade de 0,5 de ser especificada em uma direção contrária). A melhor posição pessoal $pbest_i$ inicialmente tem o mesmo valor de $\vec{x}_i$. Estabelece-se um limite máximo para a velocidade correspondente aos limites das variáveis de decisão. Armazenar a população inicial em PList.
2. Avaliar a posição de cada partícula do enxame. Contador $k = 0$.
3. $k = k + 1$.
4. Usar o método de classificação de não dominância usada em (DEB et al., 2002) para identificar as partículas que representam soluções ótimas de Pareto e armazená-las em NDomPList.
5. Usar o método de *Niche Count* para calcular o parâmetro de aglomeração de cada partícula e organizar NDomPList de acordo com este critério.
6. Para $i = 0 : N$
 - a) Selecionar aleatoriamente um $\vec{p}_g$ para a i -ésima partícula dentre os melhores indivíduos da NDomPList organizada.
 - b) Calcular a nova velocidade e a nova posição para cada partícula de acordo com as eq.(26) e eq.(27) respectivamente.
 - c) Adicionar a nova partícula P_i e a sua nova posição x_i a uma população temporal armazenada em nextPopList. Este arquivo tem uma dimensão $2N$ devido à coexistência de P_i e x_i nele.
 - d) Se $i < N$, ir ao item a).
7. Usar o método de classificação de não dominância usada em (DEB et al., 2002) para identificar as partículas que representam soluções Ótimas de Pareto em nextPopList e

armazená-las em NDomPList. As partículas restantes são armazenadas em nextPopListRest.

8. Esvaziar PList para a próxima iteração.
9. Selecionar aleatoriamente membros de NDomPList e adicioná-los a PList sem exceder N.
10. Laço para quando o tamanho de PList for menor do que N:
 - a) usar o método de classificação de não dominância usada em (DEB et al., 2002) para identificar as partículas que representam soluções ótimas de Pareto de NDomListRest e armazená-las em NextNDomList.
 - b) adicionar os membros de nextNDomList a PList até quando o tamanho de PList for menor do que N.
 - c) copiar nextPopListRest num novo arquivo nextPopListRestCopy e esvaziar nextPopListRest.
 - d) preencher nextPopListRest com as partículas que ficam em nextPopListRest após tirar as não dominadas.
 - e) ir ao item a) se ainda o tamanho de PList for menor do que N.
11. Se $k <$ máximas iterações, ir ao item 2.
12. Obter a Frente Ótima de Pareto da última população.

As validações desenvolvidas pelo autor mostraram que o algoritmo NSPSO é eficiente e rápido na solução de problemas de otimização multi-objetivo de validação. Mais detalhes sobre os parâmetros e as validações desenvolvidas se encontram em (LI, 2003).

CAPÍTULO 4

4. QUIMIOTAXIA DE BACTÉRIAS

As definições principais relacionadas ao processo de quimiotaxia de bactérias assim como à estratégia quimiotáctica de colônias de bactérias, são explicadas neste capítulo. Apresenta-se também uma revisão bibliográfica das aplicações de engenharia da estratégia quimiotáctica e o modelo de sinais de comunicação entre bactérias utilizado neste trabalho.

4.1 O QUE É A QUIMIOTAXIA?

Em sentido geral, a quimiotaxia é o movimento quimicamente dirigido que desenvolvem alguns seres vivos. Esse movimento dirigido e as substâncias químicas envolvidas nele são usados por alguns organismos unicelulares, insetos, mamíferos e até mesmo pelo homem, com diversos fins, como por exemplo, na busca de nutrientes, a fim de evitar predadores, para gerar comunicação entre indivíduos na formação de colônias ou grupos, para atração sexual, ou na demarcação territorial (MURRAY, 2002). Além desta definição ampla, na literatura científica o termo quimiotaxia é quase sempre usado para se referir ao movimento celular em resposta a gradientes de concentração de químicos presentes no ambiente.

4.2 AS BACTÉRIAS

Uma bactéria é um organismo unicelular do tipo procariótico, ou seja, não possui organelas ligadas à membrana celular nem um verdadeiro núcleo, e seu DNA consta unicamente de uma molécula circular.

Sua estrutura é basicamente conformada por um corpo central de dimensões microscópicas que pode adotar muitas formas diferentes (YOUNG, 2006), e cuja grandeza pode variar desde $0,01 \mu m^3$ como no caso da *Pelagibacter ubique* (RAPPE et al., 2002) até um volume 10^{10} vezes maior como no caso da *Epulopiscium fishelsoni* (ANGERT; CLEMENTS; PACE, 1993). A parede celular é uma estrutura rígida constituída por moléculas de polissacarídeos limitadas por aminoácidos que, por sua vez recobrem a membrana celular (AXELROD, 1999).

Muitas bactérias são dotadas de uma série de flagelos rotativos na sua superfície celular que, ao funcionar como propulsores, as impulsionam e lhes permitem se moverem a uma velocidade de até $30 \mu m$ por segundo (EISENBACH et al., 2004).

Além da estrutura apropriada para se movimentar autonomamente, as bactérias contam com potentes receptores localizados na superfície celular, chamados de quimiorreceptores, capazes de detectar mudanças espaço temporais nas concentrações químicas do ambiente que as rodeia (ERBAN, 2005). Dessa forma, quando detectam “perturbações” externas, mudam a sua locomoção de um movimento não dirigido para um movimento dirigido. Para a sua sobrevivência, sintetizam permanentemente os sinais dos estímulos externos que lhe são vitais como o catabolismo do carbono, o fornecimento de nitrogênio, as fontes de energia e o estado fisiológico do resto da colônia (EISENBACH et al., 2004).

A reprodução das bactérias acontece através de um processo conhecido como fissão binária. Neste processo o DNA é replicado antes da divisão. Em seguida, a parede celular e a membrana plasmática crescem para dentro, dividindo a célula em duas.

4.3 O MECANISMO DA QUIMIOTAXIA DE BACTÉRIAS

A maioria das bactérias é móvel e usa a quimiotaxia como mecanismo de movimentação que lhe permite tanto encontrar fontes de nutrientes quanto fugir dos ambientes nocivos.

Após séculos de esquecimento acerca do tema, graças ao trabalho de Adler (1966) na década de 60, o interesse no estudo e pesquisa sobre a quimiotaxia de bactérias vem aumentando. Dentre todas as bactérias, a *E. Coli* e a *Salmonella typhimurium* têm sido as mais analisadas, sendo encontrado na literatura científica um grande número de referências a elas (BERG; BROWN, 1972), (SOURJIK, 2004), (POLEZHAEV et al., 2006). Outros tipos de bactérias e células também têm sido objetos de estudo, tais como: a *Pseudomonas putida* em (HILPERT, 2005); a *Thiovulum majus* em (THAR; KUHL, 2005); a *Myxococcus xanthus* em (SPIRO; PARKINSON; OTHMER, 1997) e (MAURIELLO; ZUSMAN, 2007); a *Deleya marina* em (SHEA; NUNLEY; SMITH-SOMERVILLE, 1991) e (YAGI; MARUYAMA, 1998); e a célula do sangue humano *neutrophil* em (ONSUM; ARKIN, 2005), dentre outras.

A quimiotaxia faz parte da estratégia total de sobrevivência das bactérias, a qual foi resumida de forma muito apropriada por Young (2006, p.664):

“para crescerem, as bactérias precisam se alimentar; para se reproduzir, elas precisam se dividir; se as coisas estiverem boas onde elas se encontram, elas querem ficar ali; se as coisas estiverem melhores num outro lugar, elas querem ir até lá; se forem ameaçadas, elas precisam fugir; e se o mundo ao redor mudar elas precisam mudar”.

Embora no universo bacteriano existam outros mecanismos de movimentação, para este trabalho os mecanismos relevantes são os sistemas quimiotáticos constituídos por um conjunto de flagelos. Para esses sistemas, Mitchell (2002) identifica quatro estratégias quimiotáticas diferentes:

- Corrida e giros: é uma estratégia extensamente pesquisada, entre outras coisas por ser a estratégia aplicada pela bactéria *E. Coli*.

- Corrida e arco: as bactérias re-orientam sua direção girando um arco curto. A principal diferença da estratégia anterior reside no fato de que a movimentação é produzida pela re-orientação e não pela corrida. Até o momento tem-se observado esta estratégia unicamente na bactéria *Thiovulum Majus*.
- Corrida e parada: a parada substitui o giro e o arco das estratégias anteriores. A re-orientação acontece sob a ação de movimentos brownianos. É aplicada pela bactéria *Rhodobacter sphaeroides*.
- Corrida e reversa: geralmente as reversas são mais curtas do que as corridas, as bactérias movem-se em uma única direção. Aplicada por algumas bactérias marinhas e aquáticas.

Das quatro estratégias identificadas, a de corrida e giros é a mais conhecida e também a mais fácil de traduzir em regras simples. Como fica evidente, a estratégia está baseada em dois tipos de movimentos - a corrida e o giro. Contudo, neste trabalho, o termo “corrida” será denominado de “natação”. Como esta estratégia é uma das fontes de inspiração deste trabalho, ela será detalhada a seguir.

A bactéria detecta os gradientes químicos do entorno usando seus receptores e faz uma comparação espaço temporal dos gradientes encontrados, ou seja, usa a sua “memória” (SEGALL; BLOCK; BERG, 1986), e dependendo das condições externas encontradas toma uma decisão e gira os flagelos adotando já seja uma natação ou um giro.

Natação:

Os flagelos das bactérias têm forma helicoidal, podem girar a grandes velocidades (de 270 a 600 revoluções por segundo) (KUDO; MAGARIYAMA; AIZAWA, 1990). Podem ainda se deter momentaneamente e mudar o sentido do giro de forma controlada (EISENBACH, 1990). Como consequência, quando todos eles giram no mesmo sentido, atuam como propulsores, movimentando a bactéria suavemente para frente num movimento quase retilíneo. No caso da *E. Coli* e a da *Salmonella typhimurium*, os flagelos giram no sentido horário.

Giro:

Quando os flagelos giram na direção contrária à da natação, desestabilizam-se devido a sua forma helicoidal, impulsionando a célula em diversas direções ao mesmo tempo. Assim o corpo da bactéria faz um giro aleatório. Na figura 11 ilustram-se esses conceitos para a bactéria *E. Coli*.

Figura 11. Natação e giros de bactérias

Usando diversas combinações desses dois movimentos com variações na longitude e duração dos mesmos, a bactéria movimenta-se no seu ambiente.

No contexto da quimiotaxia, as substâncias químicas presentes no ambiente da bactéria que lhe servem como alimento ou fonte de nutrientes e energia, chamam-se quimioatraentes. As substâncias nocivas que repelem a bactéria chamam-se quimiorepelentes. Dependendo do tipo de substância encontrada e do gradiente dessa, as respostas quimiotáticas de uma bactéria que aplica a estratégia natação-giro são:

- Caso o ambiente for neutro (nem atraente nem repelente), a bactéria alterna os movimentos de natação e giro, descrevendo um movimento aleatório (*random walk*) e, desta forma, continua explorando.
- Caso encontre um quimioatraente, mas sem gradiente, a longitude e velocidade médias da natação aumentam, e a duração dos giros diminui.

- c) Caso encontre o gradiente de um quimioatraente ou de um quimiorepelente, a bactéria alterna os movimentos de natação e giro. Se o estímulo for positivo (gradiente positivo no caso do quimioatraente, e negativo no quimiorepelente), a probabilidade de giros diminui; no caso de um estímulo negativo, essa probabilidade aumenta, desta forma a bactéria busca os estímulos positivos e foge dos estímulos negativos. Isso constitui um movimento aleatório dirigido (*biased random walk*). Na figura 12 ilustra-se esse comportamento.

Figura 12. Movimento quimiotático de bactérias

4.4 APLICAÇÕES EM ENGENHARIA DA ESTRATÉGIA QUIMIOTÁCTICA

Nas áreas de bioquímica, biologia, genética e outras, a modelagem da quimiotaxia de bactérias e a sua aplicação são um problema amplamente estudado, contudo, o mesmo não ocorre na área das aplicações de engenharia, aonde só na última década vem-se publicando trabalhos científicos a respeito. Nesta seção apresenta-se um resumo dos trabalhos publicados mais relevantes aplicados a diversos campos da engenharia e na área específica da otimização.

Em (DELGADO, 2000) apresentam-se duas aplicações de redes neurais SOM (Self Organising Map). Usam-se para o treinamento da rede, um algoritmo híbrido resultante da combinação da fórmula clássica da SOM para aprendizagem, e um algoritmo estocástico de otimização baseado em quimiotaxia.

Em (KIRCHNER; SCHADSCHNEIDER, 2002) inspirado no processo de quimiotaxia, é modelada e simulada a formação de trilhas, baseado no estudo dos processos de evacuação simples, usando os autômatos celulares estocásticos para a dinâmica de pedestres, de modo que se possa determinar e classificar os comportamentos manifestados na simulação. Consegue-se desenvolver um modelo de alta velocidade de simulação devido à simplicidade do mesmo, aplicável na otimização de procedimentos de evacuação.

Em (MARQUES; NUNES; DE ALMEIDA, 2002) desenvolvem-se três estratégias de navegação para um robô com sensores olfativos que se move num ambiente com presença de turbulência e várias fontes de odor. Uma das estratégias é baseada na quimiotaxia, usando comandos muito simples que executam os movimentos de natação e giro. Conclui-se que o comportamento é aceitável, além de ser muito simples de se implementar e exigir mínima capacidade de memória.

Em (RUSSELL et al., 2003) são implementados e avaliados quatro algoritmos de locomoção para robô, três deles baseados na quimiotaxia de diferentes espécies. O robô conta com sensores químicos, um sensor de fluxo de ar e contatos para detectar obstáculos. Os algoritmos foram implementados em robôs reais, concluindo-se que aquele baseado na quimiotaxia de bactérias é o mais simples de se implementar e é o que menos precisa de controle dos movimentos do robô, contudo seu campo de aplicação está restrito a ambientes

nos quais exista pouca ou nenhuma turbulência que perturbe as variações suaves das concentrações de químicos.

Com a finalidade de monitorar e estudar ecossistemas em (DHARIWAL; SUKHATME; REQUICHA, 2004), desenvolve-se a estratégia de movimento de um robô, sobre a base de um algoritmo de caminhada aleatória dirigida (*random biased walk*), o qual é simulado num grupo de robôs e posteriormente é implementado em robôs reais, utilizando uma fonte de luz como gerador do gradiente.

Em (ONSUM; ARKIN, 2005) desenvolve-se um algoritmo para controlar o movimento de um robô autônomo, baseado na quimiotaxia das células do sangue humano, chamadas *neutrophil*. O *neutrophil* é, neste caso, modelado como um sistema colaborativo de controle. Além de sugerir uma forma de implementar o algoritmo com componentes eletrônicos, demonstra-se que o algoritmo é robusto frente a perturbações.

Em (RIGANTI; SALVINI, 2007) aplicam-se diferentes algoritmos heurísticos de otimização na solução de problemas de engenharia elétrica, relacionados com sistemas de potência e de energia. Entre os algoritmos aplicados está o BCA (*Bacterial Chemotaxis Algorithm*) (MÜLLER et al., 2002). Os algoritmos heurísticos são testados e seu desempenho comparado.

4.4.1 Aplicações da estratégia quimiotáctica em otimização

Como se evidencia, a quimiotaxia é uma estratégia otimizada de movimentos. Em (BREMERMAN, 1974) é realizada uma das primeiras referências ao mecanismo da quimiotaxia de bactérias como um algoritmo de otimização, constituído dos seguintes passos: Considerando uma concentração não homogênea de quimioatraentes, a bactéria:

1. Gira e nada em linha reta em uma direção aleatória.
2. Continua até que o gradiente comece a decrescer, caso essa direção seja a de um gradiente positivo do quimioatraente. Depois de medir o decrescimento, a bactéria gira novamente e, como resultado, nada em uma nova direção aleatória.

3. Pára e depois gira por um tempo e nada em outra direção, caso a nova direção seja a de um gradiente decrescente do quimioatraente.
4. Assume que as direções aleatórias sucessivas são independentes e uniformemente distribuídas.

Neste trabalho pioneiro analisa-se o algoritmo para gradientes em duas e três direções, considerando superfícies esféricas e cilíndricas.

Em (MONTAGUE; WARDB, 1994) aplica-se um algoritmo inspirado no trabalho de Bremmerman (1974) na otimização de biorreatores. Utilizou-se este algoritmo para encontrar o perfil de alimentação, a concentração inicial e a adição adequada de enzimas que maximiza a função de desempenho de reatores.

Em (MÜLLER et al., 2002) desenvolve-se um novo algoritmo de otimização baseado em quimiotaxia, tomando como referência o trabalho de Bremermann (1974). Neste trabalho, ao contrário do algoritmo original, os valores de concentração têm relevância e, por isso, a bactéria não permanece nadando na mesma direção quando encontra gradientes positivos de quimioatraentes. O algoritmo é aplicado na otimização de funções unimodais e multimodais, assim como na otimização do projeto inverso de aerofólios.

Em (PASSINO, 2002) foi introduzido um novo algoritmo baseado no comportamento de busca de alimento desenvolvido pela bactéria *E. Coli*. Neste trabalho, além de desenvolver uma explicação muito completa sobre o fenômeno da quimiotaxia desta bactéria, o algoritmo implementado considera processos naturais como a locomoção da bactéria e mecanismos de reprodução, eliminação e dispersão. Também o processo de comunicação entre as bactérias influencia a locomoção destas, aproximando-se assim da idéia de que a quimiotaxia é um fenômeno de colônias de bactérias, mais do que comportamento individual delas. Na atualidade, esse algoritmo que é aplicado para otimização mono-objetivo é conhecido como BFOA pelas suas siglas em inglês *Bacterial Foraging Optimization Algorithm*.

Em (LIU; PASSINO, 2002) apresenta-se o conceito de métodos de otimização distribuída não baseados em gradiente. Analisa-se a quimiotaxia das bactérias *E. coli* e *M. Xanthus*. Para a *E. Coli*, apresenta-se novamente o algoritmo de Passino (2002), mas para a *Myxococcus xanthus* apresenta-se um novo algoritmo do tipo autômato celular estocástico, no qual as bactérias

podem se mexer num espaço tridimensional, onde ganha muita importância o comportamento coletivo das bactérias. Para a *E. Coli*, também são consideradas a reprodução, eliminação e dispersão. Em publicações posteriores (GAZI; PASSINO, 2002), (PASSINO, 2005) desenvolveram-se validações de convergência e análise de estabilidade do algoritmo BFOA.

Desde a sua publicação inicial, têm sido propostas novas versões com modificações do algoritmo BFOA (DE CASTRO; VON ZUBEN, 2004), (TANG; WU, 2006), (BISWAS et al., 2007), (CHEN et al., 2007), (DASGUPTA et al., 2008).

O algoritmo original e algumas de suas versões também têm sido aplicados com sucesso na solução de diversos problemas de engenharia. A seguir mencionam-se alguns destes trabalhos. Em (MISHRA, 2005) é apresentada uma aplicação para a estimação dos harmônicos em sistemas de potencia. Em (KIM; CHO, 2006) o BFOA é utilizado na afinação de um controlador PID de um AVR. Em (LI et al., 2007) apresenta-se a solução ótima para um OPF (*Optimal Power Flow*). Em (KIM; JUNG; CHO, 2007), desenvolve-se um algoritmo de simulação da quimiotaxia da bactéria *E. Coli*, com a aplicação de autômatos celulares e considerando processos de difusão dos sinais químicos e transmissão de informação entre bactérias. Em (WU et al., 2007) é apresentada uma aplicação do BFOA na solução de problemas de *job shop scheduling*. Em (LIN; LIU; FANG, 2007) são estimados os parâmetros das funções e os pesos de um conjunto de regras de um sistema de lógica nebulosa. Em (FANG; LIU; LIU, 2007) é implementada a estimação dos parâmetros de um sistema de modelagem não linear do fenômeno de fratura. Em (ACHARYAL; PANDA, 2008) é apresentada a avaliação de componentes independentes para separar sinais misturados. Em (PANIGRAHI; RAVIKUMAR PANDI, 2008) o BFOA é utilizado para solucionar problemas do tipo *economic load dispatch* e em (GUNEY; BASBUG, 2008) é desenvolvida uma aplicação para arranjos lineares de antenas no controle da amplitude dos elementos.

