

Investigação de Novas Abordagens em Sistemas Imunes Artificiais para Otimização

Lucas de Souza Batista

Dissertação submetida à
Escola de Engenharia da
Universidade Federal de Minas Gerais
para o grau de Mestre em Engenharia Elétrica

“Obras-primas não são fruto de um nascimento solitário. Elas são a consequência de vários anos de pensamento em comum, de tal modo que a experiência da massa está por trás de uma única voz.”

— Virgínia Woolf, 1882–1941 (escritora britânica)

Investigação de Novas Abordagens em Sistemas Imunes Artificiais para Otimização

Resumo

O custo computacional do processo de otimização de dispositivos eletromagnéticos está diretamente relacionado ao número de avaliações da função objetivo. Isso tem motivado o estudo de novos métodos que sejam capazes de determinar resultados eficientes com o menor número de avaliações possíveis. Esta dissertação propõe dois algoritmos imunes para otimização mono e multi-objetivo. A versão mono-objetivo, nomeada “Distributed Clonal Selection Algorithm - DCSA”, implementa um operador principal chamado *hipermutação somática distribuída*, enquanto a versão multi-objetivo, nomeada “Multi-Objective Clonal Selection Algorithm - MCSA”, além do operador anterior, implementa também um operador de *edição de receptores*. A *hipermutação somática*, composta por diferentes funções densidade de probabilidade, normal, uniforme e caótica, efetua uma busca local balanceada ao redor das soluções de maior afinidade, além de favorecer a melhor distribuição das soluções ao longo da extensão da fronteira Pareto-ótima no MCSA. Já a *edição de receptores*, implementada com base na evolução diferencial, efetua implicitamente uma pesquisa dinâmica sobre a região factível, garantindo um melhor refinamento local das soluções ótimas, e favorecendo o aumento da velocidade de convergência do método. Os parâmetros dos algoritmos de otimização são submetidos a análises de sensibilidade, o que permite determinar faixas aceitáveis aos mesmos. Além disso, os operadores imunes sugeridos são avaliados quanto ao ganho que cada um proporciona ao desempenho dos métodos. Os algoritmos imunes propostos são validados por meio da solução de problemas analíticos com diferentes características de otimização, tais como, alta suavidade, multimodalidade, múltiplas variáveis e restrições, apresentando soluções eficientes quando comparados a outros métodos evolucionários conhecidos. Finalmente são realizados testes com problemas eletromagnéticos de alto custo computacional associado, resultando mais uma vez soluções de boa qualidade, e também um menor esforço de máquina, em relação ao número de avaliações realizadas, quando comparados a outros algoritmos da literatura.

Investigation of New Approaches in Artificial Immune Systems for Optimization

Abstract

The computational cost of the optimization process of electromagnetic devices is directly related to the number of objective function evaluations. This has motivated the study of new methods that are capable of determining efficient results with a fewer number of function evaluations. This dissertation proposes two new immune algorithms for mono and multi-objective optimization. The mono-objective version, named “Distributed Clonal Selection Algorithm - DCSA”, implements a main operator called *distributed somatic hypermutation*, while the multi-objective version, named “Multi-Objective Clonal Selection Algorithm - MCSA”, implements in addition a *receptor editing* operator. The *somatic hypermutation*, composed of different probability density functions, Gaussian, uniform and chaotic, performs a balancing local search around the high affinity solutions, and also facilitates the best distribution of the solutions throughout the extension of the Pareto-optimal front in the MCSA. The *receptor editing* operator, based on the differential evolution technique, implicitly performs a dynamic search over the feasible region, ensuring the best local refinement of the optimal solutions, and helping the increase of the convergence speed of the method. The optimization parameters of the algorithms have been subjected to sensitivity analysis, which has provided a range of acceptable values for them. Furthermore, the suggested immune operators have been assessed in order to determine the effect of each one in the performance of the methods. The proposed immune algorithms have been validated through the solution of analytical problems with different optimization features, such as, strong smoothness, multimodality, high dimensions and constraints, presenting efficient solutions when compared to other known evolutionary methods. Finally, tests with electromagnetic problems of high computational cost have been performed, resulting in very good solutions with less machine effort, regarding the number of function evaluations.

Agradecimentos

Em primeiro lugar agradeço a Deus, quem guiou os meus passos desde a minha infância, quando ainda nem pensava em me formar em engenharia elétrica, e muito menos, em defender um título de mestre numa universidade tão conceituada quanto a UFMG. Agradeço sinceramente pelas inúmeras portas que foram abertas, e também pelas oportunidades que certamente virão.

Agradeço ao meu pai João Batista, quem sempre me proporcionou muitas alegrias, além de me mostrar as grandes virtudes da paciência e mansidão, e à minha mãe Maria Clarisberte, quem há muito vem me ensinando a lutar pelos sonhos tão almejados. Estas duas vidas são os principais responsáveis pelo meu caráter, e sei que mesmo diante de alguns desentendimentos, eles sempre torceram por mim.

Sou grato também aos meus familiares pelo apoio e carinho, principalmente à minha tia Airam e às minhas irmãs Paula e Arielly, as quais sempre me ajudaram e animaram. Agradeço especialmente aos meus tios Jaci e Solange, os quais nunca me desampararam e sempre se mostraram grandes amigos.

Agradeço ao meu orientador Jaime A. Ramírez, quem direcionou meus primeiros passos no campo da otimização evolucionária. Além de ter se mostrado um amigo, representa um dos maiores responsáveis pela concretização deste trabalho. Sou grato ainda aos grandes professores que ajudaram na minha formação, principalmente Oriane Magela, Rodney Saldanha, Walmir Caminhas, João Vasconcelos, Maria Helena, José Osvaldo e Antônio Emílio.

Agradeço também ao meu colaborador e amigo Frederico Guimarães, quem idealizou parte fundamental desta dissertação, além de ter participado da produção de importantes artigos. Sou grato ainda aos amigos do LEAT, do GOPAC, e também às discussões sempre produtivas realizadas às sextas-feiras sob a direção do professor Ricardo Takahashi, ao qual sou também agradecido.

Finalmente, agradeço à minha sempre namorada e amiga Érica Lombardi, a pessoa mais importante da minha vida, quem nunca deixou de mostrar seu amor, carinho e seriedade, sendo sempre companheira e muito paciente, principalmente por ter tolerado as inúmeras vezes que estive ausente. Não poderia deixar de mencionar também o apoio e ajuda dos seus pais Jésus Diniz e Fátima Lombardi, os quais sempre me abençoaram.

Ao CNPq e Capes pelo apoio financeiro, sem o qual esse trabalho não teria sido possível.

Sumário

Lista de Figuras	ix
Lista de Tabelas	xii
Lista de Algoritmos	1
1 Introdução	2
1.1 Apresentação Inicial	2
1.2 Contexto Histórico	3
1.2.1 Problema de Otimização Genérico	4
1.2.2 Algoritmo Evolucionário Unificado	4
1.2.3 Algumas Aplicações	5
1.3 Contribuições da Dissertação	6
1.4 Estrutura do Trabalho	7
2 Sistema Imune Biológico e Artificial	9
2.1 Introdução	9
2.2 Sistema Imune Natural - Uma Visão Geral	10
2.2.1 Linhas de Defesa do Sistema Imune	11
2.2.2 Princípio da Seleção Clonal	13
2.3 Sistemas Imunes Artificiais	15
2.3.1 Conceitos Básicos	15
2.3.2 Evolução Paralela e Otimização Multimodal	17
2.4 Conclusão	20
3 Otimização Evolucionária Mono-Objetivo	21
3.1 Introdução	21
3.2 Definição do Problema de Otimização	22
3.3 Condições de Otimalidade	23
3.4 Tratamento de Restrições em Algoritmos Evolucionários	23

3.5	Estrutura Geral de um AE Mono-Objetivo	26
3.6	Sistemas Evolucionários Mono-Objetivo	27
3.6.1	“Clonal Algorithm” - CLONALG	28
3.6.2	“B-Cell Algorithm” - BCA	31
3.6.3	“Real-Coded Clonal Selection Algorithm” - RCSA	33
3.6.4	“Real-Biased Genetic Algorithm” - RBGA	35
3.6.5	“Differential Evolution Algorithm” - DEA	38
3.6.6	Outras Vertentes	41
3.7	Conclusão	41
4	Otimização Evolucionária Multi-Objetivo	43
4.1	Introdução	43
4.2	Definição do Problema de Otimização	44
4.3	Condições de Otimalidade	46
4.4	Estrutura Geral de um AE Multi-Objetivo	46
4.5	Sistemas Evolucionários Multi-Objetivo	49
4.5.1	“Non-Dominated Sorting Genetic Algorithm” - NSGA-II	49
4.5.2	“Strength Pareto Evolutionary Algorithm” - SPEA-II	52
4.5.3	“Pareto Envelope-based Selection Algorithm” - PESA	54
4.5.4	“Multi-Objective Clonal Selection Algorithm” - MOCSA	56
4.5.5	“Multi-Objective Differential Evolution” - MODE	57
4.5.6	Outras Vertentes	60
4.6	Conclusão	61
5	Sistemas Imunes Artificiais: Novas Abordagens	63
5.1	Introdução	63
5.2	“Distributed Clonal Selection Algorithm”	64
5.2.1	Teoria do Caos em Otimização	64
5.2.2	Descrição do algoritmo DCSA	68
5.2.3	Análise de Sensibilidade para Calibração dos Parâmetros	74
5.3	“Multi-Objective Clonal Selection Algorithm”	88
5.3.1	Descrição do algoritmo MCSA	88
5.3.2	Calibração dos Parâmetros do MCSA	95
5.4	Conclusão	96
6	Resultados	97
6.1	Introdução	97

6.2	Resultados referentes ao DCSA	98
6.2.1	Problemas Analíticos	98
6.2.2	Problemas Eletromagnéticos	101
6.2.3	Discussão	113
6.3	Resultados referentes ao MCSA	114
6.3.1	Técnicas de Avaliação de Desempenho	114
6.3.2	Problemas Analíticos	118
6.3.3	Problema Eletromagnético	128
6.3.4	Discussão	135
6.4	Conclusão	135
7	Conclusões	137
	Conclusões	137
	Propostas de Continuidade	139
A	Literatura Especializada	143
	Referências	180

Lista de Figuras

2.1	Teoria da Seleção Clonal	14
2.2	Diagrama de blocos do ciclo de funcionamento de um algoritmo imune . .	18
2.3	Exemplos de otimização de funções multimodais	19
3.1	Ilustração das condições de Kuhn-Tucker para o caso mono-objetivo . . .	24
3.2	Taxa de mutação no CLONALG em função da afinidade normalizada . .	29
3.3	Distribuição da população no CLONALG	30
3.4	Operador “contiguous somatic hypermutation”	32
3.5	Distribuição da população no RCSA	34
3.6	Operação de cruzamento real-polarizado	37
3.7	Sistema de variação diferencial implementado no DEA	40
4.1	Ilustração das condições de Kuhn-Tucker para o caso multi-objetivo . . .	47
4.2	Ilustração do conjunto Pareto-ótimo em um problema bi-objetivo	48
4.3	Estratégia de avaliação de densidade empregado no PESA	55
4.4	Estratégia de variação empregada pelo MODE	59
5.1	Mapeamento gerado por um mapa logístico	67
5.2	Mapeamento gerado por um neurônio caótico	69
5.3	Distribuição da população no DCSA	70
5.4	Sistema de mutação adotado pelo DCSA	71

5.5	Sensibilidade referente ao parâmetro N_{pop} - $\mathcal{P}1$ e $\mathcal{P}2$	78
5.6	Sensibilidade referente ao parâmetro N_{pop} - $\mathcal{P}3$ e $\mathcal{P}4$	79
5.7	Sensibilidade referente aos parâmetros N_{dN} , N_{dU} e N_{dC} - $\mathcal{P}1$ e $\mathcal{P}2$	81
5.8	Sensibilidade referente aos parâmetros N_{dN} , N_{dU} e N_{dC} - $\mathcal{P}3$ e $\mathcal{P}4$	82
5.9	Sensibilidade referente aos parâmetros γ_N , γ_U e γ_C - $\mathcal{P}1$ e $\mathcal{P}2$	83
5.10	Sensibilidade referente aos parâmetros γ_N , γ_U e γ_C - $\mathcal{P}3$ e $\mathcal{P}4$	84
5.11	Sensibilidade referente ao parâmetro β - $\mathcal{P}1$ e $\mathcal{P}2$	85
5.12	Sensibilidade referente ao parâmetro β - $\mathcal{P}3$ e $\mathcal{P}4$	86
5.13	Número de clones gerados no MCSA em função de $\bar{\xi}$	90
5.14	Efeito da aplicação do operador de mutação diferencial	93
5.15	Diagrama de blocos da evolução da população no MCSA	95
6.1	Velocidade de convergência observada nos problemas analíticos	100
6.2	Configuração do transformador de impedância de guia de ondas (WIT)	102
6.3	Velocidade de convergência observada no problema de microondas (WIT)	104
6.4	“Superconducting Magnetic Energy Storage” (SMES)	106
6.5	Configuração otimizada obtida para o SMES 3D mono-objetivo	109
6.6	Configuração otimizada obtida para o SMES 8D mono-objetivo	112
6.7	“Nondominated Combined Set Ratio” (NDCSR)	115
6.8	“Hierarchical Cluster Counting” (HCC)	116
6.9	“S-Metric” ou “Hypervolume”	117
6.10	Diagrama de Sheldon (“box plot”)	120
6.11	Análise de desempenho frente aos probs. bi-objetivo - parte 1	122
6.12	Análise de desempenho frente aos probs. bi-objetivo - parte 2	123
6.13	Efeito do operador <i>hipermutação somática</i> no MCSA - parte 1	124

6.14	Efeito do operador <i>hipermutação somática</i> no MCSA - parte 2	125
6.15	Efeito do operador <i>edição de receptores</i> no MCSA - parte 1	127
6.16	Efeito do operador <i>edição de receptores</i> no MCSA - parte 2	128
6.17	Análise de desempenho obtida a partir do prob. bi-obj. SMES 3D	130
6.18	Análise de desempenho obtida a partir do prob. bi-obj. SMES 8D	130
6.19	Fronteira Pareto-ótimo obtida para o SMES multi-objetivo	131
6.20	Configuração otimizada obtida para o SMES 3D multi-objetivo	133
6.21	Configuração otimizada obtida para o SMES 8D multi-objetivo	134

Lista de Tabelas

5.1	Faixa de valores para a análise de sensibilidade dos parâmetros do DCSA	77
6.1	Parâmetros de otimização usados na solução dos probs. analíticos	98
6.2	Parâmetros de otimização usados na solução dos probs. eletromagnéticos	101
6.3	Resultados obtidos para o dispositivo de microondas (WIT) - DCSA . . .	103
6.4	Resultados obtidos para o dispositivo de microondas (WIT) - RCSA . . .	103
6.5	Resultados obtidos para o dispositivo de microondas (WIT) - BCA . . .	103
6.6	Variáveis de otimização para o problema SMES 3D	107
6.7	Soluções ótimas encontradas para o problema mono-objetivo SMES 3D .	108
6.8	Variáveis de otimização para o problema SMES 8D	110
6.9	Soluções encontradas para o problema mono-objetivo SMES 8D	111
6.10	Formulação dos problemas analíticos multi-objetivo	119
6.11	Parâmetros para solução dos probs. analíticos - MCSA e NSGA-II	120
6.12	Comparação com os melhores resultados multi-objetivo SMES	132

Lista de Algoritmos

1.1	Algoritmo evolucionário unificado	5
2.1	Ciclo básico de funcionamento de um algoritmo imunológico	17
3.1	Ciclo básico de funcionamento de um AE mono-objetivo	27
3.2	“Clonal Algorithm” (CLONALG)	30
3.3	“B-Cell Algorithm” (BCA)	32
3.4	“Real-Coded Clonal Selection Algorithm” (RCSA)	35
3.5	“Real-Biased Genetic Algorithm” RBGA	38
3.6	“Differential Evolution Algorithm” (DEA)	40
4.1	Ciclo básico de funcionamento de um AE multi-objetivo	48
4.2	“Non-Dominated Sorting Genetic Algorithm” (NSGA-II)	52
4.3	“Strength Pareto Evolutionary Algorithm” (SPEA-II)	54
4.4	“Pareto Envelope-based Selection Algorithm” (PESA)	56
4.5	“Multi-Objective Clonal Selection Algorithm” (MOCSA)	58
4.6	“Multi-Objective Differential Evolution” (MODE)	60
5.1	Geração de uma perturbação caótica	72
5.2	“Distributed Clonal Selection Algorithm” (DCSA)	73
5.3	“Multi-Objective Clonal Selection Algorithm” (MCSA)	94

Capítulo 1

Introdução

*“Tudo o que um homem pode imaginar,
outros homens poderão realizar.”*

— Júlio Verne, 1828–1905 (escritor francês)

“A imaginação é mais importante que o conhecimento.”

— Albert Einstein, 1879–1955 (físico alemão)

1.1 Apresentação Inicial

Computação Natural (do inglês “Natural Computing”) refere-se ao campo de pesquisa que trabalha com técnicas computacionais inspiradas em parte pela natureza e por sistemas naturais.

A partir da observação destes sistemas torna-se possível realizar uma “modelagem” computacional dos conceitos, princípios e mecanismos naturais, sob o ponto de vista em engenharia, com o propósito principal de desenvolver ferramentas eficientes, em “software” ou “hardware”, para a solução de problemas reais de otimização, aprendizagem, classificação de padrões e outros.

Essa área de pesquisa tem frequentemente conduzido vários autores à síntese de modelos e comportamentos artificiais, resultando assim nos chamados *sistemas biologicamente inspirados*. Alguns exemplos oriundos da computação inspirada na natureza são redes

neurais artificiais, algoritmos evolucionários, algoritmos baseados em colônias, e mais recentemente, sistemas imunológicos artificiais.

Na maior parte dos casos, independente da motivação que conduz ao estudo das inúmeras vertentes mencionadas anteriormente, a ideia principal ao se desenvolver uma nova ferramenta computacional é que essa consuma o menor tempo e espaço computacionais possível ao resolver um problema real de engenharia. De forma geral, uma vez que o acesso aos dados de um programa de avaliação apresenta um custo computacional associado, quanto menor o número de acessos a esse programa para a solução do problema, melhor será o desempenho do algoritmo de otimização empregado.

Com o intuito de prover melhorias ao ramo da computação natural, essa dissertação de mestrado se dedica à investigação de novas abordagens de otimização para a solução de problemas eletromagnéticos via Sistemas Imunes Artificiais (“Artificial Immune Systems” - AIS).

1.2 Contexto Histórico

Os estudos sobre sistemas bio-inspirados a partir da imunologia natural são recentes, e o artigo publicado por (Farmer et al. 1986) é considerado o trabalho pioneiro a relatar o sistema imune como uma técnica de inteligência artificial. Nesse artigo os autores descrevem um modelo dinâmico para o sistema imune, modelo este baseado em hipóteses sobre redes imunológicas.

Outros trabalhos importantes, que apresentaram estudos relevantes sobre sistemas imunes, começaram a surgir somente uma década após o relato de Farmer. Uma vez que o interesse sobre o assunto vem crescendo consideravelmente nos últimos anos, novas ideias foram propostas (Hunt & Cooke 1996, Dasgupta 1997, McCoy & Devarajan 1997, Dasgupta 1999a, Hofmeyr & Forrest 1999, Hofmeyr 2000), no entanto, nenhum modelo geral havia sido apresentado até o momento.

Os primeiros trabalhos a formalizarem o sistema imune artificial como uma ferramenta bio-inspirada, e que apresentaram modelos de algoritmos imunes, foram publicados por (de Castro & Von Zuben 1999, de Castro & Von Zuben 2000a, de Castro & Von Zuben 2000b, de Castro & Timmis 2002c).

Desde então, a pesquisa sobre a engenharia imune tem aumentado, e estudiosos de várias áreas acadêmicas vêm apresentando complementos aos sistemas imunes artificiais,

além de propostas de algoritmos imunes cada vez mais eficientes quanto a solução de problemas reais das áreas de engenharia e matemática; ver por exemplo (de Castro & Von Zuben 2002, de Castro 2002a, de Castro & Timmis 2002b, de Castro 2002b, de Castro 2002c, de Castro & Timmis 2002a, Campelo et al. 2005, Campelo et al. 2006).

1.2.1 Problema de Otimização Genérico

A modelagem matemática de um problema a partir de um sistema real é de fundamental importância no campo de pesquisa em contexto. O engenheiro, ou projetista, deve ser capaz de identificar as funções objetivo e restrições, além de especificar as variáveis de controle e o domínio viável intrínseco as mesmas. Conhecidos esses parâmetros, o problema de otimização é definido a seguir:

$$\begin{aligned} x^* &= \arg \min_x f(\vec{x}) \in \mathbb{R}^m \\ \text{sujeito a: } \vec{x} &\in \Omega \end{aligned} \tag{1.1}$$

em que $\vec{x}$ representa as variáveis de otimização, $f(\cdot)$ as m funções objetivo, e Ω é o conjunto factível, definido matematicamente por meio das funções restritivas. Formulações mais detalhadas sobre esses conceitos são apresentadas nos capítulos 3, otimização evolucionária mono-objetivo, e 4, otimização evolucionária multi-objetivo.

1.2.2 Algoritmo Evolucionário Unificado

Os sistemas imunes artificiais compõem uma das técnicas mais recentes inseridas na família dos algoritmos evolucionários. Assim sendo, esses métodos apresentam uma estrutura unificada, comumente descrita por meio de três etapas principais, as quais correspondem a *avaliação*, *seleção* e *variação* das possíveis soluções do problema de otimização (1.1).

O início do processo de otimização é normalmente caracterizado pela geração aleatória da primeira população de possíveis soluções, as quais são distribuídas sobre a região factível definida pelo conjunto Ω . Esses pontos são então avaliados nas m funções objetivo existentes, permitindo classificá-los quanto a qualidade da solução que cada um proporciona ao problema de otimização. Após esta etapa, efetua-se a seleção das melhores soluções encontradas até o momento, podendo este processo basear-se em mecanismos de escolha determinística e, ou, estocástica. O passo seguinte visa gerar novas soluções

a partir das melhores informações reunidas na etapa anterior, o que é feito por meio da aplicação de variações com características estocásticas sobre as soluções selecionadas. Finalmente, forma-se uma nova população evoluída e o ciclo se repete até ser atingido algum critério de parada estabelecido, momento no qual o algoritmo retorna uma estimativa das melhores soluções encontradas.

Essa estrutura cíclica está claramente esquematizada no Alg. 1.1.

Algoritmo 1.1: Algoritmo evolucionário unificado.

Input: Objetivos, restrições, espaço de busca, parâmetros

Output: Estimativa das melhores soluções

```

1 begin
2    $\mathcal{P}(n) \leftarrow$  População inicial;
3   while Não critério de parada do
4      $\Phi(n) \leftarrow$  Avaliação ( $\mathcal{P}(n)$ );
5      $\mathcal{S}(n) \leftarrow$  Seleção ( $\Phi(n)$ );
6      $\mathcal{Q}(n) \leftarrow$  Variação ( $\mathcal{S}(n)$ );
7      $\mathcal{P}(n+1) \leftarrow$  Atualização ( $\mathcal{P}(n), \mathcal{Q}(n)$ );
8      $n = n + 1$ ;
9   end
10 end
```

1.2.3 Algumas Aplicações

Atualmente existem inúmeros algoritmos baseados no sistema imune natural, e muitas aplicações têm sido abordadas, dentre as quais podem-se citar controle adaptativo e distribuído, aprendizagem, reconhecimento de padrões, detecção de falhas, segurança de redes de computadores, anti-vírus e anti-spam adaptativos, navegação de robôs, análise de dados e categorização, projeto de sistemas distribuídos, e otimização em geral, assunto o qual será destacado nesse trabalho.

Devido a grande necessidade de elaboração de novas técnicas de proteção contra a ação de vírus e invasores indesejados, os estudos referentes a segurança de redes de computadores têm ganhado grande ênfase entre os pesquisadores. Alguns trabalhos iniciais sobre o assunto foram publicados por (Okamoto & Ishida 1999, Dasgupta 1999b, Gu et al. 2000, Hofmeyr & Forrest 2000), e mais recentemente por (Guzella et al. 2008, Guzella & Caminhas 2009).

Os artigos (Hunt & Cooke 1996, Castiglione et al. 2001) ilustram uma abordagem simples das técnicas de aprendizagem e reconhecimento de padrões. Alguns conceitos básicos para a compreensão e elaboração de ferramentas para a análise e categorização de dados são descritos em (Timmis et al. 2000).

Outros trabalhos têm explorado a arquitetura imune para o ajuste de controladores PID (Amaral et al. 2005) e como auxílio à navegação de robôs (Krautmacher & Dilger 2004, Luh & Liu 2004).

O desenvolvimento de sistemas para a detecção de falhas em circuitos analógicos (Amaral et al. 2004, Guzella et al. 2007) e em motores de indução (Branco et al. 2003) também representam importantes casos de aplicações com sistemas imunes artificiais.

Uma lista mais detalhada, porém não tão abrangente, de artigos referenciando outras aplicações baseadas em sistemas imunes artificiais, é comentada em (de Castro & Timmis 2002b, de Castro & Timmis 2002c, Campelo 2006, Dasgupta 2007).

1.3 Contribuições da Dissertação

Tomando-se por base o contexto histórico apresentado, o presente trabalho vem enriquecer um pouco mais os conhecimentos sobre sistemas imunes artificiais, propondo novos algoritmos e operadores que caracterizam peculiaridades ainda não observadas nos trabalhos citados anteriormente.

As publicações (Batista, Guimarães & Ramírez 2009b, Batista, Guimarães, Paul & Ramírez 2009, Batista, Guimarães & Ramírez 2009c) compõem parte essencial do corpo dessa dissertação, a qual visa principalmente a formalização e apresentação de novos algoritmos imunológicos com foco especial em problemas cuja avaliação das funções objetivo envolva cálculos complexos e computacionalmente caros, mais especificamente, dispositivos eletromagnéticos.

Primeiramente sugere-se um algoritmo imunológico mono-objetivo e com codificação real dos seus parâmetros. O método proposto caracteriza-se principalmente por apresentar um operador de mutação baseado em sequências caóticas, o que melhora a sua capacidade de exploração do espaço de busca, possibilitando ao mesmo escapar de regiões pobres e identificar, mais rapidamente, o conjunto dos melhores ótimos locais. Observe que o termo *conjunto de ótimos locais* foi utilizado pois os algoritmos imunológicos são capazes de mapear o conjunto das melhores soluções de um problema ao longo de uma

única execução. O efeito de sistemas caóticos em processos de otimização será discutido com detalhes em um outro capítulo.

Posteriormente, é também proposto um algoritmo imunológico multi-objetivo, cuja parte de suas características são herdadas da versão mono-objetivo citada. Esse método, além de implementar uma rotina de mutação com propriedades caóticas, o que desempenha importante papel quanto à uniformidade e distribuição das soluções ao longo do Pareto-ótimo estimado, sugere ainda a inserção de um mecanismo de *edição de receptores*¹, visando com isso refinar as características das soluções de elevada afinidade².

Como pode ser observado nos próximos capítulos, a comparação das abordagens propostas frente a métodos conhecidos na literatura revela ganhos consideráveis daqueles em relação ao custo computacional, além de apresentar alto desempenho diante de problemas analíticos e eletromagnéticos.

1.4 Estrutura do Trabalho

Esta dissertação está organizado em seis partes.

Capítulo 2 – Sistema Imune Biológico e Artificial: Este capítulo contém uma breve discussão sobre o tema *sistema imunológico natural*, onde são destacados os principais mecanismos de defesa do sistema imune animal e a teoria da seleção clonal (“Clonal Selection Theory” - CST). Apresenta-se também uma visão geral sobre os métodos imunes artificiais, enfatizando seus mecanismos de variação genética e seleção natural. Além disso, são abordadas algumas peculiaridades dessa classe de algoritmos relacionadas a evolução paralela e a capacidade ímpar de otimização de funções multimodais.

Capítulo 3 – Otimização Evolucionária Mono-Objetivo: O capítulo 3 define inicialmente o problema de otimização mono-objetivo, apresenta as condições necessárias de otimalidade, e mostra como são tratadas as funções de restrição em algoritmos evolucionários. Após ilustrar a estrutura geral de AE's, são então descritos alguns métodos de otimização que serviram de ferramenta para a construção desse trabalho.

Capítulo 4 – Otimização Evolucionária Multi-Objetivo: O capítulo 4 apresenta uma estrutura similar ao capítulo 3. Dessa forma, define-se inicialmente o problema de

¹Do inglês “receptor editing”, é um processo observado no sistema imunológico animal.

²Supondo-se um problema de minimização, uma solução com alta afinidade apresentará reduzido valor na função objetivo.

otimização multi-objetivo, apresenta-se uma extensão para condições necessárias de otimalidade para eficiência, e, posteriormente, são relatados alguns métodos multi-objetivo conhecidos na literatura.

Capítulo 5 – Sistemas Imunes Artificiais: Novas Abordagens: Este capítulo contém a contribuição dessa dissertação. Nele são discutidas e estruturadas as rotinas dos algoritmos propostos, mono e multi-objetivo, e além disso é realizado um estudo para o ajuste dos parâmetros dos métodos.

Capítulo 6 – Resultados: Inicialmente avalia-se o desempenho dos algoritmos propostos frente a problemas analíticos e eletromagnéticos, os quais expressam características, tais como, superfície de otimização suave, multimodalidade, múltiplas variáveis, não convexidade, desconexidade, funções restritivas e elevado custo computacional. Posteriormente, os resultados obtidos são analisados por meio de diferentes técnicas de avaliação de desempenho, e discute-se ainda a função dos novos operadores sugeridos.

Capítulo 7 – Conclusões: Conclui a dissertação e apresenta uma visão geral sobre o trabalho desenvolvido. Esse capítulo é fechado com a enumeração de sugestões a serem exploradas em trabalhos futuros.

Capítulo 2

Sistema Imune Biológico e Artificial

“Não é o mais forte que sobrevive, nem o mais inteligente, mas o que melhor se adapta às mudanças.”

— Charles Darwin, 1809–1882 (naturalista britânico)

“O que não provoca minha morte faz com que eu fique mais forte.”

— Friedrich Nietzsche, 1844–1900 (filósofo alemão)

2.1 Introdução

Este capítulo contém uma breve discussão sobre o tema *sistema imunológico natural*, onde são destacados os principais mecanismos de defesa do sistema imune animal e a teoria da seleção clonal (“Clonal Selection Theory” - CST), princípio o qual inspirou, e ainda inspira, o desenvolvimento de grande parte dos sistemas imunes artificiais. Apresenta-se também uma visão geral sobre o funcionamento dos métodos imunes, enfatizando seus mecanismos de variação genética e seleção natural, além de discorrer acerca de algumas características peculiares desta classe de algoritmos.

2.2 Sistema Imune Natural - Uma Visão Geral

Devido a grande complexidade associada ao funcionamento do sistema imune natural, existem atualmente diferentes metáforas explicativas acerca do mesmo, dentre as quais se destacam a teoria da imunologia clássica, baseada na discriminação própria/não-própria (“self-nonsel self discrimination”) (Goldsby et al. 2000, Abbas & Lichtman March 2005), o princípio da seleção clonal (“clonal selection theory”) (Burnet 1978), a teoria da seleção negativa (“negative selection”) (Nossal 1994), e também a teoria do perigo (“danger theory”) (Matzinger 1994, Matzinger 2001). Na discussão seguinte serão enfatizados sobretudo a imunologia clássica e a teoria da seleção clonal.

O sistema imune representa um conjunto de processos biológicos cuja função é proteger o organismo animal contra agentes patogênicos, identificando e eliminando tais células. O sistema imune é responsável por detectar uma extensa variedade de agentes, desde parasitas a viroses, e precisa ser capaz de distingui-los de células saudáveis e tecidos do próprio organismo, garantindo assim que o mesmo funcione corretamente. No entanto, o processo de detecção é muito complicado, podendo apresentar muitas deficiências, permitindo às vezes que células patogênicas evoluam rapidamente, comprometendo assim a saúde do organismo. Felizmente, quando o organismo se vê ameaçado, múltiplos mecanismos de defesa são ativados, agilizando o reconhecimento e a neutralização de patógenos.

O sistema imune dos vertebrados consiste de vários tipos de proteínas, células, órgãos, e tecidos, os quais interagem formando uma elaborada e dinâmica rede imunológica. Uma importante característica desta complexa *resposta imune* é que o sistema imunológico humano sofre constantes adaptações para torná-lo capaz de reconhecer patógenos específicos com maior eficiência. Este processo de adaptação recebe o nome de *imunidade adaptativa*, ou *imunidade adquirida*, sendo o mesmo responsável por desenvolver uma *memória imunológica*. A memória imune é criada devido a atuação da *resposta primária* a um patógeno específico, e, dessa forma, quando esse mesmo patógeno voltar a ameaçar o organismo, o mesmo será capaz de apresentar uma *resposta secundária* aprimorada, ou seja, mais rápida e eficiente. Esse processo de aquisição de imunidade é a base do princípio de funcionamento da vacinação.

As próximas seções apresentam uma breve discussão acerca dos principais mecanismos de defesa do sistema imune, os quais ajudaram a *modelar* os primeiros sistemas imunes artificiais que surgiram.

2.2.1 Linhas de Defesa do Sistema Imune

O sistema imune protege o organismo contra infecções por meio de linhas de defesa de diferentes graus de especificidade. A primeira linha de defesa corresponde a barreiras físicas desempenhadas pela pele, mucosas e secreções, os quais compõem a forma mais simples de prevenção contra patógenos, tais como, bactérias e viroses. Caso essa barreira inicial sofra rupturas, então, o sistema imune inato apresenta uma resposta imediata, porém não-específica. Dessa forma, se os patógenos não forem barrados pela resposta inata, os vertebrados¹ apresentam a terceira linha de proteção, o sistema imune adaptativo, o qual é ativado pelo sistema inato. O sistema imune adapta sua resposta durante a infecção, aperfeiçoando a sua capacidade de reconhecimento desse patógeno específico. Esta resposta aprimorada é então armazenada na memória imune mesmo após o patógeno ser eliminado, o que permite ao sistema imune adaptativo apresentar respostas rápidas e eficientes contra esse patógeno toda vez que o mesmo for identificado no organismo.

De forma geral, o sistema imune apresenta as seguintes características:

- **Sistema Imunológico Inato:**

Resposta não-específica;

Exposição ao antígeno conduz a uma resposta máxima e imediata;

Não possui memória imunológica;

Encontrado em quase todas as formas de vida.

- **Sistema Imunológico Adaptativo:**

Resposta específica ao antígeno;

Atraso de tempo entre a exposição e a resposta máxima;

Exposição ao antígeno ativa a memória imune;

Encontrado somente nos vertebrados (surperclasse Gnathostomata).

Sistema Imune Inato

Qualquer microorganismo ou toxina que consiga ultrapassar a primeira linha de defesa é submetido imediatamente a atuação do sistema imune inato. Esse sistema é nor-

¹Somente os pertencentes à surperclasse Gnathostomata.

malmente ativado quando bactérias ou viroses são identificados por meio de receptores capazes de realizarem reconhecimento de padrões (“pattern recognition receptors”). No entanto, a defesa oferecida pelo sistema inato não é específica, o que significa que as respostas aos patógenos são feitas de maneira genérica, além de não conferir imunidade por um longo intervalo de tempo.

Os sintomas de inflamação representam uma das primeiras respostas do sistema imune contra infecções. Esses sintomas são basicamente caracterizados por vermelhidão e inchaço, os quais são causados pelo aumento de fluxo sanguíneo nos tecidos.

A segunda ferramenta de defesa do sistema imune inato é desempenhada pelos leucócitos, ou glóbulos brancos. Os leucócitos desse sistema são compostos por fagócitos (macrófagos, neutrófilos, etc), mastócitos, eosinófilos, basófilos e células citotóxicas (matadoras) naturais. Essas células são responsáveis por identificar e eliminar patógenos, além de agirem como importantes mediadores para a ativação do sistema imune adaptativo.

Sistema Imune Adquirido

O sistema imune adaptativo se desenvolve apenas em vertebrados pertencentes à superclasse Gnathostomata, e devido à capacidade de armazenamento de uma memória imune, este sistema provê uma resposta imune rápida e eficiente contra patógenos reconhecidos.