Outros algoritmos baseados em quimiotaxia da bactéria *E. Coli* têm sido apresentados nos últimos anos. Em (NICOLAU et al., 2008) apresenta-se o algoritmo de otimização *Extremotaxis*, o qual está inspirado no mecanismo quimiotático de corrida e natação desenvolvido naturalmente por essa bactéria. O algoritmo retoma algumas influências do mecanismo PSO (*Particle Swarm Optimizer*), mas sem utilizar a comunicação entre as bactérias da colônia. A eficiência deste algoritmo é demonstrada mediante a solução de quatro problemas de otimização mono-objetivo de validação. Em (LIU, W. et al., 2008) apresenta-se

uma melhora do algoritmo BCC (*Bacterial Colony Chemotaxis*), segundo a qual se acrescentam estratégias de controle de parâmetros e modos de interação de informação para melhorar a velocidade e exatidão da convergência. Em (LEE et al., 2009) apresenta-se um algoritmo que junto com a quimiotaxia de bactérias, utiliza outros mecanismos desenvolvidos na natureza como a busca de padrões para resolver problemas de medição de operações em ambientes urbanos.

4.5 COMUNICAÇÃO ENTRE BACTÉRIAS

Durante muito tempo depois do seu descobrimento, no final do século dezenove, as bactérias e, em geral, os microorganismos foram considerados seres incapacitados de aplicar processos de comunicação intra ou intercelular. No princípio da década de sessenta, graças aos estudos da agregação de células em estruturas complexas (*fruiting body formation*) e à produção de estreptomicina, reportaram-se as primeiras observações do que poderia se considerar como comportamentos sociais em grupos de bactérias (DIGGLE; CRUSZ; CÁMARA, 2007). Na atualidade, não apenas se reconhece amplamente a existência de comunicações entre microorganismos, mas também se têm atribuído formas rudimentares de inteligência às bactérias (HELLINGWERF, 2005).

Diferentes aproximações explicativas sobre o processo de transmissão de sinais entre bactérias têm sido propostas. Contudo, um número cada vez maior de trabalhos científicos atribui ao fenômeno, conhecido como “*quorum sensing*”, uma contribuição decisiva em processos importantes, como a luminescência, a esporulação, a síntese antibiótica, a virulência (GOBBETTI et al., 2007), a movimentação da colônia como um enxame (*swarming* em inglês) (DANIELS; VANDERLEYDEN; MICHIELS, 2004) e a formação de biofilmes (PARSEK; GREENBERG, 2005).

- *Quorum sensing*: é o processo complexo pelo qual a acumulação de moléculas-sinal no ambiente que rodeia uma bactéria, capacita-na para medir o número de bactérias (densidade da população) de forma que a população como um todo possa coordenar uma ação de resposta (DIGGLE; CRUSZ; CÁMARA, 2007). As bactérias têm a capacidade de produzir, liberar e medir sinais moleculares difusíveis de baixa massa molecular, que são

usadas pela colônia como mecanismos de sinalização. Quando a densidade das células atinge um valor crítico (o quorum necessário para tomar uma decisão coletiva), o sinal dispara o interruptor dos genes controlados pelo *quorum sensing* e uma resposta coletiva é desenvolvida. Geralmente é observado que, uma vez ativado o processo, a concentração das moléculas-sinal reflete o número de células na colônia ou, pelo menos, o número de células num estado fisiológico determinado (BERNARDINI; GHEORGHE; KRASNOGOR, 2007).

Desde o seu descobrimento, muitos cientistas têm proposto modelos do fenômeno de comunicação entre bactérias. Na continuação explica-se o modelo que, devido a sua baixa complexidade e sua facilidade de adaptação à técnica de otimização topológica desenvolvida, foi implementado neste trabalho.

4.5.1 Modelo baseado em quimiotaxia e otimização de recursos para comunicação bactéria – bactéria

Em (THAR; KUHL, 2005), desenvolve-se um modelo computacional que simula o comportamento quimiotático presente na formação de padrões complexos da bactéria marinha *Thiovulum majus*. Baseia-se em observações experimentais para propor o modelo.

Sabe-se que as *Thiovulum majus*, quando já agregadas à colônia, podem induzir um campo de fluxo simétrico radial. Aquelas bactérias que estiverem nadando livres, preferem se aderir às vizinhanças próximas às bactérias já aderidas à colônia para se beneficiar dos suprimentos de oxigênio localizados e evadir uma região em forma de anel ao redor das bactérias já aderidas. Para quantificar este comportamento, propõe-se a eq.(28), que representa uma forma de chapéu radial simétrica como se ilustra na figura 13.

$$\begin{aligned} q(r) &= \frac{\sin(2\pi R^{-1}r)}{2\pi R^{-1}r} & \text{se} & \quad r \leq R \\ q(r) &= 0 & \text{se} & \quad r > R \end{aligned} \tag{28}$$

na qual r é a distância radial desde a bactéria aderida, R é a máxima distância influenciada pela bactéria aderida e $q(r)$ é o sinal enviado pela bactéria a uma distância radial r . Assim valores elevados de q correspondem a condições favoráveis de oxigênio para a bactéria.

Figura 13. Sinal bactéria – bactéria

A soma dos fluxos gerados por cada uma das bactérias aproxima-se ao campo de fluxo total gerado pela colônia. Assim um parâmetro Q do ambiente em uma localização determinada da colônia gerado pelas N bactérias aderidas é calculado pela eq.(29).

$$Q = q(r_1) + q(r_2) + \dots + q(r_N) \quad (29)$$

De acordo com o modelo proposto, as bactéria livres sempre preferem as localizações com os máximos valores de Q . Assim na simulação do comportamento quimiotático das bactérias, calcula-se o parâmetro Q para cada uma das possíveis localizações. Em seguida, as bactérias nas localizações com valores mínimos de Q , são levadas às localizações não ocupadas com os máximos valores de Q .

Com esse modelo simples consegue-se simular com bastante precisão a formação de padrões complexos em colônias de bactérias marinhas.

CAPÍTULO 5

5. OTIMIZAÇÃO TOPOLÓGICA BASEADA EM QUIMIOTAXIA DE BACTÉRIAS

Neste capítulo apresenta-se o algoritmo original para otimização topológica de estruturas bidimensionais contínuas que foi projetado, desenvolvido e implementado neste trabalho. Também se apresentam cinco aplicações do algoritmo à solução de problemas de otimização topológica.

O algoritmo foi implementado no programa Matlab 7.0 ®, num computador pessoal com 3 GHz de frequência, 3.34 GB de memória RAM e processador PENTIUM 4 de INTEL.

5.1 ALGORITMO DE OTIMIZAÇÃO TOPOLÓGICA BASEADA EM QUIMIOTAXIA DE BACTERIAS - BCBTOA

Semelhante ao ESO, o BCBTOA (*Bacterial Chemotaxis Based Topology Optimization Algorithm*) é um método heurístico de otimização topológica que emula os mecanismos de formação de padrões em colônias de bactérias e não usa um modelo de material.

É inegável que o crescimento adequado, o desenvolvimento e a sobrevivência das bactérias requerem medição precisa das condições externas do meio e de uma extensa comunicação entre os membros de uma colônia (BARDWELL et al., 2007). Como parte dos mecanismos de sobrevivência das colônias, a formação de padrões desempenha um papel fundamental. O

mecanismo da formação de padrões em colônias da bactéria marinha *Thiovulum majus* é a base para o desenvolvimento deste algoritmo.

O princípio do método é muito simples. O domínio de projeto representa o ambiente no qual podem se movimentar as bactérias de uma colônia. Como todo ambiente natural, quantidades variáveis de nutrientes ou de substâncias nocivas encontram-se nos diferentes pontos que nesse caso estão representados pelos valores da função objetivo nos diferentes elementos. Imersas ali, as bactérias desenvolvem uma estratégia quimiotáctica de forma que aumente as probabilidades de sobrevivência do maior número possível dos membros da colônia, movimentando-se desde localizações pobres em nutrientes para aderir-se em localizações mais favoráveis. A figura 14 ilustra esse procedimento.

Figura 14. Bactérias aplicando estratégia quimiotáctica em BCBTOA

A aplicação detalhada deste princípio no processo de otimização topológica de estruturas explica-se a seguir.

Início:

Define-se o domínio total de projeto (espaço total no qual podem-se movimentar quimiotacticamente as bactérias), o número total de elementos $Nelem$ no qual vai ser discretizado o domínio, e o número total de iterações desejadas I_{total} (critério de finalização). Também se definem as características do material, as condições de carga, o volume máximo $Vmax$, sendo este o volume que no final do processo deverá ser ocupado pelo projeto

otimizado. O volume máximo corresponde a uma porcentagem do domínio total de projeto expresso em número de elementos, ou seja, $V_{max} = \%Nelem$. A variável de projeto é a existência de material, assim a densidade de todos os elementos durante o processo pode adotar somente dois valores 1 ou 0, representando presença ou ausência total de material. No passo inicial do algoritmo, para todos os elementos do domínio de projeto é atribuído um valor de densidade igual a 1, portanto o domínio de projeto inicial é completamente sólido. A figura 15 ilustra este passo para uma viga simplesmente apoiada.

Figura 15. Discretização do domínio de projeto em BCBTOA

Avaliação da função objetivo:

O objetivo da otimização é encontrar a distribuição ótima de material que minimize a energia interna de deformação total da estrutura, respeitando uma restrição de volume máximo. Cada elemento da malha na qual foi discretizado o domínio de projeto, representa a possível localização de uma bactéria da colônia. Fazendo uso do método de elementos finitos, o valor da função objetivo (energia interna de deformação) para cada elemento é calculado. A figura 16 ilustra esse passo para a viga da figura 15 representando o valor da função objetivo para cada elemento de acordo com a intensidade da cor. Os elementos são ordenados de acordo com os valores da função objetivo e, seguindo esta ordem, um número igual a V_{max} dos elementos mais exigidos (maiores valores da função objetivo) são ocupados pelas bactérias

iniciais, assim as primeiras bactérias localizam-se nos pontos mais ricos do domínio de projeto.

Figura 16. Avaliação da função objetivo em BCBTOA

Comunicação entre os membros da colônia:

Mediante o envio de sinais aos outros membros da colônia, as bactérias comunicam-se entre elas. Para este algoritmo usou-se uma adaptação do sinal definido no modelo proposto por (THAR; KUHL, 2005) e que foi explicado na seção 4.5.1. Dessa forma cada bactéria desde a sua localização envia um sinal ao resto da colônia cujo valor corresponde ao valor da função de chapéu (definida em 4.5.1) vezes o valor da função objetivo avaliada nessa localização.

Com esse procedimento consegue-se que as bactérias que se encontram em fontes de nutrientes mais ricas (valores maiores da função objetivo) enviem sinais mais fortes às bactérias da sua vizinhança. Os sinais enviados por cada bactéria se espalham num determinado raio de influência R dentro do domínio de projeto, modificando as condições do ambiente completo. Este raio de influência deve ser determinado pelo usuário.

Modificação da função objetivo:

Em cada localização de uma bactéria, todos os valores dos sinais recebidos são calculados e somados ao valor real da função objetivo obtendo para essa função um valor modificado de acordo com o ambiente.

Estratégia quimiotáctica:

Define-se o elemento habitado por uma bactéria com o maior valor real da função objetivo. Desenvolve-se uma comparação dos valores reais da função objetivo em cada localização

habitada por uma bactéria com esse valor máximo. As bactérias que cumpram o critério de movimentação definido pela eq.(30) devem se movimentar até localizações mais ricas.

$$\text{Se } f_{xy} < (Fator)f_{\max} \Rightarrow bac_{xy} \text{ se movimenta} \quad (30)$$

Na qual f_{xy} é o valor da função objetivo real calculado no elemento finito localizado nas coordenadas (x, y) do domínio de projeto, $Fator$ é um número entre zero e um que deve ser determinado para cada problema de OT; $f_{\max}$ é o valor máximo da função objetivo para um elemento do domínio de projeto calculado para esta iteração; e bac_{xy} é a bactéria localizada no elemento de coordenadas (x, y) . Desta forma, os elementos ineficientes ou subcarregados não podem “sustentar” uma bactéria. Na figura 17 ilustra-se a estrutura quando têm se esvaziado os elementos ineficientes.

Figura 17. Esvaziamento dos elementos ineficientes em BCBTOA

Como sugere o modelo de quimiotaxia da bactéria marinha *Thiovulum majus* apresentado em (THAR; KUHL, 2005), aquelas localizações que têm os valores maiores de função objetivo modificada, são aptos para a aderência de bactérias. Por isso, os elementos finitos são ordenados de acordo com seus valores de função objetivo modificada; assim nas localizações mais ricas que ainda não estejam ocupadas, aderem-se as bactérias que de acordo com a eq.(30) deveram-se movimentar. Na figura 18 ilustra-se a estrutura quando as bactérias que deveram-se movimentar, aderiram-se nos elementos mais tensionados.

Figura 18. Aderência de bactérias em BCBTOA

Atualização da variável de projeto:

O seguinte passo é a atualização da variável de projeto. No modelo de elementos finitos, para aqueles elementos que após a aplicação da estratégia quimiotáctica contem uma bactéria, designa-se a eles um valor de densidade igual a 1, o qual significa a existência de material sólido nesse elemento. O resto dos elementos, que não contém uma bactéria, tem um valor de densidade zero (na verdade um valor muito baixo, porém diferente de zero para evitar as inconsistências na matriz de rigidez). O algoritmo acaba após um determinado número de iterações.

Todos os passos do algoritmo são detalhados a seguir:

1. Definir o domínio de projeto, a quantidade e o tipo de elementos nos quais vai se discretizar o domínio $Nelem$, as características do material, as condições de carga, os valores iniciais das variáveis de projeto, $Vmax$, I_{total} , o parâmetro $Fator$ e o raio de influência do sinal R .
2. Avaliar as variáveis de estado e usando o método dos elementos finitos, calcular a função objetivo f para cada elemento do domínio de projeto. Determinar o valor de f_{max} .
3. Ordenar de forma descendente, os elementos de acordo com os valores da função objetivo.
4. Preencher um numero igual ao $Vmax$ dos elementos com os maiores valores da função objetivo f com bactérias.

5. Calcular o valor do sinal enviado por cada bactéria bac_{xy} de acordo com as eq.(31) e eq.(32):

$$q(r)_{bac_{xy}} = \frac{\sin(2\pi R^{-1}r)}{2\pi R^{-1}r} f_{xy} \quad \text{se} \quad r \leq R \quad (31)$$

$$q(r)_{bac_{xy}} = 0 \quad \text{se} \quad r > R \quad (32)$$

nas quais f_{xy} é o valor da função objetivo para o elemento onde está localizada a bactéria bac_{xy} ; r é a distancia radial desde a bactéria, R é a máxima distância influenciada pela bactéria, e $q(r)_{bac_{xy}}$ é o sinal enviado pela bactéria a uma distancia radial r .

6. Calcular os valores dos sinais em cada localização. Para cada localização do domínio de projeto, somam-se todos os valores dos sinais enviados pelas bactérias que atingiram cada localização. Obtém-se o valor da função objetivo modificada f_{mod} para cada localização.
7. Aplicar o critério de movimentação. da eq.(30) e determinar a quantidade de bactérias que se devem movimentar $Bacmov$, esvaziando os elementos não eficientes.
8. Ordenar os elementos de forma descendente de acordo com o valor da função objetivo modificada f_{mod} .
9. Determinar uma quantidade igual a $Bacmov$ das melhores localizações de acordo com a função objetivo modificada que ainda não estejam habitadas e aderir a elas, as bactérias que deveram se movimentar.
10. Atualizar a densidade de cada elemento finito. A aqueles elementos habitados por bactérias, atribui-lhes um valor de densidade igual a 1, aos elementos inabitados, atribui-lhes um valor de densidade igual a zero. $I = I + 1$.
11. Se $I = I_{total}$, parar. Caso contrário, voltar ao passo 1.

5.2 VALIDAÇÕES DESENVOLVIDAS

Para comprovar a efetividade do BCBTOA, o algoritmo foi implementado em Matlab 7.0 ®, introduzindo as modificações necessárias ao programa apresentado em (SIGMUND, 2001). O algoritmo foi aplicado na solução de cinco problemas de otimização topológica de estruturas bidimensionais contínuas, todos eles relativos a vigas submetidas a cargas pontuais. Como parâmetro de comparação, os mesmos problemas foram solucionados com o algoritmo SIMP apresentado em (SIGMUND, 2001) e os resultados foram representados em gráficos.

5.2.1 Problema de Otimização Topológica (POT) 1 - Viga simplesmente apoiada

Trata-se de uma viga simplesmente apoiada nos pontos **A** e **B**, com carga pontual vertical **F** aplicada no ponto central **C** do lado superior. A figura 19 ilustra o problema. As tabelas 1 e 2 indicam os valores das variáveis e parâmetros usados para a solução deste problema com os algoritmos BCBTOA e SIMP respectivamente. Nelx é a quantidade de elementos finitos na direção horizontal e Nely é a quantidade de elementos finitos na direção vertical, Nelem é o numero total de elementos, ou seja, $Nelx \times Nely$.

Figura 19. Viga do POT 1

Tabela 1 - Valores dos parâmetros de BCBTOA no POT 1

Nelx	Nely	I_{total}	Vmax	R	Fator
60	20	20	0,4Nelem	4	0,011

Tabela 2 - Valores dos parâmetros de SIMP no POT 1

Nelx	Nely	I_{total}	Vmax	p'	rmin
60	20	25	0,4Nelem	3	1,5

A figura 20 mostra a topologia resultante com a aplicação do BCBTOA e a figura 21 mostra a topologia resultante com a aplicação do algoritmo SIMP para a solução deste problema. Nos dois casos, as figuras ilustram a metade direita da viga resultante, obviamente a outra metade é igual.

Figura 20. Topologia final do POT 1 com BCBTOA

Figura 21. Topologia final do POT 1 com SIMP

As figuras 22 e 23 mostram os valores da energia interna de deformação (função objetivo) para cada elemento da metade direita da estrutura otimizada obtida para o POT 1 com os métodos BCBTOA e SIMP, respectivamente. A intensidade das cores é proporcional ao valor da função objetivo em cada elemento.

Figura 22. Energia interna de deformação do POT 1 com BCBTOA

Figura 23. Energia interna de deformação do POT 1 com SIMP

5.2.2 POT 2 - Viga engastada em extremo direito

Trata-se de uma viga engastada no extremo direito, com carga pontual vertical F aplicada no lado inferior direito ponto central C . A figura 24 ilustra o problema. As tabelas 3 e 4 indicam os valores das variáveis e parâmetros usados para a solução deste problema com os algoritmos BCBTOA e SIMP, respectivamente.

Figura 24. Viga do POT 2

Tabela 3 - Valores dos parâmetros de BCBTOA no POT 2

Nelx	Nely	I_{total}	Vmax	R	Fator
32	20	15	0,4Nelem	5	0,0118

Tabela 4 - Valores dos parâmetros de SIMP no POT 2

Nelx	Nely	I_{total}	Vmax	p'	rmin
32	20	25	0,4Nelem	3	1,2

A figura 25 mostra a topologia resultante com a aplicação do BCBTOA e a figura 26 mostra a topologia resultante com a aplicação do algoritmo SIMP para a solução deste problema.

Figura 25. Topologia final do POT 2 com BCBTOA

Figura 26. Topologia final do POT 2 com SIMP

As figuras 27 e 28 mostram os valores da energia interna de deformação (função objetivo) para cada elemento da topologia final da estrutura otimizada obtida para o POT 2 com os métodos BCBTOA e SIMP, respectivamente. A intensidade das cores é proporcional ao valor da função objetivo em cada elemento.

Figura 27. Energia interna de deformação do POT 2 com BCBTOA

Figura 28. Energia interna de deformação do POT 2 com SIMP

5.2.3 POT 3 - Viga engastada com orifício circular

Trata-se de uma viga engastada no lado esquerdo, com carga pontual vertical $\mathbf{F}$ aplicada no lado inferior direito $\mathbf{C}$ e com um orifício circular inicial como se ilustra na figura 29. As tabelas 5 e 6 indicam os valores das variáveis e parâmetros usados para a solução deste problema com os algoritmos BCBTOA e SIMP, respectivamente.