A resposta imune adquirida é específica para cada antígeno, e é desempenhada por tipos especiais de leucócitos, chamados linfócitos. Os principais tipos de linfócitos são as células B e células T, as quais são produzidas na medula óssea e, por sua vez, são responsáveis por transportarem moléculas cujos receptores são específicos de um dado patógeno.

A capacidade de elaboração de uma memória imunológica é o que torna o sistema imune adquirido tão especial. Quando as células B e T são ativadas, estas passam por um processo de multiplicação celular e boa parte das novas células geradas compõem um grupo de células de memória que permanecem no organismo durante um longo tempo. Dessa forma, durante todo o período de vida do animal, estas células de memória poderão realizar o reconhecimento de patógenos específicos que em algum momento já tenham sido detectadas no organismo, provendo uma resposta poderosa contra as mesmas. Como já deve estar claro, o nome *adquirido* deve-se ao fato de que é formado durante o período

de vida do indivíduo, e funciona como um mecanismo que sofre constantes adaptações com o intuito de aperfeiçoar o sistema de defesa imune.

A memória imune pode ser adquirida de maneira passiva (memória imune de curta duração) ou de maneira ativa (memória imune de longa duração). Esses termos são discutidos a seguir.

- **Memória Imune Passiva**

Como os recém-nascidos não são previamente expostos a bactérias ou micróbios, infelizmente eles são vulneráveis a infecções. Entretanto, durante a gestação a mãe é capaz de prover diversos tipos de proteção passiva ao bebê, sendo assim, um tipo especial de anticorpo é transferido através da placenta diretamente da mãe para o bebê, garantindo ao mesmo altos níveis de anticorpos até o seu nascimento. Mesmo após essa etapa, a mãe continua a fornecer anticorpos por meio do leite materno, o que o protegerá contra infecções até o momento em que ele próprio seja capaz de sintetizar seus próprios anticorpos. Este é o princípio da imunidade passiva, podendo durar entre poucos dias até alguns meses.

- **Memória Imune Ativa**

A memória ativa é adquirida por meio da ativação das células B e T devido a ocorrência prévia de uma infecção. Felizmente, a imunidade ativa pode também ser gerada artificialmente através da vacinação, em que a introdução de antígenos enfraquecidos de um dado patógeno estimulam o sistema imune a produzir anticorpos específicos contra esse patógeno.

De forma geral, a maior parte das vacinas virais são baseadas na atenuação destes, enquanto muitas vacinas bacterianas são baseadas em componentes acelulares de microorganismos, incluindo toxinas inofensivas. Entretanto, como muitos antígenos derivados de vacinas acelulares não induzem uma resposta adaptativa eficaz, a maioria das vacinas bacterianas são fabricadas com substâncias complementares, que ativam também as células do sistema imune inato, maximizando assim o sistema de defesa.

2.2.2 Princípio da Seleção Clonal

A habilidade do sistema imune adquirido de reconhecer estruturas de diferentes epítomos e de se adaptar quando exposto pela primeira vez a um antígeno é explicado pela *teoria*

da *seleção clonal*. Durante várias décadas muitos imunologistas contribuíram para o estudo deste princípio, mas somente em 1950 ele foi completamente formalizado pelo australiano Frank Macfarlane Burnet.

A Fig. 2.1 apresenta uma visão geral sobre a teoria da seleção clonal de Burnet. As células-tronco hematopoéticas (1) estão ligadas à geração de diversos constituintes do sangue, o que é normalmente realizado por meio de diferenciações e recombinações

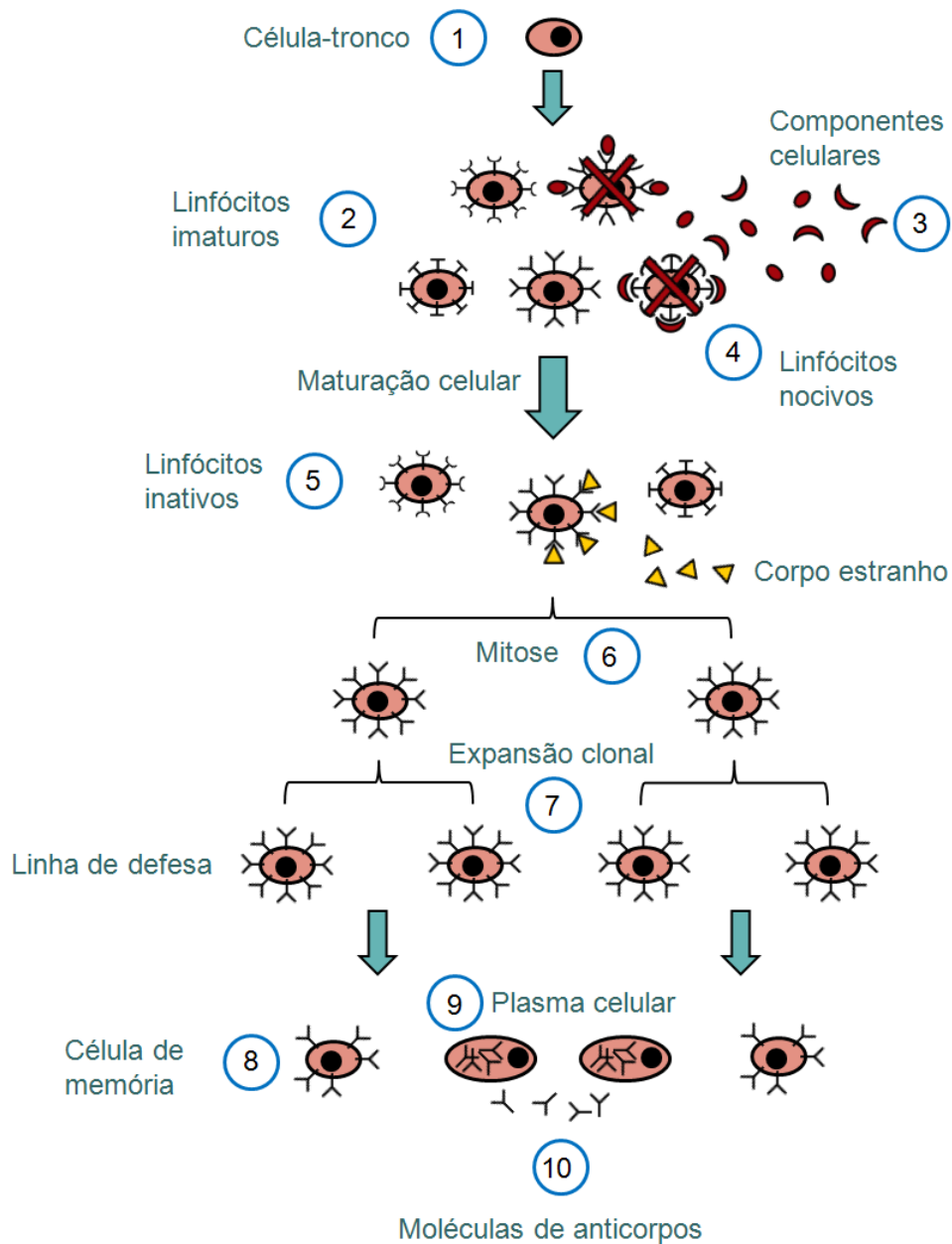


Figura 2.1: Esquema de funcionamento do princípio da seleção clonal.

genéticas. Um destes componentes corresponde aos linfócitos imaturos (2), os quais apresentam diferentes receptores de antígenos. Dentre estes linfócitos, todos aqueles que se conectarem, ou atacarem, componentes celulares do próprio organismo (3) serão eliminados (4), enquanto o restante passa por um processo de maturação celular formando um conjunto de linfócitos inativos (5). Muitos destes linfócitos nunca se acoplarão a nenhum corpo estranho, no entanto, aqueles que o fizerem serão ativados, sendo submetidos aos processos de mitose (6) e expansão clonal (7), formando assim a linha de defesa do organismo. Nesta última etapa são também elaborados as células de memória (8), o plasma celular (9) e moléculas de anticorpos (10).

Maiores detalhes sobre o sistema imune podem ser encontrados em (Talmage 1957, Burnet 1959, Burnet 1978). Um estudo detalhado sobre o assunto, sob o ponto de vista de engenharia imune, é descrito em (de Castro & Von Zuben 1999).

2.3 Sistemas Imunes Artificiais

Os sistemas imunes artificiais são fruto da modelagem de certos conceitos e mecanismos observados no sistema imune natural. Entretanto, essa *modelagem* não visa reproduzir fielmente o fenômeno analisado, mas estruturá-lo de forma a compor uma ferramenta para a solução de problemas de engenharia. Logo, nem sempre é mantida a coerência com o que se observa na teoria imunológica. A seguir são apresentados alguns conceitos sobre esses sistemas e também algumas peculiaridades relacionadas aos mesmos.

2.3.1 Conceitos Básicos

A fonte de inspiração para a formalização de muitos métodos imunes artificiais tem sido fundamentada no princípio da seleção clonal, e assim como descrito na teoria da evolução de Charles Darwin, o funcionamento da seleção clonal pode ser caracterizado por três estágios principais, os quais são *seleção natural*, *variação genética* e *diversidade genética* (Cziko 1995).

De forma similar ao verificado na maior parte dos algoritmos evolucionários, uma vez que uma população de possíveis soluções é gerada sobre o espaço de busca do problema de otimização, as mesmas são avaliadas quanto ao seu grau de afinidade desenvolvido, e quanto melhor essa afinidade, melhor é a representatividade da solução. Em sistemas

imunes artificiais, o termo *afinidade* pode ser associado ao valor retornado pela função objetivo, assim, a melhor solução de um problema de minimização é aquela de menor valor da função objetivo, a qual corresponde à solução de maior afinidade.

Dentre as soluções avaliadas, apenas as que apresentam maiores valores de afinidade são submetidas ao processo de variação. As piores soluções encontradas, aquelas de menor afinidade, são simplesmente eliminadas, ou podem ser editadas, o que possibilitaria as mesmas melhorarem o seu grau de afinidade.

Em algoritmos imunológicos, o processo de variação recebe o nome de *expansão clonal*, e essa etapa pode ser subdividida em três itens principais, sendo *clonagem*, *maturação* e *seleção elitista*. Dessa forma, cada uma das soluções selecionadas recebe um número especificado de clones (clonagem), os quais são submetidos a sucessivas mutações (maturação). O conjunto de soluções formado pelos clones maturados e pela solução original recebe o nome de subpopulação, e dentre estas soluções apenas a melhor é selecionada para prosseguir para a próxima geração (seleção elitista).

Embora o processo de maturação seja capaz de prover uma certa diversificação local, a geração de diversidade global é normalmente devida à substituição das piores soluções por novos pontos criados aleatoriamente sobre o espaço de busca, os quais possibilitam explorar novas regiões ainda não visitadas, aumentando assim a capacidade do algoritmo de escapar de ótimos locais e possivelmente encontrar soluções de alta afinidade.

Outros detalhes sobre o princípio de funcionamento dos sistemas imunes artificiais podem ser encontrados em (de Castro & Von Zuben 2000b, de Castro & Von Zuben 2002).

O Alg. 2.1 apresenta a estrutura básica de um algoritmo imune. De maneira geral, tem-se que após a geração da população inicial, o processo iterativo inicia-se com a avaliação das possíveis soluções do problema de otimização. Uma parcela dessa população, compreendida pelas melhores soluções encontradas, é então selecionada e, posteriormente, submetida à etapa de expansão clonal, permitindo ao método exercer mutações sobre as informações reunidas e gerar soluções aperfeiçoadas. Por fim, com o objetivo de evitar uma convergência prematura do algoritmo, gera-se diversidade sobre o espaço de busca, possibilitando a exploração de novas áreas prósperas. Uma nova população evoluída é estabelecida, e o ciclo prossegue até que se verifique o critério de parada especificado. A Fig. 2.2 faz uma analogia a esse algoritmo e mostra o diagrama de blocos associado, onde são evidenciados os principais passos abordados anteriormente.

Algoritmo 2.1: Ciclo básico de funcionamento de um sistema imune artificial.

Input: Objetivo, restrições, espaço de busca, parâmetros**Output:** Estimativa das melhores soluções

```

1 begin
2    $\mathcal{P}(n) \leftarrow$  População inicial;
3   while Não critério de parada do
4      $\Phi(n) \leftarrow$  Avaliação ( $\mathcal{P}(n)$ );
5      $\mathcal{S}(n) \leftarrow$  Seleção ( $\Phi(n)$ );
6      $\mathcal{Q}(n) \leftarrow$  Expansão Clonal ( $\mathcal{S}(n)$ );
7      $\mathcal{D}(n) \leftarrow$  Geração de Diversidade;
8      $\mathcal{P}(n+1) \leftarrow$  Atualização ( $\mathcal{P}(n), \mathcal{Q}(n), \mathcal{D}(n)$ );
9      $n = n + 1$ ;
10  end
11 end

```

2.3.2 Evolução Paralela e Otimização Multimodal

Assim como outros algoritmos evolucionários, os algoritmos imunes apresentam evolução paralela das suas soluções, excelente desempenho quanto ao custo computacional e capacidade de aprendizado de novas informações. Além disso, os métodos imunes apresentam uma característica peculiar em relação aos demais métodos evolucionários, que se refere a sua capacidade de desenvolvimento de uma busca multimodal e independente, com memorização dos melhores ótimos locais encontrados.

Uma vez que o sistema de variação empregado nos métodos imunes (expansão clonal) baseia-se exclusivamente em mecanismos de mutação ou maturação celular, isto é, não há necessidade de troca de informações entre as soluções, as subpopulações evoluem de forma independente, e normalmente caminham para regiões distintas do espaço de busca, o que possibilita o mapeamento de ótimos locais ao longo do processo de otimização.

Para ilustrar essa capacidade dos sistemas imunes quanto ao mapeamento de múltiplos ótimos durante a otimização de funções multimodais, a Fig. 2.3 ilustra o resultado obtido para quatro problemas conhecidos, os quais correspondem as funções *Rastrigin* (P1), *Multi* (P2), *Roots* (P3) e *Schaffer* (P4), equação (2.1). Em ambos os casos tem-se em mente a maximização da função objetivo. O método imune empregado para a solução dos problemas de otimização foi o “Clonal Algorithm” (CLONALG) (de Castro & Von Zuben 2002), no entanto, como esse exemplo é apenas ilustrativo, esse algoritmo imune não será discutido neste capítulo. Um estudo amplo sobre a otimização de funções

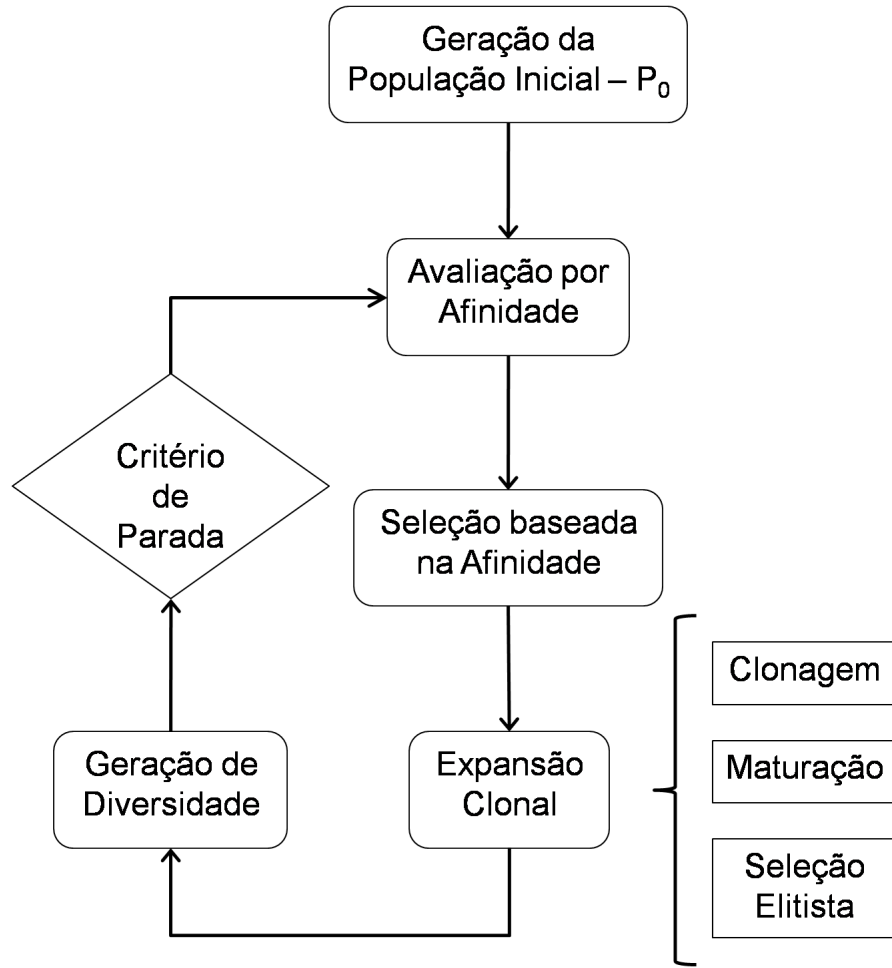
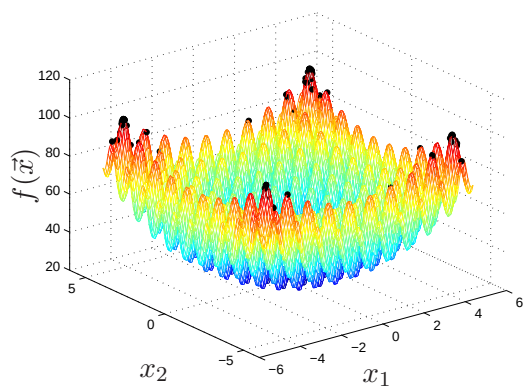
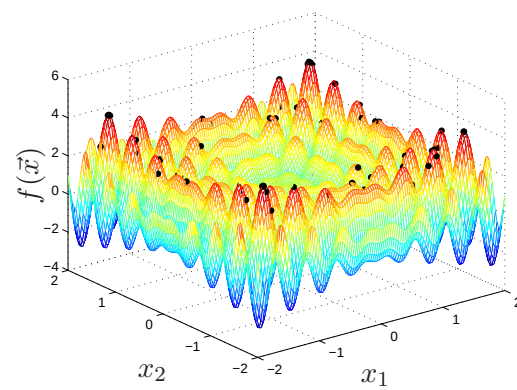
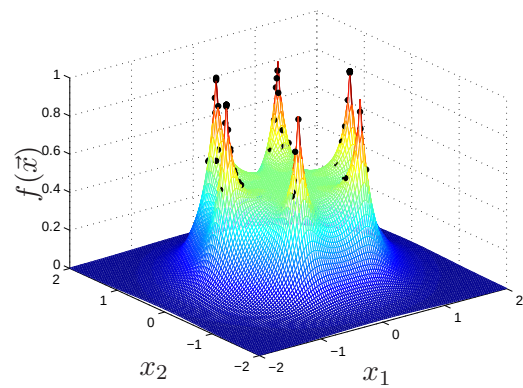
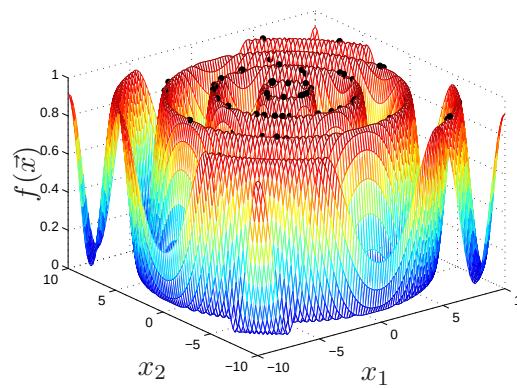


Figura 2.2: Diagrama de blocos do ciclo básico de funcionamento de um algoritmo imune.

multimodais com base em sistemas imunes artificiais é apresentado em (de Castro & Timmis 2002a).

$$\begin{aligned}
 P1 : x^* &= \arg \max_x f(\vec{x}) = 40 + \sum_{i=1}^n x_i^2 - 10 \cos(2\pi x_i), & \vec{x} &\in [-5.12, 5.12] \\
 P2 : x^* &= \arg \max_x f(\vec{x}) = x_1 \sin(4\pi x_1) - x_2 \sin(4\pi x_2) + 1, & \vec{x} &\in [-2, 2] \\
 P3 : x^* &= \arg \max_x f(\vec{x}) = \frac{1}{1 + |(x_1 + jx_2)^6 - 1|}, & \vec{x} &\in [-2, 2] \quad (2.1) \\
 P4 : x^* &= \arg \max_x f(\vec{x}) = 0.5 + \frac{\sin^2\left(\sqrt{x_1^2 + x_2^2}\right) - 0.5}{1 + 0.001(x_1^2 + x_2^2)}, & \vec{x} &\in [-10, 10]
 \end{aligned}$$

(a) Função *Rastrigin*(b) Função *Multi*(c) Função *Roots*(d) Função *Schaffer***Figura 2.3:** Exemplos de otimização de funções multimodais.

2.4 Conclusão

Com o propósito de situar o leitor quanto ao assunto a ser tratado ao longo desse trabalho, o presente capítulo discorreu brevemente sobre alguns conceitos e princípios que sustentam a teoria do sistema imunológico animal. Baseado na forma de atuação de alguns dos mecanismos imunes naturais, apresentou-se também a base de funcionamento dos sistemas imunes artificiais, caracterizando especialmente a *modelagem* dos principais operadores dessa classe de algoritmos e algumas peculiaridades da mesma.

O capítulo seguinte se dedica a descrição de alguns métodos evolucionários mono-objetivo, dentre os quais explora-se principalmente os sistemas imunes artificiais. Os mecanismos de variação genética e seleção natural apresentados anteriormente são implementados nesses algoritmos, possibilitando maiores esclarecimentos acerca da atuação destes operadores no desempenho dos métodos.

Capítulo 3

Otimização Evolucionária Mono-Objetivo

“Para ser um bom observador é preciso ser um bom teórico.”

— Charles Darwin, 1809–1882 (naturalista britânico)

“A ciência consiste em substituir o saber que parecia seguro por uma teoria, ou seja, por algo problemático.”

— José Ortega y Gasset, 1883–1955 (filósofo espanhol)

3.1 Introdução

Este capítulo visa abordar algumas definições, conceitos e algoritmos relacionados à otimização evolucionária mono-objetivo. Partindo-se deste princípio, define-se inicialmente o problema de otimização mono-objetivo, apresenta-se as condições necessárias de otimilidade, e mostra-se como são tratadas as funções de restrição em sistemas evolucionários. Após a apresentação da estrutura geral dos algoritmos evolucionários, são então descritos alguns métodos de otimização de grande importância na literatura, e comentadas suas características mais relevantes.

3.2 Definição do Problema de Otimização

O problema de otimização mono-objetivo pode ser definido como:

$$\begin{aligned} x^* &= \arg \min_x f(\vec{x}) \\ \text{sujeito a: } &\vec{x} \in \Omega \end{aligned} \quad (3.1)$$

em que o conjunto Ω representa a região factível, sendo matematicamente definida por:

$$\Omega = \begin{cases} g_i(\vec{x}) \leq 0; & i = 1, \dots, p \\ h_j(\vec{x}) = 0; & j = 1, \dots, q \end{cases} \quad (3.2)$$

O espaço de busca é limitado pelas variáveis de decisão $lb_k \leq x_k \leq ub_k$, $k = 1, \dots, n$, em que lb_k , ub_k são os limites inferior e superior da variável k , e n é o número de parâmetros do problema. Tem-se ainda $\vec{x} \in \mathbb{R}^n$, $f(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^1$, $g(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^p$ e $h(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^q$.

As funções $g_i(\vec{x})$ e $h_j(\vec{x})$ representam as restrições de desigualdade e igualdade, respectivamente. A função $f(\vec{x})$ é o objetivo do problema de otimização, e a solução é o ponto $\vec{x} \in \Omega$ que fornece o menor valor da função objetivo¹.

Para um problema de otimização definido conforme (3.1) e (3.2), as condições de otimalidade local e global são fornecidas a seguir (Bazaraa et al. 1979):

Definição 3.1. (Solução ótima local) Supondo $x^* \in \Omega$ e sua vizinhança $\mathcal{V}_\epsilon(x^*)$ definida por um raio $\epsilon > 0$, se $f(x^*) \leq f(\vec{x})$ para cada $\vec{x} \in \mathcal{V}_\epsilon(x^*)$, então, x^* é chamado *mínimo local*. $\square$

Definição 3.2. (Solução ótima global) Supondo $x^* \in \Omega$, se $f(x^*) \leq f(\vec{x})$ para todo $\vec{x} \in \Omega$, então, x^* é chamado *mínimo global*. $\square$

De forma geral, observa-se que um ótimo local é aquele cuja afinidade não pode ser melhorada efetuando-se perturbações locais nessa solução; além disso, um ótimo global é também um ótimo local.

¹Por convenção é adotado um problema de minimização, podendo ser facilmente convertido para maximização fazendo $-f(\vec{x})$.

3.3 Condições de Otimalidade

As condições de otimalidade descritas por Kuhn e Tucker em 1951 fornecem condições necessárias para que um ponto x^* seja solução do problema (3.1).

A seguinte proposição serve de base para diversos algoritmos de otimização (Bazaraa et al. 1979):

Proposição 3.1. (*Condições necessárias de Kuhn-Tucker*) Supondo $x^* \in \Omega$, esse ponto será um ótimo local do problema (3.1) se existir um conjunto de multiplicadores de Lagrange $\mu_i^* \in \mathbb{R}^p$ e $\lambda_j^* \in \mathbb{R}^q$, tal que a equação (3.3) seja satisfeita. $\square$

$$\begin{aligned} \nabla f(x^*) + \sum_{i=1}^p \mu_i \nabla g_i(x^*) + \sum_{j=1}^q \lambda_j \nabla h_j(x^*) &= 0 \\ \mu_i &\geq 0, \quad \mu_i g_i(x^*) = 0, \quad i = 1, \dots, p \\ h_j(\vec{x}) &= 0, \quad j = 1, \dots, q \end{aligned} \tag{3.3}$$

A Fig. 3.1 mostra uma interpretação geométrica para a condição de Kuhn-Tucker considerando-se restrições de desigualdade. Observe que no ponto de solução x^* existem duas restrições ativas, e que é possível determinar os valores dos multiplicadores μ_1^* e μ_2^* positivos, tal que a soma dos vetores gradiente em x^* se anule.

3.4 Tratamento de Restrições em Algoritmos

Evolucionários

Os algoritmos evolucionários (AE) são métodos criados para resolverem problemas ir-restritos. Assim sendo, torna-se necessário a inserção de um mecanismo extra para que esses métodos sejam capazes de lidar com restrições. Visto isso, inúmeras técnicas para o tratamento de restrições foram propostas nos últimos anos, dentre as quais se destacam quatro famílias:

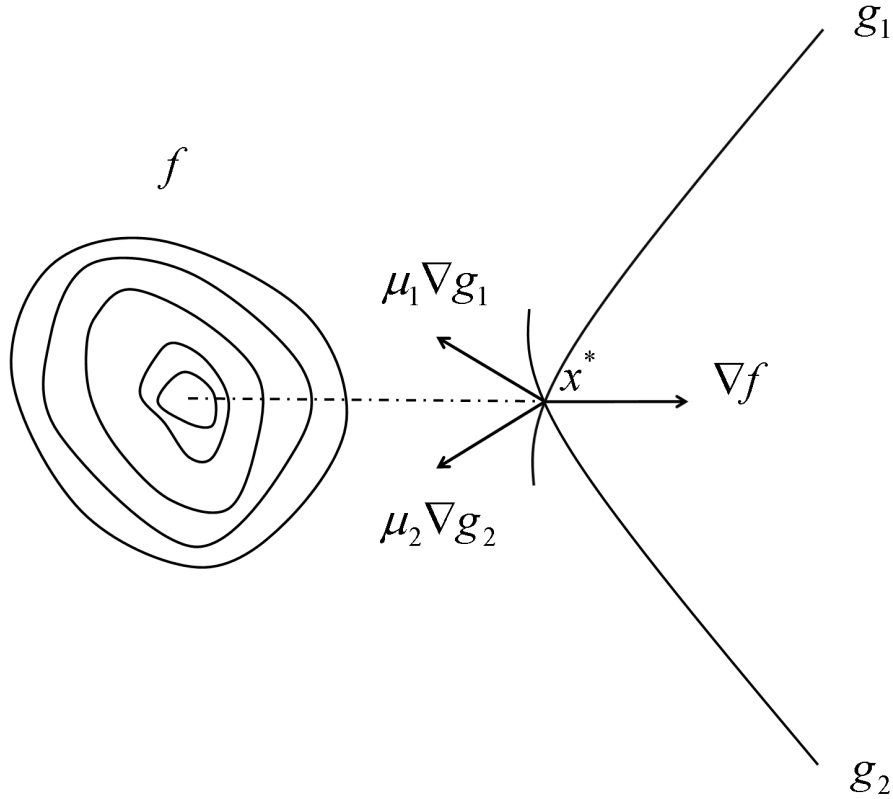


Figura 3.1: Ilustração das condições de Kuhn-Tucker para o caso mono-objetivo.

1. Uso de funções de penalidades;
2. Emprego somente de soluções pertencentes ao conjunto factível;
3. Separação entre objetivos e restrições;
4. Métodos híbridos.

Neste trabalho optou-se por considerar a maneira tradicional para o tratamento de restrições, ou seja, *funções de penalidade*. Métodos que empregam funções de penalidade basicamente transformam um problema com restrições em um único problema irrestrito. De forma geral, as restrições são acopladas à função objetivo por meio de parâmetros de penalidade, e dessa maneira, torna-se possível penalizar qualquer violação das restrições.

As restrições de igualdade e desigualdade são tratadas de formas diferentes, sendo assim, considere inicialmente o problema de otimização P_1 (3.4):

- P_1

$$\begin{aligned} \min f(\vec{x}) \\ \text{sujeito a: } h(\vec{x}) = 0 \end{aligned} \tag{3.4}$$

Intuitivamente, sabe-se que a restrição só será satisfeita quando a solução $x^* \in \Omega$ anular a igualdade ou aproximá-la sensivelmente de zero, ou seja, quando $h(x^*) \cong 0$. Por outro lado, enquanto a restrição for violada, deve existir uma penalidade de forma a forçar o algoritmo a procurar por soluções melhores. Assim sendo, o problema de otimização irrestrito associado pode ser escrito como mostrado em P'_1 (3.5):

- P'_1

$$\begin{aligned} \min f(\vec{x}) + \alpha |h(\vec{x})| \\ \text{sujeito a: } \vec{x} \in \Omega \end{aligned} \tag{3.5}$$

Suponha-se agora um problema de otimização conforme descrito em P_2 (3.6):

- P_2

$$\begin{aligned} \min f(\vec{x}) \\ \text{sujeito a: } g(\vec{x}) \leq 0 \end{aligned} \tag{3.6}$$

Observa-se facilmente que a solução $x^* \in \Omega$ não violará a restrição de desigualdade somente quando $g(x^*) \leq 0$. Dessa forma, deverá existir uma penalidade sobre essa restrição enquanto a mesma retornar um valor maior do que zero. Então, o problema de otimização irrestrito associado pode ser elaborado conforme evidenciado em P'_2 (3.7):

- P'_2

$$\begin{aligned} \min f(\vec{x}) + \alpha \max \{0, g(\vec{x})\} \\ \text{sujeito a: } \vec{x} \in \Omega \end{aligned} \tag{3.7}$$

Caso todas as restrições de um problema de otimização sejam da forma $g_i(\vec{x}) \leq 0$, $i = 1, \dots, p$ e $h_j(\vec{x}) = 0$, $j = 1, \dots, q$, então, define-se uma função de penalidade geral $\rho(\vec{x})$ (3.8):

$$\rho(\vec{x}) = \sum_{i=1}^p [\max\{0, g_i(\vec{x})\}]^r + \sum_{j=1}^q |h_j(\vec{x})|^r \quad (3.8)$$

em que r é um expoente inteiro e positivo.

Finalmente, escreve-se a função auxiliar de penalidade (3.9). Por simplicidade, em todos os testes realizados nesse trabalho é considerado $r = 1$ e $\alpha = 100$, conforme adotado por (Campelo et al. 2005).

$$f'(\vec{x}) = f(\vec{x}) + \alpha\rho(\vec{x}) \quad (3.9)$$

Note que $f'(\vec{x})$ incorpora uma soma de erros oriundos de $h(\vec{x})$ e $g(\vec{x})$, o que provavelmente compromete o desempenho do método. O plausível seria, então, expressar a restrição de igualdade por meio de duas restrições de desigualdade, e além disso, empregar diferentes penalidades (α) em função das características de cada uma das restrições.

3.5 Estrutura Geral de um AE Mono-Objetivo

Os métodos evolucionários são assim chamados, pois a aplicação de mecanismos naturais sobre um conjunto de soluções de um problema de otimização permite evoluir as características dessas soluções de forma a melhorar, a cada geração, os valores de aptidão retornados pela função objetivo.

Estes mecanismos naturais se baseiam principalmente nos princípios de seleção, variação e geração de diversidade, os quais exercem uma pressão sobre as soluções do problema, de maneira que estas são submetidas a um processo gradual de adaptação ao ambiente de otimização em questão.

Observe que embora os algoritmos evolucionários sejam comumente classificados como estocásticos, pois são compostos por operações heurísticas, essa nomenclatura não significa que esses métodos realizam uma convergência aleatória. Na verdade, esses algoritmos forçam uma busca direcionada para as regiões do ambiente de otimização onde localizam-se os melhores valores de aptidão.

O Alg. 3.1 apresenta a estrutura básica de um algoritmo evolucionário mono-objetivo. De forma geral, o mecanismo de seleção visa destacar as melhores soluções encontradas

até o momento, enquanto a variação, normalmente composta por operações de mutações e, ou, cruzamentos, tem em mente explorar novas regiões do espaço de busca, e combinar informações das soluções de forma a melhorá-las. Por fim, as soluções são atualizadas mantendo-se o tamanho da população constante.

Algoritmo 3.1: Ciclo básico de funcionamento de um AE mono-objetivo.

Input: Objetivo, restrições, espaço de busca, parâmetros

Output: Estimativa da melhor solução

```

1 begin
2    $\mathcal{P}(n) \leftarrow$  População inicial;
3   while Não critério de parada do
4      $\Phi(n) \leftarrow$  Avaliação ( $\mathcal{P}(n)$ );
5      $\mathcal{S}(n) \leftarrow$  Seleção ( $\Phi(n)$ );
6      $\mathcal{Q}(n) \leftarrow$  Variação ( $\mathcal{S}(n)$ );
7      $\mathcal{P}(n+1) \leftarrow$  Atualização ( $\mathcal{P}(n), \mathcal{Q}(n)$ );
8      $n = n + 1$ ;
9   end
10 end

```

3.6 Sistemas Evolucionários Mono-Objetivo

Nesta seção são apresentados cinco métodos de otimização muito conhecidos na literatura, dentre os quais apenas quatro são empregados para a análise de desempenho do algoritmo proposto. Inicialmente são descritos três métodos baseados na teoria da seleção clonal (seção 2.2.2), sendo estes o “Clonal Algorithm” (CLONALG) (de Castro & Von Zuben 2002), “B-Cell Algorithm” (BCA) (Kelsey & Timmis 2003) e “Real-Coded Clonal Selection Algorithm” (RCSA) (Campelo et al. 2005, Campelo 2006). Posteriormente são apresentados o “Real-Biased Genetic Algorithm” (RBGA) (Takahashi et al. 2003, Takahashi 2004), inspirado a partir da teoria da evolução de Charles Darwin, e o “Differential Evolution Algorithm” (DEA) (Storn & Price 1997, Price et al. 2005), uma técnica que tem ganhado grande destaque nos últimos anos. Observe que optou-se por empregar os nomes dos métodos em inglês, o que evitaria a tradução incoerente destes para o português.

3.6.1 “Clonal Algorithm” - CLONALG

O “Clonal Algorithm” (CLONALG) é um método bio-inspirado da teoria da seleção clonal, e foi desenvolvido inicialmente com o intuito de realizar tarefas de aprendizagem e reconhecimento de padrões. Entretanto, visto o desempenho alcançado, este método foi posteriormente empregado na otimização de problemas, principalmente na otimização combinatória e multimodal. Uma descrição detalhada deste método, bem como exemplos de aplicações, são encontrados em (de Castro & Von Zuben 2002).