Figura 29. Viga do POT 3

Tabela 5 - Valores dos parâmetros de BCBTOA no POT 3

Nelx	Nely	I_{total}	Vmax	R	Fator
45	30	14	0,5Nelem	5	0,0014

Tabela 6 - Valores dos parâmetros de SIMP no POT 3

Nelx	Nely	I_{total}	Vmax	p'	rmin
45	30	25	0,5Nelem	3	1,5

A figura 30 mostra a topologia resultante com a aplicação do BCBTOA e a figura 31 mostra a topologia resultante com a aplicação do algoritmo SIMP para a solução deste problema.

Figura 30. Topologia final do POT 3 com BCBTOA

Figura 31. Topologia final do POT 3 com SIMP

As figuras 32 e 33 mostram os valores da energia interna de deformação (função objetivo) para cada elemento da topologia final da estrutura otimizada obtida para o POT 3 com os métodos BCBTOA e SIMP, respectivamente. A intensidade das cores é proporcional ao valor da função objetivo em cada elemento.

Figura 32. Energia interna de deformação do POT 3 com BCBTOA

Figura 33. Energia interna de deformação do POT 3 com SIMP

5.2.4 POT 4 - Viga engastada com carga central

Trata-se de uma viga engastada no extremo direito, com carga pontual vertical F aplicada no ponto central C do lado direito. A figura 34 ilustra o problema. As tabelas 7 e 8 indicam os

valores das variáveis e parâmetros usados para a solução deste problema com os algoritmos BCBTOA e SIMP, respectivamente.

Figura 34. Viga do POT 4

Tabela 7 - Valores dos parâmetros de BCBTOA no POT 4

Nelx	Nely	I_{total}	Vmax	R	Fator
50	26	10	0,5Nelem	5	0,00064

Tabela 8 - Valores dos parâmetros de SIMP no POT 4

Nelx	Nely	I_{total}	Vmax	p'	rmin
50	26	25	0,5Nelem	3	1,5

A figura 35 mostra a topologia resultante com a aplicação do BCBTOA e a figura 36 mostra a topologia resultante com a aplicação do algoritmo SIMP para a solução deste problema.

Figura 35. Topologia final do POT 4 com BCBTOA

Figura 36. Topologia final do POT 4 com SIMP

As figuras 37 e 38 mostram os valores da energia interna de deformação (função objetivo) para cada elemento da topologia final da estrutura otimizada obtida para o POT 4 com os métodos BCBTOA e SIMP, respectivamente. A intensidade das cores é proporcional ao valor da função objetivo em cada elemento.

Figura 37. Energia interna de deformação do POT 4 com BCBTOA

Figura 38. Energia interna de deformação do POT 4 com SIMP

5.2.5 POT 5 - Viga simplesmente apoiada com carga central

Trata-se de uma viga simplesmente apoiada nos pontos **A** e **B**, com carga pontual vertical **F** aplicada no ponto central **C** do lado inferior. A figura 39 ilustra o problema. As tabelas 9 e 10 indicam os valores das variáveis e parâmetros usados para a solução deste problema com os algoritmos BCBTOA e SIMP, respectivamente.

Figura 39. Viga do POT 5

Tabela 9 - Valores dos parâmetros de BCBTOA no POT 5

Nelx	Nely	I_{total}	Vmax	R	Fator
50	25	10	0,5Nelem	6	0,0045

Tabela 10 - Valores dos parâmetros de SIMP no POT 5

Nelx	Nely	I_{total}	Vmax	p'	rmin
50	25	25	0,5Nelem	3	1,5

A figura 40 ilustra a progressão do processo de otimização com o método BCBTOA. A figura 41 mostra a topologia resultante com a aplicação do BCBTOA e a figura 42 mostra a topologia resultante com a aplicação do algoritmo SIMP para a solução deste problema.

Figura 40. Etapas da otimização topológica com BCBTOA

Figura 41. Topologia final do POT 5 com BCBTOA

Figura 42. Topologia final do POT 5 com SIMP

As figuras 43 e 44 mostram os valores da energia interna de deformação (função objetivo) para cada elemento da topologia final da estrutura otimizada obtida para o POT 5 com os métodos BCBTOA e SIMP, respectivamente. A intensidade das cores é proporcional ao valor da função objetivo em cada elemento.

Figura 43. Energia interna de deformação do POT 5 com BCBTOA

Figura 44. Energia interna de deformação do POT 5 com SIMP

5.3 ANÁLISE DOS RESULTADOS

Poderia se pensar que o valor final da função objetivo para as estruturas obtidas com a aplicação dos dois métodos serviria como parâmetro de comparação. Contudo, devido ao fato de que as estruturas encontradas com o método SIMP não são as definitivas, porque faltam ainda os processos de filtragem e interpretação dos resultados, os valores da energia interna de distorção calculados para essas estruturas que tem elementos com densidades intermédias não são um parâmetro válido de comparação.

As comparações entre os dois métodos que podem ser feitas com estes resultados são apenas intuitivas, baseadas nas observações dos desenhos das estruturas finais obtidas com a aplicação dos dois algoritmos. Porém, como podem ser observados nas figuras, os resultados obtidos são promissores e mostram um inegável potencial no método BCBTOA.

As figuras que mostram os valores da energia interna de deformação (função objetivo) para cada elemento para o método SIMP, apresentam o mesmo padrão de comportamento quando comparadas com as figuras das topologias finais obtidas com esse método, o qual é de se esperar devido à proporcionalidade direta entre a função objetivo otimizada e a densidade relativa dos elementos.

As figuras que mostram os valores da energia interna de deformação (função objetivo) para cada elemento para o método BCBTOA apresentam uma distribuição dos valores da função objetivo menos uniforme quando comparados com o método SIMP. Contudo, essa observação não é conclusiva, novamente devido a que as topologias obtidas com o método SIMP não são as definitivas.

Nos cinco exemplos desenvolvidos neste trabalho, o BCBTOA precisou de um número de iterações menor do que o SIMP para chegar a resultados bastante similares.

É importante destacar que os resultados obtidos com o método BCBTOA não requerem processos posteriores de filtragem, já que em cada elemento do projeto final há apenas uma condição de duas possíveis - o elemento é sólido ou vazio. Além disso, os projetos finais, pelo menos no caso dos exemplos apresentados, não apresentaram tabuleiros de xadrez.

A diferença básica de procedimento em relação ao método SIMP é que o BCBTOA consiste em um método heurístico que não usa um modelo de material.

A diferença principal em relação ao método BESO reside no fato de que no BCBTOA não se inicia com o domínio de projeto como um volume sólido para paulatinamente retirar e adicionar material. Devido a isso o BESO requer que as modificações feitas à estrutura em cada iteração sejam muito pequenas, aumentando o custo computacional. No BCBTOA, desde o começo e para cada iteração, somente um número de elementos correspondente ao

volume final desejado pode ter uma densidade igual a um, com isso se consegue redistribuir importantes quantidades de material a cada iteração.

A seleção dos parâmetros *Fator* e *R* foi realizada inteiramente mediante ensaio e erro, sendo que não foi possível estabelecer uma regra ou critério para essa seleção.

Para os cinco exemplos apresentados o algoritmo funcionou satisfatoriamente, mas seriam necessárias mais aplicações para determinar a generalidade dele.

CAPÍTULO 6

6. OTIMIZAÇÃO MULTI-OBJETIVO BASEADA EM QUIMIOTAXIA DE BACTÉRIAS

Como já foi explicado no capítulo quatro, o movimento quimiotático das bactérias é uma estratégia de busca, cuja possível implementação como processo de otimização, não é difícil de intuir. Neste capítulo expõe-se o algoritmo original de otimização multi-objetivo baseado em quimiotaxia de bactérias que foi projetado, desenvolvido e implementado neste trabalho. Este algoritmo aplica comparações espaço temporais, regras de movimentação básicas e uma estratégia simples de comunicação entre bactérias.

6.1 ALGORITMO DE OTIMIZAÇÃO MULTI-OBJETIVO BASEADO EM QUIMIOTAXIA DE BACTÉRIAS - BCMOA

O algoritmo de otimização multi-objetivo apresentado nesta seção é o fruto de inúmeras tentativas com diversas estratégias e enfoques até conseguir um desempenho altamente satisfatório. Como exemplo, no APÊNDICE B apresenta-se o resumo do artigo intitulado “*Optimización multiobjetivo de un eje con algoritmo basado en quimiotaxis de bacterias*” (Otimização multi-objetivo de um eixo com algoritmo baseado em quimiotaxia de bactérias), o qual foi apresentado no 8º Congresso Ibero-americano de Engenharia Mecânica que se realizou na cidade de Cusco, Peru, em outubro de 2007. Neste artigo explica-se uma das muitas propostas testadas antes de se encontrar um algoritmo competitivo e eficiente.

De aqui para frente, o algoritmo de otimização multi-objetivo desenvolvido será chamado pela sigla BCMOA, correspondente ao seu nome em inglês (*Bacterial Chemotaxis Multiobjective Optimization Algorithm*). O BCMOA trabalha com regras que simulam de forma muito simplificada as respostas quimiotáticas das bactérias.

Início:

No BCMOA define-se um espaço k -dimensional de variáveis no qual cada coordenada representa o valor de uma das k variáveis de decisão. Este espaço de variáveis para efeitos da simulação do processo natural de quimiotaxia, representará a “superfície de movimentação” da colônia de bactérias. Define-se também um espaço de funções que representa os valores das funções objetivo avaliadas para cada localização das bactérias no espaço de variáveis. A colônia de bactérias esta conformada por S membros que inicialmente são localizados em posições aleatórias na superfície de movimentação.

Avaliação das funções objetivo e classificação por não dominância:

Os valores das funções objetivo são calculados para cada bactéria na localização inicial. Simulando o ambiente das bactérias reais, os valores das funções objetivo representam a quantidade de nutrientes presentes nessa localização. Aplicando o procedimento de classificação rápida por não dominância (*fast nondominated sorting procedure*) (DEB et al., 2002), as atuais localizações são classificadas em Frentes Ótimas de Pareto (FOP). As bactérias cujas localizações representam soluções não dominadas são inclusas dentro do primeiro conjunto de Frente Ótima de Pareto ou FOP1 e são chamadas de bactérias fortes devido a que o seu ambiente é rico em nutrientes tendo elas suficiente comida. As bactérias restantes são classificadas em diferentes conjuntos de FOP de acordo com o critério de não dominância e são chamadas de bactérias fracas devido a que o seu ambiente é pobre em nutrientes.

Estratégia quimiotáctica para bactérias fortes:

Fazendo uso da sua memória espaço-temporal, as bactérias fortes comparam a sua atual localização com a anterior pelo critério de não dominância (DEB et al., 2002). Como resultado dessas comparações cada bactéria forte reage de acordo com algum dos dois possíveis movimentos: se uma das localizações (anterior ou atual) domina a outra, a bactéria movimenta-se até a localização não dominada e desde ali faz um giro muito curto em uma direção aleatória sem se afastar muito da atual fonte de nutrientes. De outro lado, se nenhuma

das localizações atual ou anterior domina a outra, desde a sua atual localização, a bactéria faz um giro maior, também em direção aleatória.

A figura 45 ilustra a estratégia quimiotática de movimentos para as bactérias fortes. O espaço de busca representado na figura tem duas variáveis. Os círculos com linha pontuada representam as localizações anteriores das bactérias; os círculos verdes representam as localizações atuais das bactérias e os círculos roxos representam as localizações das bactérias depois de aplicar a estratégia quimiotática de movimentos.

Figura 45. Estratégia quimiotática para bactérias fortes em BCMOA

Comunicação dentro da colônia e estratégia quimiotática para bactérias fracas:

Simulando a comunicação natural entre bactérias, as bactérias fortes enviam um sinal ao resto das bactérias da colônia, indicando que na sua localização existe uma fonte de nutrientes. As bactérias fracas tomam vantagem desse intercambio de informação da seguinte forma: as bactérias fracas sabem que ao se movimentar na direção de uma bateria forte vão encontrar uma fonte de nutrientes, então cada bactéria fraca seleciona aleatoriamente uma bactéria forte

e determina a direção que leva desde sua atual localização até a localização da bactéria forte selecionada. Conhecendo essa direção, avança até uma localização aleatória muito próxima da bactéria forte e desde ali se movimenta um passo longo em linha reta (passo de natação) seguindo a direção anteriormente determinada.

A figura 46 ilustra a estratégia quimiotática de movimentos para as bactérias fracas. O espaço de busca representado na figura tem duas variáveis. Os pontos azuis representam as bactérias fracas. Os pontos verdes representam as bactérias fortes e os pontos roxos representam as bactérias fracas depois de aplicar a estratégia quimiotática de movimentos.

Figura 46. Estratégia quimiotática para bactérias fracas em BCMOA

Com a aplicação da estratégia quimiotática para todas as bactérias, considera-se que a colônia fez um passo quimiotático o qual representa uma iteração do algoritmo.

Preservação da diversidade:

Para estimular a diversidade das soluções encontradas, desenvolveram-se validações com a aplicação do parâmetro de densidade den_i proposto em (KUMAR, 2007). Para uma solução, o cálculo deste parâmetro é baseado na distância até sua vizinha mais próxima na FOP à qual pertence, dando um estimativo da densidade das soluções nessa frente. Os maiores valores representam as melhores soluções. No BCMOA quando o tamanho da FOP1 é maior ou igual do que o 50% do tamanho da colônia, o parâmetro de densidade é calculado para as bactérias fortes. Na aplicação da estratégia quimiotáctica, as bactérias fracas selecionam a sua bactéria forte dentre aquelas que conformam o 20% das melhores bactérias fortes ordenadas de acordo com o parâmetro de densidade.

Tamanho do passo:

A seleção do tamanho do passo mostrou ser um dos elementos críticos no projeto do algoritmo. Se o tamanho dos passos for muito pequeno, o processo de busca pode ficar preso num ótimo local, de outro lado se o tamanho dos passos for muito grande o processo de busca pode simplesmente não encontrar o ótimo global. Tomando isso em consideração, foram definidas equações para os tamanhos dos giros curtos ST pelos seus nomes em inglês (*Short Tumble*), giros longos LT (*Long Tumble*) e passos de natação SW (*Swim*). O cálculo do tamanho destes passos requer quase nenhuma intervenção do usuário e são atualizados automaticamente na medida em que avança o processo.

Define-se primeiro o fator Fac que diminui linearmente desde um até zero como mostra a eq.(33):

$$Fac(j) = (CHS_{\max} - j) / CHS_{\max} \quad (33)$$

na qual $CHS_{\max}$ é o valor máximo de passos quimiotáticos permitidos (critério de finalização) e j é o índice para os passos quimiotáticos.

Seja, k o índice para os parâmetros do problema, $POFI(j)$ o conjunto de soluções não dominadas, $\theta_k(j)_{strong}$ a localização de uma bactéria forte no parâmetro k , $\theta_k(j)_{weak}$ a localização de uma bactéria fraca no parâmetro k . Os tamanhos dos passos giro curto (ST), giro longo

(LT) e natação (SW) para cada parâmetro k estão definidos pelas eq.(34), eq.(35) e eq.(36), assim:

$$ST_k(j) = 0,001(\max(POF1(j))_k - \min(POF1(j))_k)Fac(j) \quad (34)$$

$$LT_k(j) = 0,01(\max(POF1(j))_k - \min(POF1(j))_k)Fac(j) \quad (35)$$

$$SW_k(j) = (\theta_k(j)_{strong} - \theta_k(j)_{weak})Fac(j) \quad (36)$$

na qual $\max(POF1)_k$ e $\min(POF1)_k$ são os valores máximo e mínimo para o parâmetro k dentre o conjunto de soluções $POF1(j)$.

Manipulação de restrições:

Para a manipulação de restrições utilizou-se uma estratégia simples. Para cada possível solução, calculam-se os valores das restrições e uma penalização constante por cada restrição violada é aplicada a aquelas soluções que violem as restrições. Assim, cada função objetivo penalizada é a função objetivo calculada, somada às penalizações. Dessa forma, durante o processo de classificação por não dominância, as soluções que violem restrições serão consideradas piores do que aquelas que tenham piores valores de funções objetivo, mas que não violam as restrições.

Todos os passos do algoritmo são detalhados a seguir:

1. Definir os parâmetros S (número de bactérias da colônia), $CHS_{\max}$ (número máximo de passos quimiotáticos), i (índice para cada bactéria), j (índice para cada passo quimiotático) e k (índice para os parâmetros). Iniciar a população; $j=1$.

A primeira localização para cada bactéria $\theta^i(j) \in \mathcal{R}^p$ é iniciada com valores reais randômicos dentro dos limites especificados para cada variável de decisão.

- a) Armazenar cada bactéria $b^i(j)$ e sua atual localização $\theta^i(j)$ num listado Bac .
- b) Atualizar os valores de $ST(j)$, $LT(j)$ e $SW(j)$ de acordo com as eq.(34), eq.(35) e eq.(36), respectivamente.

2. Para cada bactéria $b^i(j)$ no listado Bac , avaliar $J(\theta^i(j))$ que representa o valor das funções objetivo na sua atual localização e armazená-lo num listado $ObjFunVa$.
3. Classificar a localização de todas as bactérias aplicando o critério de não dominância a $J(\theta(j))$. Armazenar cada bactéria que foi classificada como não dominada num listado $POFI(j)$ e armazenar todas as bactérias dominadas num listado $Bacdom$.
4. Para cada bactéria em $POFI(j)$, gerar um vetor randômico $\Delta^i(j) \in \mathcal{R}^p$ no qual cada elemento $\Delta_k^i, k=1,2,\dots,p$, é um numero randômico dentre $[-1, 1]$.
5. Para cada bactéria em $POFI(j)$, comparar pelo critério de não dominância os valores atuais das suas funções objetivo $J(\theta(j))$ com os valores prévios $J(\theta(j))_{prev}$ armazenados em $ObjFunVal_{prev}$. Para $j=1$; $ObjFunVal_{prev}=ObjFunVal$. As novas localizações para cada bactéria forte estão definidas pelas eq.(37), eq.(38) e eq.(39).

Se $\theta^i(j)_{prev}$ domina a $\theta^i(j)$,

$$\theta_k^i(j+1) = \theta_k^i(j)_{prev} + ST_k(j)\Delta_k^i(j) \quad (37)$$

Se pelo contrario $\theta^i(j)$ domina a $\theta^i(j)_{prev}$,

$$\theta_k^i(j+1) = \theta_k^i(j) + ST_k(j)\Delta_k^i(j) \quad (38)$$

Se tanto $\theta^i(j)_{prev}$ quanto $\theta^i(j)$ são não dominadas,

$$\theta_k^i(j+1) = \theta_k^i(j) + LT_k(j)\Delta_k^i(j) \quad (39)$$

6. Para cada localização de uma bactéria fraca $\theta(j)$ em $Bacdom$, selecionar randomicamente a localização de uma bactéria forte $\theta(j)_{strong}$ de $POFI(j)$. Assim as novas localizações das bactérias fracas estão definidas pela eq.(40).

$$\theta_k^i(j+1) = \theta_k^i(j)_{strong} + \theta_k^i(j)_{strong} r1_k + SW_k(j) \quad (40)$$

na qual $r1_k$ é um numero randômico dentre $[-0,01, 0,01]$.

7. Aplicar uma estratégia de barreira de contenção. Devido à aplicação da estratégia quimiotáctica, algumas bactérias podem ter-se movimentado até posições fora dos limites permitidos, então deve se aplicar uma estratégia que faça com que essas bactérias voltem ao espaço de busca admissível. Neste algoritmo aplica-se uma adaptação da estratégia da parede absorvente (ROBINSON; RAHMAT-SAMII, 2004). A estratégia da parede absorvente original foi desenvolvida para a técnica PSO e consiste basicamente em levar a partícula até o limite da dimensão e uma vez ali, anular a sua velocidade. A figura 47 ilustra esta técnica adaptada ao BCMOA.

Figura 47. Método da Parede Absorvente

Nesse exemplo, a bactéria usa a estratégia quimiotáctica para se movimentar no espaço de busca. Nesse movimento traspassa o limite horizontal direito, e a estratégia da parede absorvente elimina o erro, levando a bactéria precisamente até esse limite, sem modificar suas outras coordenadas. Na figura 47, $\theta^i(j)$ é a localização inicial da bactéria i ; $\theta'^i(j+1)$ é a localização dessa bactéria depois do passo quimiotático (movimenta-se além do limite horizontal), e $\theta^i(j+1)$ é a localização definitiva da bactéria após aplicar a estratégia de parede absorvente.

8. $ObjFunVal_{prev} = ObjFunVal$.
9. Se $j \leq CHS_{max}$, $j = j + 1$ e ir até o passo 1a). Noutro caso ir até o passo 10
10. Deter o algoritmo.

Após a finalização do algoritmo, é recomendável adicionar um passo correspondente ao análise das soluções ótimas obtidas.

CAPÍTULO 7

7. VALIDAÇÕES DE DESEMPENHO DO ALGORITMO BCMOA

Neste capítulo apresentam-se as validações desenvolvidas para testar o desempenho do algoritmo projetado e as medições comparativas realizadas. Para comprovar o desempenho do algoritmo proposto, este foi implementado no programa Matlab 7.0 ®, num computador pessoal com 3 GHz de frequência, 3.34 GB de memória RAM e processador PENTIUM 4 de INTEL. Também foram implementados os algoritmos NSPSO e NSGA-II.

7.1 PROBLEMAS DE VALIDAÇÃO

Em 1999 Deb (1999a) foi o primeiro autor a abordar acerca da necessidade de desenvolver validações para os AOMO de forma sistemática. Atualmente existem problemas de validação, cuja Frente Ótima de Pareto é bem determinada. Alguns destes problemas têm se tornado em uma referência na avaliação do desempenho dos AOMO, já que são usados extensamente para desenvolver validações comparativas entre os algoritmos existentes e as novas propostas. Os problemas sem restrições resolvidos com o algoritmo desenvolvido neste trabalho foram: o problema SCH (SCHAFFER, 1987), o problema FON (FONSECA; FLEMING, 1998), o problema POL (POLONI, 1997), o problema KUR (KURSAWE, 1990), e da serie ZDT (ZITZLER; LAUMMANS; THIELE, 2002), os problemas ZDT1, ZDT2, ZDT3 e ZDT4. Também foram resolvidos os problemas de validação com restrições CONSTR, SRN e TNK (DEB et al., 2002). Em todos os casos os objetivos devem ser minimizados.

Os detalhes dos problemas são apresentados na Tabela 11 e na Tabela 12.

Tabela 11 - Problemas de validação sem restrições - continua

Problema	n	Limites Variav	Funções Objetivo	Ótimos
SCH	1	$[-10^3, 10^3]$	$f_1(x) = x^2$	$x \in [0, 2]$
FON	3	$[-4, 4]$	$f_2(x) = (x - 2)^2$ $f_1(x) = 1 - \exp\left(-\frac{3}{\sum_{i=1}^3 \left(x_i - \frac{1}{\sqrt{3}}\right)^2}\right)$ $f_2(x) = 1 - \exp\left(-\frac{3}{\sum_{i=1}^3 \left(x_i + \frac{1}{\sqrt{3}}\right)^2}\right)$	$x_1 = x_2 = x_3 \in \left[-\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}\right]$
POL	2	$[-\pi, \pi]$	$f_1(x) = \left[1 + (A_1 - B_1)^2 + (A_2 - B_2)^2\right]$ $f_2(x) = \left[(x_1 + 3)^2 + (x_2 + 1)^2\right]$ $A_1 = 0,5 \sin(1) - 2 \cos(1) + \sin(2) - 1,5 \cos(2)$ $A_2 = 1,5 \sin(1) - \cos(1) + 2 \sin(2) - 0,5 \cos(2)$ $B_1 = 0,5 \sin(x_1) - 2 \cos(x_1) + \sin(x_2) - 1,5 \cos(x_2)$ $B_2 = 1,5 \sin(x_1) - \cos(x_1) + 2 \sin(x_2) - 0,5 \cos(x_2)$	
KUR	3	$[-5, 5]$	$f_1(x) = \sum_{i=1}^{n-1} \left(-10 \exp\left(-0,2\sqrt{x_i^2 + x_{i+1}^2}\right)\right)$ $f_2(x) = \sum_{i=1}^n \left(x_i ^{0,8} + 5 \sin(x_i^3)\right)$	
ZDT1	30	$[0, 1]$	$f_1(x) = x_1$ $f_2(x) = g(x) \left[1 - \sqrt{\frac{x_1}{g(x)}}\right]$ $g(x) = 1 + 9 \frac{\left(\sum_{i=2}^n x_i\right)}{(n-1)}$	$x_1 \in [0, 1]$ $x_i = 0$ $i = 2, \dots, 30$
ZDT2	30	$[0, 1]$	$f_1(x) = x_1$ $f_2(x) = g(x) \left[1 - \left(\frac{x_1}{g(x)}\right)^2\right]$ $g(x) = 1 + 9 \frac{\left(\sum_{i=2}^n x_i\right)}{(n-1)}$	$x_1 \in [0, 1]$ $x_i = 0$ $i = 2, \dots, 30$
ZDT3	30	$[0, 1]$	$f_1(x) = x_1$ $f_2(x) = g(x) \left[1 - \sqrt{\frac{x_1}{g(x)}} - \frac{x_1}{g(x)} \sin(10\pi x_1)\right]$ $g(x) = 1 + 9 \frac{\left(\sum_{i=2}^n x_i\right)}{(n-1)}$	$x_1 \in [0, 1]$ $x_i = 0$ $i = 2, \dots, 30$

Tabela 11 - Problemas de validação sem restrições - conclusão

Problema	p	Limites Variav	Funções Objetivo	Ótimos
ZDT4	10	$x_1 \in [0, 1]$ $x_i \in [-5, 5]$	$f_1(x) = x_1$ $f_2(x) = g(x) \left[1 - \sqrt{\frac{x_1}{g(x)}} \right]$ $g(x) = 1 + 10(n-1) + \sum_{i=2}^n \left[x_i^2 - 10 \cos(4\pi x_i) \right]$	$x_1 \in [0, 1]$ $x_i = 0$ $i = 2, \dots, 10$

Tabela 12 - Problemas de validação com restrições

Problema	n	Limites Variav	Funções Objetivo	Restrições
CONSTR	2	$x_1 \in [0, 1, 1.0]$ $x_2 \in [0, 5]$	$f_1(x) = x_1$ $f_2(x) = (1 + x_2) / (x_1)$	$g_1(x) = x_2 + 9x_1 \geq 6$ $g_2(x) = -x_2 + 9x_1 \geq 1$
SRN	2	$x_i \in [-20, 20]$ $i = 1, 2$	$f_1(x) = (x_1 - 2)^2 + (x_2 - 1)^2 + 2$ $f_2(x) = 9x_1 - (x_2 - 1)^2$	$g_1(x) = x_1^2 + x_2^2 \leq 225$ $g_2(x) = x_1 - 3x_2 \leq -10$
TNK	2	$x_i \in [0, \pi]$ $i = 1, 2$	$f_1(x) = x_1$ $f_2(x) = x_2$	$g_1(x) = -x_1^2 - x_2^2 + 1 +$ $0, 1 \cos(16 \arctan(x_1 / x_2)) \leq 0$ $g_2(x) = (x_1 - 0, 5)^2 +$ $(x_2 - 0, 5)^2 \leq 0, 5$

7.2 VALIDAÇÕES APLICADAS AO BCMOA

A população inicial foi de 100 indivíduos para todos os problemas de validação. Para os problemas SCH, FON, POL e KUR, realizaram-se 100 iterações. Para os problemas da série ZDT e os problemas com restrições realizaram-se 250 iterações. Assim foram feitas 10000 e 25000 avaliações das funções objetivo respectivamente. Excepcionalmente para o problema ZDT4 o parâmetro $r1_k \in [-1, 1]$.

Nas figuras 48 a 55 apresentam-se os resultados para uma execução do algoritmo para cada um dos problemas de validação sem restrições. Em todas as figuras, a linha preta contínua representa a FOP do problema, e os pontos representam a última localização das bactérias encontrada com a aplicação do BCMOA.

Figura 48. Soluções encontradas para o problema SCH com BCMOA

Figura 49. Soluções encontradas para o problema FON com BCMOA

Figura 50. Soluções encontradas para o problema POL com BCMOA

Figura 51. Soluções encontradas para o problema KUR com BCMOA

Figura 52. Soluções encontradas para o problema ZDT1 com BCMAO

Figura 53. Soluções encontradas para o problema ZDT2 com BCMAO

Figura 54. Soluções encontradas para o problema ZDT3 com BCMOA

Figura 55. Soluções encontradas para o problema ZDT4 com BCMOA

Para apresentar detalhes gráficos do processo de otimização com o BCMOA, na figura 56 ilustram-se quatro etapas diferentes do processo para o problema ZDT1.

Figura 56. Iterações de ZDT1 com BCMOA. (a) Após 10 movimentos; (b) após 50 movimentos; (c) após 150 movimentos; (d) após 200 movimentos

Com o objetivo de conhecer melhor o comportamento do algoritmo BCMOA, as figuras 57 à 62 mostram os gráficos da quantidade de bactérias que aplicam um determinado passo quimiotático por cada iteração. Os valores representados são os valores médios obtidos após 10 execuções completas do algoritmo BCMOA sem utilização da técnica de preservação da diversidade para seis problemas de validação.

Figura 57. Passos quimiotáticos em SCH

Figura 58. Passos quimiotáticos em FON

Figura 59. Passos quimiotáticos em ZDT1

Figura 60. Passos quimiotáticos em ZDT2

Figura 61. Passos quimiotáticos em ZDT3

Figura 62. Passos quimiotáticos em ZDT4

Nas figuras 63 a 65 apresentam-se os resultados de uma execução do algoritmo para os três problemas de validação com restrições. Em todas as figuras, a zona cinza representa a região admissível e os pontos representam a última localização das bactérias encontrada com a aplicação do BCMOA.

Figura 63. Soluções encontradas para o problema CONSTR com BCMOA

Figura 64. Soluções encontradas para o problema SRN com BCMOA

Figura 65. Soluções encontradas para o problema TNK com BCMOA

7.2.1 Análise dos resultados das validações aplicadas ao BCMOA

Como se pode observar nas figuras 48 a 55 e figuras 63 a 65, o algoritmo BCMOA cumpre com os dois objetivos de um AOMO, já que não somente são encontradas soluções ótimas muito próximas à FOP, mas também essas soluções estão distribuídas em toda a Frente Ótima encontrada.

As figuras 57 à 62 mostram um padrão de comportamento no referente ao tipo de passo quimiotático utilizado. Durante as primeiras iterações predomina o passo de natação, enquanto que nas últimas iterações do processo de otimização, as bactérias aplicam majoritariamente o passo de giro longo. Em todos os problemas de validação resolvidos, o passo de giro curto apresenta uma incidência baixa, mas constante durante todo o processo de otimização.

As figuras 63 à 65 mostram que também para problemas com restrições o BCMOA consegue encontrar soluções próximas da FOP distribuídas em toda a Frente, mesmo quando essa é descontínua como no caso do problema TNK.

7.3 MÉTRICAS COMPARATIVAS DO DESEMPENHO

Devido à existência de outros AOMO que propõem diferentes procedimentos e cujas bases têm inspirações tão diversas, é necessário fazer uma avaliação dos resultados obtidos com este novo algoritmo proposto quando comparados com os resultados obtidos por alguns dos melhores AOMO conhecidos até agora. Precisa-se estabelecer se o desempenho do BCMOA é melhor ou pior do que outros algoritmos já existentes. Não existe ainda um algoritmo multi-objetivo “universal”, ou seja, não existe um algoritmo que se desempenhe melhor do que todos os outros para todos os tipos de POMO. Também não existe um consenso absoluto sobre as métricas que devem ser usadas para medir o desempenho dos algoritmos.

Neste trabalho aplicam-se os critérios de convergência e diversidade propostos em (DEB et al., 2002), os critérios de proximidade e a quantidade de soluções ótimas propostos em (LI,

2003) e os tempos de execução dos algoritmos. A seguir esclarece-se o cálculo destas métricas.

7.3.1 Diversidade Δ

O objetivo é quantificar a expansão das soluções ótimas ao longo da extensão total da FOP. Primeiro calcula-se a distância euclidiana d_i entre soluções consecutivas no conjunto de soluções não dominadas obtidas com o algoritmo. Depois se calcula a média destas distâncias $\bar{d}$. Do conjunto de soluções não dominadas, encontram-se as soluções extremas. Após usa-se a eq.(41) para calcular a não uniformidade da distribuição das soluções.

$$\Delta = \frac{d_f + d_l + \sum_{i=1}^{N-1} |d_i - \bar{d}|}{d_f + d_l + (N-1)\bar{d}} \quad (41)$$

Na qual d_f e d_l são as distâncias euclidianas entre as soluções extremas e as soluções de contorno do conjunto de soluções não dominadas encontradas e N é o número de soluções. A figura 66 ilustra estes conceitos.

7.3.2 Convergência

Para aplicar este critério é preciso conhecer muito bem a localização da FOP do problema. Primeiro devem se selecionar 500 soluções da FOP uniformemente distribuídas. Para cada solução não dominada obtida com o algoritmo, calcula-se a distância euclidiana mínima desde a solução até cada uma das 500 soluções ótimas selecionadas. O valor médio destas distâncias é a métrica de convergência. A figura 67 ilustra este conceito.

Figura 66. Métrica de diversidade

Figura 67. Métrica de convergência

7.3.3 Proximidade

Busca-se medir a proximidade do conjunto de soluções não dominadas encontradas com a FOP. Encontra-se a distância euclidiana média entre o conjunto das soluções ótimas encontradas FOP^* e a FPO de acordo com a eq.(42):

$$Prox = \frac{\left(\sum_{i=1}^{|FOP^*|} d_i^M \right)^{\frac{1}{M}}}{|FOP^*|} \quad (42)$$

Na qual M é o número de objetivos, d_i é a distância euclidiana entre a solução i do conjunto de soluções não dominadas FOP^* e o membro mais próximo da FOP.

7.3.4 Quantidade de soluções ótimas

As métricas anteriores não consideram a quantidade de soluções ótimas obtidas, o qual é um parâmetro importante para avaliar a eficácia do algoritmo. Espera-se que ao finalizar o processo de otimização a totalidade dos membros da população pertençam a FOP.

7.3.5 Tempos de execução

Para oferecer uma comparação entre os tempos de execução dos três algoritmos, com o auxílio da função “*cputime*” de Matlab, contabilizou-se o tempo de execução para 100 iterações aplicadas à solução dos problemas SCH e FON e 250 iterações para os problemas da serie ZDT.

7.4 APLICAÇÃO DAS MÉTRICAS COMPARATIVAS

Selecionaram-se dois algoritmos muito populares e representativos das técnicas dos AG e da PSO a fim de comparar com o desempenho do BCMOA na solução dos problemas de validação sem restrições SCH, FON, ZDT1, ZDT2, ZDT3 e ZDT4. Os algoritmos selecionados são o NSGA-II e o NSPSO, os quais foram explicados detalhadamente nas seções 3.4.3.1 e 3.4.5.1 respectivamente.

Para todos os algoritmos, a população inicial foi de 100 indivíduos. Para os problemas SCH e FON realizaram-se 100 iterações de cada algoritmo, e para os problemas da série ZDT realizaram-se 250 iterações para garantir que em todos os casos fossem feitas 10000 e 25000 avaliações das funções objetivo, respectivamente.

Para o algoritmo NSPSO, seguindo as recomendações de (LI, 2003), as constantes $c_1 = c_2 = 2$, ω diminui gradualmente desde 1 até 0,4, e V_{MAX} corresponde aos limites das variáveis de projeto. Para o algoritmo NSGA-II, seguindo as recomendações de (DEB et al., 2002), utiliza-se SBX (*simulated binary crossover*) e mutação polinomial. A probabilidade de cruzamento é de 0,9; a probabilidade de mutação é $1/NVP$, na qual NVP é o número de variáveis de projeto e os índices de distribuição para cruzamento e mutação são iguais a 20.

Devido à necessidade do conhecimento preciso da localização da FOP para a aplicação das métricas usadas, não foram comparados os resultados obtidos para os problemas POL e KUR. Para o algoritmo BCMOA foram desenvolvidas validações sem a utilização de um processo de estímulo à diversidade e com a aplicação do parâmetro de densidade den_i proposto em (KUMAR, 2007). Para os problemas de validação SCH, FON, ZDT1, ZDT2, ZDT3 e ZDT4, as Tabelas da 13 à 15 mostram os resultados obtidos em 10 execuções totais de cada algoritmo.

7.4.1 Métrica diversidade

Tabela 13 - Resultados da métrica de diversidade

		SCH	FON	ZDT1	ZDT2	ZDT3	ZDT4
BCMOA	Média	$7,60 \times 10^{-1}$	$4,85 \times 10^{-1}$	$5,98 \times 10^{-1}$	$6,89 \times 10^{-1}$	$11,58 \times 10^{-1}$	$6,27 \times 10^{-1}$
	Variância	$1,00 \times 10^{-3}$	$1,00 \times 10^{-3}$	$4,10 \times 10^{-3}$	$8,13 \times 10^{-3}$	$3,79 \times 10^{-3}$	$3,44 \times 10^{-2}$
BCMOA	Média	$7,55 \times 10^{-1}$	$4,81 \times 10^{-1}$	$5,59 \times 10^{-1}$	$5,29 \times 10^{-1}$	$9,32 \times 10^{-1}$	$4,71 \times 10^{-2}$
Estímulo Div.	Variância	$9,80 \times 10^{-4}$	$2,00 \times 10^{-3}$	$2,57 \times 10^{-3}$	$5,07 \times 10^{-3}$	$1,24 \times 10^{-3}$	$9,66 \times 10^{-4}$
NSGA-II	Média	$3,82 \times 10^{-1}$	$4,14 \times 10^{-1}$	$4,06 \times 10^{-1}$	$4,39 \times 10^{-1}$	$7,24 \times 10^{-1}$	$10,14 \times 10^{-1}$
	Variância	$1,00 \times 10^{-3}$	$9,80 \times 10^{-4}$	$1,26 \times 10^{-3}$	$1,19 \times 10^{-3}$	$5,27 \times 10^{-4}$	$1,09 \times 10^{-2}$
NSPSO	Média	$8,66 \times 10^{-1}$	$5,81 \times 10^{-1}$	$6,38 \times 10^{-1}$	$5,80 \times 10^{-1}$	$9,43 \times 10^{-1}$	$9,75 \times 10^{-1}$
	Variância	$8,60 \times 10^{-4}$	$3,16 \times 10^{-2}$	$2,69 \times 10^{-3}$	$1,08 \times 10^{-3}$	$8,08 \times 10^{-3}$	$2,98 \times 10^{-4}$

7.4.2 Métrica convergência

Tabela 14 - Resultados da métrica de convergência

		SCH	FON	ZDT1	ZDT2	ZDT3	ZDT4
BCMOA	Média	$3,28 \times 10^{-3}$	$2,77 \times 10^{-3}$	$1,19 \times 10^{-3}$	$8,37 \times 10^{-4}$	$3,56 \times 10^{-3}$	$5,94 \times 10^{-1}$
	Variância	$2,16 \times 10^{-8}$	$3,41 \times 10^{-8}$	$6,27 \times 10^{-8}$	$1,25 \times 10^{-8}$	$1,08 \times 10^{-7}$	$3,28 \times 10^{-2}$
BCMOA	Média	$3,12 \times 10^{-3}$	$2,57 \times 10^{-3}$	$1,23 \times 10^{-3}$	$8,18 \times 10^{-4}$	$3,64 \times 10^{-3}$	$4,18 \times 10^{-1}$
Estímulo Div.	Variância	$1,16 \times 10^{-8}$	$4,81 \times 10^{-8}$	$7,11 \times 10^{-8}$	$1,76 \times 10^{-9}$	$3,04 \times 10^{-8}$	$3,05 \times 10^{-4}$
NSGA-II	Média	$3,14 \times 10^{-3}$	$2,36 \times 10^{-3}$	$4,02 \times 10^{-3}$	$2,71 \times 10^{-3}$	$5,43 \times 10^{-3}$	$102,90 \times 10^{-1}$
	Variância	$4,64 \times 10^{-8}$	$1,24 \times 10^{-8}$	$3,14 \times 10^{-7}$	$7,69 \times 10^{-8}$	$1,00 \times 10^{-7}$	$476,30 \times 10^{-1}$
NSPSO	Média	$3,40 \times 10^{-3}$	$2,84 \times 10^{-3}$	$3,81 \times 10^{-3}$	$4,59 \times 10^{-3}$	$7,77 \times 10^{-3}$	$57,19 \times 10^{-1}$
	Variância	$7,40 \times 10^{-8}$	$9,04 \times 10^{-8}$	$3,22 \times 10^{-6}$	$3,24 \times 10^{-6}$	$7,20 \times 10^{-5}$	$87,02 \times 10^{-2}$