Este método possui codificação binária, e cada variável de otimização é representada por meio de $L = 64$ bits. Assim sendo, considerando um problema definido por n parâmetros de otimização, cada solução estimada possuirá um tamanho igual a nL bits.

Assim como observado nos algoritmos evolucionários, o CLONALG inicia-se espalhando N_{pop} pontos (anticorpos) sobre o espaço de busca do problema de otimização, sendo os mesmos gerados aleatoriamente segundo uma função densidade de probabilidade uniforme. Estes anticorpos são, então, avaliados na função objetivo e classificados em ordem decrescente de afinidade, isto é, os melhores pontos são postos no início do vetor população.

Conforme explicado pela teoria da imunologia clássica (Goldsby et al. 2000, Abbas & Lichtman March 2005), apenas os anticorpos de maior afinidade são selecionados para estabelecerem a linha de defesa do organismo. Dessa forma, dentre os pontos avaliados, somente os N_{sel} melhores são submetidos ao processo de expansão clonal.

Cada ponto selecionado recebe um número de clones específico N_C^i dado pela equação (3.10):

$$N_C^i = round\left(\frac{\beta N_{pop}}{i}\right) \quad (3.10)$$

em que β é o fator multiplicativo de clonagem, i representa a posição do anticorpo no vetor população ordenado, e a função $round(\cdot)$ arredonda o seu argumento para o inteiro mais próximo.

Cada clone imaturo, ainda com o mesmo valor de afinidade do anticorpo original, é então submetido ao mecanismo de maturação celular, em que os seus bits são alterados dada uma probabilidade de ocorrência de mutação α . O valor dessa taxa de mutação é controlado, o que possibilita exercer maiores alterações sobre os clones de baixa afinidade, além de não prejudicar a qualidade daqueles de elevada afinidade. A expressão de α é

mostrada em (3.11):

$$\alpha = \exp(-\rho \bar{f}) \quad (3.11)$$

em que $\bar{f} \in [0, 1]$ representa o valor normalizado da afinidade e ρ controla o decaimento da taxa de mutação. Desejando que o melhor indivíduo ($\bar{f} = 1$) apresente uma taxa de mutação igual $\alpha = 0.05$, escolheu-se $\rho = 3$. A Fig. 3.2 ilustra a curva da taxa de mutação em função do valor normalizado da afinidade.

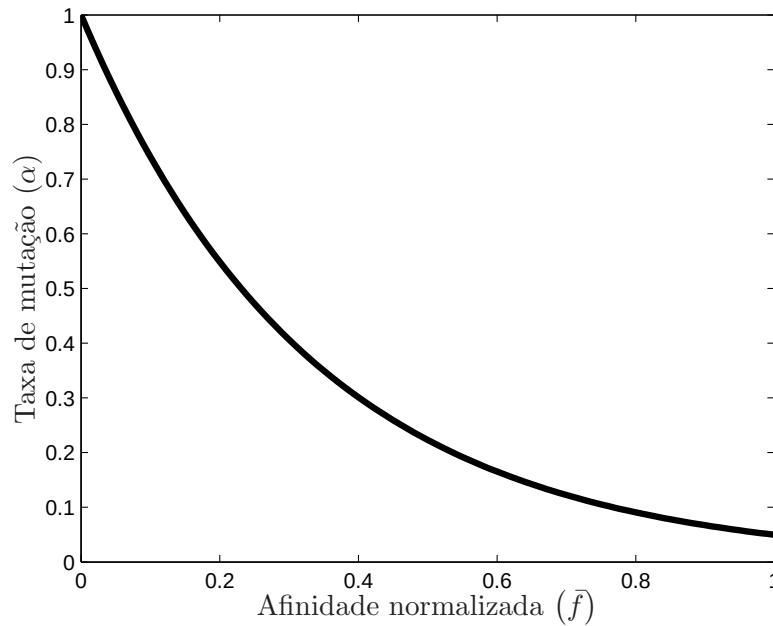


Figura 3.2: Taxa de mutação no CLONALG em função da afinidade normalizada ($\rho = 3$).

Uma vez que o processo de maturação celular foi concluído, é efetuada uma seleção elitista para determinar quais indivíduos se propagarão para a próxima geração. Essa seleção avalia todas as subpopulações (conjunto de soluções formado por cada anticorpo original e os seus clones maturados), armazenando apenas o melhor ponto de cada uma delas e eliminando os demais.

Buscando manter diversidade no espaço de busca, os piores N_{rep} anticorpos presentes na população corrente são substituídos por novos pontos gerados aleatoriamente. De forma geral, tem-se que $N_{sel} + N_{rep} < N_{pop}$, o que significa que parte da população não sofre alterações, possibilitando a esses indivíduos melhorarem o seu grau de afinidade nas gerações seguintes. O tamanho da população se mantém constante ao longo das gerações. Uma visão geral da distribuição da população está ilustrada na Fig. 3.3.

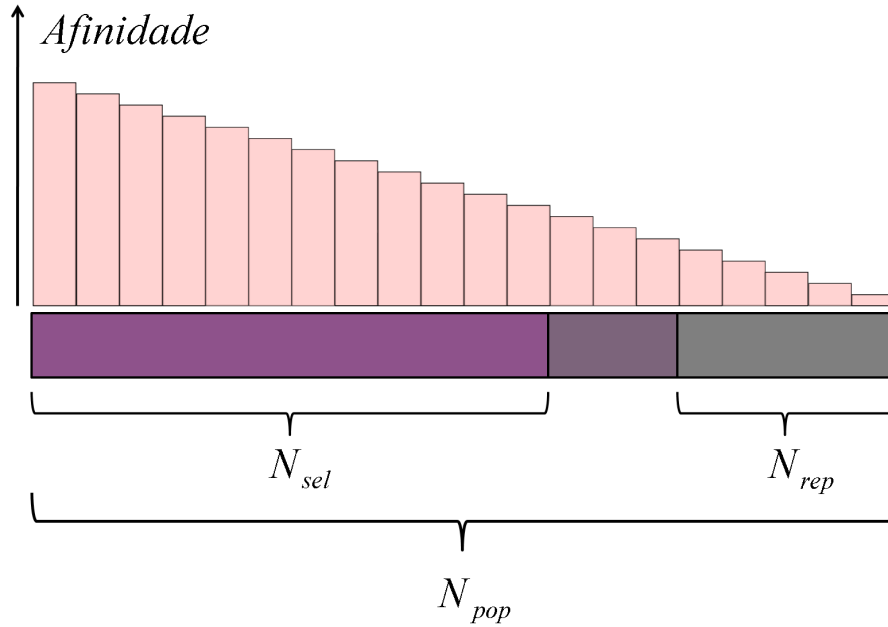


Figura 3.3: Distribuição da população no CLONALG.

O Alg. 3.2 mostra o ciclo de funcionamento do CLONALG implementado nesse trabalho.

Algoritmo 3.2: Estrutura de funcionamento do CLONALG.

Input: Objetivo, restrições, espaço de busca, N_{pop} , N_{sel} , N_{rep} , β , ρ , L

Output: Estimativa das melhores soluções

```

1 begin
2    $\mathcal{P}(n) \leftarrow$  População inicial;
3    $\Phi(n) \leftarrow$  Avaliação ( $\mathcal{P}(n)$ );
4   while Não critério de parada do
5      $\mathcal{S}(n) \leftarrow$  Seleção ( $\mathcal{P}(n)$ ,  $\Phi(n)$ ,  $N_{sel}$ );
6      $\mathcal{C}(n) \leftarrow$  Clonagem ( $\mathcal{S}(n)$ ,  $\beta$ );
7      $\mathcal{C}'(n) \leftarrow$  Maturação ( $\mathcal{C}(n)$ ,  $\rho$ );
8      $\Phi'(n) \leftarrow$  Avaliação ( $\mathcal{C}'(n)$ );
9      $\mathcal{S}(n) \leftarrow$  Seleção por subpopulação ( $\Phi(n)$ ,  $\Phi'(n)$ );
10     $\mathcal{R}(n) \leftarrow$  Geração de diversidade ( $N_{rep}$ );
11     $(\mathcal{P}(n+1), \Phi(n+1)) \leftarrow$  Atualização ( $\mathcal{P}(n)$ ,  $\mathcal{S}(n)$ ,  $\mathcal{R}(n)$ );
12     $n = n + 1$ ;
13  end
14 end

```

3.6.2 “B-Cell Algorithm” - BCA

O “B-Cell Algorithm” (BCA) assemelha-se ao CLONALG em diversos aspectos. Por exemplo, a representação empregada é exatamente a mesma, ou seja, cada variável de otimização é escrita por meio de uma cadeia binária de $L = 64$ bits. No entanto, uma importante característica que os distigue é que o BCA apresenta um mecanismo de mutação que atua de forma similar ao que se observa em cadeias de DNA mutantes. Visto que nas células de DNA as mutações ocorrem em regiões adjacentes, ao invés de em pontos isolados, este método propõe um operador chamado “contiguous somatic hypermutation” (CSH), e como forma de geração de diversidade é utilizado ainda um segundo operador nomeado “metadynamics”, os quais são descritos nas próximas linhas e detalhados em (Kelsey & Timmis 2003).

Após o espalhamento de N_{pop} células B geradas aleatoriamente sobre o espaço de busca, estas são avaliadas na função objetivo e armazenados os seus valores de afinidade. No BCA todas as células B são selecionadas para serem clonadas, as quais recebem o mesmo número de clones η_C , embora não seja necessário. Uma vez que o BCA apresenta maior velocidade de convergência com tamanhos pequenos de população, conforme Kelsey $N_{pop} \in [3, 5]$, o número de clones por célula é normalmente escolhido como sendo igual ao tamanho da população $\eta_C = N_{pop}$.

O mecanismo de maturação celular é desempenhado pelos operadores “metadynamics” e CSH. O primeiro deles é responsável pela manutenção de diversidade no espaço de busca. Assim sendo, toma-se uma única célula clonada de cada subpopulação, e cada bit desta é submetido a uma probabilidade de mutação ρ_{meta} .

Terminado esta etapa todos os clones são submetidos ao operador “contiguous somatic hypermutation”. Dessa forma, dado um clone escolhe-se um lote aleatório, de tamanho também aleatório $T \in [1, T_{max}]$, em que verifica-se a probabilidade de mutação ρ_{meta} de cada bit, até que todo o sítio seja percorrido. Esse esquema de mutação em regiões contíguas de uma cadeia de DNA está ilustrado na Fig. 3.4, onde é também exemplificada a ocorrência de mutações em pontos isolados dessa cadeia.

Finalmente os clones maturados são avaliados na função objetivo e a célula de melhor afinidade de cada subpopulação é selecionada para prosseguir para a próxima geração, enquanto as demais são eliminadas, mantendo assim uma população de tamanho constante.

O Alg. 3.3 apresenta o ciclo iterativo do BCA.

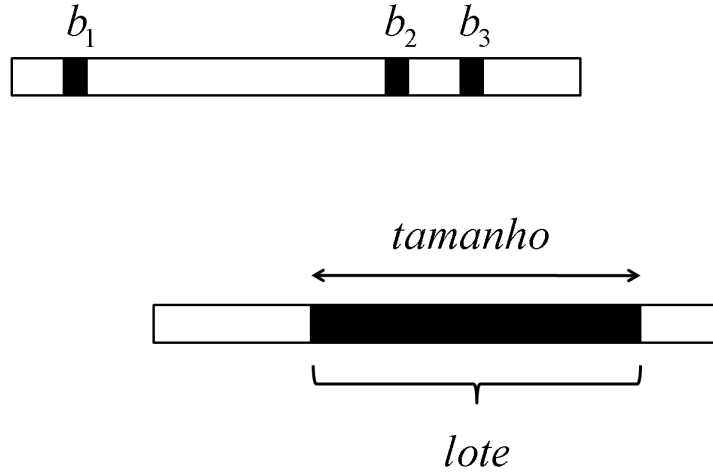


Figura 3.4: Mutações em pontos isolados e em regiões contíguas (CSH) - figura adaptada de (Kelsey & Timmis 2003).

Algoritmo 3.3: Estrutura de funcionamento do BCA.

Input: Objetivo, restrições, espaço de busca, N_{pop} , ρ_{meta} , η_C , L , T_{max}

Output: Estimativa das melhores soluções

```

1 begin
2    $\mathcal{P}(n) \leftarrow$  População inicial;
3    $\Phi(n) \leftarrow$  Avaliação ( $\mathcal{P}(n)$ );
4   while Não critério de parada do
5      $\mathcal{C}(n) \leftarrow$  Clonagem ( $\mathcal{P}(n)$ ,  $\Phi(n)$ ,  $\eta_C$ );
6      $\mathcal{C}'(n) \leftarrow$  "Metadynamics" ( $\mathcal{C}(n)$ ,  $\rho_{meta}$ );
7      $\mathcal{C}''(n) \leftarrow$  "Somatic Contiguous Hypermutation" ( $\mathcal{C}'(n)$ ,  $\rho_{meta}$ ,  $T_{max}$ );
8      $\Phi'(n) \leftarrow$  Avaliação ( $\mathcal{C}''(n)$ );
9      $\mathcal{S}(n) \leftarrow$  Seleção por subpopulação ( $\Phi(n)$ ,  $\Phi'(n)$ );
10    ( $\mathcal{P}(n+1)$ ,  $\Phi(n+1)$ )  $\leftarrow$  Atualização ( $\mathcal{S}(n)$ );
11     $n = n + 1$ ;
12  end
13 end

```

3.6.3 “Real-Coded Clonal Selection Algorithm” - RCSA

Tanto o CLONALG quanto o BCA possuem operadores de mutação que atuam sobre uma cadeia binária, o que requer a conversão em diversos momentos entre as representações binária e real, tornando estes métodos lentos e pouco eficientes para a otimização de problemas eletromagnéticos, e também problemas associados com a codificação binária (Takahashi et al. 2003). Buscando contornar essas ineficiências propôs-se o “Real-Coded Clonal Selection Algorithm” (RCSA), o qual além de apresentar codificação real dos parâmetros, possui excelente desempenho na otimização de dispositivos eletromagnéticos. Este algoritmo e suas peculiaridades são mostrados em (Campelo et al. 2005, Campelo 2006).

O RCSA inicia-se com a geração de uma população inicial de N_{pop} anticorpos, os quais são espalhados aleatoriamente sobre o espaço de busca. Estes pontos são então avaliados na função objetivo e, posteriormente, classificados em ordem decrescente do valor de afinidade.

Similar ao que se observa no CLONALG, os N_{sel} anticorpos de melhor afinidade são escolhidos para serem clonados, e cada um destes pontos recebe um número de cópias N_C^i proporcional a sua posição no vetor população. A expressão para o cálculo deste número de cópias, evidenciada em (3.12), é idêntica à adotada no CLONALG.

$$N_C^i = round \left(\frac{\beta N_{pop}}{i} \right) \quad (3.12)$$

Cada clone é então submetido ao processo de maturação celular, o qual consiste na adição de um ruído normal (Gaussiano) em pelo menos um dos parâmetros de otimização. De forma geral, o modelo matemático deste mecanismo de maturação é dado pela equação (3.13):

$$\begin{aligned} x_k^{mut} &= x_k + \nu_k \\ \nu_k &= \alpha \cdot S_k \cdot P \end{aligned} \quad (3.13)$$

em que ν_k é o ruído adicionado à k -ésima variável; $\alpha \in [0, 1]$ representa o tamanho da perturbação; S_k é a diferença entre os limites superior e inferior da respectiva variável ($ub_k - lb_k$); e P representa uma função densidade de probabilidade Gaussiana ($N(0, 1)$), com média zero e desvio padrão unitário.

A adição deste ruído normal desempenha uma busca local em torno do indivíduo mutado, o que permite um refinamento da solução nessa vizinhança. Observe que o raio desta vizinhança é determinado pela constante α . O anticorpo original e os seus clones maturados são então chamados de subpopulação.

Os clones maturados são finalmente avaliados na função objetivo, e somente o melhor de cada subpopulação é escolhido para prosseguir para a próxima geração. Com o intuito de manter diversidade e explorar novas áreas no espaço de busca, os anticorpos não selecionados para serem clonados, aqueles de baixa afinidade, são eliminados e substituídos por novos pontos gerados aleatoriamente, o que garante o tamanho fixo da população.

Diferente do que se observa no CLONALG, em que uma parte da população segue inalterada para a geração seguinte, o RCSA apresenta uma distribuição da população conforme mostrado na Fig. 3.5, onde $N_{sel} + N_{rep} = N_{pop}$.

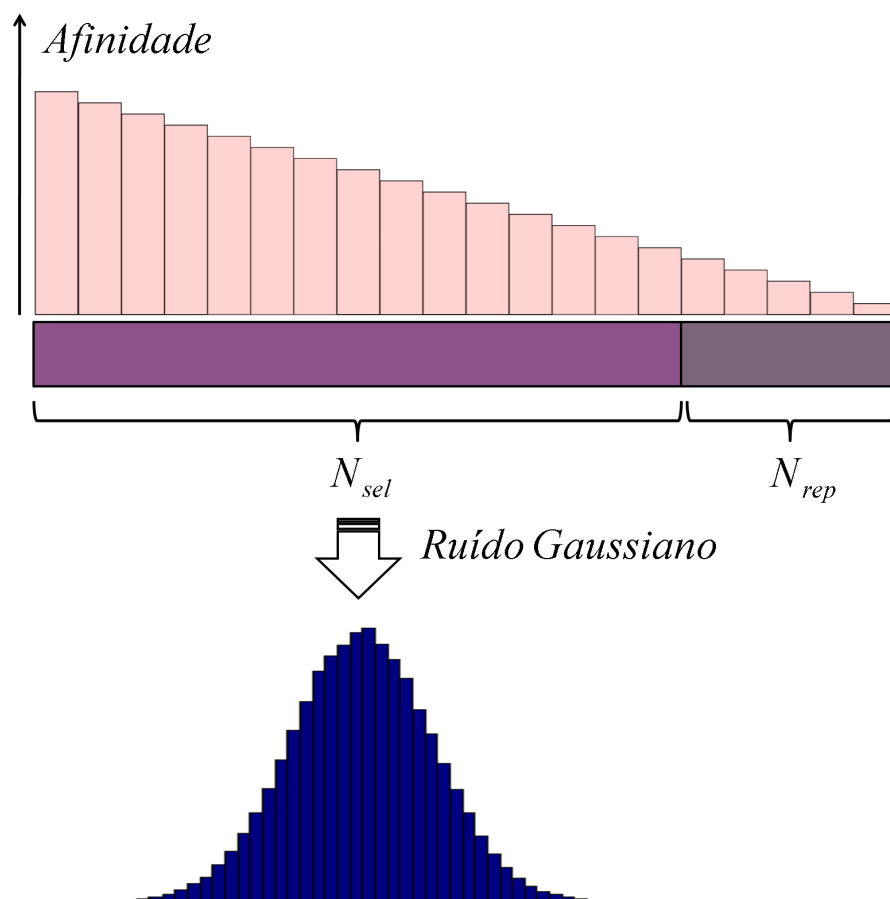


Figura 3.5: Distribuição da população no RCSA.

A estrutura cíclica de funcionamento do RCSA é apresentada no Alg. 3.4.

Algoritmo 3.4: Estrutura de funcionamento do RCSA.

Input: Objetivo, restrições, espaço de busca, N_{pop} , N_{sel} , β
Output: Estimativa das melhores soluções

```

1 begin
2    $\mathcal{P}(n) \leftarrow$  População inicial;
3    $\Phi(n) \leftarrow$  Avaliação ( $\mathcal{P}(n)$ );
4   while Não critério de parada do
5      $\mathcal{S}(n) \leftarrow$  Seleção ( $\mathcal{P}(n)$ ,  $\Phi(n)$ ,  $N_{sel}$ );
6      $\mathcal{C}(n) \leftarrow$  Clonagem ( $\mathcal{S}(n)$ ,  $\beta$ );
7      $\mathcal{C}'(n) \leftarrow$  Maturação ( $\mathcal{C}(n)$ ,  $\rho$ );
8      $\Phi'(n) \leftarrow$  Avaliação ( $\mathcal{C}'(n)$ );
9      $\mathcal{S}(n) \leftarrow$  Seleção por subpopulação ( $\Phi(n)$ ,  $\Phi'(n)$ );
10     $\mathcal{R}(n) \leftarrow$  Geração de diversidade ( $N_{rep}$ );
11     $(\mathcal{P}(n+1), \Phi(n+1)) \leftarrow$  Atualização ( $\mathcal{S}(n)$ ,  $\mathcal{R}(n)$ );
12     $n = n + 1$ ;
13  end
14 end

```

3.6.4 “Real-Biased Genetic Algorithm” - RBGA

O “Real-Biased Genetic Algorithm” (RBGA) adota uma representação real dos seus parâmetros de otimização, e implementa o que se convencionou chamar *cruzamento real polarizado*, o qual produz dois novos indivíduos como resultado da operação de cruzamento entre dois indivíduos pais. O nome *polarizado* é devido ao fato de que um dos indivíduos gerados tem maior probabilidade de se localizar nas proximidades do ancestral de melhor valor de aptidão, enquanto o segundo possui probabilidade uniforme de se localizar entre os indivíduos pais. Um estudo detalhado desse operador de cruzamento real-polarizado é apresentado em (Takahashi et al. 2003, Takahashi 2004).

Assim como os demais algoritmos mencionados, o RBGA inicia-se com a geração aleatória de N_{pop} indivíduos sobre o espaço de busca do problema de otimização. Esses pontos são então avaliados na função objetivo e submetidos a um mecanismo de seleção.

Nesse trabalho optou-se pela *seleção por roleta*, em que os indivíduos ocupam fatias correspondentes aos seus valores de aptidão em uma roleta. Dessa forma, escolhe-se pontos aleatórios nessa roleta até que toda a população tenha se formado, ou seja, até atingir N_{pop} indivíduos. De forma geral, observa-se que os pontos de maior aptidão possuem maiores chances de serem selecionados, propagando assim os melhores resultados

encontrados para a próxima geração. Como a seleção de N_{pop} pontos se dá sobre uma população de mesmo tamanho, então, torna-se possível e comum a repetição de alguns indivíduos, que normalmente são os melhores. No entanto, esse fato não compromete o desempenho do método.

Antes que se inicie o processo de variação, efetua-se ainda uma seleção elitista, que tem como propósito armazenar as melhores soluções encontradas até o momento. Assim sendo, caso estas soluções não sejam selecionadas para comporem a próxima geração, elas são deterministicamente incluídas na população, substituindo indivíduos escolhidos aleatoriamente. Nesse trabalho considerou-se o elitismo apenas da melhor solução.

O mecanismo de variação inicia-se dividindo a população em duas partes, com o objetivo de formar grupos de dois indivíduos para a etapa de cruzamento. O cruzamento entre dois indivíduos pais ocorre somente se verificada a probabilidade ρ_{cruz} . No caso positivo são gerados dois novos indivíduos segundo a equação (3.14):

$$\begin{aligned}\vec{x}_g &= \alpha \vec{x}_1 + (1 - \alpha) \vec{x}_2 \\ -0.1 &\leq \alpha \leq 1.1\end{aligned}\tag{3.14}$$

em que $\vec{x}_g$ é o novo indivíduo gerado, $\vec{x}_1$ e $\vec{x}_2$ os ancestrais, e a aptidão de $\vec{x}_2$ deve ser melhor do que a de $\vec{x}_1$.

Para a escolha do valor de α , verifica-se se o cruzamento será ou não polarizado dada a probabilidade $\rho_{cruz-pol}$. Dessa forma, se o cruzamento não for polarizado, os indivíduos filhos são gerados adotando-se α com distribuição uniforme de probabilidade dentro do intervalo de valores possíveis. Caso contrário, apenas um dos indivíduos filhos é gerado conforme citado, e para o segundo adota-se (3.15):

$$\alpha = 1.4\beta_1\beta_2 - 0.2\tag{3.15}$$

em que $\beta_1, \beta_2 \in [0, 1]$ são escolhidos aleatoriamente com distribuição de probabilidade uniforme.

Os indivíduos filhos são sempre gerados sobre o segmento de reta que une $\vec{x}_1$ e $\vec{x}_2$, considerando-se uma extrapolação deste segmento de 10% em ambas as extremidades. A Fig. 3.6 ilustra esta operação.

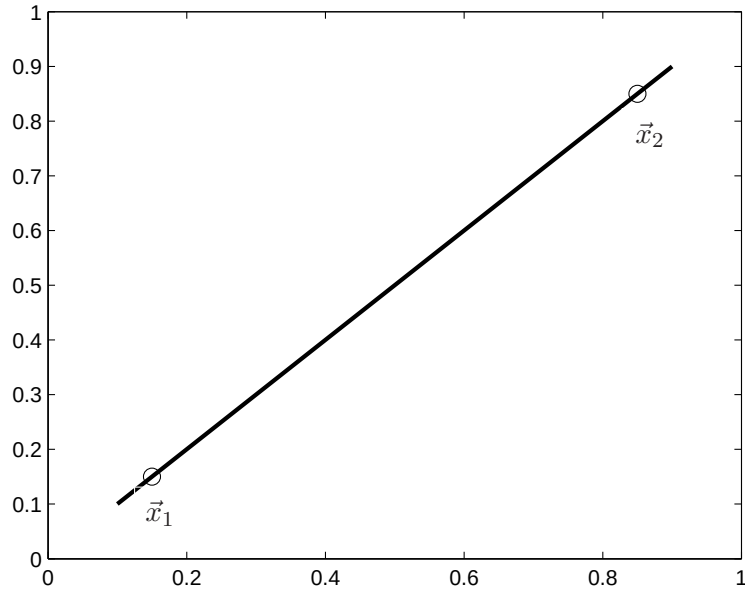


Figura 3.6: Operação de cruzamento real-polarizado.

O processo de mutação consiste na adição de um ruído Gaussiano em cada um dos indivíduos da população, dada a probabilidade ρ_{mut} . Então, caso essa probabilidade seja observada para um dado ponto $\vec{x}$, faz-se conforme mostrado em (3.16):

$$\begin{aligned} x_k^{mut} &= x_k + \nu_k \\ \nu_k &= 0.05\beta_k S_k \end{aligned} \tag{3.16}$$

em que ν_k é o ruído adicionado à k -ésima variável; β_k é um número aleatório com distribuição Gaussiana, média zero e variância unitária; e S_k é a diferença entre os limites superior e inferior da respectiva variável ($ub_k - lb_k$).

Finalmente, o contador de gerações é incrementado e o ciclo se repete até ser alcançado algum critério de parada. Para a função de ajuste adotou-se $\gamma = 1.8$, conforme recomendado em (Takahashi 2004).

O Alg. 3.5 ilustra a estrutura básica do RBGA implementado nesse trabalho.

Algoritmo 3.5: Estrutura de funcionamento do RBGA.

Input: Objetivo, restrições, espaço de busca, N_{pop} , ρ_{cruz} , $\rho_{cruz-pol}$, ρ_{mut}
Output: Estimativa da melhor solução

```

1 begin
2    $\mathcal{P}(n) \leftarrow$  População inicial;
3   while Não critério de parada do
4      $\Phi(n) \leftarrow$  Avaliação ( $\mathcal{P}(n)$ );
5      $\mathcal{E}(n) \leftarrow$  Elitismo ( $\mathcal{P}(n), \Phi(n)$ );
6      $\mathcal{S}(n) \leftarrow$  Seleção por roleta ( $\mathcal{P}(n), \Phi(n)$ );
7      $\mathcal{Q}(n) \leftarrow$  Cruzamento ( $\mathcal{S}(n), \rho_{cruz}, \rho_{cruz-pol}$ );
8      $\mathcal{Q}'(n) \leftarrow$  Mutação ( $\mathcal{Q}(n), \rho_{mut}$ );
9      $\mathcal{P}(n+1) \leftarrow$  Atualização ( $\mathcal{Q}'(n)$ );
10     $n = n + 1$ ;
11  end
12 end

```

3.6.5 “Differential Evolution Algorithm” - DEA

Quando se propõe alguma técnica de otimização, esta deve ser capaz de satisfazer quatro requisitos fundamentais, os quais são:

1. Habilidade para lidar com funções não-diferenciáveis, não-lineares e multimodais;
2. Desempenho altamente paralelo;
3. Reduzido número de parâmetros de controle e fácil ajuste destes; e
4. Capacidade de convergência para o ótimo do problema.

Alguns estudos sobre *evolução diferencial* surgiram em 1990, mas tornou-se popular somente com a publicação de (Storn & Price 1997), onde os autores mostram a robustez deste método em relação aos quatro tópicos mencionados anteriormente.

Além de apresentar uma implementação simples e de fácil entendimento, este algoritmo possui poucos parâmetros de ajuste, o que é um contraste diante da peculiar velocidade de convergência desempenhada e alta qualidade das soluções retornadas. O princípio de funcionamento desse método é descrito a seguir, sendo melhor caracterizado em (Storn & Price 1997, Price et al. 2005).

O “Differential Evolution Algorithm” (DEA) inicia-se com a geração aleatória, considerando-se uma função densidade de probabilidade uniforme, de $\vec{x}_{i,G}$, $i = 1, \dots, N$ vetores sobre a região de busca do problema de otimização, em que N representa o

tamanho da população e G a geração corrente. Estes pontos são então avaliados na função objetivo e os seus valores de custo são armazenados.

O processo de variação tem início com a etapa de mutação, em que cada vetor alvo $\vec{x}_{i,G}$, $i = 1, \dots, N$ terá um vetor mutante associado, obtido pela equação (3.17):

$$\begin{aligned} \vec{v}_{i,G} &= \vec{x}_{r1,G} + \omega (\vec{x}_{r2,G} - \vec{x}_{r3,G}) \\ i &= 1, \dots, N \end{aligned} \quad (3.17)$$

sendo $r1, r2, r3 \in [1, N]$ índices aleatórios e mutuamente distintos, e $\omega \in [0, 2]$ um fator real, constante ou variável, que controla o tamanho do vetor diferencial $(\vec{x}_{r2,G} - \vec{x}_{r3,G})$.

A etapa de cruzamento é realizada com o objetivo de aumentar a diversidade dos pontos mutados $\vec{v}_{i,G}$. Matematicamente, tem-se que o novo ponto gerado $\vec{u}_{i,G}$, chamado vetor teste, é obtido como evidenciado na equação (3.18):

$$\begin{aligned} u_{ji,G} &= \begin{cases} v_{ji,G} & \text{se } (\beta_j \leq CR) \parallel (j = \delta_i) \\ x_{ji,G} & \text{se } (\beta_j > CR) \& (j \neq \delta_i) \end{cases} \\ j &= 1, \dots, n \end{aligned} \quad (3.18)$$

em que n é o número de parâmetros de otimização (dimensão do problema), $\beta_j \in [0, 1]$ é um real aleatório obtido de uma distribuição uniforme para a variável de otimização j , $\delta_i \in [1, n]$ é um índice aleatório sorteado para o vetor teste i , e $CR \in [0, 1]$ representa uma constante de cruzamento escolhida pelo usuário. Como em algum momento a igualdade $j = \delta_i$ será verificada, isto garante que pelo menos um dos parâmetros do vetor teste ($\vec{u}_{i,G}$) seja herdado do vetor mutante ($\vec{v}_{i,G}$).

Observe que o vetor mutante ($\vec{v}_{i,G}$) representa apenas um caminho por meio do qual o vetor teste ($\vec{u}_{i,G}$) é gerado. Dessa forma, o processo de seleção se realiza entre um vetor alvo predeterminado ($\vec{x}_{i,G}$) e o vetor teste correspondente, selecionando dentre estes o que possuir o menor custo. A Fig. 3.7 mostra o processo de variação implementado pelo DEA.

Finalmente, o Alg. 3.6 ilustra o ciclo de funcionamento do DEA.

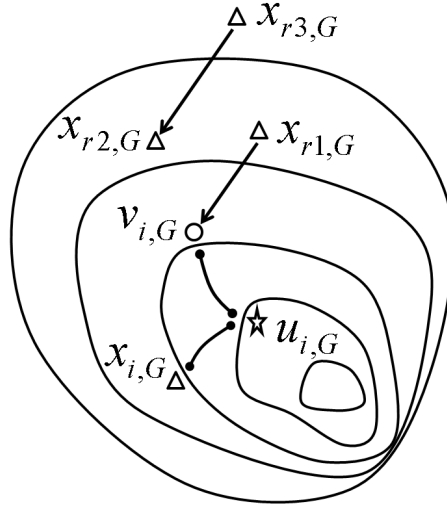


Figura 3.7: Sistema de variação diferencial implementado no DEA.

Algoritmo 3.6: Estrutura de funcionamento do DEA.

Input: Objetivo, restrições, espaço de busca, N , ω , CR

Output: Estimativa da melhor solução

```

1 begin
2    $\mathcal{P}(n) \leftarrow$  População inicial;
3    $\Phi(n) \leftarrow$  Avaliação ( $\mathcal{P}(n)$ );
4   while Não critério de parada do
5      $\mathcal{D}(n) \leftarrow$  Mutação ( $\mathcal{P}(n), \omega$ );
6      $\mathcal{D}'(n) \leftarrow$  Cruzamento ( $\mathcal{P}(n), \mathcal{D}(n), CR$ );
7      $\Phi'(n) \leftarrow$  Avaliação ( $\mathcal{D}'(n)$ );
8      $\mathcal{S}(n) \leftarrow$  Seleção ( $\Phi(n), \Phi'(n)$ );
9      $(\mathcal{P}(n+1), \Phi(n+1)) \leftarrow$  Atualização ( $\mathcal{S}(n)$ );
10     $n = n + 1$ ;
11  end
12 end

```

3.6.6 Outras Vertentes

Embora apenas cinco métodos populares da área de otimização tenham sido tratados nos itens anteriores, várias outras classes de algoritmos possuem fundamental importância nessa linha de pesquisa, dentre as quais se destacam:

- “Simulated Annealing” (SA)
Inicialmente proposto por (Kirkpatrick et al. 1983), essa técnica faz analogia ao processo metalúrgico de resfriamento lento de metais, permitindo aos átomos desse material encontrar níveis mais estáveis de energia;
- “Ant Colony Optimization” (ACO)
Proposto por (Coloni et al. 1992), originou-se da observação de formigas à procura de alimento, onde estas mostram-se hábeis na determinação do caminho mais curto entre a comida encontrada e a colônia;
- “Particle Swarm Optimization” (PSO)
Descrito pela primeira vez em (Kennedy & Eberhart 1995), baseia-se em princípios da psicologia social, mais especificamente, essa técnica simula uma percepção do comportamento humano.

Como pode ser observado, são várias as técnicas de otimização empregadas atualmente, entretanto, nos últimos anos tem-se percebido a necessidade de sugestões para a implementação de novos operadores que incrementem as qualidades dos métodos tradicionais, seja melhorando as características das soluções encontradas ou aumentando a velocidade de convergência para o ótimo. Este é também um assunto a ser tratado nessa dissertação.

3.7 Conclusão

Ao longo do capítulo corrente realizou-se uma explanação envolvendo sistemas de otimização mono-objetivo, onde discutiu-se, principalmente, ideias fundamentais acerca do funcionamento de métodos imunes. Além disso, foram abordadas técnicas clássicas relacionadas a algoritmos genéticos e algoritmos de evolução diferencial.

Conforme apresentado nos tópicos anteriores, os métodos estudados expressam diferentes características funcionais, dentre as quais pode-se citar, por exemplo, o tipo de codificação adotado, as técnicas de mutação e, ou, cruzamento, normalmente inspiradas

a partir de diferentes fenômenos naturais, e também algumas peculiaridades observadas durante o processo de otimização de um problema específico, tais como, capacidade de busca local, global, e velocidade de convergência. Todos esses fatores são de grande relevância para o desempenho do método, e sendo assim, serão considerados quando da implementação do algoritmo a ser proposto nesta dissertação.

Como esse capítulo abordou apenas sistemas mono-objetivo, e pretende-se investigar também sistemas multi-objetivo, o capítulo seguinte apresentará a base necessária para a otimização de problemas com múltiplos objetivos, e também alguns métodos de grande relevância na literatura.

Capítulo 4

Otimização Evolucionária Multi-Objetivo

“Toda arte nada mais é do que a imitação da natureza.”

— Lucius Annaeus Seneca, 4a.C.–65d.C. (filósofo latino)

“A arte e a ciência têm o seu ponto de encontro no método.”

— Edward Bulwer-Lytton, 1803–1873 (escritor inglês)

4.1 Introdução

Neste capítulo são tratados assuntos referentes à otimização evolucionária multi-objetivo. Tendo isso em vista, define-se inicialmente o problema de otimização multi-objetivo, apresenta-se uma extensão para condições necessárias de otimalidade para eficiência, estrutura-se um modelo geral dessa classe de algoritmos, e, posteriormente, são relatados alguns métodos de renome na literatura, destacando suas principais características de otimização.