7.4.3 Métrica proximidade

Tabela 15 - Resultados da métrica de proximidade

		SCH	FON	ZDT1	ZDT2	ZDT3	ZDT4
BCMOA	Média	$3,78 \times 10^{-4}$	$3,62 \times 10^{-4}$	$2,02 \times 10^{-4}$	$1,00 \times 10^{-4}$	$5,23 \times 10^{-4}$	$6,51 \times 10^{-2}$
	Variância	$1,57 \times 10^{-10}$	$8,03 \times 10^{-10}$	$3,64 \times 10^{-9}$	$1,97 \times 10^{-10}$	$3,99 \times 10^{-9}$	$2,34 \times 10^{-4}$
BCMOA	Média	$3,66 \times 10^{-4}$	$3,58 \times 10^{-4}$	$2,40 \times 10^{-4}$	$9,73 \times 10^{-5}$	$5,13 \times 10^{-4}$	$4,96 \times 10^{-2}$
Estímulo Div.	Variância	$1,89 \times 10^{-10}$	$9,28 \times 10^{-9}$	$4,14 \times 10^{-9}$	$1,55 \times 10^{-11}$	$8,18 \times 10^{-10}$	$1,47 \times 10^{-6}$
NSGA-II	Média	$3,68 \times 10^{-4}$	$2,94 \times 10^{-4}$	$5,56 \times 10^{-4}$	$4,18 \times 10^{-4}$	$6,71 \times 10^{-4}$	$10,50 \times 10^{-1}$
	Variância	$3,45 \times 10^{-10}$	$4,23 \times 10^{-10}$	$7,95 \times 10^{-9}$	$9,64 \times 10^{-9}$	$9,13 \times 10^{-10}$	$4,84 \times 10^{-1}$
NSPSO	Média	$4,51 \times 10^{-4}$	$3,64 \times 10^{-4}$	$4,37 \times 10^{-4}$	$3,50 \times 10^{-4}$	$6,16 \times 10^{-4}$	$4,05 \times 10^{-1}$
	Variância	$2,68 \times 10^{-8}$	$1,86 \times 10^{-9}$	$3,40 \times 10^{-8}$	$1,65 \times 10^{-8}$	$2,95 \times 10^{-8}$	$4,32 \times 10^{-2}$

Devido ao desconhecimento da localização exata da FOP para os problemas KUR e POL, o que dificulta a aplicação das métricas nestes casos, as figuras 68 e 69 ilustram uma superposição dos resultados obtidos com o algoritmo NSGA-II (pontos azuis) e com o algoritmo BCMOA (pontos roxos) para estes problemas, de forma que se possa fazer uma comparação visual dos resultados.

A figura 70 ilustra os resultados obtidos com os algoritmos BCMOA (pontos roxos), NSGA-II (pontos azuis) e NSPSO (pontos pretos) para o problema ZDT4, após 250 iterações.

Figura 68. . Soluções encontradas para o problema POL com BCMOA e NSGA-II

Figura 69. Soluções encontradas para o problema KUR com BCMOA e NSGA-II

Figura 70. Soluções encontradas para o problema ZDT4 com BCMOA, NSGA-II e NSPSO

7.4.4 Quantidade de soluções ótimas

Para todos os problemas de validação e para os três algoritmos, a população final completa pertence à FOP, pelo qual esta métrica de comparação não permite estabelecer diferenças no desempenho dos algoritmos sendo trivial sua análise.

7.4.5 Tempos de execução

A Tabela 16 mostra os resultados em segundos para os algoritmos BCMOA (sem e com técnica de preservação da diversidade), NSPSO e NSGA-II. Apresenta-se a média dos resultados obtidos em 10 execuções totais de cada algoritmo.

Tabela 16 - Resultados dos tempos de execução

		SCH	FON	ZDT1	ZDT2	ZDT3	ZDT4
BCMOA	Média (s)	28,22	28,18	89,22	102,70	94,68	80,86
BCMOA	Média (s)	27,13	27,52	90,59	104,63	94,22	84,66
Estímulo Div.							
NSPSO	Média (s)	83,79	135,34	193,67	245,91	244,40	127,66
NSGA-II	Média (s)	100,40	98,28	245,82	249,14	262,53	301,43

7.4.6 Análise dos resultados das métricas comparativas

A Tabela 13 mostra que a aplicação da técnica de preservação de diversidade no BCMOA tem um efeito positivo na distribuição das soluções ótimas encontradas. Daqui mesmo pode se ver que o NSGA-II apresenta a melhor distribuição para todos os problemas de validação (benchmark problems), excetuando o ZDT4 para o qual o BCMOA mostrou o melhor desempenho. O algoritmo NSPSO apresentou o pior desempenho nesta métrica para todos os problemas de validação.

Os resultados apresentados pela Tabela 14 não são conclusivos no referente à influência positiva da utilização da estratégia de preservação da diversidade sobre a convergência do algoritmo BCMOA. A tabela mostra que para os problemas ZDT1 e ZDT3 a convergência do BCMOA foi melhor quando a técnica de preservação foi utilizada, não se comportando desta forma para os problemas restantes. De outro lado, quando comparado com os outros dos algoritmos, as duas versões do BCMOA (sem e com a utilização da técnica de preservação da distribuição) apresentaram a melhor convergência para os problemas da série ZDT. Embora o NSGA-II apresentou a melhor convergência para o problema FON, aconteceu o oposto para o problema ZDT4 para o qual apresentou a pior convergência.

A Tabela 15 mostra que o desempenho do BCMOA na métrica de proximidade é melhor quando é utilizada a técnica de preservação de diversidade para todos os problemas de validação excetuando o ZDT1. Essa tabela mostra também que o melhor desempenho nesta métrica para os problemas da série ZDT foi apresentado pelo BCMOA nas suas duas versões,

seguido pelo algoritmo NSPSO e finalmente pelo NSGA-II. Para os problemas SCH e FON, o algoritmo NSGA-II mostrou o melhor desempenho dos três, seguido pelas duas versões do BCMOA (com a utilização da técnica de preservação de diversidade e sem ela) e finalmente o NSPSO mostrou o pior desempenho para esses dois problemas.

Da Tabela 16 pode se observar que a aplicação da técnica de preservação da diversidade não modifica consideravelmente os tempos de execução do BCMOA, sendo que para os casos dos problemas SCH, FON e ZDT3, as médias dos tempos foram menores quando se utilizou a técnica de preservação do que a execução sem a utilização da técnica. Os outros resultados dessa tabela mostram que as duas versões do BCMOA precisam de um tempo de execução muito menor do que os tempos requeridos para a execução dos algoritmos NSPSO e NSGA-II para os seis problemas de validação. Assim por exemplo, para o problema FON, o tempo de execução requerido pelo BCMOA é 4,91 vezes menor do que o requerido pelo NSPSO e 3,57 vezes menor do que o requerido pelo NSGA-II; para o problema ZDT4 o BCMOA precisa 1,57 vezes menos tempo de execução do que o NSPSO e 3,73 vezes menos do que o requerido pelo NSGA-II. Pela sua vez, o algoritmo NSPSO precisou de tempos menores de execução do que o algoritmo NSGA-II para os problemas SCH, ZDT1, ZDT2, ZDT3 e ZDT4.

CAPITULO 8

8. APLICACÕES DO ALGORITMO BCMOA EM PROJETO MECÂNICO

Os resultados satisfatórios com os problemas de validação do algoritmo BCMOA permitem extrapolar seu uso na solução de problemas práticos de projeto mecânico. Neste capítulo, mediante um exemplo simples, explica-se detalhadamente a abordagem que permite aplicar o algoritmo BCMOA na solução de POMO de projeto mecânico. Finalmente apresentam-se três aplicações do algoritmo na solução de problemas de otimização dimensional de eixos com diferentes condições de carregamento e restrições. As quatro aplicações apresentadas foram implementadas em Matlab 7.0.

8.1 FORMULAÇÃO DE PROBLEMAS DE PROJETO MECÂNICO (PPM) COMO PROBLEMAS DE OTIMIZAÇÃO MULTI-OBJETIVO

Supõe-se que há o seguinte problema de projeto mecânico:

Deseja-se projetar uma peça de aço de forma retangular com um orifício circular no centro como a que se ilustra na figura 71. Conhece-se o material (aço com $S_y = 462$ MPa), as condições de carregamento e a geometria da peça, com exceção do diâmetro d do orifício. O problema consiste em encontrar o conjunto de soluções que maximizem simultaneamente o diâmetro do orifício e o fator de segurança do projeto.

Figura 71. Desenho esquemático do PPM 1

Esse é claramente um problema de otimização multi-objetivo. Por um lado procura-se o máximo valor do diâmetro que se pode furar no centro da placa e por outro, deseja-se maximizar o fator de segurança. Os dois objetivos são contraditórios; no caso de se aumentar o tamanho do diâmetro do orifício, a secção transversal será menor e como consequência, as tensões normais máximas às que a placa está submetida aumentarão e farão com que o fator de segurança diminua.

Como já foi mencionado, para os POMO não existe uma solução única. A tarefa de otimização consiste em balancear os dois objetivos até encontrar valores aceitáveis para cada um deles. Para essa geometria e essas condições de carregamento, sabe-se que a placa está submetida a tensões normais máximas σ que se calculam com a eq.(43):

$$\sigma = K \frac{P}{A} \quad (43)$$

Na qual P é a carga normal, A é a área da secção transversal da peça e K é o fator de concentração de tensões dado pela eq.(44) (BEER; JHONSTON; DEWOLF, 2002):

$$K \cong 3,0039 - 3,753 \left(\frac{d}{0,08} \right) + 7,9735 \left(\frac{d}{0,08} \right)^2 - 9,2659 \left(\frac{d}{0,08} \right)^3 + 1,8145 \left(\frac{d}{0,08} \right)^4 - 2,9684 \left(\frac{d}{0,08} \right)^5 \quad (44)$$

Finalmente, o fator de segurança do projeto, está definido pela eq.(45):

$$FS = \frac{S_y}{\sigma} \quad (45)$$

Na qual S_y é a resistência ao escoamento do material.

Além dos objetivos, devem ser consideradas as restrições do problema. Em primeiro lugar não são aceitáveis projetos que tenham um fator de segurança inferior a 1, pois de acordo com as teorias de falha estática, isso significa que o projeto não é seguro e a placa falhará por fluência plástica. Por outro lado, a equação (44) só é válida enquanto a relação entre o diâmetro do orifício e a largura da placa permaneça inferior ou igual a 0,65 (BEER; JHONSTON; DEWOLF, 2002). De acordo com isso, a formulação do POMO é a seguinte:

Maximizar:

$$f_1(x) = d \quad (46)$$

$$f_2(x) = FS = \frac{S_y}{\sigma} = \frac{462MPa}{\left(K \frac{P}{A} \right)} = \left(\frac{462 \times 10^6}{K * \left(\frac{3 \times 10^6}{0,08 - d} \right)} \right) \quad (47)$$

Sujeito às seguintes restrições:

$$g_1(x) = \frac{S_y}{\sigma} - 1 \geq 0 \quad (48)$$

$$g_2(x) = 0,65 - \frac{d}{0,08} \geq 0 \quad (49)$$

Nas quais $0 \leq d < 0,08$.

A figura 72 apresenta as soluções ótimas encontradas após 100 iterações do algoritmo BCMOA para o PPM 1, com uma população inicial de 100 bactérias.

Figura 72. Soluções ótimas para o PPM 1

Como pode se ver na figura 72, a aplicação do algoritmo BCMOA permite obter tantas soluções ótimas como bactérias tenha a população inicial utilizada.

Aplicando esse mesmo procedimento para definir os problemas, o algoritmo BCMOA é muito útil na solução de problemas de otimização de projeto mecânico. A seguir explicam-se detalhadamente três exemplos de aplicação do algoritmo BCMOA na otimização dimensional de eixos submetidos a diversos tipos de carregamentos e com diferentes variáveis de projeto.

8.2 OTIMIZAÇÃO DIMENSIONAL DE EIXOS

Um eixo é um elemento de máquina rotatório cujas funções principais são as de transmitir movimento, pares de torção e potencia. A importância destes elementos de máquina reside no fato de que a sua presença é indispensável em uma alta percentagem das máquinas.

Em condições de operação, como consequência das forças normais, as forças de flexão e as forças torsionais que atuam sobre ele, o eixo é submetido a uma combinação de tensões normais e cisalhantes aplicadas ciclicamente. Devido à presença destas tensões cíclicas multiaxiais, no projeto de eixos devem ser tomados em conta dois tipos gerais de critérios: i) critérios de resistência os quais se referem à prevenção de falhas devido à resistência a fluência do material e a resistência à fadiga e ii) critérios de rigidez os quais se focam em manter as deformações do elemento dentro de valores permitidos.

Por outro lado, embora teoricamente os elementos rotativos devam ser projetados para serem intrinsecamente balanceados pela sua geometria, com os processos de manufatura existentes hoje, é impossível produzir qualquer eixo com a massa perfeitamente balanceada. Esse desbalance residual junto com a carga variante no tempo, fazem com que o elemento vibre e entre em ressonância quando alcança uma velocidade de rotação igual à frequência natural. A amplitude da resposta vibratória pode eventualmente levar o sistema à falha catastrófica. As severas consequências de operar um sistema rotatório perto da sua velocidade crítica, fazem com que no processo de projeto de um eixo devam ser consideradas precauções especiais a fim de evitar excitar o sistema em suas frequências críticas ou perto delas.

A natureza das cargas envolvidas na sua operação e a dependência que tem a vida útil das máquinas rotatórias do correto funcionamento dos eixos, tornam o projeto destes elementos uma tarefa sistemática e complexa. A metodologia tradicional de projeto é um processo altamente iterativo (NORTON, 2004) que começa com uma geometria estimada (diâmetros e comprimentos das seções) tomando em conta os requerimentos funcionais, a localização dos apoios e as tensões. Essa geometria preliminar é verificada de acordo com as deflexões admissíveis e é adotado um procedimento de correção o qual usualmente consiste em incrementar a rigidez do eixo mediante o aumento dos diâmetros das seções. Essa nova geometria é verificada de novo, desta vez considerando o valor mínimo admissível para a

frequência natural. Esse processo continua até encontrar um projeto final com um fator de segurança satisfatório.

O resultado final do processo tradicional de projeto de eixos é uma geometria que satisfaz os requerimentos de projeto, mas que não é necessariamente um projeto ótimo.

A seguir aplica-se o algoritmo BCMOA na solução de três problemas de projeto mecânico de eixos, eliminando assim o desgastante processo iterativo e resultando em várias soluções ótimas para cada problema.

8.2.1 PPM 2 - Eixo submetido a torque variável

A seguir explicam-se as condições iniciais e os dados conhecidos para o segundo problema de projeto mecânico resolvido, que consiste em otimizar a geometria de um eixo de máquina ferramenta submetido a um torque variável, pelo tanto deve se considerar a resistência à fadiga quando existem componentes alternantes e medias para os momentos fletores e os torques.

Precisa-se projetar a geometria de um eixo como o que se mostra na figura 73. No ponto A encontra-se instalada uma polia para correia trapezoidal e no ponto C encontra-se instalada uma engrenagem de dentes retos. O eixo estará apoiado em mancais nos pontos B e D. A potencia do motor varia desde um valor mínimo até um valor máximo. Os outros dados relevantes, incluindo as faixas de valores que podem assumir as variáveis, apresentam-se na Tabela 17 e na Tabela 18.

Figura 73. Desenho esquemático dos eixos dos PPM 2 e PPM 3

Tabela 17 - Parâmetros fixos do PPM 2

Polia A	Diâmetro: 0,3 m
	Peso: 220 N
Engrenagem de dentes retos C	Diâmetro: 0,2 m
	Peso: 70 N
	Ângulo de pressão α : 20°
Motor	Potencia máxima: 7500 watts
	Potencia mínima: 1100 watts
Material	Aço AISI 1040

Tabela 18 - Variáveis do PPM 2

a	b	c	Vel	d_1	d_2	d_3	L_1
(m)	(m)	(m)	(rpm)	($\times 10^{-2}$ m)	($\times 10^{-2}$ m)	($\times 10^{-2}$ m)	(m)
0,15	0,2	0,45	1200	[0,5, 2,5]	[2,5, 5]	[0,5, 2,5]	[0,2, 0,5]

O problema de otimização multi-objetivo consiste em minimizar simultaneamente o peso do eixo e a deflexão lateral máxima. O eixo deve cumprir os requerimentos de resistência à fadiga com um fator de segurança mínima de 2,2 para carga flutuante. Além disso, a deflexão lateral máxima não pode superar os $12,7 \times 10^{-5}$ m.

8.2.1.1 Definição do PPM 2 como um POMO

Neste problema, as variáveis de decisão são os diâmetros das três seções do eixo d_1 , d_2 , d_3 e o comprimento da seção central do eixo L_1 . Para esse caso $L_2 = (L - L_1) / 2$.

Dadas as condições iniciais explicadas anteriormente, as equações que definem o POMO são as seguintes:

Minimizar:

$$f_1(x) = 60482,33(d_1^2 L_2 + d_2^2 L_1 + d_3^2 L_2) \quad (47)$$

$$f_2(x) = \sqrt{(y_{xy})^2 + (y_{xz})^2} \quad (48)$$

Sujeito a:

$$g_1(x) = d_1 - \left\{ \frac{32N_f}{\pi} \left[\frac{\sqrt{(K_{f1}M_{a1})^2 + \frac{3}{4}(K_{fs1}T_{a1})^2}}{S_f} + \frac{\sqrt{(K_{fm1}M_{m1})^2 + \frac{3}{4}(K_{fsm1}T_{m1})^2}}{S_{ut}} \right] \right\}^{\frac{1}{3}} \geq 0 \quad (49)$$

$$g_2(x) = d_2 - \left\{ \frac{32N_f}{\pi} \left[\frac{\sqrt{(K_{f2}M_{a2})^2 + \frac{3}{4}(K_{fs2}T_{a2})^2}}{S_f} + \frac{\sqrt{(K_{fm2}M_{m2})^2 + \frac{3}{4}(K_{fsm2}T_{m2})^2}}{S_{ut}} \right] \right\}^{\frac{1}{3}} \geq 0 \quad (50)$$

$$g_3(x) = d_3 - \left\{ \frac{32N_f}{\pi} \left[\frac{\sqrt{(K_{fs}M_{a3})^2 + \frac{3}{4}(K_{fs}T_{a3})^2}}{S_f} + \frac{\sqrt{(K_{fm3}M_{m3})^2 + \frac{3}{4}(K_{fsm3}T_{m3})^2}}{S_{ut}} \right] \right\}^{\frac{1}{3}} \geq 0 \quad (51)$$

$$g_4(x) = 0,000127 - y_{\max} \geq 0 \quad (52)$$

A eq.(47) representa o peso do eixo, na qual os valores das dimensões d e L são os diâmetros e os comprimentos das três seções do eixo.

A eq.(48) representa a deflexão lateral máxima do eixo a qual é calculada com a aplicação de integração numérica.

As eq.(49), eq.(50) e eq.(51) representam os diâmetros mínimos admissíveis de acordo com a teoria de falha por fadiga (NORTON, 2004). Nessas equações K_f e K_{fm} são fatores de concentração de tensão de fadiga por flexão para componentes media e alternantes, respectivamente; K_{fs} e K_{fsm} são fatores de concentração torcional de fadiga para componentes media e alternantes, respectivamente; M_m e M_a são as componentes media e alternante do momento flexor, respectivamente; T_m e T_a são as componentes media e alternante do torque, respectivamente; S_f é a resistência à fadiga, S_{ut} é a resistência ultima a tração e N_f é o fator de segurança à fadiga desejado.

Na eq.(52) $y_{\max}$ é a deflexão lateral máxima do eixo.

8.2.1.2 Soluções obtidas com BCMOA para o PPM 2

Após 500 iterações do algoritmo BCMOA com uma população inicial de 100 bactérias, obtiveram-se 100 soluções que pertencem à mesma Frente Ótima. Na Tabela 19 apresentam-se nas duas primeiras linhas os valores para as variáveis de projeto das soluções extremas, aquela com o menor valor para o peso e a solução para o menor valor de deflexão lateral. As

linhas terceira e quarta mostram outros dois exemplos das soluções ótimas obtidas, selecionados aleatoriamente dentre a população.

Tabela 19 – Exemplos de soluções para o PPM 2

d_1 (m)	d_2 (m)	d_3 (m)	L_2 (m)	<i>Peso</i> (N)	<i>Deflexão</i> (m)
$1,55 \times 10^{-2}$	$3,20 \times 10^{-2}$	$2,21 \times 10^{-2}$	$20,07 \times 10^{-2}$	21,28	$1,24 \times 10^{-4}$
$1,01 \times 10^{-2}$	$5,00 \times 10^{-2}$	$2,53 \times 10^{-2}$	$47,61 \times 10^{-2}$	74,64	$1,52 \times 10^{-5}$
$1,54 \times 10^{-2}$	$3,83 \times 10^{-2}$	$2,54 \times 10^{-2}$	$20,00 \times 10^{-2}$	28,50	$6,45 \times 10^{-5}$
$1,47 \times 10^{-2}$	$4,58 \times 10^{-2}$	$2,53 \times 10^{-2}$	$31,79 \times 10^{-2}$	47,76	$2,68 \times 10^{-5}$

A figura 74 mostra um desenho esquemático comparativo a escala das geometrias definitivas dos quatro eixos da Tabela 19. A figura 75 mostra a Frente Ótima obtida e os valores das funções objetivo para as 100 soluções ótimas obtidas.

Figura 74. Desenho esquemático comparativo de geometrias de eixos do PPM 2

Figura 75. Soluções ótimas para o PPM 2

8.2.2 PPM 3 - Eixo com torques constantes, variando o ponto de apoio e considerando a velocidade crítica

Considerando a configuração apresentada na figura 73, precisa-se encontrar a geometria ótima do eixo que minimize simultaneamente o peso e a deflexão lateral máxima. Neste caso, a potência do motor permanece constante. Além das restrições referentes aos diâmetros mínimos admissíveis para um fator de segurança à fadiga de 2,2, e à deflexão lateral máxima, a velocidade crítica do eixo não pode ser inferior a quatro vezes a velocidade de trabalho. Neste caso, além dos diâmetros das seções, pretende-se encontrar a localização ótima do mancal no ponto B.

Os dados dos parâmetros fixos e as faixas de valores que podem adotar as variáveis relevantes para esse problema apresentam-se na Tabela 20 e na Tabela 21.

Tabela 20 - Parâmetros fixos do PPM 3

Polia A	Diâmetro: 0,3 m
	Peso: 220 N
Engrenagem de dentes retos C	Diâmetro: 0,2 m
	Peso: 70 N
	Ângulo de pressão α : 20°
Motor	Potencia: 7500 watts
Material	Aço AISI 1040

Tabela 21 - Variáveis do PPM 3

b	c	L_1	L_2	Vel	d_1	d_2	d_3	a
(m)	(m)	(m)	(m)	(rpm)	($\times 10^{-2}$ m)	($\times 10^{-2}$ m)	($\times 10^{-2}$ m)	(m)
0,15	(0,65- a)	0,2	0,3	1600	[0,5, 2,5]	[2,5, 5]	[0,5, 2,5]	[0,05, 0,5]

8.2.2.1 Definição do PPM 3 como um POMO

Neste problema, as variáveis de decisão são os diâmetros das três seções do eixo d_1 , d_2 , d_3 e a grandeza a que define a localização do mancal em B. Para esse caso $L_2 = (L - L_1) / 2$. Dadas as condições iniciais explicadas anteriormente, as equações que definem o POMO são as seguintes:

Minimizar:

$$f_1(x) = 60482,33(d_1^2 L_2 + d_2^2 L_1 + d_3^2 L_2) \quad (47)$$

$$f_2(x) = \sqrt{(y_{xy})^2 + (y_{xz})^2} \quad (48)$$

Sujeito a:

$$g_1(x) = d_1 - \left\{ \frac{32N_f}{\pi} \left[\left(k_f \frac{M_{a1}}{S_f} \right)^2 + \frac{3}{4} \left(\frac{T_{m1}}{S_y} \right)^2 \right]^{\frac{1}{2}} \right\}^{\frac{1}{3}} \geq 0 \quad (53)$$

$$g_2(x) = d_2 - \left\{ \frac{32N_f}{\pi} \left[\left(k_f \frac{M_{a2}}{S_f} \right)^2 + \frac{3}{4} \left(\frac{T_{m2}}{S_y} \right)^2 \right]^{\frac{1}{2}} \right\}^{\frac{1}{3}} \geq 0 \quad (54)$$

$$g_3(x) = d_3 - \left\{ \frac{32N_f}{\pi} \left[\left(k_f \frac{M_{a3}}{S_f} \right)^2 + \frac{3}{4} \left(\frac{T_{m3}}{S_y} \right)^2 \right]^{\frac{1}{2}} \right\}^{\frac{1}{3}} \geq 0 \quad (55)$$

$$g_4(x) = 0,000127 - y_{\max} \geq 0 \quad (56)$$

$$g_5(x) = \omega_n - 4\omega_t \geq 0 \quad (57)$$

As eq.(47) e eq.(48) que são repetidas neste caso, representam o peso e a deflexão lateral máxima ao igual que para o PPM 2.

As eq.(53), eq.(54) e eq.(55) representam os diâmetros mínimos admissíveis de acordo com a teoria de falha por fadiga (NORTON, 2004). Nessas equações K_f é o fator de concentração de tensão de fadiga por flexão para a componente alternante, M_a é a componente alternante do momento flexor, T_m é a componente media do torque, S_f é a resistência à fadiga, S_y é a resistência de escoamento em tração e N_f é o fator de segurança à fadiga desejado.

A eq.(56) representa a restrição de deflexão máxima admissível, a qual não pode ser maior do que $12,7 \times 10^{-5}$ m. A deflexão lateral máxima é calculada por integração numérica.

A eq.(57) representa a restrição no valor mínimo da primeira velocidade crítica do eixo que não pode ser inferior a quatro vezes a velocidade de trabalho. A frequência natural é calculada mediante o método de Rayleigh (NORTON, 2004).

8.2.2.2 Soluções obtidas com BCMOA para o PPM 3

Após 500 iterações do algoritmo BCMOA com uma população inicial de 100 bactérias, obtiveram-se 100 soluções que pertencem à mesma Frente Ótima. Na Tabela 22 apresentam-se nas duas primeiras linhas os valores para as variáveis de projeto das soluções extremas,

aquela com o menor valor para o peso e a solução para o menor valor de deflexão lateral. As linhas terceira e quarta mostram outros dois exemplos das soluções ótimas obtidas, selecionados aleatoriamente dentre a população.

Tabela 22 – Exemplos de soluções para o PPM 3

d_1 ($\times 10^{-2}$ m)	d_2 ($\times 10^{-2}$ m)	d_3 ($\times 10^{-2}$ m)	a ($\times 10^{-2}$ m)	Peso (N)	Deflexão (m)
2,81	4,00	3,14	5,03	60,13	$1,28 \times 10^{-5}$
4,96	4,99	3,55	7,65	105,34	$6,30 \times 10^{-7}$
3,07	4,35	3,38	5,89	70,94	$5,58 \times 10^{-6}$
4,40	4,93	3,66	7,44	98,50	$8,02 \times 10^{-7}$

A figura 76 mostra um desenho esquemático comparativo a escala das geometrias definitivas dos quatro eixos da Tabela 22. A linha vertical na primeira seção dos eixos representa a localização ótima do mancal em B. A figura 77 mostra a Frente Ótima obtida e os valores das funções objetivo para as 100 soluções ótimas obtidas.

Figura 76. Desenho esquemático comparativo de geometrias de eixos do PPM 3

Figura 77. Soluções ótimas para o PPM 3

8.2.3 PPM 4 - Eixo com torque constante, considerando a velocidade crítica calculada por elementos finitos

A ultima aplicação do algoritmo BCMOA na otimização dimensional de um elemento mecânico refere-se ao eixo de uma máquina ferramenta como o que se apresenta na figura 78.

O eixo está apoiado em mancais nos pontos A e E. No ponto B está instalada uma polia para correia trapezoidal e nos pontos C e D encontram-se engrenagens de dentes retos que distribuem o torque até as ferramentas. O torque permanece constante.

Para esse caso, o objetivo da otimização é simultaneamente minimizar o peso de eixo e maximizar fator de segurança à fadiga, objetivos que são logicamente contraditórios entre si. Além de ter como restrições os valores mínimos dos diâmetros de acordo com a resistência a fadiga, busca-se manter a velocidade crítica mínima do eixo em uma faixa de valores, essa velocidade não pode ser inferior a quatro vezes nem superior a vinte vezes a velocidade de trabalho. A velocidade crítica dos eixos é calculada com o método dos elementos finitos de

acordo como modelo exposto em (NELSON; MCVAUGH, 1976). Os valores ótimos da geometria que se procuram são os diâmetros das quatro seções do eixo.

Figura 78. Desenho esquemático do eixo do PPM 4

As Tabelas 23 e 24 mostram os dados dos parâmetros fixos e as faixas de valores que podem adotar as variáveis relevantes para esse problema.

Tabela 23 - Parâmetros fixos do PPM 4

Polia B	Diâmetro: 0,2 m Largura: 0,06 m Material: aço
Engrenagem de dentes retos C	Diâmetro: 0,4 m Largura: 0,05 m Ângulo de pressão $\alpha : 20^\circ$ Material: aço
Engrenagem de dentes retos D	Diâmetro: 0,2 m Largura: 0,05 m Ângulo de pressão $\alpha : 20^\circ$ Material: aço
Motor	Potencia: 1100 watts
Material do eixo	Aço AISI 1040
$l = a = b = c = d$	0,1 m

Tabela 24 - Variáveis do PPM 4

ω (rpm)	d_1 ($\times 10^{-3}$ m)	d_2 ($\times 10^{-3}$ m)	d_3 ($\times 10^{-3}$ m)	d_4 ($\times 10^{-3}$ m)	$\omega_n \min$ (rpm)	$\omega_n \max$ (rpm)
2200	[5, 30]	[5, 30]	[5, 30]	[5, 30]	4ω	20ω

8.2.3.1 Definição do PPM 4 como um POMO

Neste problema, as variáveis de decisão são os diâmetros das quatro seções do eixo d_1 , d_2 , d_3 e d_4 . Dadas as condições iniciais explicadas anteriormente, as equações que definem o POMO são as seguintes:

Minimizar:

$$f_1(x) = 60482,33(d_1^2 l + d_2^2 l + d_3^2 l + d_4^2 l) \quad (58)$$

$$f_2(x) = \frac{1}{FS} = \frac{d_{\min \text{admissivel}}}{d_{\min \text{obtido}}} \quad (59)$$

Sujeito a:

$$g_i(x) = d_i - \left\{ \frac{32N_f}{\pi} \left[\left(k_f \frac{M_{ai}}{S_f} \right)^2 + \frac{3}{4} \left(\frac{T_{mi}}{S_y} \right)^2 \right]^{\frac{1}{2}} \right\}^{\frac{1}{3}} \geq 0 \quad (60)$$

para $i=1, \dots, 4$

$$g_5(x) = \omega_n - 4\omega_i \geq 0 \quad (61)$$

$$g_6(x) = 20\omega_i - \omega_n \geq 0 \quad (62)$$

As eq.(58) e eq.(59) representam o peso e o inverso do fator de segurança mínimo a fadiga, assim ao minimizar a função objetivo $f_2(x)$ está se maximizando o fator de segurança.

A eq.(60) representa os diâmetros mínimos admissíveis de acordo com a teoria de falha por fadiga (NORTON, 2004). Nessa equação i varia desde 1 até 4; assim representa as quatro primeiras restrições do problema. Nessas equações K_f é o fator de concentração de tensão de fadiga por flexão para a componente alternante, M_a é a componente alternante do momento flexor, T_m é a componente media do torque, S_f é a resistência à fadiga, S_y é a resistência de escoamento em tração e N_f é o fator de segurança à fadiga mínimo desejado, neste caso 2,2.

As eq.(61) e eq.(62) representam a faixa admitida de valores para a velocidade critica do eixo. Para esse caso, a frequência foi calculada com a ajuda do método de elementos finitos implementado em Matlab.

8.2.3.2 Soluções obtidas com BCMOA para o PPM 4

Após 500 iterações do algoritmo BCMOA com uma população inicial de 100 bactérias, obtiveram-se 100 soluções que pertencem à mesma Frente Ótima. Na Tabela 25 apresentam-se nas duas primeiras linhas os valores para as variáveis de projeto das soluções extremas, aquela com o menor valor para o peso e a solução para o maior valor de fator de segurança e mais dos exemplos aleatórios de soluções ótimas.

Tabela 25 - Exemplos de soluções para o PPM 4

d_1 ($\times 10^{-2}$ m)	d_2 ($\times 10^{-2}$ m)	d_3 ($\times 10^{-2}$ m)	d_4 ($\times 10^{-2}$ m)	<i>Peso</i> (N)	<i>Fator</i> <i>Segurança</i>	ω_h (rpm)
1,73	2,05	1,88	1,70	8,25	2,21	$2,16 \times 10^4$
2,48	2,91	2,69	2,44	16,68	6,27	$4,37 \times 10^4$
2,28	2,68	2,48	2,22	14,25	4,86	$3,70 \times 10^4$
2,30	2,73	2,50	2,28	14,71	5,09	$3,82 \times 10^4$

A figura 79 mostra um desenho esquemático comparativo a escala das geometrias definitivas dos quatro eixos da Tabela 25. A figura 80 mostra a Frente Ótima obtida e os valores das funções objetivo para as 100 soluções ótimas obtidas. A figura 81 mostra a Frente Ótima obtida e os valores das funções objetivo peso e inverso do Fator de Segurança, cuja minimização simultânea constituiu o problema de otimização multi-objetivo resolvido com BCMOA.

Figura 79. Desenho esquemático comparativo de geometrias de eixos do PPM 4

Figura 80. Soluções ótimas para o PPM 4

Figura 81. Soluções ótimas para o PPM 4: Peso - Inverso do Fator de Segurança

CAPÍTULO 9

9. CONCLUSÕES

9.1 CONTRIBUIÇÕES

- O projeto moderno de engenharia é um processo sistemático e complexo do qual faz parte a otimização matemática. Considera-se que um projeto é ótimo quando a solução que representa é o resultado da busca do melhor candidato dentro de um conjunto de alternativas que se encontram em um espaço de grande tamanho. Um processo de busca e exploração em espaços desconhecidos que foi altamente otimizado durante milhões de anos pela evolução natural é a quimiotaxia de bactérias. Seguindo a recente tendência de desenvolver sistemas artificiais que emulem comportamentos naturais, neste trabalho desenvolveram-se com sucesso duas técnicas de otimização baseadas na quimiotaxia de bactérias. Assim, a principal contribuição deste trabalho é a demonstração do potencial da quimiotaxia de bacterias como uma poderosa fonte de inspiração de técnicas de otimização distribuídas que resultem competitivas frente a outras técnicas bio-inspiradas já existentes.
- Foi projetado, desenvolvido e implementado o BCBTOA o qual é um algoritmo heurístico de otimização topológica para estruturas bidimensionais contínuas baseado no processo da quimiotaxia de bactérias.
- Com a aplicação do BCBTOA foram implementados e resolvidos com sucesso cinco problemas de otimização topológica de estruturas bidimensionais contínuas.

- Foi projetado, desenvolvido e implementado o BCMOA, o qual é um algoritmo de otimização multi-objetivo baseado no processo da quimiotaxia de bactérias, introduzindo com este, pela primeira vez, uma aplicação deste processo biológico na área da otimização multi-objetivo bio-inspirada.
- Com a aplicação do BCMOA, foram implementados e resolvidos com sucesso onze problemas de otimização multi-objetivo de validação que permitiram comprovar a eficiência e competitividade do algoritmo.
- Com a aplicação do BCMOA, foram implementados e resolvidos três problemas de otimização multi-objetivo de eixos, demonstrando a versatilidade e aplicabilidade do algoritmo na solução de problemas de projeto mecânico.

9.2 RESULTADOS

- O algoritmo BCBTOA consegue minimizar a função de energia interna de distorção das cinco estruturas otimizadas.
- Os resultados obtidos com a utilização do algoritmo BCBTOA na solução dos problemas de otimização topológica de estruturas bidimensionais contínuas, mostram padrões de desempenho altamente satisfatórios quando comparados com os resultados obtidos mediante o algoritmo SIMP.
- A interpretação física das estruturas resultantes com o a aplicação do algoritmo BCBTOA é direta, já que em cada elemento, de forma excludente, ou existe material ou há um vazio.
- O algoritmo BCMOA cumpre satisfatoriamente os dois objetivos de qualquer AOMO: i) para uma diversidade de problemas de otimização multi-objetivo é capaz de encontrar simultaneamente um número importante de soluções ótimas que se localizam muito perto ou mesmo sobre a FOP; ii) estas soluções mostram uma boa distribuição em toda a extensão da FOP encontrada.

- Os resultados obtidos mostram que para o algoritmo BCMOA o processo de exploração do espaço de busca é executado pelo passo de natação enquanto que o processo de refinamento é executado majoritariamente pelo passo de giro longo.
- O algoritmo BCMOA requer mínima intervenção do usuário. No caso dos quinze problemas resolvidos neste trabalho, unicamente para o problema ZDT4 teve que ser modificado o parâmetro $r1_k$.
- O algoritmo BCMOA é capaz de resolver problemas de otimização multi-objetivo de alta complexidade como o ZDT4, problemas que envolvem um numero importante de variáveis como o ZDT1 e o ZDT2, aqueles nos quais a FOP é não ocnítua como o FON, o KUR e o ZDT3, assim como problemas complexos com restrições como o CONSTR, o SRN e o TNK.
- Os resultados obtidos com a aplicação das métricas de comparação do desempenho dos algoritmos BCMOA, NSGA-II e NSPSO demonstram claramente a competitividade do algoritmo aqui desenvolvido. Na métrica de diversidade, o algoritmo BCMOA se mostrou com desempenho inferior em relação ao algoritmo NSGA-II, mas sempre superou ao algoritmo NSPSO. Nas métricas de convergência e de proximidade, o algoritmo BCMOA mostrou ser superior aos outros dois algoritmos na solução dos problemas da serie ZDT. Finalmente, na métrica dos tempos de execução, o algoritmo BCMOA foi bem superior aos outros dois algoritmos em todos os problemas de validação sem restrições, o que ratifica seu desempenho altamente satisfatório, o qual o converte em uma outra ferramenta válida entre as técnicas de otimização multi-objetivo bio-inspiradas.
- Com a aplicação do algoritmo BCMOA na solução de problemas de projeto mecânico, é possível obter um número de soluções ótimas finais igual ao número de bactérias iniciais da colônia, resultando em muitas e variadas opções, entre as quais o engenheiro pode selecionar aquela que melhor satisfaça os seus critérios como projetista.

9.3 EXPERIÊNCIAS ADQUIRIDAS

- Por serem algoritmos altamente intuitivos, os algoritmos bio-inspirados apresentados neste trabalho requereram inúmeros esforços e tentativas, incluindo mudanças de enfoque e sintonização de parâmetros até encontrar um desempenho satisfatório.
- A métrica de proximidade não permite detectar quando, por alguma falha no processo de otimização, várias ou todas as soluções encontradas convergem para um mesmo ponto, sendo nestes casos “enganosa” a informação obtida a partir desta métrica.
- O algoritmo BCBTOA mostrou-se extremamente sensível às mudanças nos parâmetros *Fator* e *R*. O parâmetro *Fator* define o valor mínimo admissível da função objetivo para um elemento finito da estrutura contínua, que permite considerar esse elemento como apto para a permanência de uma bactéria nele. O parâmetro *R* define a máxima distancia influenciada pelo sinal enviado por uma bactéria. Assim a definição correta destes parâmetros determina o sucesso do processo de otimização. Os valores desses parâmetros reportados neste trabalho são o resultado de uma sintonização por ensaio e erro, não se podendo estabelecer para estes uma regra geral para a sua estimação.