4.2 Definição do Problema de Otimização

O problema de otimização multi-objetivo pode ser definido como:

$$\begin{aligned} \mathcal{X}^* = \arg \min_x f(\vec{x}) &\triangleq \{f_1(\vec{x}), \dots, f_m(\vec{x})\} \\ \text{sujeito a: } \vec{x} &\in \Omega \end{aligned} \quad (4.1)$$

em que $\mathcal{X}^*$ representa o *conjunto Pareto-ótimo*, e Ω é a região factível, matematicamente definida por:

$$\Omega = \begin{cases} g_i(\vec{x}) \leq 0; & i = 1, \dots, p \\ h_j(\vec{x}) = 0; & j = 1, \dots, q \end{cases} \quad (4.2)$$

O espaço de busca é descrito pelo conjunto $\mathcal{X}$:

$$\mathcal{X} = \begin{cases} lb_k \leq x_k \leq ub_k \\ k = 1, \dots, n \end{cases} \quad (4.3)$$

em que n representa o número de parâmetros do problema e lb_k, ub_k são os limites inferior e superior da k -ésima variável de otimização, respectivamente. Além disso, segue que $\vec{x} \in \mathbb{R}^n$, $g(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^p$, $h(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^q$, e $f(\cdot) : \mathcal{X} \rightarrow \mathcal{Y}$ mapeia o espaço de parâmetros $\mathcal{X} \subset \mathbb{R}^n$ dentro do espaço de objetivos $\mathcal{Y} \subset \mathbb{R}^m$.

As funções $g_i(\vec{x})$ e $h_j(\vec{x})$ representam as restrições de desigualdade e igualdade, respectivamente. A função vetorial $f(\vec{x})$ compõe os objetivos do problema de otimização, e o conjunto solução $\mathcal{X}^* \in \Omega$ definido no espaço de parâmetros é representado pelos pontos $x^* \in \mathcal{X}^*$, cuja relação de compromisso custo/benefício diante dos objetivos encontra-se estável, não sendo possível melhorar nenhum critério de otimização sem tornar algum outro pior.

Algumas definições importantes acerca do conjunto Pareto-ótimo são dadas a seguir (Takahashi 2004):

Definição 4.1. (Dominância) Supondo $\vec{x}_1, \vec{x}_2 \in \mathcal{X}$ dois pontos do espaço de parâmetros, diz-se que $\vec{x}_1$ domina $\vec{x}_2$ se $f(\vec{x}_1) \leq f(\vec{x}_2)$ e $f(\vec{x}_1) \neq f(\vec{x}_2)$. Equivalentemente, diz-se que $f(\vec{x}_1) \in \mathcal{Y}$ domina $f(\vec{x}_2) \in \mathcal{Y}$ no espaço de objetivos. Essa relação de dominância é comumente escrita como $\vec{x}_1 \prec \vec{x}_2$ ou $f(\vec{x}_1) \prec f(\vec{x}_2)$. $\square$

Definição 4.2. (Solução Pareto-ótima) Suponto x^* um ponto do espaço de parâmetros, diz-se que $x^* \in \mathcal{X}^*$ é uma solução Pareto-ótima do problema multi-objetivo se não existe qualquer outra solução factível $\vec{x} \in \Omega$ do espaço de parâmetros, tal que $f(\vec{x}) \leq f(x^*)$ e $f(\vec{x}) \neq f(x^*)$, ou seja, x^* não é dominado por nenhum outro ponto factível. $\square$

De forma similar às definições de localidade e globalidade em problemas de otimização mono-objetivo, são apresentadas a seguir as definições para a formulação multi-objetivo de soluções Pareto-ótima local e global.

Definição 4.3. (Solução Pareto-ótima local) Suponto x^* um ponto factível do espaço de parâmetros, diz-se que $x^* \in \mathcal{X}^*$ é uma solução localmente Pareto-ótima do problema multi-objetivo em uma dada vizinhança $\mathcal{V}_\epsilon(x^*)$ definida por um raio $\epsilon > 0$, se não existe qualquer outra solução $\vec{x} \in \mathcal{V}_\epsilon(x^*)$ que satisfaça $f(\vec{x}) \leq f(x^*)$ e $f(\vec{x}) \neq f(x^*)$, ou seja, se x^* não é dominado por nenhum outro ponto dessa vizinhança. $\square$

Definição 4.4. (Solução Pareto-ótima global) Suponto x^* um ponto factível do espaço de parâmetros, diz-se que $x^* \in \mathcal{X}^*$ é uma solução globalmente Pareto-ótima do problema multi-objetivo se não existe nenhum outro ponto factível $\vec{x} \in \Omega$, tal que $f(\vec{x}) \leq f(x^*)$. Além disso, uma solução Pareto-ótima global é também uma solução Pareto-ótima local. $\square$

Finalmente, com base nas definições citadas anteriormente escreve-se matematicamente o conjunto Pareto-ótimo $\mathcal{X}^*$ para o problema geral de otimização multi-objetivo:

$$\mathcal{X}^* = \{x^* \in \Omega : \nexists \vec{x} \in \Omega \mid f(\vec{x}) \leq f(x^*) \text{ e } f(\vec{x}) \neq f(x^*)\} \quad (4.4)$$

Note que quando se trata de problemas com variáveis reais, a cardinalidade do conjunto $\mathcal{X}^*$ é igual a infinito ($|\mathcal{X}^*| = \infty$), sendo limitada apenas na otimização combinatoria ($|\mathcal{X}^*| < \infty$).

4.3 Condições de Otimalidade

As condições de otimalidade descritas por Kuhn-Tucker podem ser estendidas para problemas multi-objetivo, fornecendo as condições necessárias de Kuhn-Tucker para eficiência (Takahashi 2004). A proposição seguinte expressa essas condições.

Proposição 4.1. (*Condições necessárias de Kuhn-Tucker para eficiência*)

Uma solução factível $x^* \in \mathcal{X}^*$ satisfaz as condições necessárias de Kuhn-Tucker para eficiência se a equação (4.5) é atendida e, além disso, devem existir $\nu^* \geq 0$, $\mu^* \geq 0$ e $\lambda^* \geq 0$, com pelo menos uma desigualdade estrita $\nu_l^* > 0$. $\square$

$$\begin{aligned} \sum_{l=1}^m \nu_l \nabla f_l(x^*) + \sum_{i=1}^p \mu_i \nabla g_i(x^*) + \sum_{j=1}^q \lambda_j \nabla h_j(x^*) &= 0 \\ g_i(x^*) &\leq 0, \quad \mu_i g_i(x^*) = 0, \quad i = 1, \dots, p \\ h_j(\vec{x}) &= 0, \quad j = 1, \dots, q \end{aligned} \tag{4.5}$$

A Fig. 4.1 mostra uma interpretação geométrica para a condição de Kuhn-Tucker considerando-se uma restrição de desigualdade e dois objetivos. Observe que no ponto de solução x^* essa restrição está ativa, e que é possível determinar os valores dos multiplicadores ν_1^* , ν_2^* e μ^* positivos, tal que a soma dos vetores gradiente em x^* se anule.

4.4 Estrutura Geral de um AE Multi-Objetivo

Eficiência Pareto, ou *otimalidade Pareto*, é um conceito importante em economia, com larga aplicação nas engenharias e ciências sociais. Este termo originou-se após os estudos de Vilfredo Pareto (1848-1923), um renomado economista italiano, e pioneiro em otimização com múltiplos objetivos.

Em poucas palavras, dado um certo conjunto de pontos $\mathcal{C}$ representado no espaço de objetivos, o conjunto Pareto-ótimo associado $\mathcal{X}^*$ é definido pelos pontos não-dominados daquele conjunto. Observe que qualquer ponto de Pareto deve ser melhor que todos os outros em pelo menos um objetivo, e além disso, não deve ser dominado por nenhum outro, embora não precise dominar algum.

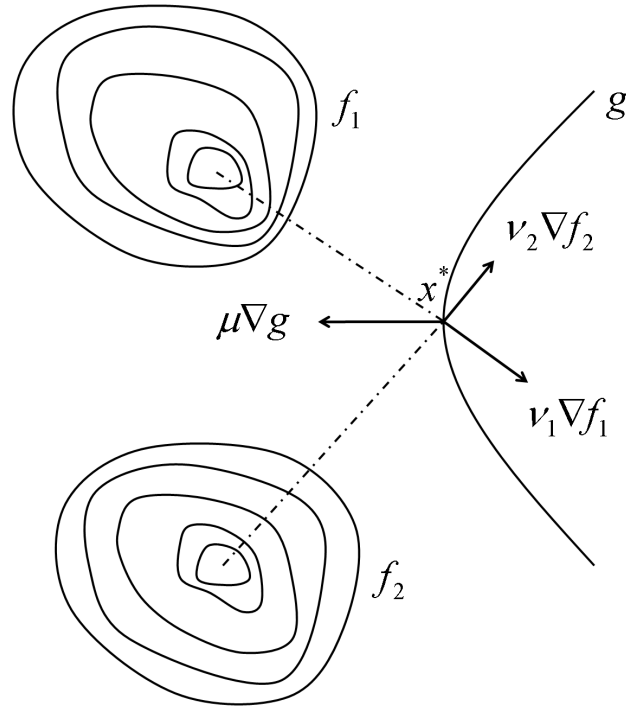


Figura 4.1: Ilustração das condições de Kuhn-Tucker para o caso multi-objetivo.

Na otimização multi-objetivo, além da importância em se conhecer o conjunto Pareto-ótimo, torna-se indispensável a determinação dos conjuntos de fronteiras $\mathcal{F}_i^1$, o que possibilita classificar os pontos em $\mathcal{C}$, atribuindo um valor de aptidão aos mesmos, com base na fronteira a qual pertencem. Dessa forma, os pontos da primeira fronteira $\mathcal{F}_1$ são ditos melhores do que os da segunda $\mathcal{F}_2$, e por sua vez estes são melhores do que os da terceira $\mathcal{F}_3$, e assim por diante. A Fig. 4.2 ilustra a distribuição de um conjunto de pontos em fronteiras, sendo que a primeira delas representa o Pareto-ótimo estimado dada a tarefa de minimização dos objetivos f_1 e f_2 . O Pareto-ótimo real $\mathcal{X}^*$ está representado pela curva contínua indicada.

O Alg. 4.1 apresenta a estrutura básica de um algoritmo evolucionário multi-objetivo com população de memória. De forma geral, após a inicialização aleatória de uma população de pontos, e a criação de um arquivo externo, esses pontos são avaliados nos objetivos e posteriormente classificados em fronteiras, possibilitando aplicar os mecanismos de seleção e variação (cruzamento e, ou, mutação), gerando assim a nova população. O arquivo externo é então atualizado, e o processo se repete até que seja verificado al-

¹ $\mathcal{F}_1$ representa o Pareto-ótimo estimado, e $\mathcal{F}_i$, $\forall i > 1$, corresponde ao melhor conjunto de pontos na ausência das fronteiras que a antecedem.

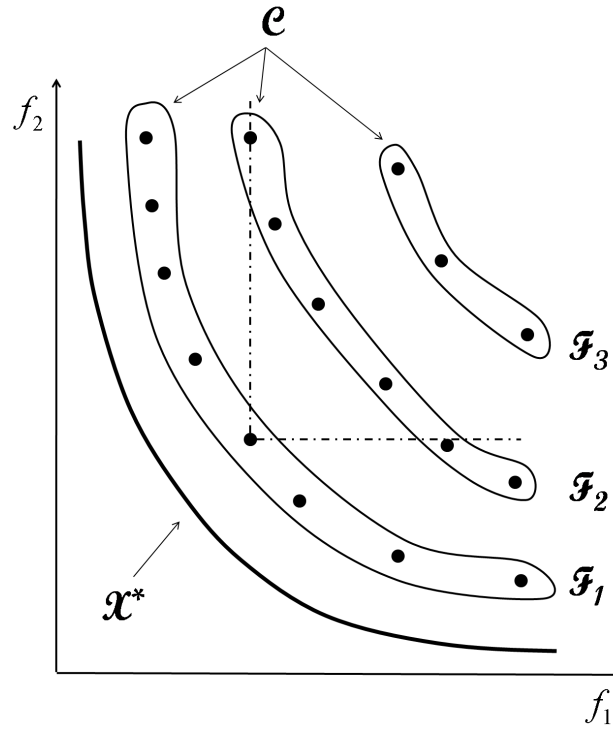


Figura 4.2: Ilustração dos conjuntos de fronteiras em um problema bi-objetivo.

um critério de parada especificado. Nesse ponto, o algoritmo retorna o Pareto-ótimo estimado, armazenado na população de arquivo.

Algoritmo 4.1: Ciclo básico de funcionamento de um AE multi-objetivo.

Input: Objetivos, restrições, espaço de busca, parâmetros

Output: Estimativa do conjunto Pareto-ótimo $\mathcal{A}$

```

1 begin
2    $\mathcal{P}(n) = \{\vec{p}_1, \dots, \vec{p}_N\} \leftarrow$  População inicial;
3    $\mathcal{A}(n) = \emptyset \leftarrow$  Arquivo inicial;           /* armazena as melhores soluções */
4   while Não critério de parada do
5      $\Phi(n) \leftarrow$  Avaliação ( $\mathcal{P}(n)$ );
6      $\mathcal{F}(n) \leftarrow$  Classificação ( $\Phi(n)$ );
7      $\mathcal{S}(n) \leftarrow$  Seleção ( $\mathcal{F}(n)$ );
8      $\mathcal{P}(n+1) \leftarrow$  Variação ( $\mathcal{S}(n)$ );
9      $\mathcal{A}(n+1) \leftarrow$  Atualização ( $\mathcal{A}(n) \cup \mathcal{P}(n)$ );
10     $n = n + 1$ ;
11  end
12 end

```

4.5 Sistemas Evolucionários Multi-Objetivo

Nessa seção são apresentadas algumas técnicas multi-objetivo consideradas importantes pela comunidade acadêmica envolvida na área da otimização computacional. Dentre estas, estão presentes três métodos baseados na teoria da evolução de Charles Darwin, os quais são “Non-Dominated Sorting Genetic Algorithm” (NSGA-II) (Deb et al. 2000, Deb et al. 2002), “Strength Pareto Evolutionary Algorithm” (SPEA-II) (Zitzler et al. 2001) e “Pareto Envelope-based Selection Algorithm” (PESA) (Corne et al. 2000). Além desses, são discutidos outros dois métodos, sendo o primeiro inspirado a partir do princípio da seleção clonal, “Multi-Objective Clonal Selection Algorithm” (MOCSA) (Guimarães et al. 2007), e o segundo baseia-se em uma evolução diferencial, “Multi-Objective Differential Algorithm” (MODE) (Xue et al. 2003b, Xue et al. 2005). Embora apenas uma dessas técnicas seja empregada durante a análise dos resultados, considerou-se importante a apresentação das mesmas, visto a sua contribuição na elaboração de trabalhos futuros.

4.5.1 “Non-Dominated Sorting Genetic Algorithm” - NSGA-II

Um dos primeiros algoritmos evolucionários multi-objetivo foi proposto por (Srinivas & Deb 1994), o qual é chamado “Non-Dominated Sorting Genetic Algorithm” (NSGA). Embora este método apresentasse muitas vantagens em relação aos que o precederam, o mesmo recebeu várias críticas em função da alta complexidade computacional associada e devido a ausência de um mecanismo elaborado de elitismo. Buscando reduzir esses problemas, propôs-se uma versão aperfeiçoada do NSGA, a qual foi nomeada NSGA-II (Deb et al. 2000). Essa nova versão, além de apresentar melhorias quanto a velocidade de convergência, proporcionou a redução da complexidade computacional de $O(mN^3)$ para $O(mN^2)$, sendo m o número de objetivos e N o tamanho da população. Esse método é descrito nas próximas linhas e detalhado em (Deb et al. 2000, Deb et al. 2002).

O NSGA-II inicia-se com a geração aleatória de uma população $\mathcal{P}_0$ de tamanho N sobre o espaço de busca. Cria-se também a população externa, ou arquivo $\mathcal{A}_0 = \emptyset$, com o objetivo de armazenar a cada geração o Pareto-ótimo estimado. Este arquivo possui tamanho máximo igual a L . A população inicial é avaliada e, então, ordenada de acordo com o princípio de não-dominância, em que cada solução recebe um valor de aptidão associado à fronteira em que se encontra, sendo igual a um para o melhor nível, igual a dois para o segundo nível, e assim por diante, até que toda a população

tenha sido classificada. Essa operação de classificação em fronteiras é realizada por uma rotina chamada “fast non-dominated sorting”. Feito isso, aplica-se em sequência os operadores genéticos de seleção por torneio binário, cruzamento e mutação, criando a primeira população de descendentes $\mathcal{Q}_0$ de tamanho N .

A etapa cíclica do NSGA-II cria primeiramente uma população combinada $\mathcal{R}_n = \mathcal{P}_n \cup \mathcal{Q}_n$ de tamanho $2N$. Essa população é novamente classificada em fronteiras, e a nova população $\mathcal{P}_{n+1}$ é formada adicionando-se, uma a uma, as melhores fronteiras, até atingir N soluções. Como normalmente a última fronteira não precisará ser inserida por completo, o que acarretaria um número superior a N soluções em $\mathcal{P}_{n+1}$, somente as melhores soluções dessa fronteira são selecionadas, o que é realizado por meio de um mecanismo de nicho, conhecido como “crowding-distance assignment”.

A população $\mathcal{P}_{n+1}$ é, então, submetida aos operadores de seleção por torneio binário, cruzamento e mutação, criando assim novos descendentes $\mathcal{Q}_{n+1}$. Observe que o critério de seleção executado durante a etapa cíclica baseia-se não somente no valor de aptidão relacionado à fronteira a qual a solução pertence, mas também considera-se o operador de nicho. Logo, entre dois indivíduos de fronteiras diferentes, seleciona-se aquele de melhor aptidão, e entre indivíduos da mesma fronteira, escolhe-se o que apresentar o maior “crowding distance”.

Uma discussão detalhada acerca das rotinas “fast non-dominated sorting” e “crowding-distance assignment” é encontrada em (Deb et al. 2000, Deb et al. 2002).

Os operadores de cruzamento e mutação implementados nesse trabalho são, respectivamente, “simulated binary crossover” (SB $\mathcal{X}$) e mutação polinomial. O SB $\mathcal{X}$ implementa um operador de cruzamento com codificação real, cujo poder de busca é similar ao desempenhado por um cruzamento binário de um único parâmetro de uma solução. Esse operador é descrito a seguir e foi proposto por (Deb & Agrawal 1995), sendo amplamente discutido em (Deb & Goyal 1996, Deb & Beyer 2001, Deb et al. 2007).

Escolhe-se aleatoriamente dois indivíduos pais $\bar{x}_{i,G}^1$ e $\bar{x}_{i,G}^2$ pertencentes à geração corrente G . A probabilidade de cruzamento $\rho_c \leq U(0, 1)$ entre esses pontos é testada, e caso não seja satisfeita os indivíduos pais são diretamente inseridos na população de descendentes. Entretanto, uma vez verificada essa probabilidade, o cruzamento é então realizado em cada variável de otimização dada a probabilidade de ocorrência $\nu = U(0, 1)$. De forma geral, tem-se que para cada variável j determina-se um fator de dispersão β_j em função do índice de distribuição de cruzamento η_c , escolhido pelo usuário. O valor de β_j é obtido conforme mostrado na equação (4.6).

$$\beta_j = \begin{cases} \frac{1}{(2\nu_j)\eta_c + 1} & \text{se } \nu_j \leq 0.5 \\ \left(\frac{1}{2(1 - \nu_j)}\right) \frac{1}{\eta_c + 1} & \text{caso contrário} \end{cases} \quad (4.6)$$

Após esta etapa os descendentes são determinados como evidenciado na equação (4.7). Esse processo é repetido até que a população de descendentes tenha tamanho igual a N .

$$\begin{aligned} x_{ij,G+1}^1 &= 0.5 [(1 + \beta_j) x_{ij,G}^1 + (1 - \beta_j) x_{ij,G}^2] \\ x_{ij,G+1}^2 &= 0.5 [(1 - \beta_j) x_{ij,G}^1 + (1 + \beta_j) x_{ij,G}^2] \end{aligned} \quad (4.7)$$

A mutação consiste da adição de um fator de perturbação δ a um dado ponto selecionado aleatoriamente da população que sofreu cruzamento, em que δ possui distribuição segundo uma função densidade de probabilidade polinomial (Deb & Goyal 1996). Após o sorteio aleatório de $\vec{x}_{i,G}$, a probabilidade de mutação $\rho_m \leq U(0, 1)$ é testada, e caso não seja satisfeita o indivíduo selecionado não sofre modificações. Entretanto, se ρ_m for atendida, testa-se a probabilidade de mutação de cada variável j dada uma taxa de ocorrência $u_j = U(0, 1)$. O vetor de perturbação (equação (4.8)) é obtido em função do índice de distribuição de mutação η_m , escolhido pelo usuário. O indivíduo mutado é calculado por meio da equação (4.9). Esse processo é repetido até que N pontos tenham sido selecionados.

$$\delta_j = \begin{cases} \frac{1}{(2u_j)\eta_m + 1} - 1 & \text{se } u_j < 0.5 \\ 1 - [2(1 - \eta_m)] \frac{1}{\eta_m + 1} & \text{caso contrário} \end{cases} \quad (4.8)$$

$$x_{ij,G+1} = x_{ij,G} + \delta_j \quad (4.9)$$

O Alg. 4.2 mostra o ciclo básico de funcionamento do NSGA-II.

Algoritmo 4.2: Estrutura de funcionamento do NSGA-II.

Input: Objetivos, restrições, espaço de busca, N , L , ρ_c , ρ_m , η_c , η_m
Output: Estimativa do conjunto Pareto-ótimo $\mathcal{A}(n)$

```

1 begin
2    $\mathcal{P}(n) = \{\vec{p}_1, \dots, \vec{p}_N\} \leftarrow$  População inicial;
3    $\mathcal{A}(n) = \emptyset \leftarrow$  Arquivo inicial;
4    $\Phi_P(n) \leftarrow$  Avaliação ( $\mathcal{P}(n)$ );
5    $\mathcal{F}(n) \leftarrow$  “Fast Non-Dominated Sorting” ( $\Phi_P(n)$ );
6    $\mathcal{S}(n) \leftarrow$  Seleção ( $\mathcal{F}(n)$ );
7    $\mathcal{Q}(n) \leftarrow$  Variação ( $\mathcal{S}(n), \rho_c, \eta_c, \rho_m, \eta_m$ );
8    $\Phi_Q(n) \leftarrow$  Avaliação ( $\mathcal{Q}(n)$ );
9   while Não critério de parada do
10     $\mathcal{R}(n) = \mathcal{P}(n) \cup \mathcal{Q}(n)$ ;
11     $\Phi(n) = (\Phi_P(n) \cup \Phi_Q(n))$ ;
12     $\mathcal{F}(n+1) \leftarrow$  “Fast Non-Dominated Sorting” ( $\Phi(n)$ );
13     $\mathcal{I}(n+1) \leftarrow$  “Crowding Distance” ( $\Phi(n)$ );
14     $\mathcal{P}(n+1) \leftarrow$  Redução ( $\mathcal{F}(n), \mathcal{I}(n), N$ ); /* seleção dos N melhores */
15     $\mathcal{A}(n+1) \leftarrow$  Atualiza ( $\mathcal{A}(n) \cup \mathcal{P}(n), L$ ); /* soluções não-dominadas */
16     $\mathcal{S}(n+1) \leftarrow$  Seleção ( $\mathcal{F}(n), \mathcal{I}(n)$ );
17     $\mathcal{Q}(n+1) \leftarrow$  Variação ( $\mathcal{S}(n), \rho_c, \eta_c, \rho_m, \eta_m$ );
18     $n = n + 1$ ;
19  end
20 end

```

4.5.2 “Strength Pareto Evolutionary Algorithm” - SPEA-II

O “Strength Pareto Evolutionary Algorithm” (SPEA) (Zitzler & Thiele 1999) representa um algoritmo genético multi-objetivo muito conhecido na literatura, mas que também sofreu várias críticas em sua primeira versão. Visto isso, propôs-se uma versão aperfeiçoada, nomeada SPEA-II (Zitzler et al. 2001), a qual se destaca por empregar uma técnica elaborada para a criação da função de aptidão, além de incorporar informações relacionadas à densidade física das soluções. Diferente do que se observa em outros métodos multi-objetivo, o SPEA-II mantém uma população externa de tamanho fixo, e somente os indivíduos desse arquivo são submetidos ao processo de seleção. O princípio de funcionamento desse método é descrito a seguir.

A etapa de inicialização é responsável pela geração aleatória de uma população $\mathcal{P}_0$ de tamanho N , e pela criação de um arquivo vazio $\mathcal{A}_0 = \emptyset$. Já na etapa cíclica, avalia-se o vetor $\mathcal{P}_n \cup \mathcal{A}_n$, atribuindo um valor de aptidão a cada solução. Com base nesses dados, atualiza-se a população do arquivo inserindo as soluções não-dominadas do conjunto

$\mathcal{P}_n \cup \mathcal{A}_n$ em $\mathcal{A}_{n+1}$. Como o tamanho do arquivo deve ser fixo, sendo igual a L , caso o número de soluções não-dominadas exceda esse valor, elimina-se aquelas pertencentes às regiões mais densas, até que restem somente L . Esse esquema de nicho baseia-se em uma adaptação do “k-neighbor method” (Silverman 1986). Em contraposição, se o número de soluções não-dominadas for inferior a L , então adiciona-se as melhores soluções dominadas pertencentes a $\mathcal{P}_n \cup \mathcal{A}_n$ em $\mathcal{A}_{n+1}$, até que a população do arquivo esteja completa. Esse processo garante que seja mantida diversidade na população externa, o que permite executar o operador de seleção sobre esse arquivo, gerando a nova população $\mathcal{P}_{n+1}$ de tamanho N . Finalmente são aplicados os operadores de cruzamento e mutação, e incrementado o contador de gerações.

No sistema de avaliação efetuado pelo SPEA-II, cada solução inserida no conjunto $\mathcal{P}_n \cup \mathcal{A}_n$ recebe um valor $s(i)$ que representa o número de soluções que o indivíduo $\vec{p}_i$ domina, sendo matematicamente definido pela equação (4.10):

$$s(i) = |\{j : \vec{p}_j \in \mathcal{P}_n \cup \mathcal{A}_n \mid \vec{p}_i \prec \vec{p}_j\}| \quad (4.10)$$

em que $|\cdot|$ denota a cardinalidade do conjunto em seu argumento.

Além disso, faz-se necessário o cálculo do valor de aptidão bruto $b(i)$, o qual corresponde ao somatório dos $s(j)$ de todas as soluções $\vec{p}_j$ que dominam $\vec{p}_i$. A definição de $b(i)$ é dada na equação (4.11).

$$b(i) = \sum_{\vec{p}_j \in \mathcal{P}_n \cup \mathcal{A}_n, \vec{p}_j \prec \vec{p}_i} s(j) \quad (4.11)$$

Como mencionado anteriormente, o cálculo da densidade estimada (equação (4.12)) baseia-se em uma adaptação do “k-neighbor method”. Dessa forma, para cada indivíduo $\vec{p}_i$ calcula-se a distância, no espaço de objetivos, em relação aos k vizinhos mais próximos pertencentes ao conjunto $\mathcal{P}_n \cup \mathcal{A}_n$, e armazena o resultado na variável σ_i^k , sendo $k = \sqrt{N + L}$.

$$d(i) = \frac{1}{\sigma_i^k + 2} \quad (4.12)$$

Finalmente, o valor de aptidão associado a cada indivíduo $\vec{p}(i)$ é fornecido pela equação (4.13).

$$\Phi(p_i) = b(i) + d(i) \quad (4.13)$$

O Alg. 4.3 apresenta o ciclo de funcionamento do SPEA-II.

Algoritmo 4.3: Estrutura de funcionamento do SPEA-II.

Input: Objetivos, restrições, espaço de busca, N , L
Output: Estimativa do conjunto Pareto-ótimo $\mathcal{A}(n)$

```

1 begin
2    $\mathcal{P}(n) = \{\vec{p}_1, \dots, \vec{p}_N\} \leftarrow$  População inicial;
3    $\mathcal{A}(n) = \emptyset \leftarrow$  Arquivo inicial;
4   while Não critério de parada do
5      $\Phi(n) \leftarrow$  Avaliação ( $\mathcal{P}(n) \cup \mathcal{A}(n)$ );
6      $\mathcal{A}(n) \leftarrow$  Atualização ( $\mathcal{P}(n) \cup \mathcal{A}(n), \Phi(n), L$ ); /* melhores soluções */
7      $\mathcal{S}(n) \leftarrow$  Seleção ( $\mathcal{A}(n), N$ );
8      $\mathcal{P}(n+1) \leftarrow$  Variação ( $\mathcal{S}(n)$ );
9      $n = n + 1$ ;
10  end
11 end

```

4.5.3 “Pareto Envelope-based Selection Algorithm” - PESA

O “Pareto Envelope-based Selection Algorithm” (PESA) é mais um conhecido método evolucionário multi-objetivo, o qual se destaca, principalmente, por realizar os processos de seleção e geração de diversidade por meio de um único e simples esquema baseado na construção de um “hyper-grid” no espaço de objetivos do problema de otimização. Apresenta-se a seguir uma breve descrição deste método, sendo melhor exposto em (Corne et al. 2000).

A etapa de inicialização caracteriza-se pela geração aleatória e avaliação da população interna de cromossomos $\mathcal{P}_0$, de tamanho N . Cria-se também o arquivo externo $\mathcal{A}_0$, inicialmente vazio. Já na etapa cíclica, o arquivo é atualizado, sendo preenchido com as soluções não-dominadas pertencentes à população interna corrente. Feito isso, os cromossomos da população corrente são deletados, e novas soluções são selecionadas do arquivo até que a nova população $\mathcal{P}_n$ possua N novos pontos. Esses cromossomos selecionados são então submentidos aos processos de cruzamento e mutação, gerando

diversidade no espaço de busca. A população resultante do mecanismo de variação é comparada com as soluções do arquivo, e mantém-se apenas as não-dominadas, respeitando o tamanho máximo da população externa L . O ciclo se repete até a verificação de algum critério de parada.

Como mencionado anteriormente, os mecanismos de seleção e manutenção de diversidade são baseados em um “hyper-grid” do espaço de objetivos, a partir do qual se define o chamado fator de compressão (“squeeze factor”). A Fig. 4.3 ilustra algumas soluções de Pareto de um problema de minimização bi-objetivo, as quais estão inseridas em caixas, ou “grids”, uniformemente distribuídas no domínio de objetivos normalizados. De forma geral, o fator de compressão $s(i)$ expressa a característica de densidade de soluções em uma dada caixa i . Logo, pela Fig. 4.3 tem-se que $s(i) = 3$, $s(j) = 2$, $s(k) = 1$, etc.

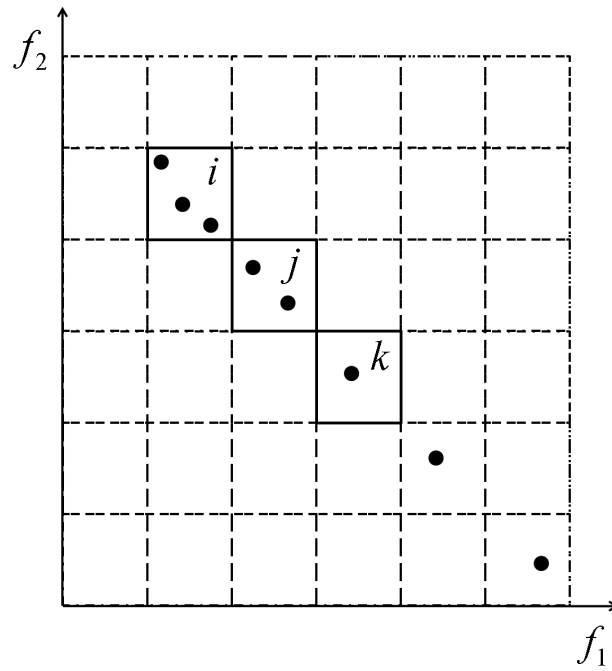


Figura 4.3: Estratégia de avaliação de densidade empregado no PESA.

Durante o mecanismo de atualização, os elementos da população interna são apresentados um a um ao arquivo, e somente os cromossomos não-dominados são nele inseridos. Caso em algum momento a população externa exceda o tamanho limite L , então elimina-se uma solução aleatória pertencente ao “grid” mais denso, ou seja, de maior valor de s . Esse processo garante que as soluções sejam melhor distribuídas ao longo do conjunto Pareto-ótimo estimado.

O PESA emprega um sistema de seleção por torneio binário, e dentre dois cromossomos escolhidos aleatoriamente do arquivo, seleciona-se aquele localizado na região menos densa do espaço de objetivos, ou seja, permanecerá a solução que estiver contida no “grid” que apresentar o menor valor de s . Dessa forma, o algoritmo é forçado a explorar regiões pouco pesquisadas do espaço de busca.

O Alg. 4.4 evidencia o ciclo básico de funcionamento do PESA.

Algoritmo 4.4: Estrutura de funcionamento do PESA.

Input: Objetivos, restrições, espaço de busca, N , L
Output: Estimativa do conjunto Pareto-ótimo $\mathcal{A}(n)$

```

1 begin
2    $\mathcal{P}(n) = \{\vec{p}_1, \dots, \vec{p}_N\} \leftarrow$  População inicial;
3    $\mathcal{A}(n) = \emptyset \leftarrow$  Arquivo inicial;
4   while Não critério de parada do
5      $\Phi(n) \leftarrow$  Avaliação ( $\mathcal{P}(n) \cup \mathcal{A}(n)$ );
6      $\mathcal{A}(n) \leftarrow$  Atualiza ( $\mathcal{P}(n) \cup \mathcal{A}(n)$ ,  $\Phi(n)$ ,  $L$ ); /* pontos não-dominadas */
7      $\mathcal{S}(n) \leftarrow$  Seleção ( $\mathcal{A}(n)$ ,  $N$ );
8      $\mathcal{P}(n+1) \leftarrow$  Variação ( $\mathcal{S}(n)$ );
9      $n = n + 1$ ;
10  end
11 end

```

4.5.4 “Multi-Objective Clonal Selection Algorithm” - MOCSA

Em função do nascimento recente do estudo de sistemas imunes artificiais, são poucos os trabalhos que sugerem métodos multi-objetivo baseados nesta teoria; ver por exemplo (Coello & Cortés 2002, Coello & Cortés 2005, Gong et al. 2007, Guimarães et al. 2007). Visto que uma das contribuições dessa dissertação consiste da apresentação de um novo algoritmo multi-objetivo inspirado no princípio da seleção clonal, discute-se a seguir um método imunológico multi-objetivo, chamado “Multi-Objective Clonal Selection Algorithm” (MOCSA) (Guimarães et al. 2007), o qual pode ser considerado uma extensão do método RCSA (ver seção 3.6.3) aplicado a problemas multi-objetivo.

O ponto de partida do MOCSA consiste na geração e avaliação de uma população inicial de tamanho N , distribuída aleatoriamente sobre o espaço de busca. Além disso, cria-se um arquivo externo inicialmente vazio. Dessa forma, inicia-se o ciclo iterativo com a classificação dos anticorpos em fronteiras não-dominadas, o que é feito conforme executado pelo NSGA-II. Após esta etapa, os N_{sel} melhores anticorpos são selecionados,

sendo que aqueles pertencentes a uma dada fronteira i recebem o mesmo número de clones N_C^i :

$$N_C^i = \text{round}\left(\frac{\beta N}{i}\right) \quad (4.14)$$

em que β é um fator de clonagem e $\text{round}(\cdot)$ arredonda o seu argumento para o inteiro mais próximo.

Os clones são então maturados segundo uma função densidade de probabilidade Gaussiana, e N_{rep} novos anticorpos são inseridos na população em substituição aqueles não selecionados para a clonagem, o que garante a manutenção de diversidade e exploração de novas regiões do espaço de busca. A nova população (anticorpos originais + clones maturados + anticorpos inseridos) é novamente classificada em fronteiras, e os indivíduos pertencentes ao Pareto-ótimo estimado são armazenados na população de arquivo.