9.4 TRABALHOS FUTUROS

- A busca por melhores desempenhos dos algoritmos desenvolvidos, é uma tarefa incessante. Em quanto ao algoritmo BCBTOA podem-se dedicar esforços no estabelecimento de regras gerais para a estimação dos parâmetros *Fator* e *R*, assim como ao desenvolvimento e ensaio de novos critérios de finalização do algoritmo.
- Em quanto ao algoritmo BCMOA, podem se experimentar novas formas de calcular os tamanhos de passo, e novos métodos de estímulo à preservação da diversidade que tenham maior impacto nos resultados finais.

- Finalmente, as aplicações do BCMOA na otimização de projetos de engenharia podem ser estendidas a diversos campos, da mesma forma que já aconteceu com outros AOMO bio-inspirados.

REFERÊNCIAS

ACHARYAL, D.P.; PANDA, G. (2008). Effect of finite register length on Bacterial Foraging Optimization based ICA and Constrained Genetic Algorithm based ICA algorithm. In: PROCEEDINGS OF THE IEEE INTERNATIONAL CONFERENCE ON SIGNAL PROCESSING, COMMUNICATIONS AND NETWORKING. Chennai, India, 4-6 de janeiro de 2008, p.244-249.

AMOS, M.; HODGSON, D.; GIBBONS, A. (2005). Bacterial self-organisation and computation. Submetido a *International Journal of Unconventional Computing*. Disponível em: <http://arxiv.org/abs/q-bio/0512017>. Acesso em: março de 2006.

ADLER, J. (1966). Chemotaxis in bacteria. *Science*. v.153, p.708-716.

ANGERT, E.R.; CLEMENTS, K.; PACE, N. (1993). The largest bacterium. *Nature*. v.362, n.6417, p.239-241.

AXELROD, A. (1999). *Introduction to Biology: General Biology the Big 10 Way*. Blacklick, OH, Estados Unidos da América: McGraw-Hill Professional Book Group.

BARDWELL, L. et al. (2007). Mathematical models of specificity in cell signaling. *Biophysical Journal*. v.9, p.3425-3441.

BEER, F.; JHONSTON, E.R.; DEWOLF, J.T. (2002). *Mecanica de materiais*. 3a Ed. Mexico: McGraw-Hill.

BENDSØE, M.; KIKUCHI, N. (1988). Generating optimal topologies in structural design using a homogenization method. *Computer methods in applied mechanics and engineering*. v.71, n.2, p.197-224.

BENDSØE, M.P.; DIAZ, A.; KIKUCHI, N. (1993). Topology and generalized layout optimization of elastic structures. In: BENDSØE, M.P.; MOTA SOARES, C.A. (Eds.). TOPOLOGY DESIGN OF STRUCTURES. PROCEEDINGS OF. NATO ARW. Dordrecht: Kluwer. Sesimbra, Portugal, p.159-206.

BERG, H.C.; BROWN, D. (1972). Chemotaxis in Escherichia Coli analyzed by three-dimensional tracking. *Nature*. v.239, n.5374, p.500-504.

BERNARDINI, F.; GHEORGHE, M.; KRASNOGOR, N. (2007). Quorum sensing P systems. *Theoretical Computer Science*. v.371, n.1-2, p.20-33.

BISWAS, A. et al. (2007). A synergy of differential evolution and bacterial foraging optimization for global optimization. *Neural Network World*. v.17, n.6, p.607-626.

BONABEAU, E.; DORIGO, M.; THERAULAZ, G. (1999). *Swarm Intelligence: From Natural to Artificial Systems*. Cary, NC, Estados Unidos da América: Oxford University Press Incorporated.

BREMERMANN, H. (1974). Chemotaxis and optimization. *J Franklin Institute*. v.297, p.397-404.

CARLISLE, A.; DOZIER, G. (2001). An off-the-self PSO. In: PROCEEDINGS OF THE WORKSHOP ON PARTICLE SWARM OPTIMIZATION. Indianapolis, Estados Unidos da América, abril de 2001, p.1-6.

CHEN, T.C. et al. (2007). A Novel Optimization Approach: Bacterial-GA Foraging. In: PROCEEDINGS OF THE SECOND INTERNATIONAL CONFERENCE ON INNOVATIVE COMPUTING, INFORMATION AND CONTROL. Washington D.C., Estados Unidos da América, 5-7 de setembro de 2007, p.391-394.

CHINCHULUUN, A.; PARDALOS, P. (2007). A survey of recent developments in multiobjective optimization. *Annals of Operation Research*. v.154, n.1, p.29-50.

COELLO, C.A.C.; TOSCANO, G.; SALAZAR, M. (2004). Handling Multiple objectives with Particle Swarm Optimization. *IEEE Transactions on Evolutionary Computation*. v.8, n.3, p.256-279.

COELLO, C.A.C. (2006). Evolutionary multiobjective optimization: A historical view of the field. *IEEE Computational Intelligence Magazine*. v.1, n.1, p.28-36.

DANIELS, R.; VANDERLEYDEN, J.; MICHIELS, J. (2004). Quorum sensing and swarming migration in bacteria. *FEMS Microbiology Reviews*. v.28, n.3, p.261-289.

DANTZIG, G.; THAPA, M. (1997). *Linear Programming: Introduction*. New York: Springer-Verlag New York Incorporated.

DAS, R.; JONES, R.; CHANDRA, S. (2007). Damage tolerance based shape design of a stringer cutout using evolutionary structural optimisation. *Engineering Failure Analysis*. v.4, n.1, p.118-137.

DAS, R.; JONES, R.; XIE, Y. (2005). Design of structures for optimal static strength using ESO. *Engineering Failure Analysis*. v.12, n.1, p.61-80.

DASGUPTA, S. et al. (2008). Adaptive Computational Chemotaxis in Bacterial Foraging Algorithm. . In: PROCEEDINGS OF THE 2008 INTERNATIONAL CONFERENCE ON COMPLEX, INTELLIGENT AND SOFTWARE INTENSIVE SYSTEMS 2008, 4-7 de março de 2008, p.64-71.

DE CASTRO, L.; TIMMIS, J. (2002). An artificial immune network for multimodal function optimization. In: PROCEEDINGS OF THE IEEE CONGRESS ON EVOLUTIONARY COMPUTATION (CEC'02), 2002, Havaí, Estados Unidos da América, maio de 2002, v.1, p.699-674.

DE CASTRO, L.; VON ZUBEN, F. (2004). *Recent Developments in Biologically Inspired Computing*. Hershey, PA, Estados Unidos da América: Idea Group Publishing.

DE CASTRO, L. (2007). Fundamentals of natural computing: an overview. *Physics of Life Reviews*. v.4, n.1, p.1-36.

DE LIMA, C. (2002). Projeto de mecanismos flexíveis usando o método de otimização topológica. 145 p. Dissertação (Mestrado em Engenharia). Escola Politécnica da Universidade de São Paulo, São Paulo, 2002.

DEB, K. (1999). Evolutionary algorithms for multi-criterion optimization in engineering design. In: PROCEEDINGS OF EVOLUTIONARY ALGORITHMS IN ENGINEERING AND COMPUTER SCIENCE (EUROGEN-99). Jyväskylä, Finlandia, 29 de maio - 03 de junho de 1999, p.135-161.

DEB, K. (1999a). Multiobjective genetic algorithms: Problem difficulties and construction of test functions. *Evolutionary Computation*. v.7, n.3, p.205-230.

DEB, K. (2001). *Multi-Objective Optimization using Evolutionary Algorithms*. New York: John Wiley & Sons.

DEB, K. et al. (2002). A fast and elitist multiobjective Genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*. v.6, n.2, p.182-197.

DELGADO, A. (2000). Control of Nonlinear Systems Using a Self-Organizing Neural Network. *Neural Computing & Applications*. v.9, n.2, p.113-123.

DHARIWAL, A.; SUKHATME, G.S.; REQUICHA, A.A.G. (2004). Bacterium-inspired robots for environmental monitoring. In: PROCEEDINGS OF THE 2004 IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, New Orleans, Estados Unidos da América, 26 de abril - 1 de maio de 2004, v.2, p.1436-1443.

DIGGLE, S.; CRUSZ, S.A.; CÁMARA, M. (2007). Quorum sensing. *Current Biology*. v. 17, n. 21, p. R907-R910.

EFFATI, S.; JAFARZADEH, M. (2007) Nonlinear neural networks for solving the shortest path problem. *Applied Mathematics and Computation*. v.189, n.1, p.567-574.

EHRGOTT, M.; GANDIBLEUX, X. (Ed.) (2002). *Multiple Criteria Optimization: State of the Art Annotated Bibliographic Surveys*. Secaucus, NJ, USA: Kluwer Academic Publishers.

EISENBACH, M. (1990). Functions of the flagellar modes of rotation in bacterial motility and chemotaxis. *Molecular Microbiology*. v.4, n.2, p.161-167.

EISENBACH, M. et al. (2004). *Chemotaxis*. Singapura: Imperial College Press.

ERBAN, R. (2005). From individual to collective behavior in biological systems. 200 p. Tese (Doutorado). University of Minnesota, Estados Unidos da América, 2005.

ESCHENAUER, H.; OLHOFF, N. (2001). Topology optimization of continuum structures: A review. *Applied Mechanics Review*. v.54, n.4, p.331-390.

FANG, Y.; LIU, Y.; LIU, J. (2007). A novel simplified foraging optimization algorithm for parameter identification of nonlinear system model. In: PROCEEDINGS OF THE IEEE INTERNATIONAL CONFERENCE ON AUTOMATION AND LOGISTICS. Jinan, China, 18-21 de agosto de 2007, p.798-802.

FONSECA, C.M.; FLEMING, P.J. (1993). Genetic Algorithms for multiobjective optimization: formulation, discussion and generalization. In: PROCEEDINGS OF THE FIFTH INTERNATIONAL CONFERENCE ON GENETIC ALGORITHMS. Forrest, S. (Ed.). University of Illinois at Urbana- Champaign. Morgan Kauffman Publishers, San Mateo, California, Estados Unidos da América, p.416-423.

FONSECA, C.M., FLEMING, P.J. (1998). Multiobjective optimization and multiple constraint handling with evolutionary algorithms—Part II: Application example. *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*. v.28, n.1, p.38-47.

GAZI, V.; PASSINO, K.M. (2002). Stability analysis of swarms in an environment with an attractant/repellent profile. In: PROCEEDINGS OF THE AMERICAN CONTROL CONFERENCE. Anchorage, Alaska, Estados Unidos da América, maio de 2002, p.1819-1824.

GOBBETTI, M. et al. (2007). Cell–cell communication in food related bacteria. *International Journal of Food Microbiology*. v.120, n.1-2, p.34-45.

GOLDBERG, D.E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, Massachusetts: Addison-Wesley.

GUNEY, K.; BASBUG, S. (2008). Interference suppression of linear antenna arrays by amplitude-only control using a bacterial foraging algorithm. *Progress in Electromagnetics Research*. v.79, p.475-497.

HAIBA, M. et al. (2005). Evolutionary structural optimisation of dynamically loaded components in consideration of fatigue life. *Advances in Engineering Software*. v.36, n.1, p.49-57.

HELLINGWERF, K.J. (2005). Bacterial observations: a rudimentary form of intelligence?. *TRENDS in Microbiology*. v.13, n.4, p.152-158.

HILPERT, M. (2005). Lattice-Boltzmann model for bacterial chemotaxis. *Mathematical Biology*. v.51, n.3, p.302-332.

HOLLAND, J. (1975). *Adaptation in Natural Artificial Systems*. Ann Arbor: University of Michigan Press.

KENNEDY, J.; EBERHART, R.C. (1995). Particle swarm optimization. In: PROCEEDINGS OF THE IEEE INTERNATIONAL CONFERENCE ON NEURAL NETWORKS IV. Piscataway: IEEE, p.1942–1948.

KICINGER, R.; ARCISZEWSKI, T.; DE JONG, K. (2005). Evolutionary computation and structural design: A survey of the state-of-the-art . *Computers and Structures*. v.83, n.23-24, p.1943-1978.

KIM, D.H.; CHO, J.H. (2006). A biologically inspired intelligent PID controller tuning for AVR systems. *International Journal of Control, Automation, and Systems*. v.4, n.5, p.624-636.

KIM, T.H.; JUNG, S.H.; CHO K.H. (2007). Investigations into the design principles in the chemotactic behavior of Escherichia Coli. *BioSystems*. v.91, n.1, p.171-182.

KIRCHNER, A.; SCHADSCHNEIDER, A. (2002). Simulation of evacuation processes using a bionics-inspired cellular automaton model for pedestrian dynamics. *Physica A: Statistical Mechanics and its Applications*. v.312, n.1-2, p.260-276.

KONGA, M.; TIANA, P.; KAO, Y. (2008). A new ant colony optimization algorithm for the multidimensional Knapsack problem. *Computers & Operations Research*, Oxford, v.35, n.8, p.2672-2683.

KUDO, S.; MAGARIYAMA, Y.; AIZAWA, S.I. (1990). Abrupt changes in flagellar rotation observed by laser dark-field microscopy. *Nature*. v.346, n.6285, p.677-680.

KUMAR, P.; BANDYOPADHYAY, S.; KUMAR, S. (2007). Multi-Objective Particle Swarm Optimization with time variant inertia and acceleration coefficients. *Information Sciences*. v.177, n.22, p.5033-5049.

KURSAWE, F. (1990). A variant of evolution strategies for vector optimization. In: PROCEEDINGS OF THE 1ST WORKSHOP ON PARALLEL PROBLEM SOLVING FROM NATURE. *Lecture Notes In Computer Science*. v.496, p.193-197.

LEE, U. et al. (2009). Bio-inspired multi-agent data harvesting in a proactive urban monitoring environment. *Ad Hoc Networks*. v.7, n.4, p.725-741.

LI, X. (2003). A non-dominated Sorting Particle Swarm Optimizer. In: PROCEEDING OF GENETIC AND EVOLUTIONARY COMPUTATION CONFERENCE 2003 (GECCO'03), *Lecture Notes in Computer Science*. Chicago, Estados Unidos da América, 12-16 de Julho 2003, p. 37-48.

LI, M.S. et al. (2007). Bacterial Foraging Algorithm with varying population for optimal power flow. In: PROCEEDINGS OF EVOWORKSHOPS. Valencia, Espanha, 11-13 de abril de 2007, p.267-261.

LIN, S.; LIU, Y.; FANG, Y. (2007). Weighted fuzzy rule-based system combined with a novel simplified E. Coli Foraging Optimization Algorithm. In: PROCEEDINGS OF 22ND IEEE INTERNATIONAL SYMPOSIUM ON INTELLIGENT CONTROL PART OF IEEE MULTI-CONFERENCE ON SYSTEMS AND CONTROL. Singapura, 1-3 de outubro de 2007, p.267-270.

LIU, W. et al. (2008). Simulation and Study of Self-adaptive Bacterial Colony Chemotaxis Algorithm. In: PROCEEDINGS OF THE 2008 FOURTH INTERNATIONAL CONFERENCE ON NATURAL COMPUTATION. v.7, p.678-682.

LIU, X. et al. (2008). Genetic evolutionary structural optimization. *Journal of Constructional Steel Research*. v.64, n.3, p.305-311.

LIU, Y.; PASSINO, K. (2002). Biomimicry of social foraging bacteria for distributed optimization: models, principles, and emergent behaviors. *Journal of optimization theory and applications*. v.115, n.3, p.603-628.

MARQUES, L.; NUNES, U.; DE ALMEIDA, A. (2002). Olfaction-based mobile robot navigation. *Thin Solid Films*. v.418, n.1, p.51-58.

MAURIELLO, E.; ZUSMAN, D. (2007). Polarity of motility systems in *Myxococcus xanthus*. *Current Opinion in Microbiology*. v.10, n.6, p.624-629.

MICHELL, A.G.M. (1904). The limits of economy of material in frame structures. *Philos Mag*. v.8, p.589-597.

MISHRA, S. (2005). A hybrid least square-fuzzy bacterial foraging strategy for harmonic estimation. *IEEE Transactions on Evolutionary Computation*. v.9, n.1, p.61-73.

MITCHELL, M. (1995). *An Introduction to Genetic Algorithms*. MA: MIT Press.

MITCHELL, J.G. (2002). The energetics and scaling of search strategies in bacteria. *The American Naturalist*. v.160, n.6, p.727-740.

MONTAGUE, G.A.; WARDB, A.C. (1994). A sub-optimal solution to the optimisation of bioreactors using the chemotaxis algorithm. *Process Biochemistry*. v.29, n.6, p.489-496.

MÜLLER, S.D. et al. (2002). Optimization based on bacterial chemotaxis. *IEEE Transactions on Evolutionary Computation*. v.6, n.1, p.16-29.

MURRAY, J. (2002). *Mathematical Biology: An Introduction*. Secaucus, NJ, USA: Springer-Verlag, New York.

NELSON, H.D.; MCVAUGH, J.M. (1976). The dynamics of rotor-bearing systems using finite elements. *Journal of Engineering for industry*. v.98, n.2, p.593-600.

NICOLAU JR. et al. (2008). 'Extremotaxis': Computing with a bacterial-inspired algorithm. *BioSystems*. v.94, p.47-54.

NOCEDAL, J.; WRIGTH, S. (1999). *Numerical Optimization*. Secaucus, NJ, USA: Springer-Verlag New York.

NORTON, R. (2004). *Design of machinery : an introduction to the synthesis and analysis of mechanisms and machines*. Estados Unidos da América: McGraw-Hill.

ONSUM, M.; ARKIN, A. (2005). Autonomous mobile robot control based on white blood cell chemotaxis. *Computational methods in systems biology*. v.3082, p.9-19.

PANIGRAHI, B.K.; RAVIKUMAR PANDI, V. (2008). Bacterial foraging optimisation: Nelder–Mead hybrid algorithm for economic load dispatch. *IET Generation, Transmission & Distribution*. v.2, n.4, p.556-565.

PARSEK, M.R.; GREENBERG, E.P. (2005). Sociomicrobiology: the connections between quorum sensing and biofilms. *Trends in Microbiology*. v.13, n.1, p.27-33.

PASSINO, K. (2002). Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Systems Magazine*. v.22, n.3, p.52-67.

PASSINO, K.M. (2005). *Biomimicry for optimization, control and Automation*. London: Springer-Verlag.

PEDAMALLU, C.S.; OZDAMAR, L. (2008). Investigating a hybrid simulated annealing and local search algorithm for constrained optimization. *European Journal of Operational Research*. v.185, n.3, p.1230-1245.

POLEZHAEV, A. et al. (2006). Spatial patterns formed by chemotactic bacteria Escherichia Coli. *International journal of developmental biology*. v.5, n.2-3, p.309-314.

POLI, R.; KENNEDY, J.; BLACKWELL, T. (2007). Particle swarm optimization. *Swarm Intelligence*. v.1, p.33-57.

POLONI, C. (1997). Hybrid GA for multiobjective aerodynamic shape optimization. *Genetic Algorithms in Engineering and Computer Science*. New York: John Wiley & Sons.

PRADYUMN, S.; DEB, K. (2007). On finding multiple Pareto-optimal solutions using classical and evolutionary generating methods. *European Journal of Operational Research*. v.181, n.3, p.1630-1652.

PRAGER, W.; ROZVANY, G.I.N. (1977). Optimization of structural geometry. In: BEDNAREK, A. R.; CESARI, L. (Eds.). *Dynamical systems*. New York: Academic Press.

PROENÇA, S. (2006). *Introdução aos métodos numéricos*. Notas de Aula. Universidade de São Paulo. Escola de Engenharia de São Carlos. Pós-graduação em Engenharia de Estruturas, São Carlos, Brasil, Março de 2006.

QUERIN, O.M. et al. (2000). Computational efficiency and validation of bi-directional evolutionary structural optimisation. *Computer Methods in Applied Mechanics and Engineering*. v.189, n.2, p.559-573.