Durante o processo de seleção, o mecanismo de escolha empregado entre anticorpos de uma mesma fronteira baseia-se no “k-neighbor method”. Dessa forma, a partir de informações relacionadas a densidade de soluções em uma região específica do espaço de objetivos normalizado, prioriza-se a seleção daquelas que estiverem em áreas pouco representadas, ou seja, em áreas menos densas. De forma análoga, caso o número de soluções não-dominadas ultrapasse o tamanho limite L do arquivo externo, utiliza-se um mecanismo de supressão também baseado no “k-neighbor method”, o que possibilita identificar e eliminar as soluções das regiões mais densas do conjunto Pareto-ótimo estimado.

O Alg. 4.5 estrutura o ciclo iterativo implementado pelo MOCSA.

4.5.5 “Multi-Objective Differential Evolution” - MODE

Diante dos bons resultados encontrados pela otimização mono-objetivo baseada na evolução diferencial (ver seção 3.6.5), vários autores têm proposto extensões multi-objetivo com o emprego desta técnica (Madavan 2002, Abbass 2002, Babu & Jehan 2003, Sarker & Abbass 2004, Parsopoulos et al. 2004, Robic & Filipic 2005, Iorio & Li 2006, Hernandez-Diaz et al. 2006, Qian & Li 2008, Gong & Cai 2008, Alatas et al. 2008). Visto isso, essa subseção apresenta um MODE discutido em (Xue et al. 2003b, Xue et al. 2005), o qual vem demonstrando um alto desempenho frente a importantes problemas conhecidos na literatura.

Algoritmo 4.5: Estrutura de funcionamento do MOCSA.

Input: Objetivos, restrições, espaço de busca, N , N_{sel} , β , L
Output: Estimativa do conjunto Pareto-ótimo $\mathcal{A}(n)$

```

1 begin
2    $\mathcal{P}(n) = \{\vec{p}_1, \dots, \vec{p}_N\} \leftarrow$  População inicial;
3    $\Phi_P(n) \leftarrow$  Avaliação ( $\mathcal{P}(n)$ );
4    $\mathcal{A}(n) = \emptyset \leftarrow$  Arquivo inicial;
5   while Não critério de parada do
6      $\mathcal{S}(n) \leftarrow$  Seleção ( $\mathcal{P}(n)$ ,  $\Phi_P(n)$ ,  $N_{sel}$ );
7      $\mathcal{C}(n) \leftarrow$  Clonagem ( $\mathcal{S}(n)$ ,  $\beta$ );
8      $\mathcal{Q}(n) \leftarrow$  Maturação ( $\mathcal{C}(n)$ );
9      $\mathcal{D}(n) \leftarrow$  Diversidade ( $N - N_{sel}$ );
10     $\mathcal{R}(n) \leftarrow (\mathcal{Q}(n) \cup \mathcal{D}(n))$ ;
11     $\Phi_R(n) \leftarrow$  Avaliação ( $\mathcal{R}(n)$ );
12     $\mathcal{A}(n+1) \leftarrow$  Atualização ( $\mathcal{R}(n) \cup \mathcal{A}(n)$ ,  $L$ );
13     $\mathcal{P}(n+1) \leftarrow (\mathcal{S}(n) \cup \mathcal{R}(n))$ ;
14     $\Phi_P(n+1) \leftarrow (\Phi_S(n) \cup \Phi_R(n))$ ;
15     $n = n + 1$ ;
16  end
17 end

```

O “Multi-Objective Differential Evolution” (MODE) proposto por (Xue et al. 2003b) é muito similar aos demais métodos discutidos anteriormente, distinguindo-se apenas quanto ao mecanismo de seleção adotado, e quanto ao processo de variação, os quais são descritos a seguir.

Dado um conjunto de pontos, previamente classificado em fronteiras de Pareto, a variação é elaborada por meio de dois mecanismos de mutação, sendo estes caracterizados ou pela adição de vetores diferenciais, ou pela adição de vetores de perturbação. De forma geral, tem-se que após a escolha aleatória de um dado ponto $\vec{p}_i$ pertencente a população corrente, verifica-se se o mesmo é ou não uma solução dominada. Caso não seja uma solução dominada, então este ponto é apenas perturbado, gerando assim uma nova solução $\vec{p}_i^{mut}$. Esse mecanismo de mutação exerce uma busca local ao redor da solução $\vec{p}_i$ pertencente ao conjunto Pareto-ótimo estimado. Entretanto, se o ponto sorteado $\vec{p}_i$ for uma solução dominada, a mutação diferencial exige a escolha aleatória de um segundo ponto $\vec{p}_i^D \in \mathcal{F}_1$, o qual deve pertencer ao subconjunto $\mathcal{D} \in \mathcal{F}_1$, composto pelos pontos que dominam $\vec{p}_i$. O resultado dessa mutação diferencial é ainda perturbado, gerando assim a nova solução mutante. Em ambos os casos, o vetor de perturbação é gerado a partir da escolha aleatória de pontos pertencentes a população corrente. Esse segundo mecanismo de mutação possui fundamental importância quanto a velocidade

de convergência do método. A formulação matemática da variação implementada pelo MODE é definida na equação (4.15):

$$\vec{p}_i^{mut} = \begin{cases} \vec{p}_i + w \sum_{k=1}^K (\vec{p}_{r1}^{i,k} - \vec{p}_{r2}^{i,k}) & \text{se } \vec{p}_i \in \mathcal{F}_1 \\ \gamma \vec{p}_i^D + (1 - \gamma) \vec{p}_i + w \sum_{k=1}^K (\vec{p}_{r1}^{i,k} - \vec{p}_{r2}^{i,k}) & \text{caso contrário} \end{cases} \quad (4.15)$$

em que $\gamma \in [0, 1]$ representa o fator diferencial, w é o fator de escala do vetor de perturbação, K é o número de vetores de perturbação, e $\vec{p}_{r1}^{i,k}$, $\vec{p}_{r2}^{i,k}$ são pontos mutuamente distintos, escolhidos aleatoriamente na população. Vale mencionar que essa estratégia de mutação é aplicada sobre cada variável de otimização, dada a probabilidade de ocorrência ρ_{mut} .

A Fig. 4.4 ilustra a técnica de variação do MODE em um problema de minimização bi-objetivo. Obviamente, as direções são definidas no espaço de parâmetros, e não no domínio dos objetivos.

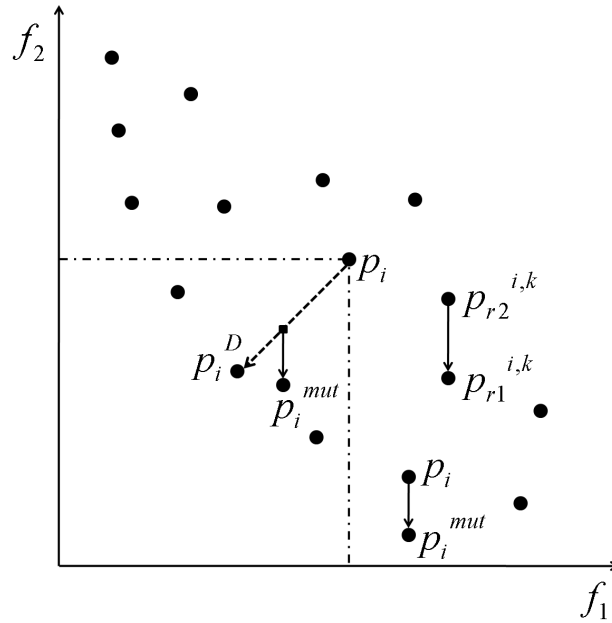


Figura 4.4: Estratégia de variação empregada pelo MODE - figura adaptada de (Xue et al. 2003b).

Durante o processo de seleção implementado pelo NSGA-II, os indivíduos pertencentes às melhores fronteiras são diretamente inseridos na próxima geração, e a métrica “crowding distance” é utilizada somente para completar a escolha dos N indivíduos necessários. Entretanto, conforme mostrado em (Xue et al. 2003a), esta estratégia eli-

tista nem sempre produz bons resultados, uma vez que o critério de diversidade não é considerado em todas as fronteiras de Pareto. Dessa forma, além de implementar as ferramentas utilizadas no NSGA-II, o MODE inclui um parâmetro extra (σ_{crowd}), que tem como objetivo especificar o quão próximas podem estar as soluções pertencentes a uma dada fronteira, possibilitando evitar a presença de pontos muito similares no interior da mesma. Esse mecanismo, além de impedir uma convergência prematura do método, permite a inserção de certos pontos dominados, outrora simplesmente descartados, garantindo um melhor nível de diversidade entre as fronteiras. Maiores detalhes sobre a importância da manutenção de diversidade entre as fronteiras de Pareto é encontrado em (Deb & Goel 2001).

O Alg. 4.6 apresenta a estrutura básica de funcionamento do MODE.

Algoritmo 4.6: Estrutura de funcionamento do MODE.

Input: Objetivos, restrições, espaço de busca, N , γ , w , K , ρ_{mut} , σ_{crowd} , L
Output: Estimativa do conjunto Pareto-ótimo $\mathcal{A}(n)$

```

1 begin
2    $\mathcal{P}(n) = \{\vec{p}_1, \dots, \vec{p}_N\} \leftarrow$  População inicial;
3    $\Phi_P(n) \leftarrow$  Avaliação ( $\mathcal{P}(n)$ );
4    $\mathcal{A}(n) = \emptyset \leftarrow$  Arquivo inicial;
5   while Não critério de parada do
6      $\mathcal{Q}(n) \leftarrow$  Variação ( $\mathcal{P}(n)$ ,  $\gamma$ ,  $w$ ,  $K$ ,  $\rho_{mut}$ );
7      $\Phi_Q(n) \leftarrow$  Avaliação ( $\mathcal{Q}(n)$ );
8      $(\mathcal{P}(n+1), \Phi_P(n+1)) \leftarrow$  Seleção ( $\mathcal{P}(n) \cup \mathcal{Q}(n)$ ,  $\sigma_{crowd}$ ,  $N$ );
9      $\mathcal{A}(n+1) \leftarrow$  Atualização ( $\mathcal{P}(n) \cup \mathcal{A}(n)$ ,  $L$ );
10     $n = n + 1$ ;
11  end
12 end

```

4.5.6 Outras Vertentes

Embora as subseções anteriores mencionem somente alguns dos métodos multi-objetivo mais famosos da literatura, vale a pena citar um pequeno histórico em relação aos primeiros trabalhos propostos.

- “Vector Evaluated Genetic Algorithm” (VEGA)

O VEGA foi o primeiro algoritmo genético implementado para a solução de problemas multi-objetivo (Schaffer 1984). Apesar de ser muito simples e inviável a sua

aplicação na prática, esse trabalho apresentou grande importância histórica, pois incentivou outros pesquisadores a elaborarem técnicas cada vez mais eficientes.

- “Multi-Objective Genetic Algorithm” (MOGA)

Durante a década de 1990 surgiram várias técnicas evolucionárias multi-objetivo, dentre as quais se destaca o MOGA (Fonseca & Fleming 1993, Fonseca & Fleming 1998b, Fonseca & Fleming 1998c), pois foi o primeiro método a empregar relações de dominância e conceitos de Pareto-ótimo. Esse algoritmo, além de classificar as soluções estimadas em fronteiras de Pareto, já incluía técnicas de nicho, possibilitando uma melhor atribuição de valores de aptidão às soluções encontradas a cada geração.

- “Niche Pareto Genetic Algorithm” (NPGA)

Um segundo importante método proposto durante a década de 1990 foi o NPGA (Horn et al. 1993, Horn et al. 1994), o qual tornou-se conhecido por combinar seleção por torneio binário e alguns conceitos de dominância e Pareto-ótimo. Assim como o MOGA, o NPGA ainda não apresentava uma população de arquivo (população externa).

Finalmente, vale mencionar que assim como é clara a necessidade de implementação de novos operadores evolucionários, mono e multi-objetivo, é também imprescindível o estudo de novas métricas para a análise de desempenho de métodos multi-objetivo, e comparação dos conjuntos Pareto-ótimo estimados.

4.6 Conclusão

O presente capítulo foi escrito objetivando-se esclarecer, ao leitor interessado, alguns aspectos pertinentes relacionados ao processo de otimização de problemas com múltiplos objetivos. De forma geral, enumerou-se algumas definições e proposições que embasam a síntese dos sistemas multi-objetivo e, posteriormente, descreveu-se vários métodos evolucionários considerados importantes pela comunidade acadêmica envolvida com pesquisa em otimização computacional.

Embora os algoritmos não tenham sido apresentados segundo a ordem cronológica em que foram propostos, os mesmos surgiram com o propósito evidente de aperfeiçoar algumas características de otimização até então pouco elaboradas, e com isso reduzir, por exemplo, os custos computacionais associados, e ainda assim alcançar soluções eficientes.

No capítulo seguinte são apresentadas duas novas abordagens imunes artificiais, para otimização mono e multi-objetivo, nas quais são levados em conta as características mencionadas anteriormente, objetivando dessa forma, a formulação e concepção de métodos robustos e eficientes.

Capítulo 5

Sistemas Imunes Artificiais: Novas Abordagens

“Nada se inventaria se nos sentíssemos satisfeitos com as coisas descobertas.”

— Lucius Annaeus Seneca, 4a.C.–65d.C. (filósofo latino)

“A mente que se abre a uma nova ideia jamais volta ao seu tamanho original.”

— Albert Einstein, 1879–1955 (físico alemão)

5.1 Introdução

Este capítulo inclui a contribuição desta dissertação. Visando facilitar a compreensão do leitor, a apresentação das novas abordagens imunológicas artificiais é realizada em duas partes principais, sendo a primeira destinada à otimização mono-objetivo, onde descreve-se o método “Distributed Clonal Selection Algorithm” (DCSA), e a segunda dedicada à otimização multi-objetivo, onde discorre-se sobre o método “Multi-Objective Clonal Selection Algorithm” (MCSA). A estrutura de cada um dos algoritmos é claramente descrita, e apresentada uma discussão detalhada sobre o ajuste dos parâmetros desses métodos. Por fim, avalia-se a atuação dos operadores imunes implementados e o seu efeito no desempenho dos algoritmos.

5.2 “Distributed Clonal Selection Algorithm”

O “Distributed Clonal Selection Algorithm” (DCSA), o qual implementa uma técnica de otimização mono-objetivo, possui representação real dos seus parâmetros, e pode ser visto como uma versão aperfeiçoada do RCSA (ver seção 3.6.3). O problema de otimização e as condições de otimalidade para o DCSA são as mesmas definidas nas seções 3.2 e 3.3, respectivamente.

Em poucas palavras, tem-se que a principal diferença entre esses métodos (RCSA e DCSA) está contida na etapa de expansão clonal. De forma geral, observa-se que enquanto o RCSA emprega somente uma distribuição Gaussiana durante a etapa de maturação dos anticorpos clonados, o DCSA utiliza diferentes funções densidade de probabilidade (*pdf*) para este mesmo fim, o que torna possível efetuar uma busca local balanceada, ou distribuída, ao redor das melhores soluções encontradas.

Essa busca local é chamada balanceada, pois o processo de mutação baseia-se em três distribuições distintas, Gaussiana, uniforme e caótica, as quais possibilitam desempenhar uma pesquisa com diferentes raios de busca em torno dos anticorpos clonados. Assim, quanto melhor for a solução, menor é a perturbação gerada sobre esta, e à medida que o valor de afinidade reduz, a perturbação torna-se mais ampla em função da *pdf* adotada.

Antes que o DCSA seja apresentado com mais detalhes, é importante que o leitor conheça um pouco sobre a aplicação de séries caóticas em métodos de otimização. A subseção seguinte mostra alguns exemplos de trabalhos que utilizam distribuições caóticas, e, além disso, fornece conceitos indispensáveis para a compreensão do método proposto.

5.2.1 Teoria do Caos em Otimização

Na matemática, a teoria do caos descreve o comportamento de certos sistemas dinâmicos que, por sua vez, exibem uma dinâmica muito sensível às suas condições iniciais. Sistemas deste tipo tornaram-se populares após o trabalho de Edward N. Lorenz, onde o autor descreve o comportamento de uma sequência caótica, atualmente conhecida como *atrator de Lorenz* (Lorenz 1963).

Apesar dos estudos relativos à teoria do caos terem se tornado um importante campo de pesquisa interdisciplinar nos últimos anos (Chen & Dong 1998), poucos artigos têm

sugerido aplicações do caos em métodos de otimização, dentre os quais se destacam (Nozawa 1992, Chen & Aihara 1995, Hayakawa et al. 1995, Wang 1996, Chen & Aihara 1997, Wang & Smith 1998, Tokuda et al. 1998, Zhou & Chen 2000, He 2002, Mingjun & Huanwen 2004), onde, na maioria das vezes, emprega-se um sistema caótico para o ajuste de parâmetros de redes neurais artificiais (*Holpfild Neural Network*) associadas a um algoritmo de recozimento simulado (*Simulated Annealing*).

Outros trabalhos têm explorado a aplicação do caos em diferentes áreas de otimização, sendo os casos mais importantes relacionados a algoritmos genéticos (Yuan et al. 2002, Yang & Chen 2002, Coelho & Alotto 2008), algoritmo de evolução diferencial (Coelho & Mariani 2006), sistema imunológico artificial (Zilong et al. 2006), colônia de formigas (Cai et al. 2007), e (Luo & Shao 2000, Dashora et al. 2008), onde avalia-se o desempenho de algoritmos evolucionários (AE) híbridos, na presença de mutações e, ou, cruzamentos com propriedades caóticas, frente as suas versões originais.

Em geral, os autores mencionados enfatizam que o emprego de sequências caóticas na arquitetura de métodos de otimização os tornam mais eficientes, melhorando sua capacidade de escaparem de ótimos locais e aumentando sua velocidade de convergência para o ótimo global, reduzindo assim o custo computacional associado. Salvo os artigos citados, esse ganho obtido ora é alcançado por meio de ajustes de parâmetros via sequências caóticas, ora por meio da substituição de distribuições comuns (e.g. normal, uniforme) por uma distribuição caótica, o que tem se mostrado uma interessante ferramenta para a exploração de novas regiões do espaço de busca.

Observadas essas características, e visto que os mecanismos de variação dos sistemas imunes artificiais se sustentam em perturbações, a aplicação de sequências caóticas nessa classe de algoritmos torna-se de grande interesse.

Acredita-se, portanto, que embora um reduzido número de pesquisadores tenham sugerido técnicas de otimização que empreguem sistemas caóticos, esse número tenderá a crescer, uma vez que resultados eficientes foram apresentados nos trabalhos relacionados.

Sistemas Caóticos

Embora existam inúmeros sistemas caóticos, nesse trabalho são mencionados apenas dois, sendo o primeiro derivado de um mapa logístico, e o segundo de um neurônio caótico.

- **Mapa Logístico**

O mapa logístico, conhecido na literatura por apresentar um complexo e caótico comportamento, é gerado a partir de uma função polinomial simples. Esse mapa tornou-se popular devido ao trabalho de pesquisa do biólogo Robert M. May (May 1976). Nesse artigo o autor apresenta um diagrama de bifurcação com o objetivo de estudar populações animais ao longo dos anos. Matematicamente, o mapa logístico é descrito pela equação (5.1):

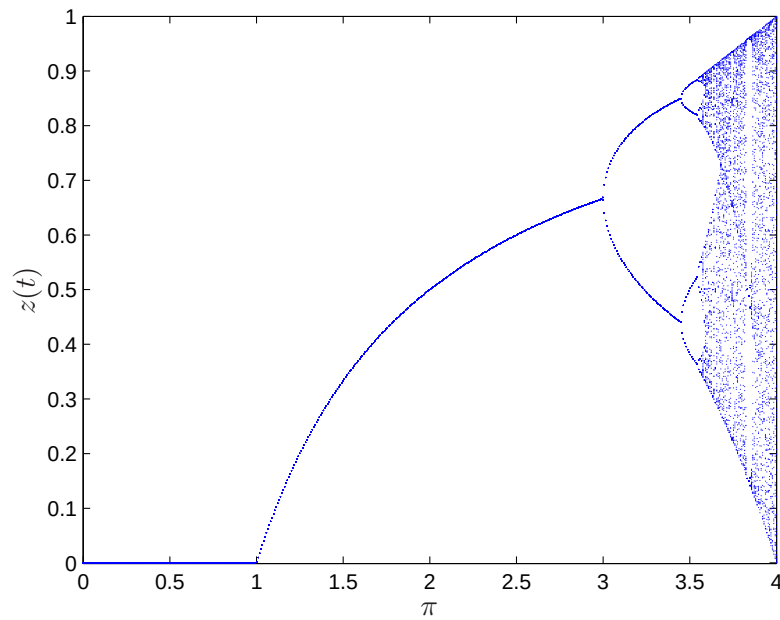
$$z_{t+1} = \pi z_t (1 - z_t) \quad (5.1)$$

em que $z_t \in [0, 1]$ representa a porcentagem viva da população de uma dada espécie no ano t , e π é um número positivo associado à uma taxa que relaciona os índices de reprodução e de mortalidade dessa população.

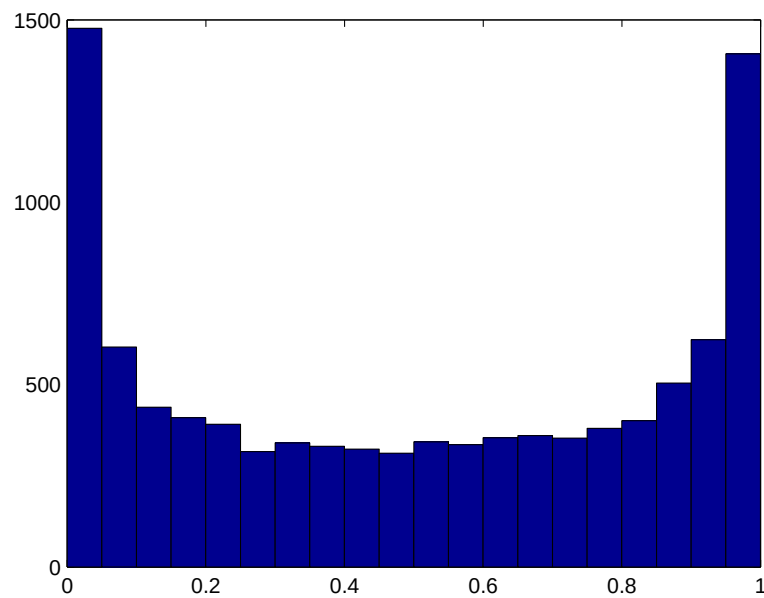
Variando-se o parâmetro π na formulação do mapa logístico, obtém-se um diagrama de bifurcação (Fig. 5.1(a)) que caracteriza graficamente a sobrevivência de uma dada população ao longo dos anos. O eixo horizontal representa o parâmetro π , enquanto o vertical representa valores assumidos por z .

Partindo desse diagrama de bifurcação pode-se obter diferentes funções densidade de probabilidade (*pdf*) com características caóticas. No entanto, a comumente citada em artigos adota o valor do parâmetro $\pi = 4$. Assim sendo, tem-se a distribuição caótica mostrada na Fig. 5.1(b).

Voltando à origem do trabalho realizado por May, tem-se que a distribuição alcançada para $\pi = 4$ revela que a população observada possui grandes chances de se reproduzir e, também, de ser extinta, o que representa o caráter caótico estudado.



(a) Diagrama de bifurcação gerado por um mapa logístico



(b) *pdf* caótica gerada por um mapa logístico com $\pi = 4$

Figura 5.1: Diagrama de bifurcação e distribuição caótica gerados a partir de um mapa logístico.

• Neurônio Caótico

Uma vez que não são muitas as referências que discutem sobre o modelo de mapas criados a partir de neurônios caóticos (Mingjun & Huanwen 2004), esse item não será tão enfatizado quanto o anterior. Matematicamente, a função de um neurônio caótico está evidenciada na equação (5.2):

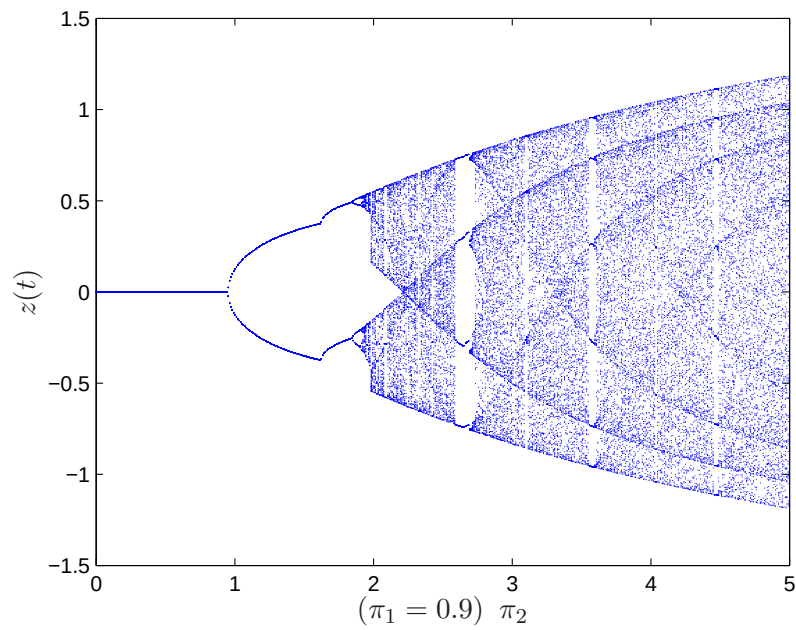
$$z_{t+1} = \pi_1 z_t - 2 \tanh[\pi_2 z_t] \exp[-3z_t^2] \quad (5.2)$$

em que z_t representa o estado interno do neurônio, $\pi_1 \in [0, 1]$ é o fator de amortecimento da membrana nervosa e π_2 é o fator de ajuste interno do sistema não-linear.

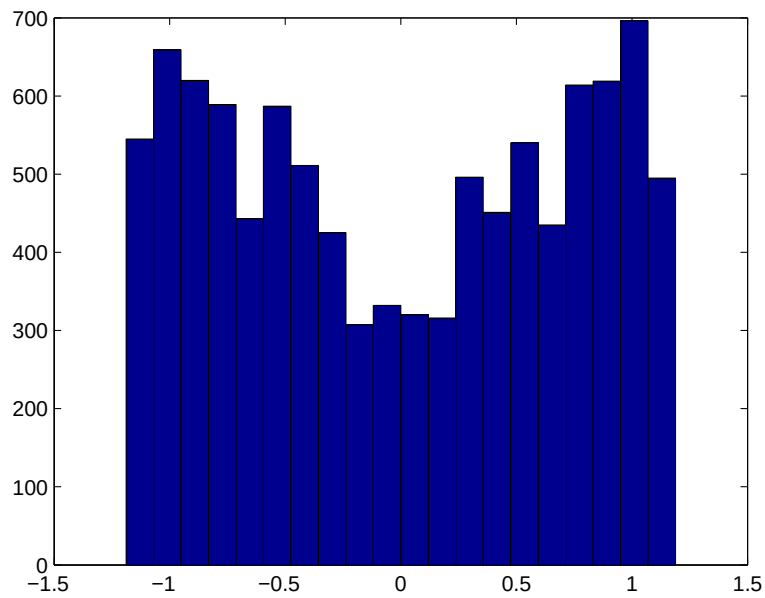
Fixando-se o parâmetro $\pi_1 = 0.9$ e variando-se o parâmetro π_2 , obtém-se o diagrama de bifurcação mostrado na Fig. 5.2(a). De forma similar ao realizado para o mapa logístico, quando fixa-se o parâmetro $\pi_2 = 5$, tem-se uma função densidade de probabilidade (*pdf*) com a característica caótica mostrada na Fig. 5.2(b).

5.2.2 Descrição do algoritmo DCSA

O “Distributed Clonal Selection Algorithm” (DCSA) começa com a geração de uma população inicial, usualmente espalhando N_{pop} pontos aleatórios sobre o espaço de busca do problema de otimização. Estes pontos (anticorpos) são avaliados na função de afinidade, que pode ser representada como $f(\vec{x})$ ou $-f(\vec{x})$, dependendo se o objetivo é de minimização ou maximização, respectivamente. O vetor população é, então, classificado em ordem decrescente de afinidade e submetido ao processo de expansão clonal. Para a execução desta etapa, o vetor população ordenado é separado em quatro grupos principais: o primeiro grupo, composto pelos $N_{dN}\%$ melhores anticorpos, é selecionado para clonagem e posterior maturação utilizando-se a distribuição Gaussiana; o segundo grupo, composto pelos $N_{dU}\%$ melhores pontos seguintes, é selecionado para clonagem e maturação utilizando-se a distribuição uniforme; o terceiro grupo, composto pelos $N_{dC}\%$ pontos seguintes, é selecionado para clonagem e maturação utilizando-se a distribuição caótica obtida por meio de um mapa logístico; e o último grupo, composto pelos $N_{REP}\%$ pontos não selecionados para clonagem, é eliminado, sendo repostos por novos pontos gerados aleatoriamente. Esta operação de substituição das piores soluções encontradas possui papel fundamental neste algoritmo, pois permite a manutenção de diversidade e a exploração de novas regiões do espaço de busca. A Fig. 5.3 apresenta o esquema de distribuição da população ordenada no DCSA, e a Fig. 5.4 mostra, em duas



(a) Diagrama de bifurcação gerado por um neurônio caótico



(b) *pdf* caótica gerada por um neurônio caótico com $\pi_1 = 0.9$ e $\pi_2 = 5$

Figura 5.2: Diagrama de bifurcação e distribuição caótica gerados a partir de um neurônio caótico.

dimensões, os resultados ilustrativos de sucessivas mutações de um mesmo anticorpo considerando-se as três distribuições adotadas pelo método.

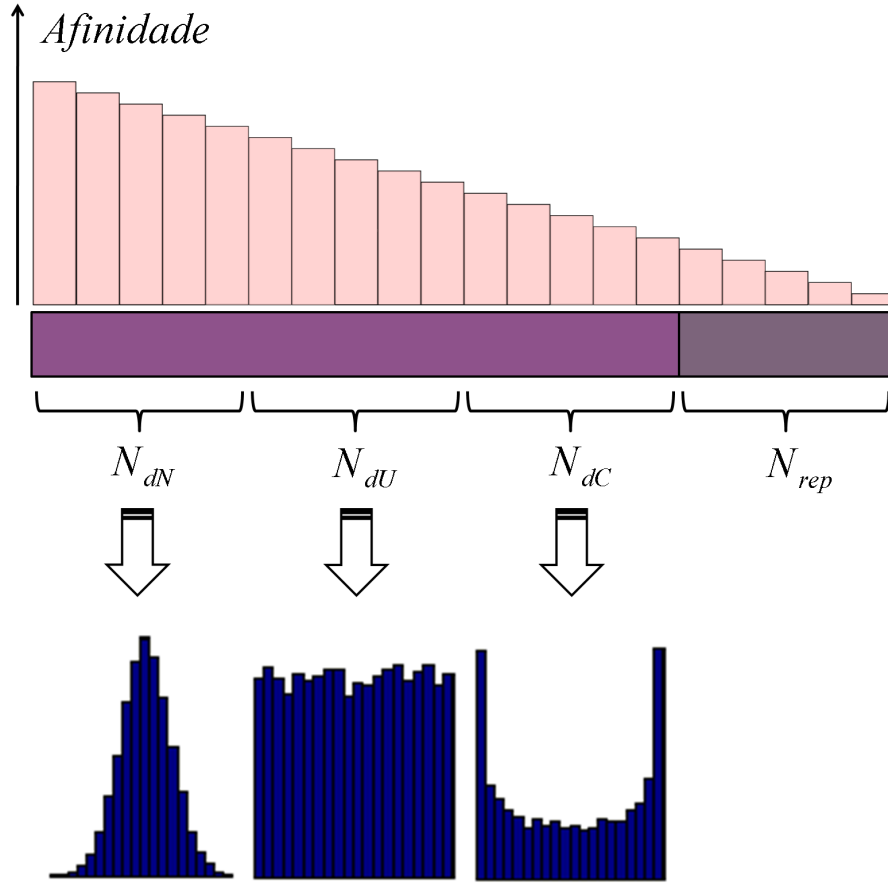


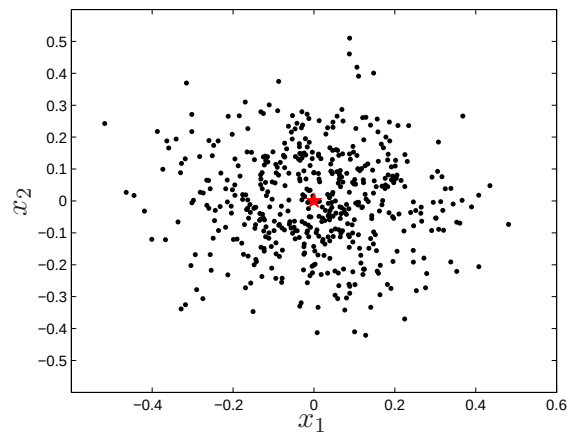
Figura 5.3: Distribuição da população no DCSA durante a etapa de expansão clonal. As *pdf's* ilustram o tipo de maturação que é exercido sobre cada um dos grupos selecionados.

Durante a fase de clonagem, cada anticorpo selecionado recebe um número de cópias N_C^i proporcional à sua posição i no vetor população ordenado (equação (5.3)):

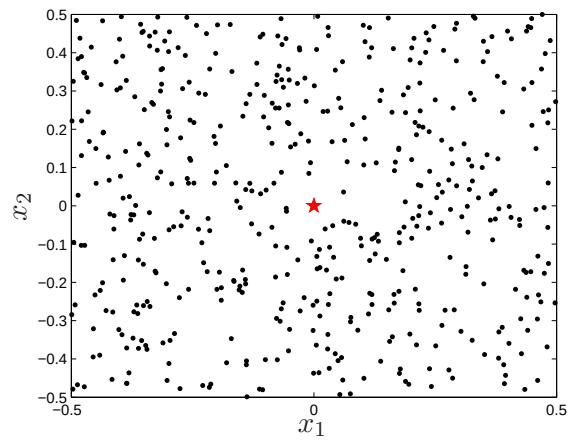
$$N_C^i = \text{round}\left(\frac{\beta N_{pop}}{i}\right) \quad (5.3)$$

em que β é o fator multiplicativo de clonagem e a função $\text{round}(\cdot)$ arredonda o seu argumento para o inteiro mais próximo.

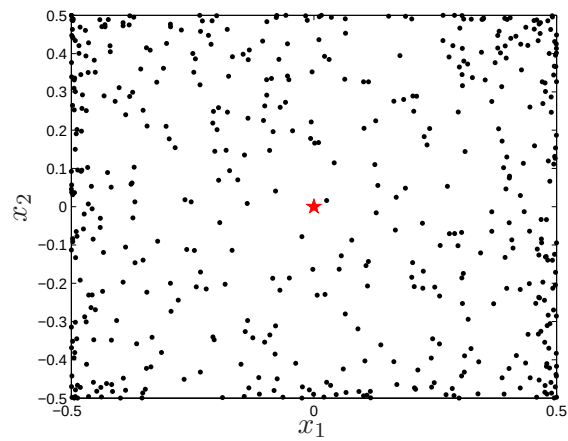
Em sequência, os clones são submetidos ao mecanismo de maturação, que é responsável pela adição de um ruído segundo uma função densidade de probabilidade específica.



(a) Mutação normal



(b) Mutação uniforme



(c) Mutação caótica

Figura 5.4: Sistema de mutação adotado pelo DCSA (as mutações são efetuadas com base em um ponto de referência localizado na origem).

Assim, um indivíduo mutante é obtido conforme mostrado na equação (5.4):

$$\begin{aligned} x_{i,k}^{mut} &= x_{i,k} + \nu_k \\ \nu_k &= \alpha \cdot S_k \cdot P \end{aligned} \quad (5.4)$$

em que ν_k é o ruído adicionado à k -ésima variável; $\alpha \in [0, 1]$ representa o tamanho da perturbação, podendo ser chamado γ_N , γ_U ou γ_C dependendo do tipo de ruído (Gaussiano para uma busca local, uniforme para uma busca simples, e caótico para uma busca mais ampla); S_k é a diferença entre os limites superior e inferior da respectiva variável ($ub_k - lb_k$); e P representa uma função densidade de probabilidade específica, com $P = N(0, 1)$ para a distribuição normal, $P = 2U(0, 1) - 1$ para a distribuição uniforme, e $P = 2C(\pi) - 1$ para a distribuição caótica, onde $C(\pi) : z(t+1) = \pi z(t)[1 - z(t)]$, $\pi = 4$, descreve uma sequência caótica obtida por meio de um sistema não-linear de um mapa logístico.

Um dado anticorpo e os seus clones maturados compõem uma subpopulação. Assim, as cópias maturadas são avaliadas na função de afinidade e somente a melhor solução de cada subpopulação é selecionada para prosseguir para a próxima geração, mantendo então uma população de tamanho constante.

O Alg. 5.1 ilustra como é gerada a perturbação caótica considerando-se um dado sistema não-linear representado pela função $\xi(\cdot)$.

Algoritmo 5.1: Geração de uma perturbação caótica.

```

1 begin
2    $z(0) \leftarrow U(0, 1);$ 
3   for  $t \leftarrow 1$  to 100 do
4      $z(t) \leftarrow \xi(z(t-1), \pi);$ 
5   end
6   for  $k \leftarrow 1$  to  $n$  do
7     seleção aleatória do índice  $i_k \in [1, 100];$ 
8      $\nu_k \leftarrow \gamma_C(ub_k - lb_k) z(i_k);$ 
9      $x_k \leftarrow x_k + \nu_k;$ 
10  end
11 end
```

Finalmente, apresenta-se no Alg. 5.2 a estrutura cíclica para implementação do DCSA.

Algoritmo 5.2: Estrutura de funcionamento do DCSA.

Input: Objetivo, restrições, espaço de busca, parâmetros

Output: Estimativa das melhores soluções

```
1  $N_{pop} \leftarrow$  tamanho da população;
2  $N_{dN} \leftarrow$  taxa da população submetida ao ruído Gaussiano;
3  $N_{dU} \leftarrow$  taxa da população submetida ao ruído Uniforme;
4  $N_{dC} \leftarrow$  taxa da população submetida ao ruído Caótico;
5  $\gamma_N \leftarrow$  representa o tamanho da perturbação normal;
6  $\gamma_U \leftarrow$  representa o tamanho da perturbação uniforme;
7  $\gamma_C \leftarrow$  representa o tamanho da perturbação caótica;
8  $\beta \leftarrow$  fator multiplicativo de clonagem;
9 begin
10    $\mathcal{P}(n) = \{\vec{p}_1, \dots, \vec{p}_N\} \leftarrow$  População inicial;
11    $\Phi(n) \leftarrow$  Avaliação ( $\mathcal{P}(n)$ );
12   while Não critério de parada do
13      $(\mathcal{P}'(n), \Phi'(n)) \leftarrow$  Ordenação ( $\mathcal{P}(n), \Phi(n)$ ); /* afinidade decrescente */
14      $(\mathcal{G}_{N_{dN}}, \mathcal{G}_{N_{dU}}, \mathcal{G}_{N_{dC}}) \leftarrow$  Seleção dos grupos ( $\mathcal{P}'(n), \Phi'(n)$ );
15      $\mathcal{C}(n) \leftarrow$  Clonagem ( $\mathcal{G}_{N_{dN}}, \mathcal{G}_{N_{dU}}, \mathcal{G}_{N_{dC}}$ );
16      $(\mathcal{Q}(n), \Phi_Q(n)) \leftarrow$  Maturação ( $\mathcal{C}(n)$ );
17      $(\mathcal{S}(n), \Phi_S(n)) \leftarrow$  Seleção por subpopulação ( $\mathcal{P}'(n), \Phi'(n), \mathcal{Q}(n), \Phi_Q(n)$ );
18      $(\mathcal{D}(n), \Phi_D(n)) \leftarrow$  Diversidade ( $N_{rep}$ );
19      $(\mathcal{P}(n+1), \Phi(n+1)) \leftarrow$  Atualização ( $\mathcal{S}(n), \Phi_S(n), \mathcal{D}(n), \Phi_D(n)$ );
20      $n = n + 1$ ;
21   end
22 end
```

5.2.3 Análise de Sensibilidade para Calibração dos Parâmetros

O estudo de sensibilidade dos parâmetros utilizados no DCSA é realizado de forma similar ao empregado em (Campelo et al. 2005, Campelo 2006). Assim sendo, a influência do ajuste destes parâmetros no desempenho do algoritmo é verificada por meio de duas métricas, sendo a primeira relacionada ao *número de avaliações da função objetivo até a convergência* (NAF), e a segunda, diz respeito à *taxa de falha de convergência* (TFC). Observa-se então que quanto menores forem os valores retornados pelas métricas, melhor será o desempenho do algoritmo.

Para uma melhor definição destas métricas são feitas as seguintes considerações:

- O critério de parada é ativado somente quando a convergência for verificada ou quando o número máximo de avaliações da função objetivo for alcançado;
- Caso o algoritmo não convirja para o ótimo até atingir o número máximo de avaliações da função objetivo, considera-se uma falha de convergência;
- Os valores médios encontrados para o NAF consideram somente os caso em que ocorreu convergência. Os valores médios apresentados para a TFC são normalizados;
- Os valores médios de cada cenário de simulação são calculados considerando-se 100 execuções do algoritmo.

Problemas Testes

Com o objetivo de determinar uma faixa aceitável para o ajuste dos parâmetros do DCSA, tal que este apresente alto desempenho diante de problemas reais de otimização, são sugeridos quatro problemas de testes, os quais compreendem algumas dificuldades, tais como multimodalidade, restrições, superfície suave de otimização, etc.

- **Problema $\mathcal{P}1$: Rosenbrock**

O primeiro problema de teste está definido na equação (5.5).

$$\begin{aligned} x^* = \arg \min_x f(\vec{x}) &\triangleq \sum_{i=1}^{n-1} \left[100 (x_i^2 - x_{i+1})^2 + (1 - x_i)^2 \right] \\ \text{sujeito a: } x_i &\in [-2.48, 2.48], \quad i = 1, \dots, n \end{aligned} \quad (5.5)$$

A função de *Rosenbrock* em duas dimensões, $n = 2$, possui um único mínimo que se localiza em uma região plana do espaço de busca. Esse fato caracteriza o nível de dificuldade deste teste, uma vez que muitos métodos apresentam uma convergência lenta para o ótimo. O mínimo global deste problema é $x^* = [1, 1]$, onde $f(x^*) = 0$, e considera-se que o algoritmo convergiu somente quando a distância Euclidiana entre o ótimo conhecido e a solução estimada for inferior ou igual a 0.05. O número máximo de avaliações da função objetivo permitido neste teste é igual a 3000. Esse critério de parada foi escolhido de maneira que permitisse analisar os resultados medianos encontrados pelas métricas NAF e TFC.

- **Problema $\mathcal{P}2$: Rastrigin**

Um outro problema de teste é definido na equação (5.6).

$$\begin{aligned} x^* = \arg \min_x f(\vec{x}) &\triangleq 10n + \sum_{i=1}^n x_i^2 - 10 \cos(2\pi x_i) \\ \text{sujeito a: } x_i &\in [-5.12, 5.12], \quad i = 1, \dots, n \end{aligned} \quad (5.6)$$

Esta função descreve o problema *Rastrigin* irrestrito, o qual possui forte multimodalidade, apresentando 10^n mínimos locais, e somente um ótimo global. Em duas dimensões ($n = 2$) tem-se $x^* = [0, 0]$, em que $f(x^*) = 0$. Novamente o número máximo de avaliações possíveis para a convergência é igual a 3000, e a distância Euclidiana mínima entre o ótimo conhecido e o ótimo estimado deve ser $\sqrt{0.02}$, conforme mostrado em (Vasconcelos et al. 2001).

• **Problema P3:**

O terceiro problema de teste é descrito na equação (5.7).

$$x^* = \arg \min_x f(\vec{x}) \triangleq 2.6164 + \frac{1}{n} \sum_{i=1}^n 0.01 [(x_i + 0.5)^4 - 30x_i^2 - 20x_i] \quad (5.7)$$

sujeito a: $x_i \in [-6, 6], i = 1, \dots, n$

Esta função possui característica multimodal com bacias mais ou menos bem espaçadas. Adota-se neste teste $n = 6$, sendo o ótimo global localizado em $x_i^* = -4.4538, \forall i \in [1, 6]$, em que $f(x^*) = 0$. A convergência só é verificada quando a distância Euclidiana entre o ótimo conhecido e o ótimo estimado for menor ou igual a 0.5, além disso, o número máximo de avaliações na função objetivo é igual a 10000.

• **Problema P4: Coello g09**

O último problema é apresentado na equação (5.8).

$$x^* = \arg \min_x f(\vec{x}) \triangleq (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2 + \dots$$

$$\dots + 10x_5^6 + 7x_6^2 + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7$$

sujeito a: $\begin{cases} g_1(\vec{x}) = -127 + 2x_1^2 + 3x_2^4 + x_3 + 4x_4^2 + 5x_5 \\ g_2(\vec{x}) = -282 + 7x_1 + 3x_2 + 10x_3^2 + x_4 - x_5 \\ g_3(\vec{x}) = -196 + 23x_1 + x_2^2 + 6x_6^2 - 8x_7 \\ g_4(\vec{x}) = 4x_1^2 + x_2^2 - 3x_1x_2 + 2x_3^2 + 5x_6 - 11x_7 \\ x_i \in [-10, 10], i = 1, \dots, n \end{cases} \quad (5.8)$

A função de *Coello* é definida em sete dimensões $n = 7$, e além disso apresenta quatro restrições de desigualdade. O ótimo global restrito deste problema encontra-se em $x^* = [2.330499, 1.951372, -0.4775414, 4.365726, -0.624487, 1.038131, 1.594227]$, onde $f(x^*) \approx 680.63$. O número máximo de avaliações da função objetivo até a convergência foi estipulado como sendo igual a 10000, e a distância Euclidiana mínima para convergência entre o ótimo restrito conhecido e o ótimo estimado foi igual a 0.5.

Estudo de Sensibilidade dos Parâmetros do DCSA

Esta subseção visa apresentar alguns resultados de convergência considerando-se os quatro problemas mencionados anteriormente, e a partir destes estipular uma faixa aceitável de valores para cada um dos parâmetros do DCSA. De forma a avaliar o efeito desses parâmetros no desempenho do algoritmo, cada parâmetro é variado isoladamente, mantendo-se os demais fixos. Conforme pode ser observado, o estudo de sensibilidade a ser realizado é muito simples, entretanto, acredita-se ser suficiente para estipular, com erros moderados, os valores desses parâmetros. A tabela 5.1 mostra os valores mínimos, máximos e fixos para cada um dos cenários de simulação.

Tabela 5.1: Faixa de valores para a análise de sensibilidade dos parâmetros do DCSA.

—	N_{pop}	N_{dN}	N_{dU}	N_{dC}	β	γ_N	γ_U	γ_C
mínimo	5.00	0.10	0.10	0.10	0.10	0.05	0.05	0.05
máximo	50.0	0.60	0.60	0.60	3.00	1.00	1.00	1.00
fixo	20.0	0.20	0.20	0.20	0.50	0.10	0.10	0.10

- Sensibilidade referente ao parâmetro N_{pop}

Pelas Figs. 5.5 e 5.6 tem-se que o desempenho do algoritmo proposto mostrou-se praticamente insensível à variação do tamanho da população quando diante do problema $\mathcal{P}1$, o que caracteriza a eficiência do método quanto a otimização de superfícies planas. Entretanto, quando diante dos problemas multimodais, $\mathcal{P}2$ e $\mathcal{P}3$, observa-se uma alta taxa de falha de convergência com valores reduzidos de N_{pop} . Por fim, durante o processo de otimização do problema restrito $\mathcal{P}4$, observa-se uma tendência de crescimento de ambos os índices, NAF e TFC, com o aumento excessivo da população. Neste caso, tem-se que uma maior população exige um número também maior de avaliações da função objetivo por geração, o que resulta em um menor número total de gerações vivenciadas pelas soluções estimadas, comprometendo assim a qualidade das mesmas. Tomando-se por base os quatro problemas estudados, percebe-se que os melhores valores encontrados para NAF e TFC correspondem aos valores de $N_{pop} \in [15, 30]$.

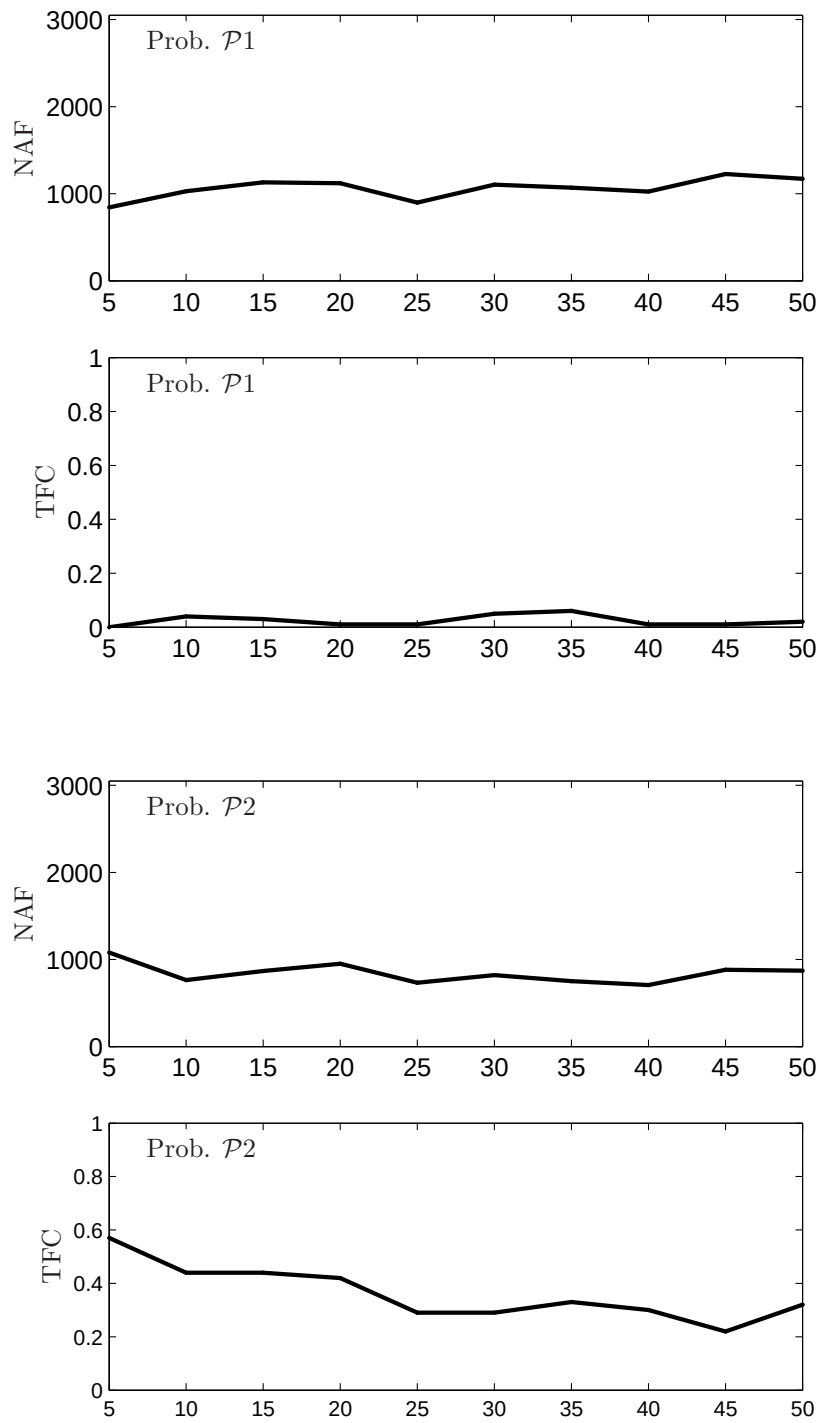


Figura 5.5: Sensibilidade referente ao parâmetro N_{pop} - $\mathcal{P}1$ e $\mathcal{P}2$.

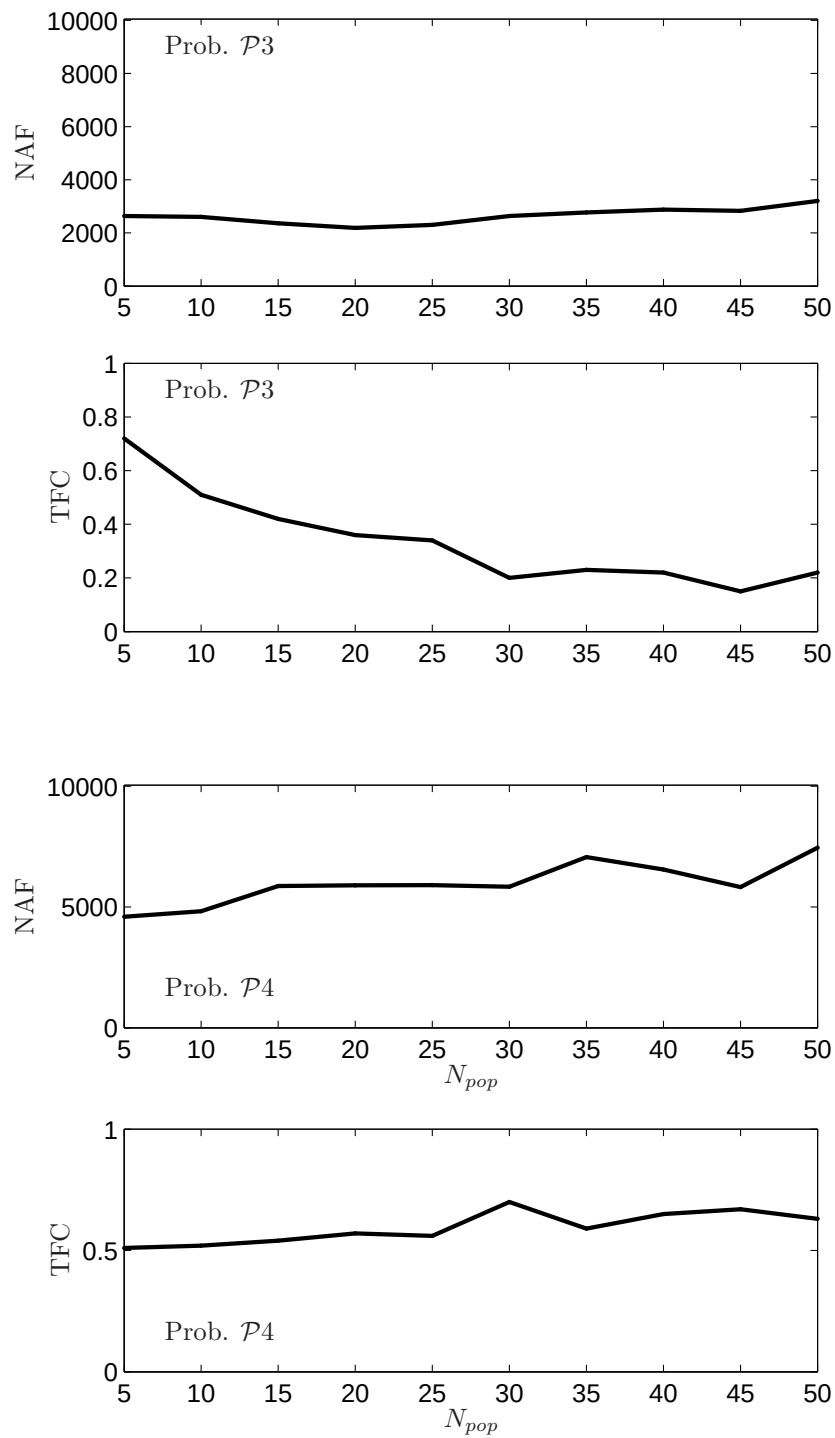


Figura 5.6: Sensibilidade referente ao parâmetro N_{pop} - $\mathcal{P}3$ e $\mathcal{P}4$.

- **Sensibilidade referente aos parâmetros N_{dN} , N_{dU} e N_{dC}**

De forma geral, observa-se pelas Figs. 5.7 e 5.8 que o desempenho do DCSA é pouco sensível à variação dos parâmetros N_{dN} , N_{dU} e N_{dC} , dando assim maior liberdade ao usuário na escolha dos mesmos, conforme a necessidade do problema de otimização. Dessa forma, dependendo das características de um dado problema, o usuário poderá especificar qual o tipo de busca a ser enfatizado, privilegiando então uma busca local mais intensa ou, se desejar, uma busca ampla mais intensa. Os resultados evidenciados pelos problemas multimodais, $\mathcal{P}2$ e $\mathcal{P}3$, mostram uma tendência de aumento da taxa de falha de convergência quando os parâmetros avaliados assumem valores superiores a 0.40, logo, sugere-se a adoção de valores tal que N_{dN} , N_{dU} , $N_{dC} \in [0.10, 0.30]$.

- **Sensibilidade referente aos parâmetros γ_N , γ_U e γ_C**

De acordo com as Figs. 5.9 e 5.10, o desempenho do DCSA mostrou-se bastante sensível ao parâmetro γ_N , entretanto, pouco vulnerável aos parâmetros γ_U e γ_C . Exceto pelo problema $\mathcal{P}2$, o método apresenta melhor desempenho para valores reduzidos de γ_N , normalmente pertencentes ao intervalo $[0.10, 0.30]$. Contudo, devido a forte característica multimodal observada em $\mathcal{P}2$, a busca torna-se mais eficiente para os valores de $\gamma_N \geq 0.30$, em que a taxa de falha de convergência aproxima-se de zero. Com relação aos demais parâmetros, sugere-se γ_U , $\gamma_C \in [0.20, 0.30]$.

- **Sensibilidade referente ao parâmetro β**

O estudo de sensibilidade do parâmetro β é apresentado nas Figs. 5.11 e 5.12. Enquanto o processo de otimização dos problemas $\mathcal{P}1$ e $\mathcal{P}4$ mostrou-se pouco sensível à variação de β , a taxa de falha de convergência nos problemas $\mathcal{P}2$ e $\mathcal{P}3$ apresenta uma suave elevação para valores de $\beta > 1$. Visto isso, com base nos problemas testes adotados tem-se que valores aceitáveis de NAF e TFC são obtidos quando $\beta \in [0.5, 1.0]$.

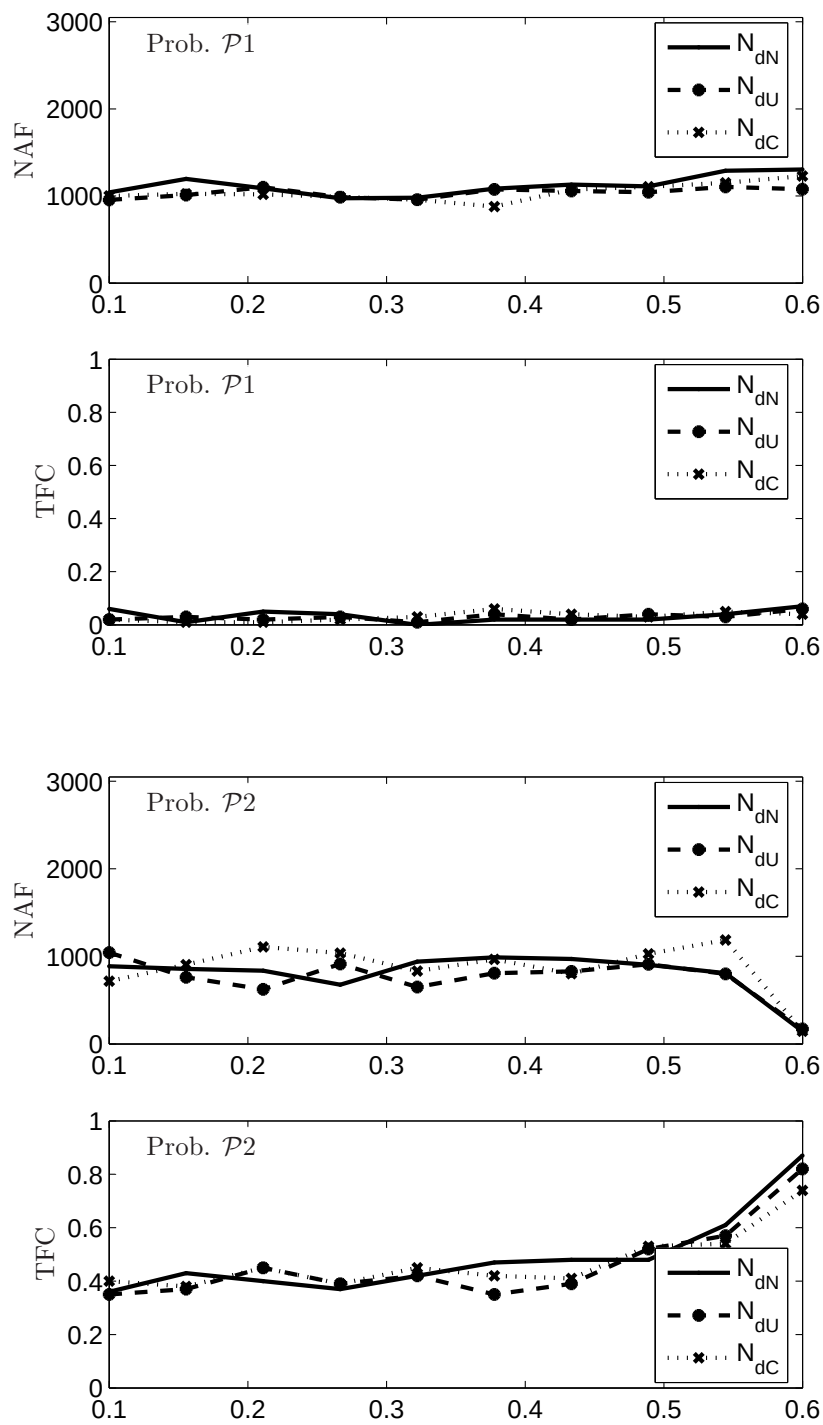


Figura 5.7: Sensibilidade referente aos parâmetros N_{dN} , N_{dU} e N_{dC} - $\mathcal{P}1$ e $\mathcal{P}2$.

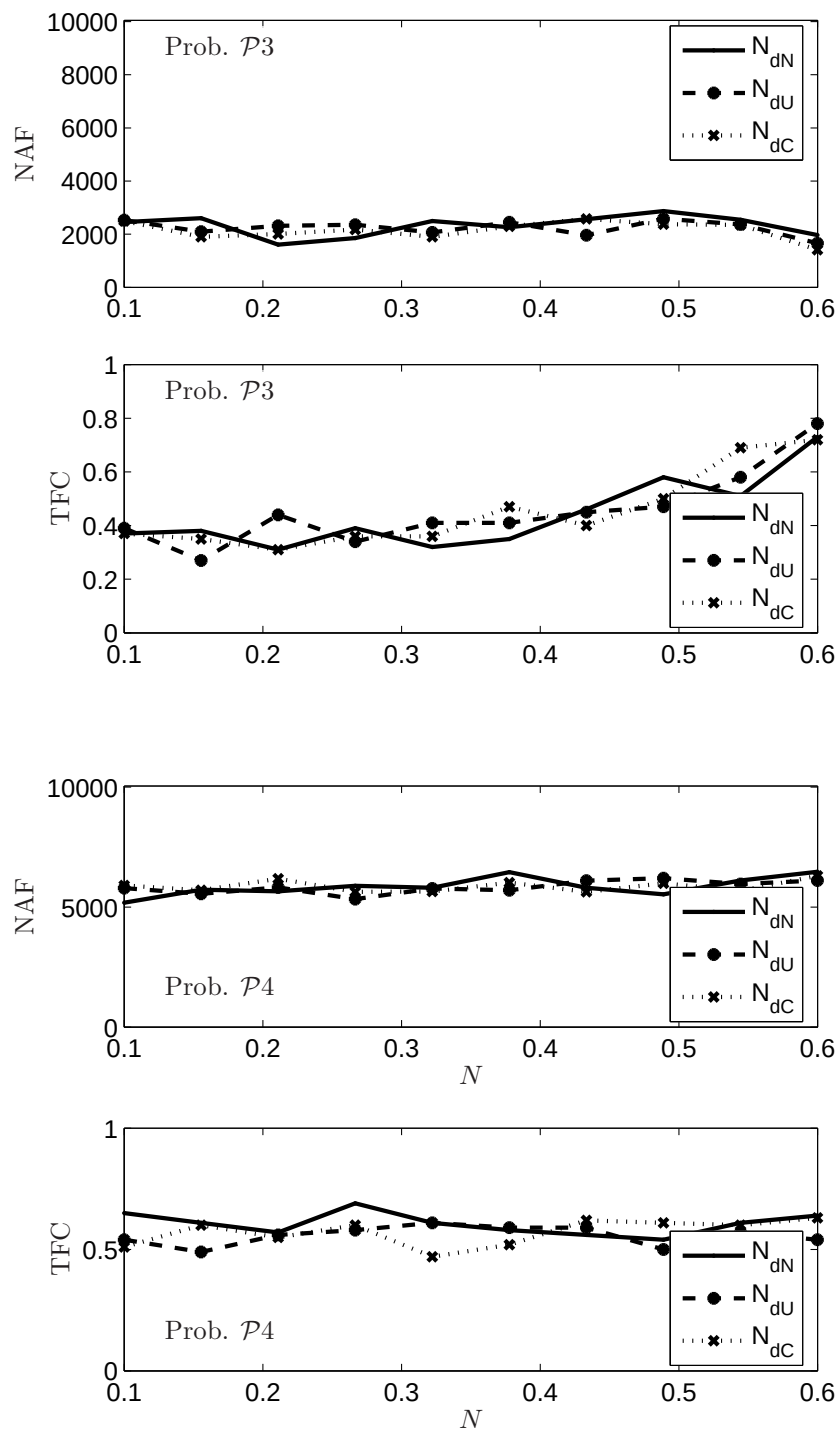


Figura 5.8: Sensibilidade referente aos parâmetros N_{dN} , N_{dU} e N_{dC} - $\mathcal{P}_3$ e $\mathcal{P}_4$.

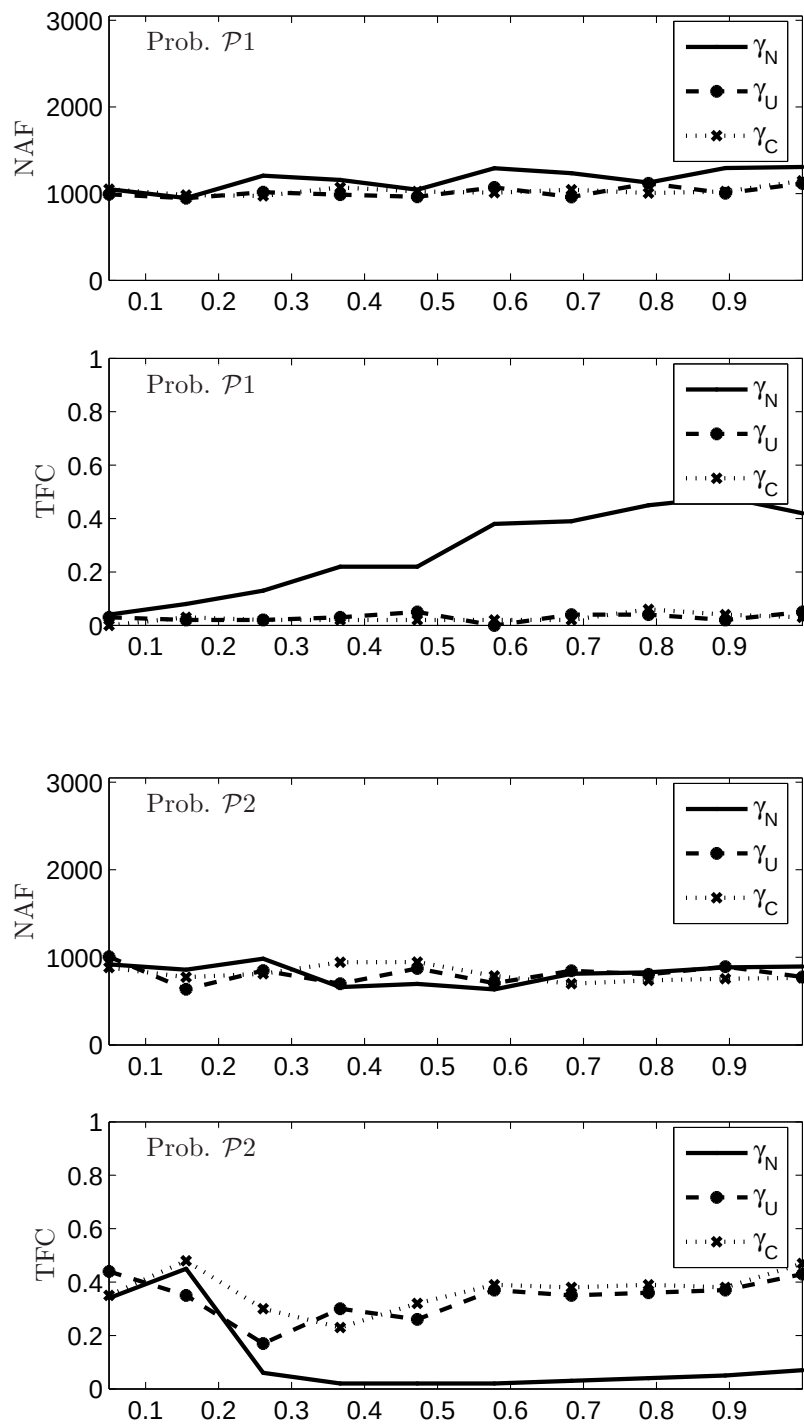


Figura 5.9: Sensibilidade referente aos parâmetros γ_N , γ_U e γ_C - $\mathcal{P}1$ e $\mathcal{P}2$.

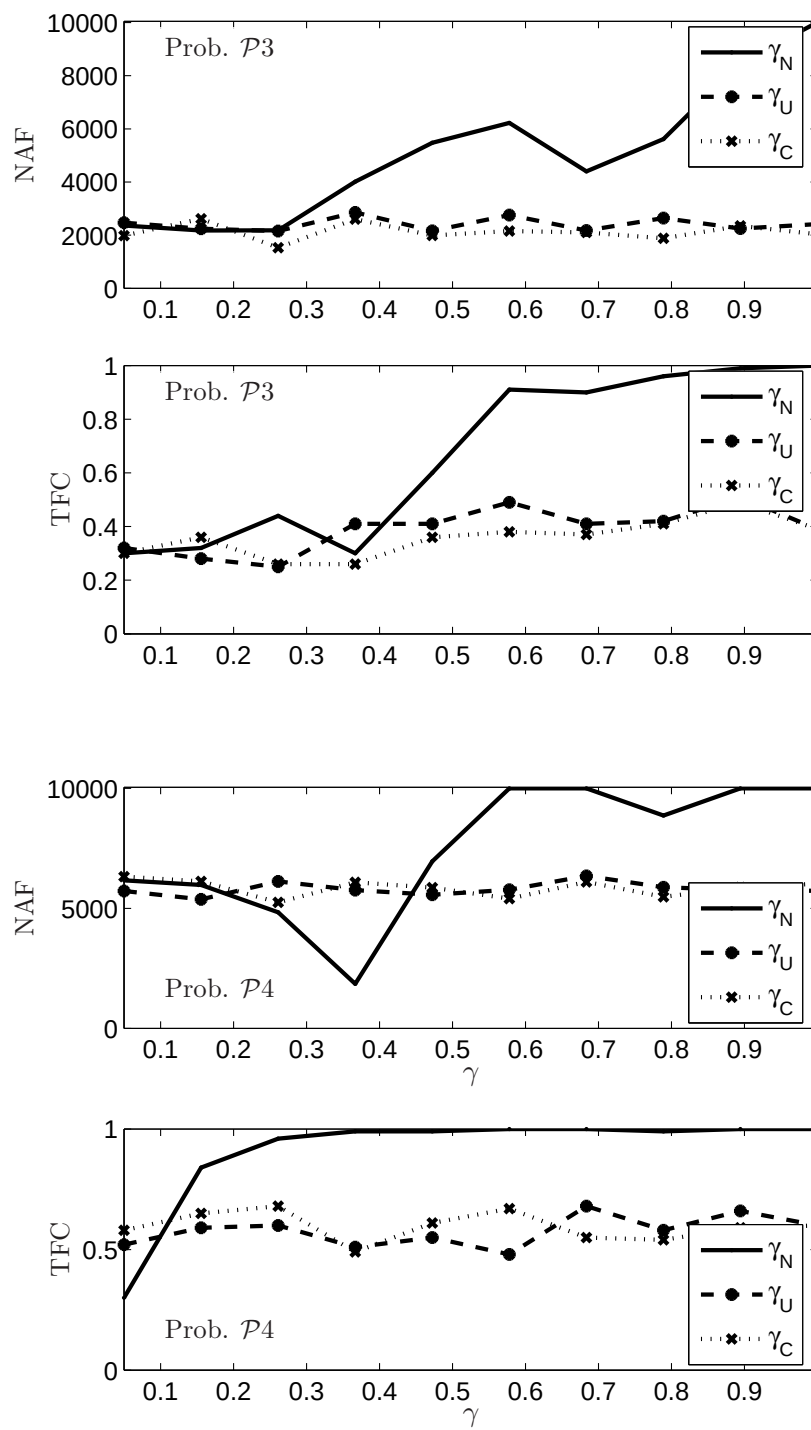


Figura 5.10: Sensibilidade referente aos parâmetros γ_N , γ_U e γ_C - P3 e P4.