RAPPE, M.S.; et al. (2002). Cultivation of the ubiquitous SAR11 marine bacterioplankton clade. *Nature*. v. 418, n. 6898, p. 630-633.

RIGANTI, F.; SALVINI, A. (2007). Comparative analysis between modern heuristics and hybrid algorithms. *COMPEL: The International Journal for Computation and Mathematics in Electrical and Electronic Engineering*. v.26, n.2, p.259-268.

ROZVANY, G.I.N. (2001). Aims, scope, methods, history and unified terminology of computer-aided topology optimization in structural mechanics. *Structural and Multidisciplinary Optimization*. v.21, n.2, p.90-108.

RUSSELL, A. et al. (2003). A comparison of reactive robot chemotaxis algorithms. *Robotics and Autonomous Systems*. v.45, n.2, p.83-97.

SANDGREN, E.; JENSEN, E.; WELTON, J. (1990). Topological design of structural components using genetic optimization methods. In: PROCEEDINGS OF THE WINTER ANNUAL MEETING OF THE AMERICAN SOCIETY OF MECHANICAL ENGINEERS, SENSITIVITY ANALYSIS AND OPTIMIZATION WITH NUMERICAL METHODS, AMD. v. 115. Dallas, Estados Unidos da América. p.31-43.

SARKER, R. (Ed.). (2002). *Evolutionary Optimization*. Secaucus, NJ, Estados Unidos da América: Kluwer Academic Publishers.

SCHAFFER, J.D. (1984). Some experiments in machine learning using vector evaluated genetic algorithm (artificial intelligence, optimization, adaptation, pattern recognition). 166p. Tese (Doutorado), Vanderbilt University, Nashville, Estados Unidos de América, 1984.

SCHAFFER, J.D. (1987). Multiple objective optimization with vector evaluated genetic algorithms. In: PROCEEDINGS OF THE FIRST INTERNATIONAL CONFERENCE ON GENETIC ALGORITHMS. J.J. Grefenstette (Ed). Hillsdale, NJ: Lawrence Erlbaum, p.93-100.

SEGALL, J.; BLOCK, S.; BERG, H. (1986). Temporal comparisons in bacterial chemotaxis. *Proceedings of the National Academy of Sciences of the United States of America*. v.83, n.23, p.8987-8991.

SHEA, C.; NUNLEY, J.W.; SMITH-SOMERVILLE, H.E. (1991). Variable expression of gliding and swimming motility in *Deleya marina*. *Canadian Journal in Microbiology*. v.37, n.11, p.808- 814.

SHIM, P.Y.; MANOOCHMURI, S. (1997). Generating optimal configurations in structural design using simulated annealing. *International Journal for Numerical Methods in Engineering*. v.40, n.6, p.1053-1069.

SIGMUND, O. (2001). A 99 line topology optimization code written in Matlab. *Structural Multidisciplinary Optimization*. v.21, p.120-127.

SOURJIK, V. (2004). Receptor clustering and signal processing in *E. Coli* chemotaxis. *Trends in Microbiology*. v.12, n.12, p.569-576.

SPIRO, P.A.; PARKINSON, J.S.; OTHMER, H.G. (1997). A model of excitation and adaptation in bacterial chemotaxis. *Proceedings of the National Academy of Sciences of the United States of America*. v.94, n.14, p.7263-7268.

TANG, W.J.; WU, Q.H. (2006). Bacterial Foraging Algorithm for dynamic environments. In: PROCEEDINGS OF THE 2006 IEEE CONGRESS ON EVOLUTIONARY COMPUTATION (CEC 2006). Vancouver, Canada, 16-21 de julho de 2006, p.1324-1330.

TANSKANEN, P. (2002). The evolutionary structural optimization method: theoretical aspects. *Computer Methods in Applied Mechanics and Engineering*. v.191, n.47-48, p.5485-5498.

THAR, R.; KÜHL, M. (2005). Complex pattern formation of marine gradient bacteria explained by a simple computer model. *FEMS Microbiology Letters*. v.246, n.1, p.75-79.

TOVAR, A. (2005). Optimización topológica con la técnica de los autómatas celulares híbridos. *Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería*. v.21, n.4, p.365-383.

VICTORIA, M. (2006). Optimización de forma y topología con malla fija y algoritmos genéticos. 221 p. Tese (Doutoral). Universidad Politécnica de Cartagena. Cartagena, Espanha, 2006.

WU, C. et al. (2007). Improved Bacterial Foraging Algorithms and their applications to job shop scheduling problems. *LNCS*. v.4431, p.562-569.

XIE, Y.; STEVEN, A. (1993). Simple evolutionary procedure for structural optimization. *Computers & Structures*. v.49, n.5, p.885-896.

YAGI, H.; MARUYAMA, A. (1998). Novel diglycosyldiacylglycerol from the Gram-negative bacterium *Deleya marina*. *Biochimica et Biophysica Acta (BBA) – Lipids and Lipid Metabolism*. v.1393, n.1, p.161-165.

YANG, X.Y.; XIE, Y.M.; STEVEN, G.P. (2005). Evolutionary methods for topology optimisation of continuous structures with design dependent loads. *Computers and Structures*. v.83, n.12, p.956-963.

YOUNG, K. (2006). The selective value of bacterial shape. *Microbiology and molecular biology reviews*. v.70, n.3, p.660-703.

ZITZLER, E.; LAUMMANS, M.; THIELE, L. (2002). SPEA2: Improving the Strength Pareto Evolutionary Algorithm. In: PROCEEDINGS OF THE EUROGEN2001 CONFERENCE. K. Giannakoglou, D. Tsahalis, J. Periaux, P. Papailou, and T. Fogarty, (Eds.), Atenas, Greça, Setembro de 2001.

ZITZLER, E.; DEB, K.; THIELE, L. (2000). Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*. v.8, n.2, p.173-195.

**APÊNDICE A - Código fonte em Matlab dos
algoritmos BCBTOA e BCMOA**

Essa é a implementação do algoritmo BCBTOA em Matlab.

```
%=====
%      BCBTOA (Bacterial Chemotaxis Based Topology Optimization Algorithm)
%      Versão de junho de 2009 - María Alejandra Guzmán Pardo
%=====

clear all
clc
nelx=60; % numero de elementos em x
nely=20; % numero de elementos em y
Nr=20; % numero total de iterações
R=4; % raio de incidência do sinal
Fator=0,011; % fator da eq.(30)
x(1:nely,1:nelx)=1; % valor inicial da densidade de cada elemento
Nelem=nelx*nely; % numero total de elementos
Volumen=Nelem; % volume inicial
Nbac=0.5*Nelem; % numero inicial de bactérias
qaa(:,,:)=zeros(nely,nelx,Nr); % matriz dos valores dos sinais
P(:,,:)=zeros(nely,nelx,Nr); % matriz de localização das bactérias
P1(:,,:)=zeros(nely,nelx,Nr); % matriz de bactérias movimentadas
P2(:,,:)=zeros(nely,nelx,Nr); % matriz de bactérias aderidas

%=====
%                               INICIO
%=====
Nge=1; % número de iteração

for Nge=1:Nr
    [U]=FECHEMO(nelx,nely,x); % cálculo da função objetivo por EF
    [KE] = lkCHEMO;
    c=zeros(nely,nelx); % mean compliance
    for ely = 1:nely
        for elx = 1:nelx
            n1 = (nely+1)*(elx-1)+ely;
            n2 = (nely+1)* elx +ely;
            Ue = U([2*n1-1; 2*n1; 2*n2-1;2*n2; 2*n2+1;2*n2+2;...
                2*n1+1 ;2*n1+2],1);
            c(ely,elx) = x(ely,elx)*Ue*KE*Ue; % mean compliance para
                                                % cada elemento
        end
    end
    ctot(1,Nge)=sum(sum(c)); % mean compliance para toda a estrutura
    cmax=max(max(c)); % valor máximo de c
    cmin=min(min(c));
    [K,L] = ind2sub(size(c),find(c~=NaN)); % subíndices de localizações(elementos)
    indices=[K';L'];

%=====
%      CALCULO DO SINAL RECEBIDO POR CADA BACTÉRIA
%=====
    for i=1:Nelem
        qq=0; qa=0;
        for j=1:Nelem
            r=sqrt((indices(1,i)-indices(1,j))^2+(indices(2,i)-indices(2,j))^2);
            if r==0 % se a distancia é zero
                qa=1; %a função vale 1
            elseif r>R %distancias maiores do que R
                qa=0; % a função vale zero
            elseif r<=R
                % a função vale zero
            end
        end
    end
end
```


Essa é a implementação do algoritmo BCMOA em Matlab.

```
%=====
%      BCMOA (Bacterial Chemotaxis Multiobjective Optimization Algorithm)
%      Versão de junho de 2009 - María Alejandra Guzmán Pardo
%=====
clear all
clc
p=30;                % número de variáveis
M=2;                 % número de funções objetivo
S=100;               % numero de bactérias da população
Nveces=250;          % numero de iterações
contador=1;          % contador de iterações
Bac=[];
Bac(:,:)=zeros(p,S); % matriz de localização das bactérias
J=zeros(M,S);        % matriz dos valores das funções objetivo
Bac(:,:)=rand(p,S);  % valores iniciais aleatórios
Bac(p+1,:)=1:1:S;    % numeração das bactérias
Bac1(:,:)=zeros(p+1,S); % matriz de novas localizações
TamPOF1=0;           % Tamanho da FOP1

while contador<Nveces
%=====
%      AVALIAR FUNÇÕES OBJETIVO
%=====
    for i=1:S
        J(1,i)=obj1(Bac(:,i));
        J(2,i)=obj2(Bac(:,i));
    end

    Fac=(Nveces-contador)/(Nveces) % fator Fac da eq.(33)

%=====
%      CLASIFICACÃO EM FOP POR NÃO DOMINÂNCIA
%=====

    [POF]=Sorting(J,S,Bac,p);

%=====
%      ESTRATEGIA QUIMIOTACTICA PARA BACTÉRIAS FORTES
%=====
    NumFren=0;
    for i=1:S
        if isempty(POF{i})==0
            NumFren=NumFren+1; % quantidade de FOP resultantes
        end
    end

    if contador==1
        Bacant(:,:)=Bac(:,:);
        Jant(:,:)=J(:,:);
    end

    POF1=size(POF{1});
    TamPOF1=POF1(1,2); % tamanho da FOP(1)

    for i=1:TamPOF1
        Jcomp(1,1)=obj1(POF{1}(1:p,i));
        Jcomp(2,1)=obj2(POF{1}(1:p,i),p);
        Jcomp(1,2)=obj1(Bacant(1:p,POF{1}(p+1,i)));
    end
end
```

```

Jcomp(2,2)=obj2(Bacant(1:p,POF{1}(p+1,i)),p);
Baccomp=[POF{1}(1:p,i) Bacant(1:p,POF{1}(p+1,i));1 2];
[POFcomp]=Sorting(Jcomp,2,Baccomp,p); %comparação por não
                                         %dominância de localização
                                         %atual e anterior

POFcomp=size(POFcomp{1});
TamPOFcomp=POFcomp(1,2);

Delta(1:p,POF{1}(p+1,i))= (2*(rand(p,1))-1);
Deltatau(:,POF{1}(p+1,i))=Delta(:,POF{1}(p+1,i))/... % vetor unitário em direção
sqrt(Delta(:,POF{1}(p+1,i))*Delta(:,POF{1}(p+1,i))); % aleatória

if TamPOFcomp==1 % uma localização domina à outra
    for zz=1:p
        Bac1(zz,POF{1}(p+1,i))=POFcomp{1}(zz,1)+...
            (Deltatau(zz,POF{1}(p+1,i))*1e-3*(max(POF{1}(zz,:))-...
            min(POF{1}(zz,:))*Fac); %eq.(37) e eq.(38)
    end
else %as duas são não dominadas
    for zz=1:p
        Bac1(zz,POF{1}(p+1,i))=POF{1}(zz,i)+...
            (Deltatau(zz,POF{1}(p+1,i))*1e-2*(max(POF{1}(zz,:))-...
            min(POF{1}(zz,:))*Fac); %eq.(39)
    end
end
Bac1(p+1,POF{1}(p+1,i))=POF{1}(p+1,i);
end

%=====
%   ESTRATEGIA QUIMIOTACTICA PARA BACTERIAS FRACAS
%=====

Bacdom=[];
JJ=[];
POF1ord=[];
for i=2:NumFren
    Bacdom=[Bacdom POF{i}(:,)]; %
end

for jj=1:TamPOF1
    JJ(:,jj)=J(:,POF{1}(p+1,jj));
end

[OBJcomp]=densidad(POF{1}(:,),JJ,TamPOF1,p,S); %cálculo do fator deni (KUMAR, 2007)
POF1ord=(sortrows(OBJcomp(:,)',p+4));

% aplicação da estratégia de estímulo à diversidade
for ii=1:S-TamPOF1
    if TamPOF1>=0.5*S
        mama=round(0.8*TamPOF1+((TamPOF1-0.8*TamPOF1)*rand));
    else
        mama=round(1+(TamPOF1-1)*rand);
    end
    for zz=1:p %cálculo de SW/Fac
        Delta(zz,Bacdom(p+1,ii))=POF1ord(zz,mama)-Bacdom(zz,ii);
    end
    for zz=1:p
        Bac1(zz,Bacdom(p+1,ii))=POF1ord(zz,mama)+POF1ord(zz,mama)*...
            (-0.01+0.02*rand)+(Delta(zz,Bacdom(p+1,ii))*Fac); % eq.(39)
    end
    Bac1(p+1,Bacdom(p+1,ii))=Bacdom(p+1,ii); %conservação do indice
end

```

```

for i=1:S
    for zz=1:p
        if Bac1(zz,i)<0 ;
            Bac1(zz,i)=0;
        elseif Bac1(zz,i)>1 ; % aplicação de parede absorvente
            Bac1(zz,i)=1;
        end
    end
end

Bacant(:,:)=Bac(:,:);
POFant{ 1 }=[];
POFant{ 1 }=POF{ 1 };
Bac(:,:)=Bac1(:,:);

contador=contador+1;

end % finalização do algoritmo

```

**APÊNDICE B- Resumos dos artigos apresentados em
congressos internacionais**

Resumo do artigo que foi apresentado no 8 Congresso Ibero-americano de Engenharia Mecânica. Cusco, Peru, Outubro 23-25 de 2007.

OPTIMIZACIÓN MULTIOBJETIVO DE UN EJE CON ALGORITMO BASADO EN QUIMIOTAXIS DE BACTERIAS

Guzmán, M.A., Delgado, A., De Carvalho, J.

RESUMEN

La quimiotaxis es el movimiento de algunos organismos vivos en respuesta a gradientes químicos presentes en su ambiente. Una bacteria es un organismo unicelular que utiliza la quimiotaxis como mecanismo de movilización para encontrar los nutrientes de que precisa para sobrevivir así como para huir de los ambientes nocivos. Gracias a los avances computacionales de la actualidad, la excelente capacidad de búsqueda de este proceso biológico puede ser modelada, simulada y aprovechada para desarrollar métodos de optimización multiobjetivo alternativos a los métodos bio-inspirados ya existentes.

En este trabajo se desarrolla un algoritmo de optimización basado en el comportamiento quimiotáctico de bacterias para encontrar soluciones óptimas a problemas multiobjetivo con y sin restricciones. El desempeño del algoritmo desarrollado es evaluado mediante su aplicación en la solución de tres problemas de optimización multiobjetivo de prueba y finalmente es aplicado en la optimización dimensional de un eje de máquina herramienta, minimizando simultáneamente su peso y su deflexión lateral máxima. Los resultados de esta aplicación son comparados con los obtenidos para el mismo problema usando el conocido algoritmo NSGA (Non-dominated Sorting Genetic Algorithm).

PALAVRAS CLAVE: Optimización multiobjetivo, quimiotaxis, ejes

Resumo do artigo que foi apresentado na International Conference on Engineering Optimization EngOpt 2008. Ro de Janeiro, Brasil, junho 1 a 5 de 2008.

A Bacterial Chemotaxis Multiobjective Optimization Algorithm

Guzmn, M.A., Delgado, A., De Carvalho, J.

ABSTRACT

This paper presents the new Bacterial Chemotaxis Multiobjective Optimization Algorithm (BCMOA) based on bacterial chemotaxis. It incorporates chemotactical responses of individuals as a viable bio-inspired optimization technique. As other multiobjective optimization algorithms, the BCMOA works with populations, in this case bacterial colonies, which are located into the search space. The locations represent the value of the objective functions for possible solutions to the optimization problem, thus the well known non-dominated sorting concept is incorporated in order to classify the location of each bacterium. The bacterial chemotactical behavior is exploited by using the ability to sense the environmental conditions and make temporal-space comparisons to the actual and previous locations, always with the goal of finding nutrient sources (optimal locations) and escape from noxious environments (dominated locations). The self organized collective behavior showed for bacterial colonies through communications and signaling between bacteria are also considered. The bacterium motile behavior is emulated by application of simple responses to the signal sense and comparisons in a way that each bacterium develops one of two possible chemotactical actions, swimming or tumbling. The proposed algorithm is validated using test functions as SCH, FON, POL, KUR, from ZDT1 to ZDT4, and different performance measures were also implemented.

Keywords: multiobjective optimization, bacterial chemotaxis, bioinspired methods.

Resumo do artigo que vai ser apresentado no 8 World Congress on Computational Mechanics que se realizara na cidade de Veneza entre os dias 29 de junho e 4 de julho de 2008.

TOPOLOGY OPTIMIZATION WITH ALGORITHM BASED ON BACTERIAL CHEMOTAXIS

***Guzmán, M.A.¹, Delgado, A.² De Carvalho, J.³**

Key Words: Topology optimization, bacterial chemotaxis, bio-inspired methods.

ABSTRACT

This paper presents a new topology optimization methodology that combines the FE method for structural analysis with an optimization strategy inspired by bacterial chemotaxis.

The principle of BCBTOA (Bacterial Chemotaxis Based Topology Optimization Algorithm) is very simple. A structure evolves to an optimum configuration by systematically redistributing the material into the design domain, reinforcing the overload zones and removing material where it is not required. Inspired by the model of self-organized collective behaviour showed by marine bacteria *Thiovulum majus* [1], the algorithm establishes rules that drive the structure to an optimum distribution of material.

The design domain is constructed by the FE method and represents the environment in which a colony of bacteria can move. Support conditions and external loads are also applied to this model. As happens in nature, in any place of the environment there are different concentrations of nutrients or noxious substances. Here, these concentrations are represented by the values of the field variables for each element (mean compliance).

An amount of bacteria equal to the desired volume is placed on the design domain. Immersed there, the bacteria develop a chemotactical strategy as a whole considering communications between them, to increase the chances of survival a maximal number of individuals from the colony. In each iteration, bacteria exchanged information regarding the concentration of

nutrients at its current location. Based on this information, bacteria located in positions with low concentrations of nutrients move toward more favorable positions. As ESO (Structural Evolutionary Optimization) [2], in this method the design variable is the density trackled directly, so the elements can be turned off or on, depending on the presence or absence of bacteria in them.

The procedure of BCBTOA is:

For the giving boundary and load conditions, define the design domain.

Perform FE analysis to evaluated the field variables.

Locate the bacteria in the best positions.

Calculate the signal between bacteria and exchange information.

Remove the bacteria in poorest positions.

Locate bacteria in the richest empty positions.

Update the density for each element.

Return to step 2 until stop criterion be satisfied.

The BCBTOA was implemented in Matlab 7.0 ® making the necessary modifications to the code developed in [3], and has been successfully applied to solve topology optimization problems in a variety of two dimentional structures in a continuum design domain as show in Figure 1 and Figure 2.

Figure 1. Topology optimization of a cantilever beam.

Figure 2. Topology optimization of a Michell-type structure.

REFERENCES

- R. Thar and M. Klh, "Complex pattern formation of marine gradient bacteria explained by a simple computer model with one degree of freedom per node", FEMS Microbiology Letters 246, pp.75-79, (2005).
- R. Das, R. Jones and Y.M. Xie. "Design of Structures for optimal static strength using ESO", Engineering Failure Analysis 12, pp. 61-80, (2005).
- O. Sogmund, "A 99 lines topology optimization code written in Matlab", Struc. Multidisc. Optim. 21, pp. 120–127, (2001).