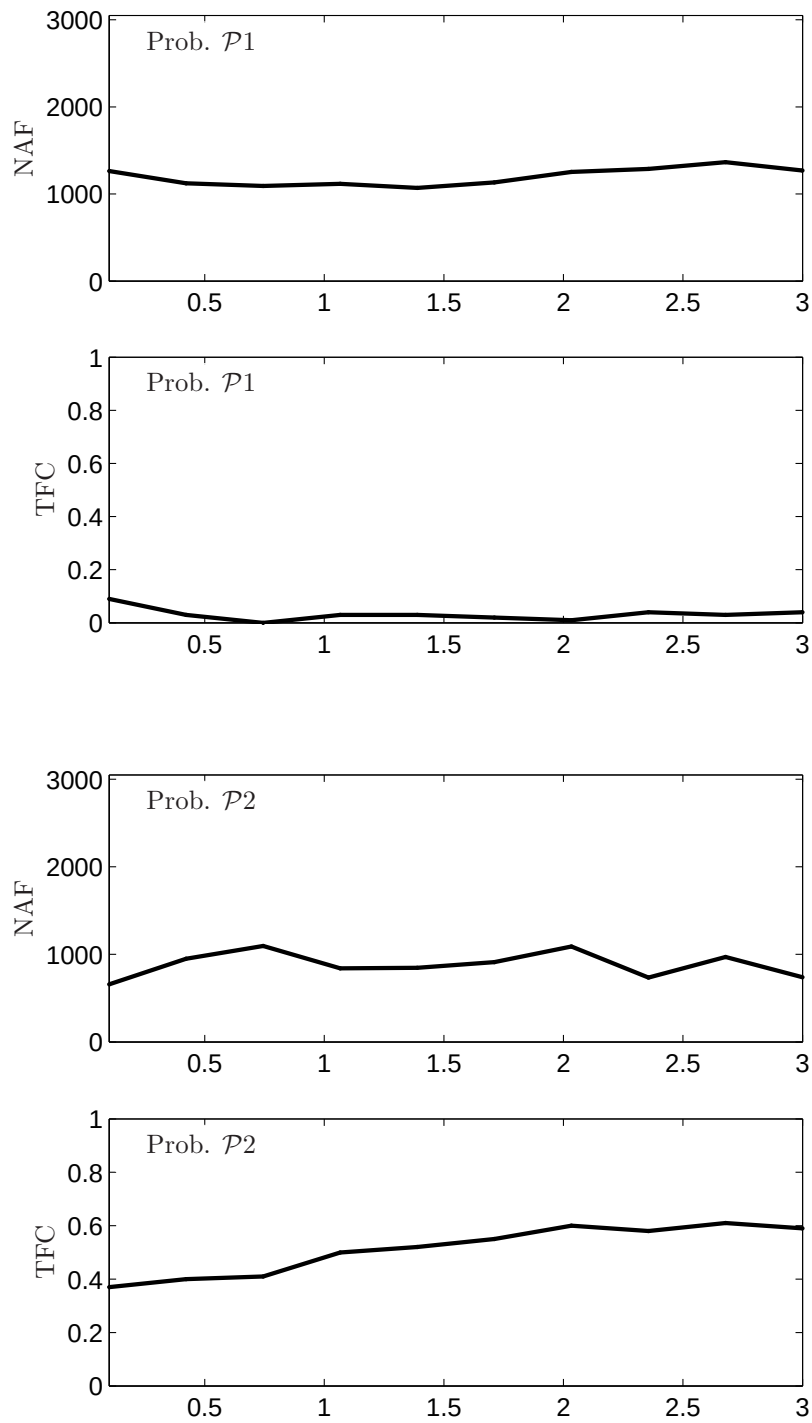


Figura 5.11: Sensibilidade referente ao parâmetro β - $\mathcal{P}1$ e $\mathcal{P}2$.

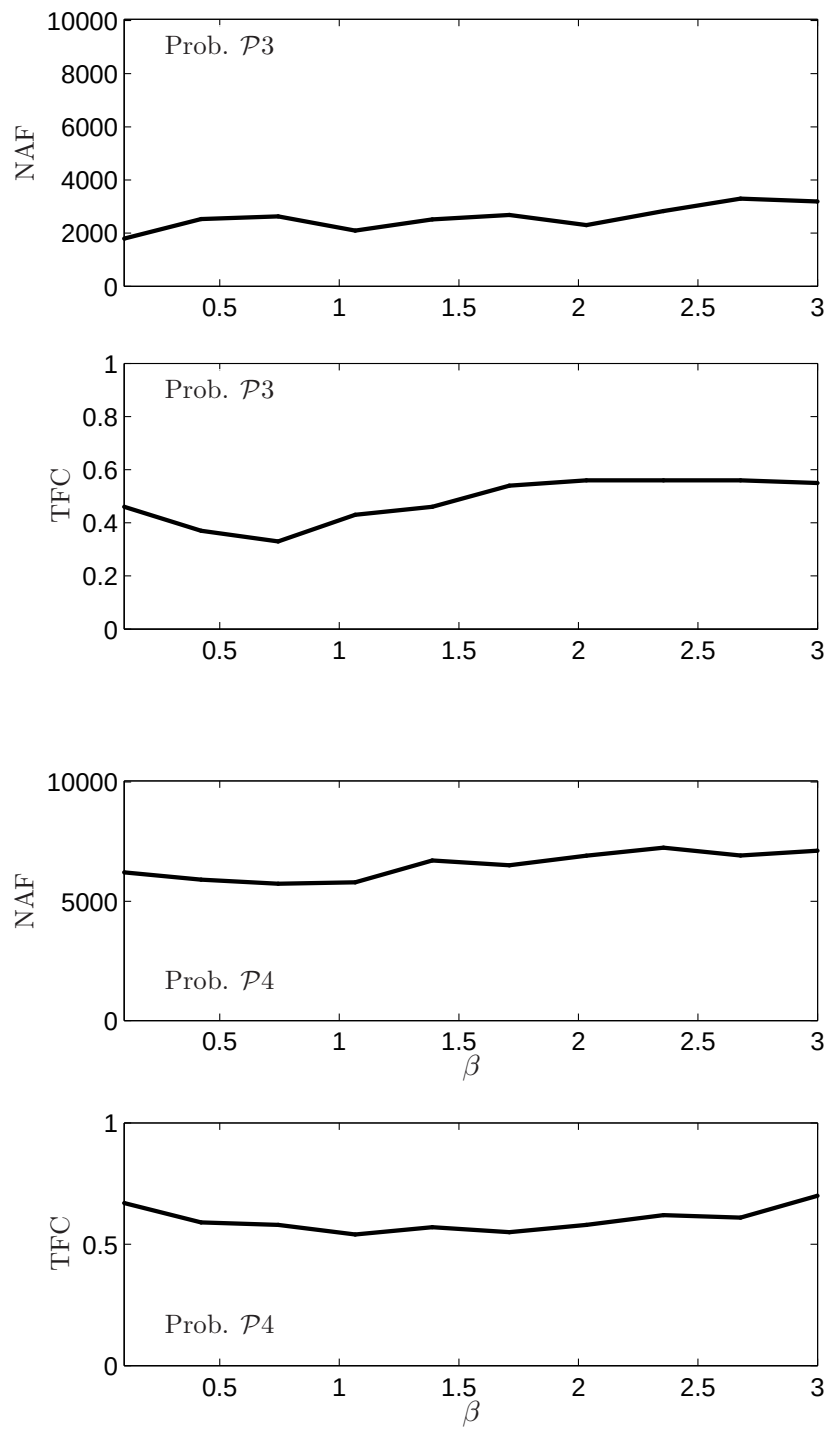


Figura 5.12: Sensibilidade referente ao parâmetro β - $\mathcal{P}3$ e $\mathcal{P}4$.

- **Sumário**

Finalmente, dados os testes realizados e os problemas empregados, sugere-se a adoção de valores para os parâmetros do DCSA, tal que:

- $N_{pop} \in [15, 30]$;
- $N_{dN}, N_{dU}, N_{dC} \in [0.10, 0.30]$;
- $\gamma_N, \gamma_U, \gamma_C \in [0.10, 0.30]$;
- $\beta \in [0.5, 1.0]$.

Nota. A contribuição apresentada nesta seção foi publicada na literatura especializada (Batista, Guimarães & Ramírez 2009b, Batista, Guimarães, Paul & Ramírez 2009), cópias no Apêndice [A](#).

5.3 “Multi-Objective Clonal Selection Algorithm”

O “Multi-Objective Clonal Selection Algorithm” (MCSA) propõe uma versão estendida do DCSA para otimização multi-objetivo. Além disso, o problema de otimização e as condições de otimalidade para o MCSA são as mesmas definidas nas seções 4.2 e 4.3, respectivamente.

Esse algoritmo se destaca principalmente pela forma como efetua o processo de *maturação por afinidade*, o qual é composto por dois mecanismos independentes, chamados *hipermutação somática* e *edição de receptores*. Assim sendo, durante a etapa de hipermutação somática são utilizadas diferentes funções densidade de probabilidade, com o objetivo de promover uma busca local balanceada sobre o espaço factível. De forma similar ao observado no DCSA, a distribuição é selecionada em função da qualidade das soluções obtidas. Já a etapa nomeada edição de receptores visa “reeditar” as melhores soluções encontradas, possibilitando às mesmas aperfeiçoarem suas características. Este operador adicional não emprega nenhuma distribuição de probabilidade explícita, mas efetua implicitamente uma busca dinâmica baseada em evolução diferencial, o que aumenta a velocidade de convergência do método.

Esses operadores e demais detalhes são discutidos durante a apresentação do algoritmo.

5.3.1 Descrição do algoritmo MCSA

Como é evidente pelo próprio nome, o MCSA baseia-se no princípio da seleção clonal e, por isso, é essencialmente definido por meio de quatro passos principais, chamados *avaliação por afinidade*, *clonagem*, *maturação por afinidade* e *geração de diversidade*. Cada um destes mecanismos é explicado nos próximos itens.

Avaliação por Afinidade

As soluções candidatas apresentadas ao algoritmo são avaliadas em todas as funções objetivo, penalizando qualquer violação de restrições, conforme mostrado na seção 3.4. Feito isso, cada solução recebe um valor escalar segundo as técnicas “fast nondominated sorting” e “crowding distance” discutidas em (Deb et al. 2000). Esse valor escalar

atribuído a cada solução representa a qualidade de sua afinidade, a qual é utilizada para ordenar a população estimada.

Clonagem

Nesta etapa, as N_{sel} melhores soluções presentes na população original de tamanho N_{pop} são selecionadas para clonagem. Essa seleção é efetuada de acordo com a fronteira de Pareto a qual cada solução pertence e segundo o valor atribuído pelo “crowding distance”. Os anticorpos escolhidos recebem então um número de clones (cópias) N_C^i dado pela equação (5.9):

$$N_C^i = round(\beta N_{pop} \exp(\bar{\xi}_i)) \quad (5.9)$$

em que $\beta \in [0, 1]$ é um parâmetro do algoritmo, $\bar{\xi}_i \in [0, 1]$ representa o valor do “crowding distance” normalizado da solução i , e a função $round(\cdot)$ arredonda o seu argumento para o inteiro mais próximo.

Observa-se que durante a execução do NSGA-II, os valores atribuídos pelo “crowding distance” às soluções localizadas nos extremos de cada fronteira são positivos infinitos. Entretanto, no algoritmo proposto esses valores são considerados iguais a 1.10 vezes o maior “crowding distance” encontrado, exceto o retornado pelos anticorpos extremos, que é positivo infinito. Dessa forma, estes índices de densidade são facilmente normalizados, sendo que aos pontos extremos é conferido $\bar{\xi} = 1$.

Segundo a equação (5.9), o número mínimo e máximo de clones gerados é respectivamente igual a βN_{pop} e $e\beta N_{pop}$, com uma suave variação entre esses limites (ver Fig. 5.13), o que garante a realização de uma busca com mais ou menos a mesma intensidade. Ainda assim, note que um número maior de clones é gerado para os anticorpos isolados e para os extremos em cada fronteira, enquanto menos clones são produzidos para as soluções pertencentes às regiões mais densas da fronteira.

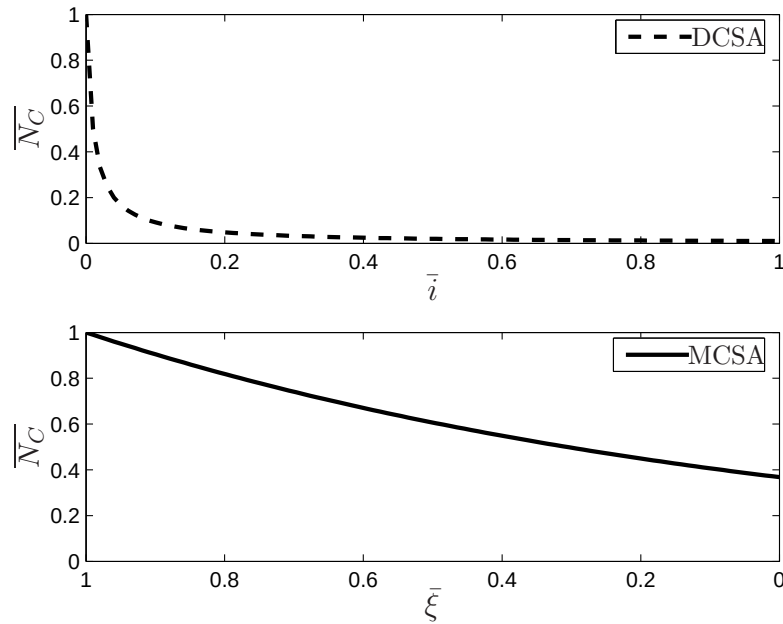


Figura 5.13: Número de clones gerados no MCSA em função do “crowding distance” normalizado $\bar{\xi}$. A título de comparação é apresentada a mesma curva (normalizada) para a versão mono-objetivo proposta (DCSA).

Maturação por Afinidade

Como sugerido por estudos recentes, o processo de maturação da resposta imune é desempenhado basicamente por dois mecanismos, os quais são *hipermutação somática* e *edição de receptores* (Nussenzweig 1998, George & Gray 1999, de Castro & Von Zuben 2000b).

De certa forma, a hipermutação somática permite ao sistema imune explorar a região local em torno de um ponto por meio de pequenos movimentos/perturbações, possibilitando a determinação de um anticorpo melhor. Já a edição de receptores provê um recurso adicional para a introdução de diversidade durante o processo de maturação, ou seja, este mecanismo oferece ao método a habilidade de escapar de ótimos locais localizados em uma dada bacia de atração. Devido a isso, acredita-se que o sistema de maturação, na ausência do operador de edição de receptores, apresentará dificuldades para determinar anticorpos com altos valores de afinidades, além de provavelmente desenvolver uma velocidade de convergência lenta.

Sendo mais específico, a hipermutação somática desempenha um importante papel na exploração de regiões locais do espaço de busca, enquanto a edição de receptores é

h  bil na detec  o de  reas pr speras (regi es de alta afinidade) em substitui  o a  reas n o muito atrativas. Essas t cnicas s o implementadas no MCSA como dois operadores independentes, os quais s o descritos a seguir.

- **Hipermuta  o Som tica**

Dada a popula  o de clones obtida a partir das N_{sel} melhores solu  es presentes na popula  o corrente, o operador de hipermuta  o consiste da adi  o de diferentes tipos de perturba  es aos mesmos. Essa etapa   similar a realizada pelo DCSA, assim sendo, os clones das primeiras N_{dN} melhores solu  es s o perturbados utilizando-se a distribui  o Gaussiana, os clones das subsequentes N_{dU} melhores solu  es s o perturbados utilizando-se a distribui  o uniforme, e os clones das N_{dC} solu  es restantes s o perturbados utilizando-se a distribui  o ca tica, tal que $N_{dN} + N_{dU} + N_{dC} = 1$. Mais uma vez, o emprego de diferentes fun  es densidade de probabilidade permite a realiza  o balanceada de um refinamento local ao redor das solu  es. Esse efeito   observado devido a considera  o de diferentes regi es de vizinha a durante a execu  o da busca local.

- **Edi  o de Receptores**

O operador de edi  o pode ser visto como um tipo especial de muta  o, efetuado de forma similar ao procedimento empregado por um algoritmo de evolu  o diferencial simples (“differential evolution algorithm” - DEA), ver se  o 3.6.5. De forma geral, tem-se que a capacidade de busca do DEA baseia-se principalmente na realiza  o de uma muta  o diferencial, na qual um vetor mutante $\vec{v}_i$   produzido a partir de tr s indiv duos distintos, selecionados aleatoriamente na popula  o:

$$\vec{v}_i = \vec{x}_{r1} + \omega (\vec{x}_{r2} - \vec{x}_{r3}) \quad (5.10)$$

Assim sendo, o MCSA adota um procedimento similar aplicado sobre os N_{sel} melhores anticorpos, o qual produz novas solu  es $\vec{v}_i$, $i \in [1, N_{sel}]$, de acordo com a equa  o (5.10), com  ndices mutuamente distintos e aleat rios $r1, r2, r3 \in [1, N_{sel}]$. Al m disso, $\omega > 0$   escolhido aleatoriamente no intervalo $[0.5, 1.0]$ e controla o tamanho do vetor diferencial $(\vec{x}_{r2} - \vec{x}_{r3})$.

Observe que o operador de edição cria implicitamente uma função densidade de probabilidade adaptativa, a qual baseia-se nos diferentes vetores diferenciais produzidos por meio das N_{sel} melhores soluções. Sendo mais claro, tem-se que no início do processo de otimização, as melhores soluções estão normalmente espalhadas em todo o domínio de busca, e com isso, os vetores de diferenças encontram-se bastante distribuídos em todas as direções e com diferentes valores de amplitudes. Essa característica é diretamente responsável pela inserção de diversidade na população. Entretanto, à medida que as melhores soluções se aproximam do conjunto Pareto-ótimo estimado, os vetores de diferenças ficam distribuídos ao longo da região do Pareto-ótimo, fazendo então com que o mecanismo de mutação deixe de atuar como um gerador de diversidade, e passe a trabalhar como um operador de busca local, a ser desempenhado ao redor das soluções de alta qualidade.

A Fig. 5.14 ilustra a ideia apresentada anteriormente. Todos os vetores de diferenças possíveis de serem obtidos a partir das N_{sel} melhores soluções estão evidenciados, sendo referenciados à origem; para este exemplo utilizou-se um problema bi-objetivo simples. Observe que os vetores de diferenças se adaptam ao longo das gerações, descrevendo a passagem entre os estágios de busca global para busca local. Para saber mais detalhes sobre a aplicação da mutação diferencial em sistema evolucionários multi-objetivo, recomenda-se a leitura do artigo (Batista, Guimarães & Ramírez 2009a).

Geração de Diversidade

Em adição à atuação dos operadores *hipermutação somática* e *edição de receptores*, realizados durante a fase de maturação por afinidade, uma pequena fração de novas células geradas aleatoriamente são adicionadas a população, matendo diversidade na mesma. Conforme discutido em (Jerne 1984), esse mecanismo é responsável pela substituição dos piores linfócitos por novas células, aproximadamente 5–8%. No MCSA, esta operação de substituição cria N_{rep} novos pontos aleatórios sobre a região de busca, desempenhando um mecanismo de geração de diversidade. De forma geral, tem-se que os operadores de edição e de diversidade desenvolvem papéis similares no início do processo de busca, entretanto, após algumas gerações a edição de receptores passa a favorecer o refinamento local das soluções de alta afinidade, enquanto o operador de substituição preserva sua característica de busca global.

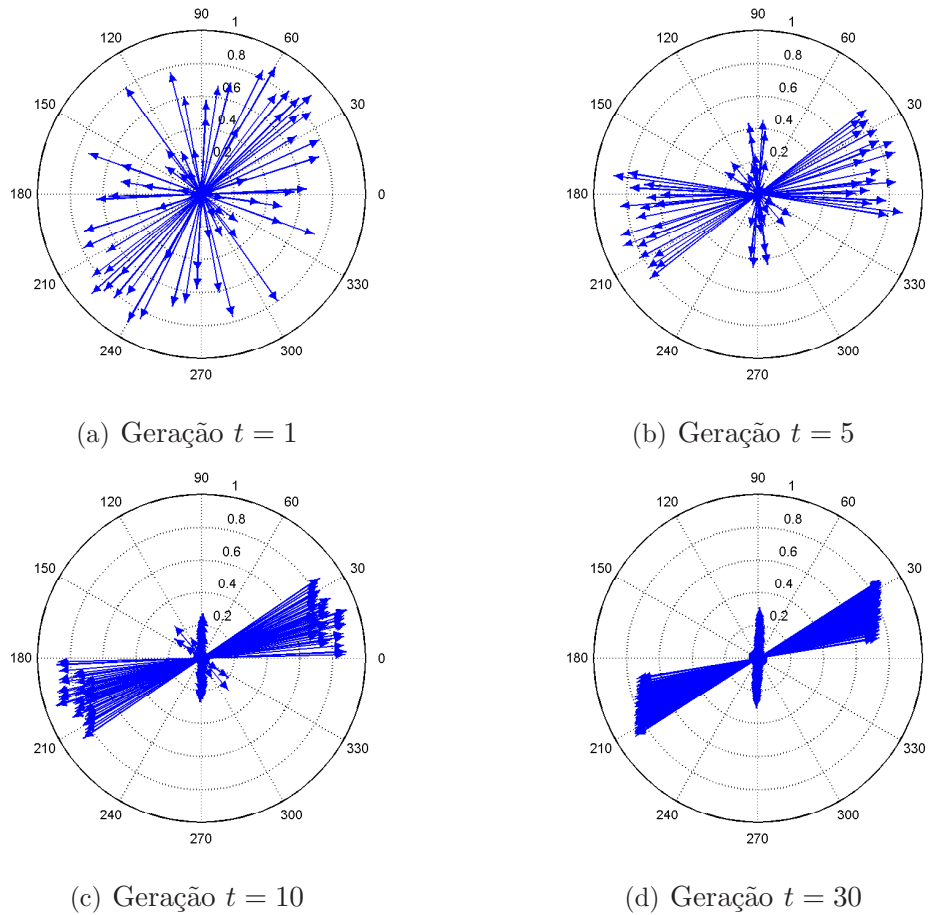


Figura 5.14: Efeito da aplicação do operador de mutação diferencial - figura adaptada de (Batista, Guimarães & Ramírez 2009a).

Todas as novas soluções - aquelas produzidas pelos operadores de hipermutação, edição e diversidade - são agrupadas às soluções presentes no arquivo externo, e então classificadas em fronteiras Pareto não-dominadas. Feito isso, a população de memória é atualizada, ou seja, esta armazena as soluções pertencentes à primeira fronteira $\mathcal{F}_1$, levando-se em conta o tamanho máximo L estipulado para o arquivo. Caso o número de soluções em $\mathcal{F}_1$ seja maior do que L , então somente aquelas de maior “crowding distance” são armazenadas. Observe que, quando possível, os N_{sel} melhores anticorpos são selecionados da primeira fronteira, entretanto, no caso em que o número de pontos em $\mathcal{F}_1$ é menor do que os N_{sel} necessários, as próximas fronteiras são consideradas, atribuindo-se $\bar{\xi} = 0$ para todos os pontos não contidos em $\mathcal{F}_1$. Note que pela equação (5.9), os anticorpos cujo $\bar{\xi} = 0$ recebem o menor número de clones.

Por fim, as N_{pop} melhores soluções, isto é, aquelas menos dominadas e com maior “crowding distance”, são selecionadas para a próxima geração, mantendo assim uma população de tamanho constante.

Algoritmo

Finalmente, os passos anteriores são sumarizados na estrutura cíclica descrita no Alg. 5.3. Apresenta-se ainda o diagrama de blocos mostrado na Fig. 5.15.

Algoritmo 5.3: Estrutura de funcionamento do MCSA.

Input: Objetivos, restrições, espaço de busca, N_{pop} , N_{sel} , $N_{dN,dU,dC}$, $\gamma_{N,U,C}$, β , L

Output: Estimativa do conjunto Pareto-ótimo $\mathcal{A}(n)$

```

1 begin
2    $\mathcal{P}(n) = \{\vec{p}_1, \dots, \vec{p}_N\} \leftarrow$  População inicial;
3    $\Phi(n) \leftarrow$  Avaliação por afinidade ( $\mathcal{P}(n)$ );
4    $\mathcal{A}(n) = \emptyset \leftarrow$  Arquivo inicial;
5   while Não critério de parada do
6      $\mathcal{F}(n) \leftarrow$  Classificação em fronteiras não-dominadas ( $\mathcal{P}(n), \Phi(n)$ );
7      $\mathcal{I}(n) \leftarrow$  Determinação dos valores do “crowding distance” ( $\mathcal{F}(n)$ );
8      $\mathcal{S}(n) \leftarrow$  Seleção dos melhores anticorpos ( $\mathcal{I}(n), N_{sel}$ );
9      $\mathcal{C}(n) \leftarrow$  Realização do processo de clonagem ( $\mathcal{S}(n)$ );
10     $\mathcal{Q}(n) \leftarrow$  Mecanismo de hipermutação somática ( $\mathcal{C}(n)$ );
11     $\mathcal{E}(n) \leftarrow$  Mecanismo de edição de receptores ( $\mathcal{S}(n)$ );
12     $\mathcal{R}(n) \leftarrow$  Substituição e geração de diversidade ( $N_{rep}$ );
13     $\mathcal{P}(n+1) \leftarrow (\mathcal{F}_1(n) \cup \mathcal{Q}(n) \cup \mathcal{E}(n) \cup \mathcal{R}(n))$ ;
14     $\Phi(n+1) \leftarrow$  Avaliação por afinidade ( $\mathcal{P}(n+1)$ );
15     $\mathcal{A}(n+1) \leftarrow$  Atualização ( $\mathcal{P}(n) \cup \mathcal{A}(n), L$ );
16     $n = n + 1$ ;
17  end
18 end
```

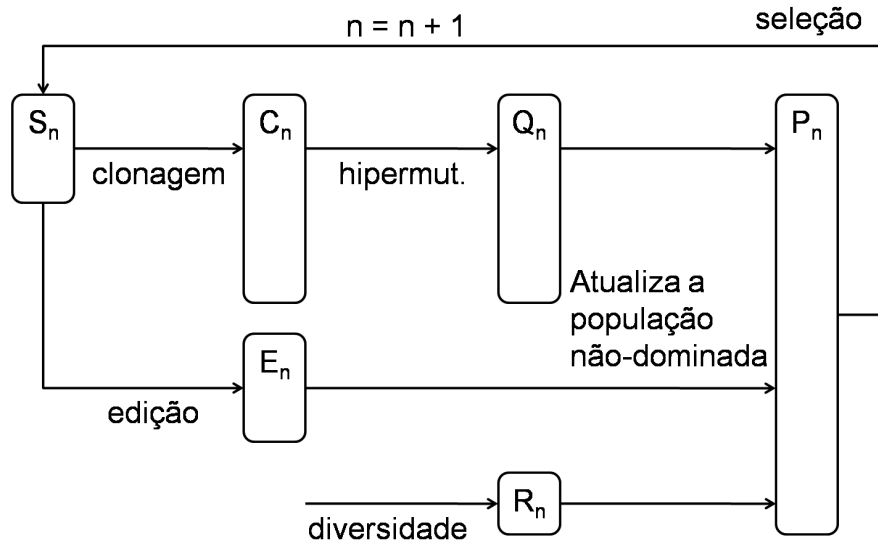


Figura 5.15: Diagrama de blocos da evolução da população no MCSA.

5.3.2 Calibração dos Parâmetros do MCSA

Como o MCSA é muito similar ao DCSA, não considera-se importante o estudo de sensibilidade dos parâmetros daquele. Embora esses métodos empreguem equações distintas para o cálculo do número de clones (rever Fig. 5.13), o parâmetro β pode ser facilmente adequado ao MCSA. Em função da característica exponencial mais suave da curva adotada para o cálculo de N_C^i na versão multi-objetivo, sugere-se menores valores para β , pertencentes ao intervalo $[0.1, 0.5]$, sendo os demais parâmetros escolhidos conforme apresentado na versão mono-objetivo ($N_{pop} \in [15, 30]$, N_{dN} , N_{dU} , $N_{dC} \in [0.10, 0.30]$ e γ_N , γ_U , $\gamma_C \in [0.10, 0.30]$).

Nota. A contribuição apresentada nesta seção foi publicada na literatura especializada (Batista, Guimarães & Ramírez 2009c), cópia no Apêndice A.

5.4 Conclusão

A contribuição dessa dissertação foi descrita ao longo do corpo deste capítulo, onde apresentou-se isoladamente as duas abordagens imunes propostas. De forma geral, após ter sido realizada uma explicação detalhada acerca do funcionamento dos métodos, e formalizada a estrutura cíclica de cada um, estudou-se a sensibilidade da versão mono-objetivo (DCSA) diante da variação de seus parâmetros de controle, sendo estabelecidas faixas aceitáveis aos mesmos. Como a versão multi-objetivo (MCSA) herdou grande parte das características do DCSA, o ajuste de seus parâmetros de otimização basearam-se também na análise de sensibilidade empregada no caso mono-objetivo, com pequenas alterações das variáveis de controle.

Embora nenhum teste tenha sido realizado até o momento, pode-se enumerar algumas expectativas com respeito ao efeito dos operadores *hipermutação somática* e *edição de receptores* no desempenho dos métodos. Assim sendo, há fortes indícios de que devido as diferentes funções densidades de probabilidades (*pdf*) adotadas na etapa de hipermutação, as soluções ótimas estimadas sejam melhor distribuídas tanto sobre a região de busca no DCSA, quanto ao longo da região de Pareto no MCSA. Além disso, espera-se que a evolução diferencial implementada na fase de edição acelere a convergência do MCSA. Essas características são melhor esclarecidas no capítulo seguinte.

Capítulo 6

Resultados

“Os estudos aperfeiçoam a natureza e são aperfeiçoados pela experiência.”

— Francis Bacon, 1561–1626 (filósofo inglês)

“Faça as coisas o mais simples que puder, porém não as mais simples.”

— Albert Einstein, 1879–1955 (físico alemão)

6.1 Introdução

Neste capítulo são apresentados os principais resultados referentes aos métodos propostos, “Distributed Clonal Selection Algorithm” (DCSA) e “Multi-Objective Clonal Selection Algorithm” (MCSA). De forma geral, a avaliação de desempenho é efetuada com base em comparações entre os sistemas imunes sugeridos e algoritmos de renome da literatura. Inicialmente, o DCSA é posto à prova diante de vários problemas analíticos com diferentes características de otimização, sendo submetido também a dois problemas eletromagnéticos, os quais correspondem a um transformador de impedância de guia de ondas retangular (“Rectangular Waveguide Impedance Transformer” - WIT), e a um dispositivo supercondutor armazenador de energia (“Superconducting Magnetic Energy Storage” - SMES). Posteriormente, o MCSA é avaliado com a ajuda de algumas técnicas de análise de desempenho, sendo posto à prova frente a problemas analíticos convexos, não convexos e desconexos, e ainda diante da versão bi-objetivo, 3D e 8D, do problema eletromagnético SMES.

6.2 Resultados referentes ao DCSA

Nesta seção o desempenho do DCSA é comparado com alguns dos métodos descritos no capítulo 3. Inicialmente são realizados testes quanto a velocidade de convergência destes algoritmos, e posteriormente são resolvidos os problemas relacionados aos dispositivos de microonda e eletromagnético.

6.2.1 Problemas Analíticos

Os métodos mono-objetivo a serem avaliados são os seguintes: DCSA, CLONALG, BCA, RCSA e RBGA. Os parâmetros de otimização de cada um destes algoritmos são apresentados na tabela 6.1.

Tabela 6.1: Parâmetros de otimização empregados na solução dos problemas analíticos.

(a) DCSA		(b) CLONALG		(c) BCA		(d) RCSA		(e) RBGA	
N_{pop}	15	N_{pop}	20	N_{pop}	04	N_{pop}	15	N_{pop}	20
N_{dN}	0.30	N_{sel}	0.70	ρ_{meta}	0.50	N_{sel}	0.60	ρ_{cruz}	0.50
N_{dU}	0.20	N_{rep}	0.20	η_C	N_{pop}	β	0.80	$\rho_{cruz-pol}$	0.30
N_{dC}	0.20	β	1.0	L	64	α	0.10	ρ_{mut}	0.02
β	0.50	ρ	3.0	T_{max}	48				
γ_N	0.10	L	64						
γ_U	0.25								
γ_C	0.25								

A capacidade de otimização destes algoritmos é verificada tomando-se por base os problemas testes $\mathcal{P}1$ e $\mathcal{P}3$ utilizados durante a análise de sensibilidade do DCSA (subseção 5.2.3). A formulação matemática destes problemas é rerepresentada a seguir.

- Problema $\mathcal{P}1$ ($n = 2$): Rosenbrock

$$x^* = \arg \min_x f(\vec{x}) \triangleq \sum_{i=1}^{n-1} \left[100 (x_i^2 - x_{i+1})^2 + (1 - x_i)^2 \right] \quad (6.1)$$

sujeito a: $x_i \in [-2.48, 2.48]$, $i = 1, \dots, n$

- Problema $\mathcal{P}3$ ($n = 3$):

$$x^* = \arg \min_x f(\vec{x}) \triangleq 2.6164 + \frac{1}{n} \sum_{i=1}^n 0.01 [(x_i + 0.5)^4 - 30x_i^2 - 20x_i] \quad (6.2)$$

sujeito a: $x_i \in [-6, 6], i = 1, \dots, n$

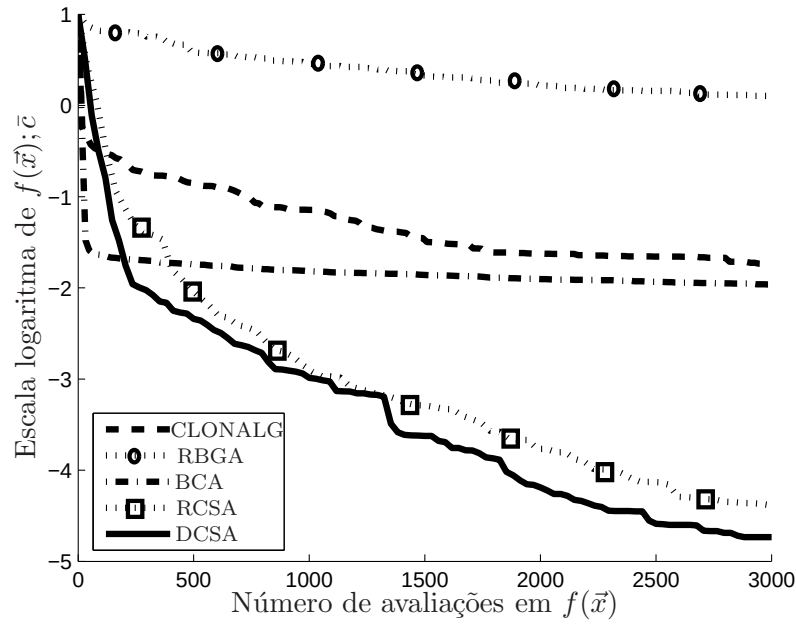
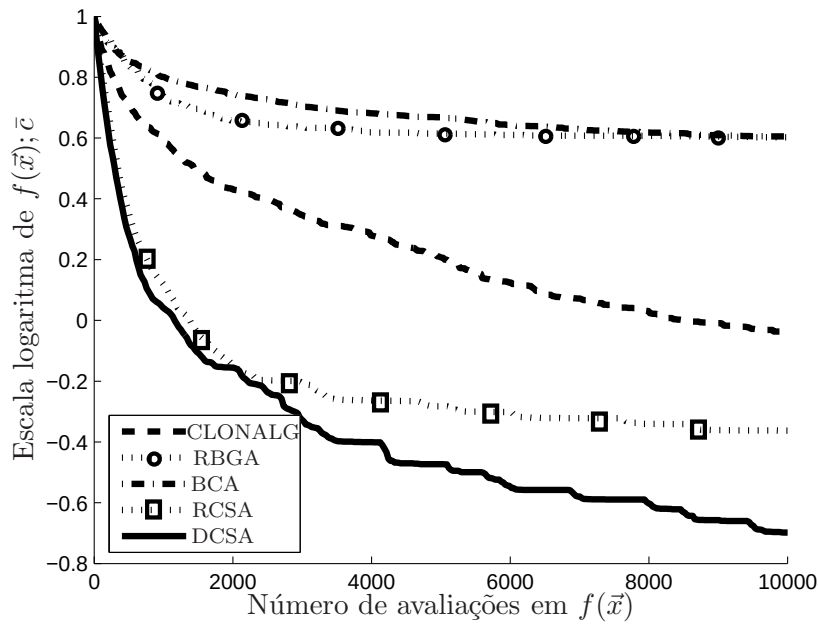
A métrica utilizada para avaliar o desempenho dos métodos baseia-se na análise das curvas médias de velocidade de convergência ($\bar{c}$). A obtenção dessa curva para cada método requer os seguintes passos:

1. Realiza-se t execuções do algoritmo, armazenando a melhor solução de cada geração no arquivo $A(n)$, $n = 1, \dots, t$;
2. Obtém-se o arquivo $\bar{A} = \text{mean}(A(1), \dots, A(t))$, correspondente à média dos t vetores;
3. Finalmente, calcula-se a curva média de velocidade de convergência segundo a equação (6.3).

$$\bar{c} = 1 + \log \left(\sqrt{\frac{\bar{A}}{\bar{A}_1}} \right) \quad (6.3)$$

A Fig. 6.1 apresenta os resultados obtidos pelos métodos avaliados. A saber, considerou-se $t = 50$ execuções, sendo adotado como critério de parada o número máximo de avaliações da função objetivo, em que estipulou-se 3000 para o problema $\mathcal{P}1$ e 10000 para $\mathcal{P}3$.

Como pode ser observado em ambos os problemas sob teste, o algoritmo proposto apresenta um desempenho muito superior ao alcançado pelos métodos CLONALG, BCA e RBGA. Nota-se, contudo, que durante o processo de otimização da função de “Rosenbrock”, os algoritmos DCSA e RCSA revelam um desempenho similar até atingir, aproximadamente, 1300 avaliações da função objetivo, instante a partir do qual o DCSA se mostra mais apto para encontrar regiões factíveis de melhor qualidade. Esse mesmo cenário é observado durante a otimização do problema multimodal $\mathcal{P}3$, em que o DCSA revela-se superior ao RCSA somente para valores de avaliações no objetivo maiores que 2500.

(a) Problema $\mathcal{P}1$: função de "Rosenbrock"(b) Problema $\mathcal{P}3$ **Figura 6.1:** Curva média de velocidade de convergência observada nos problemas analíticos.

Como discutido no capítulo 5, o DCSA representa uma extensão do RCSA, sendo diferenciado principalmente pelo emprego de funções densidade de probabilidade distintas. De forma geral, observa-se que o desempenho alcançado por estes métodos é similar, a menos da maior taxa de velocidade descrita pelo DCSA, o que certamente deve-se ao efeito desenvolvido pela distribuição caótica adotada, revelando assim o seu importante papel em sistemas de otimização.

6.2.2 Problemas Eletromagnéticos

Os experimentos realizados com os dispositivos de microonda e eletromagnético consideram somente os algoritmos DCSA, BCA e RCSA. Como o nível de dificuldade destes problemas é mais expressivo, se comparado ao esforço computacional exigido pelos problemas analíticos, são adotados os novos parâmetros de otimização mostrados na tabela 6.2.

Tabela 6.2: Parâmetros de otimização empregados na solução dos problemas eletromagnéticos.

(a) DCSA			(b) BCA			(c) RCSA		
—	<i>SMES</i>	<i>WIT</i>	—	<i>SMES</i>	<i>WIT</i>	—	<i>SMES</i>	<i>WIT</i>
N_{pop}	30	15	N_{pop}	06	04	N_{pop}	30	15
N_{dN}	0.30	0.30	ρ_{meta}	0.50	0.50	N_{sel}	0.60	0.60
N_{dU}	0.20	0.20	η_C	N_{pop}	N_{pop}	β	0.50	0.80
N_{dC}	0.20	0.20	L	64	64	α	0.10	0.10
β	0.50	0.50	T_{max}	48	48			
γ_N	0.10	0.10						
γ_U	0.25	0.25						
γ_C	0.25	0.25						

“Rectangular Waveguide Impedance Transformer” (WIT)

A configuração física do dispositivo de microonda é apresentada na Fig. 6.2, sendo que o transformador de impedância (WIT) (Paul et al. 2008) corresponde a seção retangular intermediária, situada entre as seções de guia de ondas transversais extremas. O componente emissor é definido por uma seção transversal de área $a_1 \times b_1$ e comprimento l_1 ,

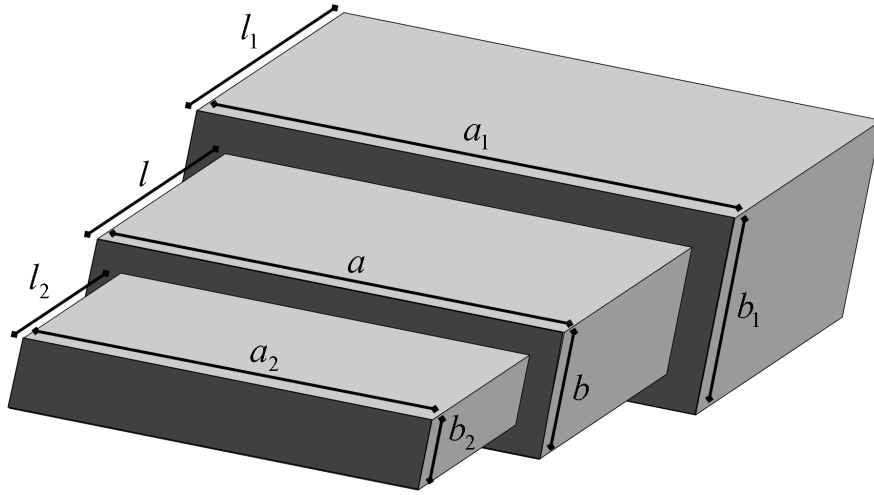


Figura 6.2: Configuração do transformador de impedância de guia de ondas retangular (WIT).

enquanto o componente receptor possui seção transversão de área $a_2 \times b_2$ e comprimento l_2 .

Dada a presente configuração, objetiva-se determinar os valores a , b e l , que definem a seção intermediária, tal que o coeficiente de reflexão seja minimizado. A saber, durante a solução deste problema, o coeficiente de reflexão é calculado por meio de parâmetros de dispersão, obtidos via análise de elementos finitos em 3 dimensões. Além disso, o dispositivo em questão é projetado para operar entre 9.75 e $10.25GHz$.

A formulação matemática do problema de otimização (WIT) é mostrada na equação (6.4):

$$x^* = \arg \min_x f(\vec{x}) \triangleq 20 \log_{10} |S_{11}|$$

$$\text{sujeito a: } \begin{cases} a_2 = 20mm < a < a_1 = 24mm \\ b_2 = 4mm < b < b_1 = 8mm \\ 5mm < l < 20mm \end{cases} \quad (6.4)$$

em que as variáveis de otimização são $\vec{x} = [a; b; l]$, e S_{11} representa o parâmetro de dispersão. Vale mencionar que para a obtenção de S_{11} ao longo da faixa de frequências estabelecidas, utiliza-se “Asymptotic Waveform Evaluation” (AWE), (Jiao et al. 1999).

As tabelas 6.3, 6.4 e 6.5 mostram um conjunto de soluções ótimas obtidas por meio de uma única execução de cada um dos algoritmos. O mesmo critério de parada foi

Tabela 6.3: Resultados obtidos para o dispositivo de microondas (WIT) - DCSA.

Parâmetro	$f(\vec{x})$	a	b	l
Unidade	dB	mm	mm	mm
DCSA	-29.9502	21.6125	5.6321	10.1186
	-29.9500	21.6123	5.6346	10.1261
	-29.9498	21.6113	5.6339	10.1099
	-29.9490	21.6119	5.6379	10.1109

Tabela 6.4: Resultados obtidos para o dispositivo de microondas (WIT) - RCSA.

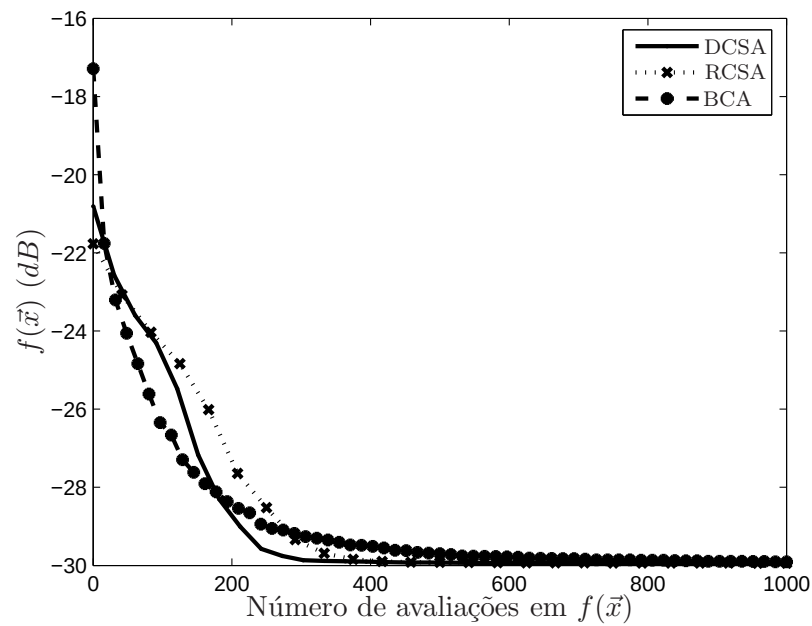
Parâmetro	$f(\vec{x})$	a	b	l
Unidade	dB	mm	mm	mm
RCSA	-29.9495	21.6149	5.6358	10.1031
	-29.9493	21.6179	5.6302	10.1209
	-29.9482	21.6176	5.6412	10.1178
	-29.9478	21.6148	5.6396	10.0980

Tabela 6.5: Resultados obtidos para o dispositivo de microondas (WIT) - BCA.

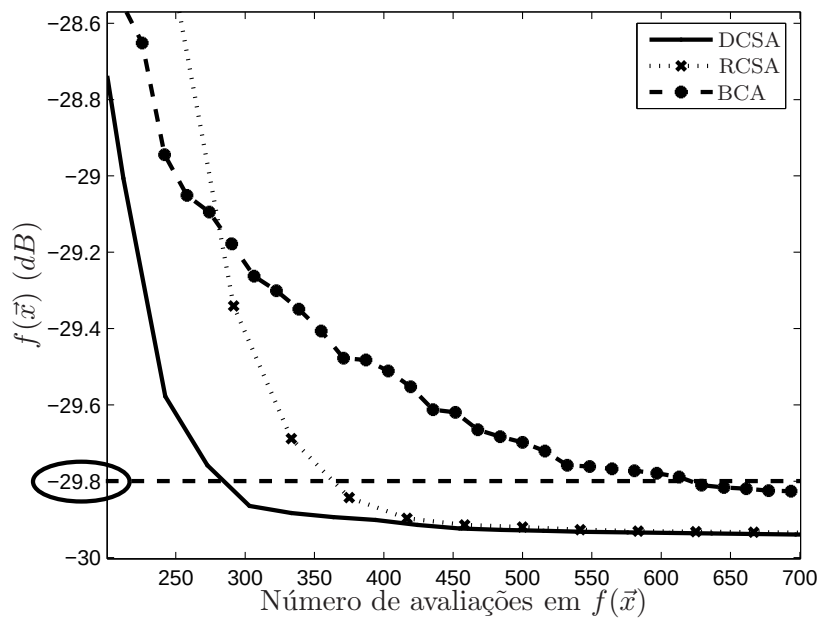
Parâmetro	$f(\vec{x})$	a	b	l
Unidade	dB	mm	mm	mm
BCA	-29.9456	21.6252	5.6306	10.1326
	-29.9408	21.6300	5.6421	10.1447
	-29.9400	21.6260	5.6321	10.1643
	-29.9337	21.5913	5.6241	10.1316

adotado, onde considerou-se um número máximo de avaliações da função objetivo igual a 1000. Como está evidente pelas tabelas apresentadas, os três métodos foram capazes de encontrar ótimos muito similares, com perdas por retorno próximas a $-30dB$.

Uma ideia mais clara a respeito do desempenho alcançado pelos métodos DCSA, RCSA e BCA, pode ser obtida a partir da análise das curvas de velocidade de convergência mostradas na Fig. 6.3. Para a realização deste teste, executou-se cada algoritmo 100 vezes, tomando-se o vetor mediano de ótimos alcançados ao longo das gerações.



(a) Curvas sem “zoom”



(b) Curvas com “zoom”

Figura 6.3: Curva média de velocidade de convergência observada no problema de microondas (WIT).

Como pode ser observado pela Fig. 6.3(a), o BCA mostra uma maior velocidade de convergência inicial, porém reduz a mesma ao longo do processo de otimização. Isso revela que embora este método seja capaz de efetuar uma busca global inicial mais eficiente, o mesmo não possui um mecanismo eficaz de refinamento local das soluções. De forma geral (ver Fig. 6.3(b)), tem-se que o DCSA encontra melhores soluções com um menor custo computacional. Por exemplo, tomando-se como referência a perda por retorno de $-29.80dB$, o número de avaliações consumidas em média por cada método foi, aproximadamente, 280, 360 e 620, para o DCSA, RCSA e BCA, respectivamente.

“Superconducting Magnetic Energy Storage” (SMES)

Em um dispositivo SMES, uma grande quantidade de corrente é armazenada no campo magnético, com o objetivo de ser fornecida ao sistema elétrico quando necessário. Em função das baixas perdas observadas na bobina supercondutora, a transferência de energia pelo dispositivo é altamente eficiente e muito rápida. Entretanto, para preservar o estado de supercondutividade, a bobina necessita de um mecanismo de refrigeração sofisticado, que seja capaz de mantê-la a temperaturas criogênicas (Schoenung et al. 1996).

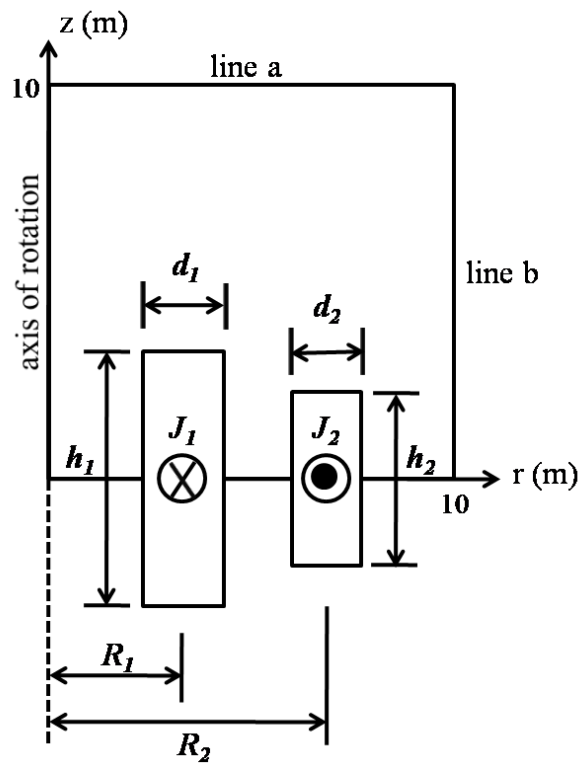
Uma definição matemática para o dispositivo eletromagnético descrito anteriormente, incluindo suas restrições físicas, é apresentada em (Alotto et al. 1996), onde os autores propõem um problema multi-objetivo, cuja avaliação dos objetivos é de alto custo computacional.

Nessa seção, emprega-se uma versão mono-objetivo para o problema 22 obtida a partir do estudo realizado em (Dias & Vasconcelos 2002), a qual consiste da minimização da densidade de fluxo magnético de dispersão a uma certa distância do dispositivo.

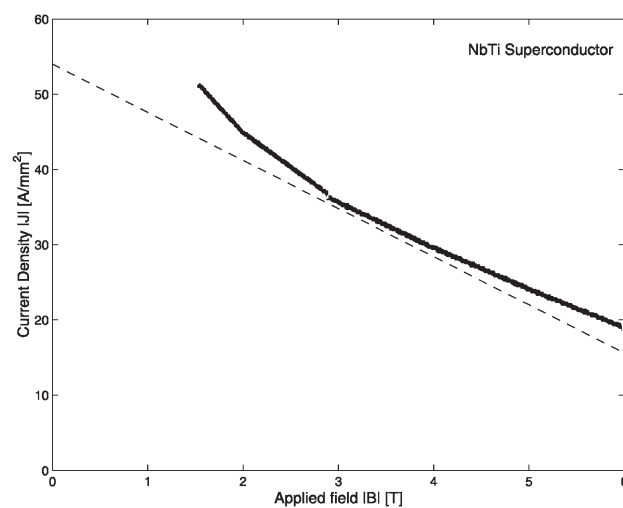
A função objetivo é definida como a minimização do valor de B_{Stray} , conforme mostrado na equação (6.5):

$$x^* = \arg \min_x f(\vec{x}) \triangleq B_{Stray} = \sqrt{\frac{\sum_{i=1}^{21} |B_{Si}|^2}{21}} \quad (6.5)$$

em que B_{Si} representa a densidade de fluxo magnético avaliado em cada um dos 21 pontos, distribuídos uniformemente ao longo das linhas a e b mostradas na Fig. 6.4(a).



(a) Configuração do dispositivo



(b) Condição de supercondutividade

Figura 6.4: “Superconducting Magnetic Energy Storage” (SMES).

Este problema possui três restrições principais. A primeira delas, de igualdade, está relacionada a quantidade de energia armazenada no SMES (equação (6.6)):

$$h(\vec{x}) = \frac{Energy - E_{ref}}{E_{ref}} = 0 \quad (6.6)$$

onde $E_{ref} = 180 MJ$; e as demais restrições são de desigualdade (equação (6.7)):

$$\begin{aligned} g_1(\vec{x}) &= |J_i| + 6.4 |B_{max_i}| - 54 \leq 0 \\ g_2(\vec{x}) &= (R_1 - R_2) + \frac{1}{2} (d_1 + d_2) < 0 \end{aligned} \quad (6.7)$$

em que $g_1(\vec{x})$ formaliza a condição de supercondutividade do dispositivo (Fig. 6.4(b)), e $g_2(\vec{x})$ garante a não-superposição entre as bobinas interna e externa.

Como auxílio à resolução deste problema, emprega-se o “software” livre FEMM, “Finite Element Method Magnetics”, (Meeker 1994–2000).

• SMES em 3 Dimensões

O sistema de otimização em 3D é definido pelos parâmetros mostrados na tabela 6.6, e conforme pode ser observado, as variáveis se relacionam somente à bobina externa, as quais são r_2 , h_2 e d_2 , mantendo-se as demais fixas.

A tabela 6.7 apresenta o conjunto de soluções ótimas encontradas ao longo de uma única execução do DCSA, em que adotou-se um critério de parada igual a 1000 avaliações no objetivo. Como está evidenciado, todas as soluções encontradas respeitam a restrição de energia com um erro percentual máximo de 0.10%. Além disso, o volume de material magnético descrito pela configuração da bobina externa, ($V = 2\pi r_2 h_2 d_2$), é comparável ao associado à solução ótima conhecida na literatura. Essa comparação torna-se relevante uma vez que o custo financeiro do metro cúbico de material supercondutor é elevado.

Tabela 6.6: Variáveis de otimização para o problema SMES 3D.

Parâmetro	r_1	h_1	d_1	r_2	h_2	d_2	J_1	J_2
Unidade	m	m	m	m	m	m	MA/m^2	MA/m^2
mínimo	—	—	—	2.6	0.408	0.1	—	—
máximo	—	—	—	3.4	2.2	0.4	—	—
fixo	2.0	1.6	0.27	—	—	—	22.5	22.5

A melhor solução encontrada pelo DCSA, primeira da tabela 6.7, é um pouco inferior ao ótimo conhecido, apresentando um erro de 1.14% em relação ao valor de B_{Stray} , entretanto, proporciona uma indução magnética máxima menor. As soluções ótimas apresentadas por (Campelo et al. 2005) e (Takahashi et al. 2003) consumiram 512 e 2400 avaliações na função objetivo, respectivamente.

A Fig. 6.5 apresenta as configurações obtidas considerando-se a melhor solução encontrada pelo DCSA e a solução ótima conhecida na literatura. Como pode ser observado, somente as bobinas externas sofreram modificações, uma vez que os parâmetros da bobina interna são fixos. Nesta figura são ilustradas também as linhas de fluxo magnético e a densidade de fluxo magnético, a qual é expressa em função das diversas tonalidades mostradas.

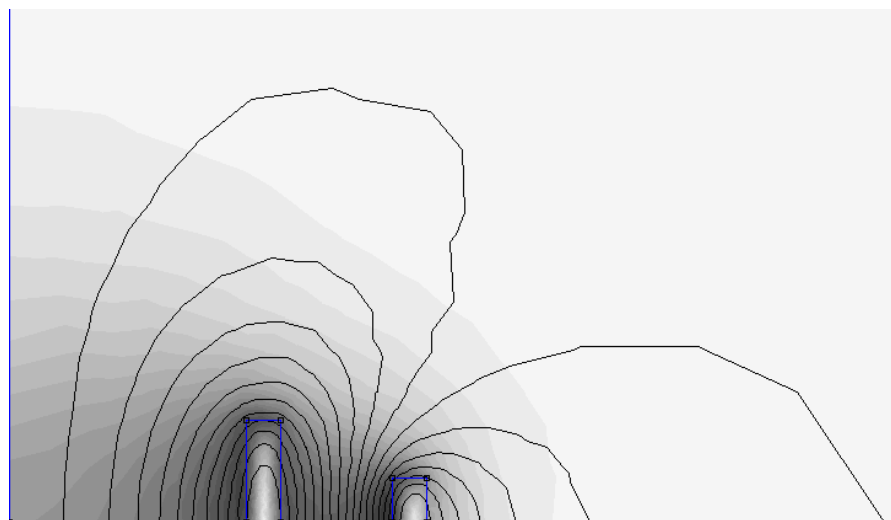
Tabela 6.7: Soluções ótimas encontradas para o problema mono-objetivo SMES 3D.

Parâmetro Unidade	R_2 m	h_2 m	d_2 m	B_{Stray} mT	B_{max} T	$Energy$ MJ	$Erro$ $\%$	$Volume$ m^3
DCSA	3.1417	0.7031	0.2665	0.8997	3.9510	180.13	0.07	3.70
—	3.0959	0.5219	0.3670	1.0398	4.4355	179.94	0.03	3.73
—	3.0440	0.5330	0.3860	1.1090	4.7369	179.87	0.07	3.93
—	3.0577	0.5154	0.3928	1.1575	4.6822	180.19	0.10	3.89
—	3.0197	0.6161	0.3496	1.4625	4.4956	180.07	0.04	4.09
—	2.9785	0.6452	0.3557	2.1186	4.7982	180.01	0.00	4.29
RCSA ¹	3.1176	0.6017	0.3152	0.8890	4.1348	179.91	0.05	3.71
RBGA ²	3.0500	0.4920	0.4000	0.9642	4.8550	175.43	2.54	3.77
TEAM ³	3.0800	0.4780	0.3940	0.8896	4.6230	180.03	0.02	3.65

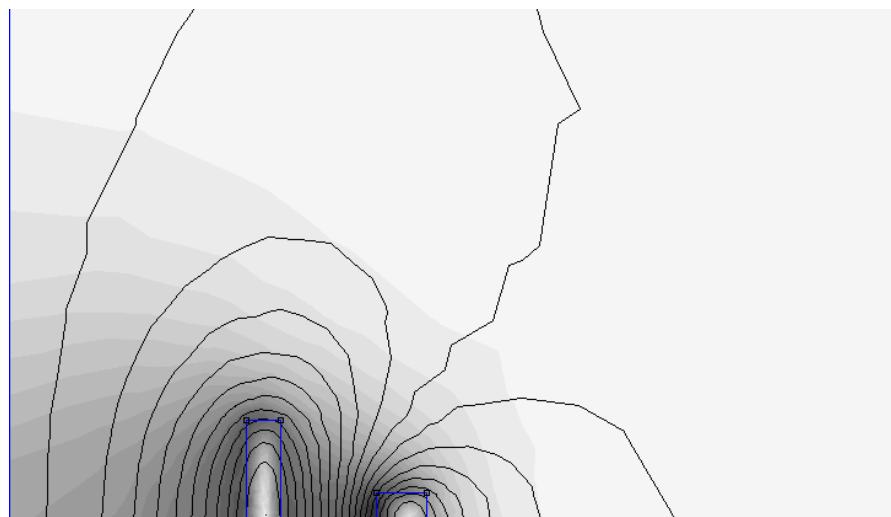
¹Resultado extraído de (Campelo et al. 2005).

²Resultado extraído de (Takahashi et al. 2003).

³Resultado extraído de (Alotto et al. 1996) - solução ótima do SMES 3D conhecida na literatura.



(a) DCSA



(b) TEAM

Figura 6.5: Configuração otimizada obtida para o SMES 3D mono-objetivo.

- SMES em 8 Dimensões

O desempenho do DCSA é também avaliado quando diante do problema eletromagnético SMES de 8 dimensões. A título de comparação são novamente utilizados os métodos RCSA e BCA. Os parâmetros de otimização, bem como a faixa de valores factíveis estabelecida para cada uma das variáveis, são mostrados na tabela 6.8.

Tabela 6.8: Variáveis de otimização para o problema SMES 8D.

Parâmetro	r_1	h_1	d_1	r_2	h_2	d_2	J_1	J_2
Unidade	m	m	m	m	m	m	MA/m^2	MA/m^2
mínimo	1.0	0.2	0.1	1.8	0.2	0.1	10	10
máximo	4.0	3.6	0.8	5.0	3.6	0.8	30	30

Os resultados do processo de otimização estão evidenciados na tabela 6.9, onde apresenta-se o conjunto de ótimos estimados, obtidos ao longo de uma única execução de cada um dos algoritmos. O mesmo critério de parada foi adotado, em que estipulou-se o número máximo de avaliações da função objetivo igual a 2000.

Embora não estejam impressos na tabela de resultados, pode-se observar que o erro percentual em relação a restrição de energia varia entre 0.30 – 2.30% para o DCSA, entre 0.30 – 1.40% para o RCSA, e entre 1.50 – 10.10% para o BCA, o que revela um desempenho similar dos dois primeiros métodos quanto ao atendimento da restrição de igualdade $h(\vec{x})$. Entretanto, se observado outros parâmetros, percebe-se que o desempenho do DCSA é superior ao dos outros métodos, pois além de encontrar soluções com densidade de fluxo de dispersão menores, os valores associados de indução magnética máxima nas bobinas também são inferiores aos observados nos ótimos determinados pelo RCSA e BCA.

Tabela 6.9: Soluções ótimas encontradas para o problema mono-objetivo SMES 8D.
(a) “Distributed Clonal Selection Algorithm”

Parâmetro	<i>DCSA</i>				TEAM ¹
$R_1[m]$	2.5037	1.8226	2.5006	1.9902	1.5703
$h_1[m]$	2.2761	1.8625	2.1108	1.7140	1.5692
$d_1[m]$	0.5226	0.3154	0.5338	0.3084	0.5943
$R_2[m]$	3.1185	2.3961	3.1077	2.6136	2.0999
$h_2[m]$	3.7406	3.1901	3.5995	3.1250	2.8368
$d_2[m]$	0.0975	0.1956	0.1025	0.1832	0.2562
$J_1[MA/m^2]$	10.3674	23.5680	10.9063	23.1993	17.3367
$J_2[MA/m^2]$	22.0921	12.7373	21.9398	12.2607	12.5738
$B_{Stray}[mT]$	0.2008	0.2537	0.2809	0.5029	0.0148
$B_{max}[T]$	3.3732	4.0817	3.5021	3.6893	—
$Energy[MJ]$	176.47	180.63	181.53	175.91	179.99

(b) “Real-Coded Clonal Selection Algorithm”

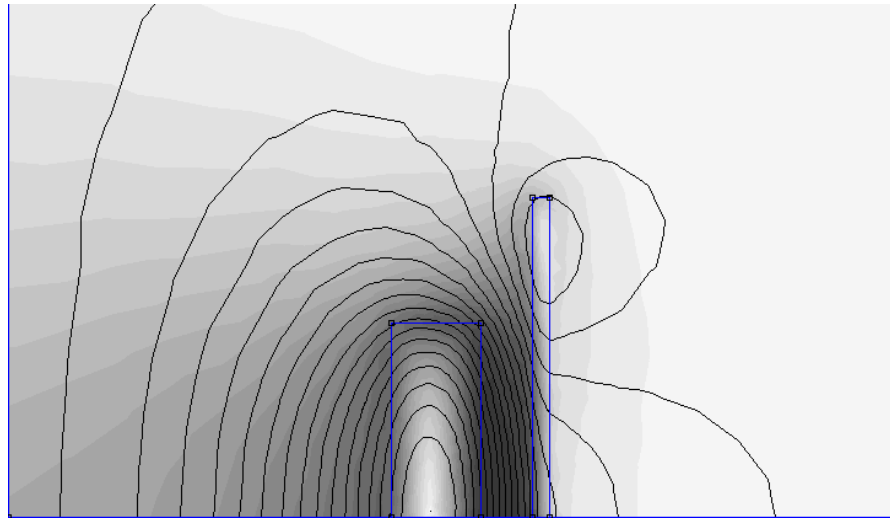
Parâmetro	<i>RCSA</i>				TEAM ¹
$R_1[m]$	1.7392	1.7034	1.7401	1.9628	1.5703
$h_1[m]$	1.5922	1.5314	1.0693	1.0385	1.5692
$d_1[m]$	0.3872	0.4111	0.5780	0.5340	0.5943
$R_2[m]$	2.5574	2.5109	3.0195	3.3036	2.0999
$h_2[m]$	1.1541	1.2042	0.6018	1.2753	2.8368
$d_2[m]$	0.2113	0.2088	0.2336	0.1672	0.2562
$J_1[MA/m^2]$	19.5459	19.6241	16.6529	17.1195	17.3367
$J_2[MA/m^2]$	24.0252	23.8510	25.9853	16.7766	12.5738
$B_{Stray}[mT]$	0.4862	0.5519	0.6757	0.8344	0.0148
$B_{max}[T]$	4.4390	4.2899	5.1607	4.7598	—
$Energy[MJ]$	179.42	181.88	178.86	177.54	179.99

(c) “B-Cell Algorithm”

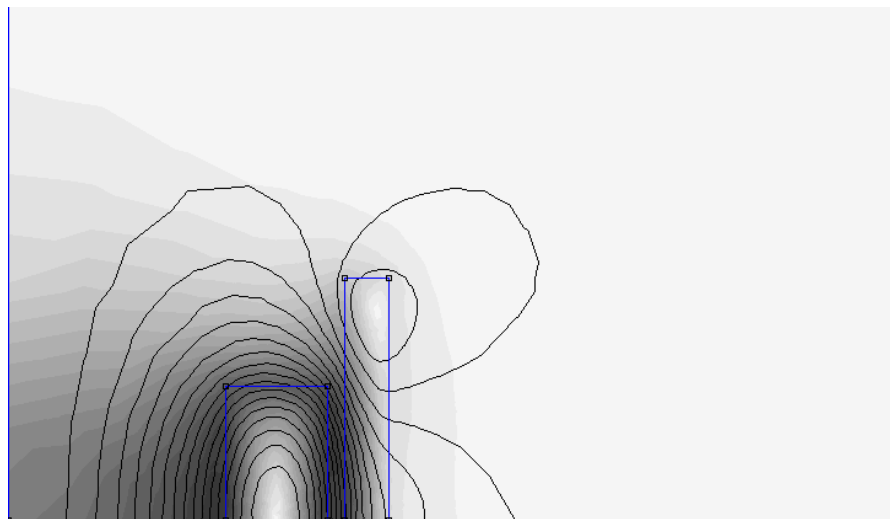
Parâmetro	<i>BCA</i>				TEAM ¹
$R_1[m]$	1.5272	2.0738	1.3137	2.0599	1.5703
$h_1[m]$	1.8302	0.9275	1.4567	1.1552	1.5692
$d_1[m]$	0.3078	0.4802	0.7024	0.3646	0.5943
$R_2[m]$	2.6475	3.1049	3.2239	3.3416	2.0999
$h_2[m]$	0.7248	1.1621	0.5208	1.7992	2.8368
$d_2[m]$	0.4784	0.1941	0.4301	0.1813	0.2562
$J_1[MA/m^2]$	23.6075	19.7495	12.9713	22.0599	17.3367
$J_2[MA/m^2]$	13.3768	18.4074	10.7866	11.2575	12.5738
$B_{Stray}[mT]$	0.5547	0.7121	0.8097	0.8852	0.0148
$B_{max}[T]$	4.4800	4.4680	6.3793	4.4897	—
$Energy[MJ]$	195.87	161.79	166.77	177.33	179.99

¹Resultado extraído de (Alotto et al. 1996) - solução ótima do SMES 8D conhecida na literatura.

A Fig. 6.6 apresenta as configurações obtidas considerando-se a primeira solução encontrada pelo DCSA (Tabela 6.9) e a solução ótima conhecida na literatura. Novamente, são ilustradas as linhas de fluxo magnético e a densidade de fluxo magnético.



(a) DCSA



(b) TEAM

Figura 6.6: Configuração otimizada obtida para o SMES 8D mono-objetivo.

6.2.3 Discussão

O método DCSA posto à prova nos itens anteriores, assim como o seu antecessor RCSA, representam sistemas imunes artificiais para otimização mono-objetivo, os quais, por sua vez, empregam uma representatividade real dos seus parâmetros e utilizam mecanismos de variação e seleção natural muito simples e eficientes. Devido a essas características, estes algoritmos são dedicados principalmente a processos de otimização envolvendo dispositivos eletromagnéticos, onde os custos computacionais são geralmente elevados.

O ganho expresso pelo DCSA, quando comparado ao RCSA, é obtido por meio do emprego de diferentes funções densidade de probabilidade durante a etapa de maturação das soluções estimadas. Essa mutação distribuída, a qual consiste no diferencial deste método, permite ao algoritmo promover uma busca mais eficiente sobre a região factível, possibilitando o escape de ótimos locais pobres, e a determinação de áreas prósperas ainda não exploradas.

Tomando-se por base todos os experimentos realizados, tanto analíticos quanto eletromagnéticos (WIT e SMES), observa-se que nenhum dos métodos utilizados dominou o DCSA, o qual demonstrou grande habilidade na determinação das soluções ótimas estimadas. Esse fato, certamente confere ao algoritmo proposto um importante papel como ferramenta de otimização de problemas reais de alto custo computacional, mais especificamente, dispositivos eletromagnéticos.

6.3 Resultados referentes ao MCSA

O desempenho obtido pelo “Multi-Objective Clonal Selection Algorithm” (MCSA) é avaliado de forma similar ao realizado para o DCSA. Assim sendo, apresenta-se inicialmente alguns testes analíticos, com a finalidade de melhor compreender a atuação dos operadores do MCSA, e, posteriormente, estuda-se o desempenho desse método frente ao problema eletromagnético SMES. Os sistemas empregados para a comparação de algoritmos multi-objetivo são descritos a seguir.

6.3.1 Técnicas de Avaliação de Desempenho

O projeto de um algoritmo multi-objetivo requer a observação de duas características principais, as quais se relacionam à *qualidade* das soluções do conjunto Pareto-ótimo estimado, produzido pelo método proposto, e ao *tempo* necessário para a verificação da convergência.

Assim como qualquer problema multi-objetivo, a avaliação da qualidade dos ótimos estimados exige a consideração de múltiplos critérios, e, infelizmente, não existem padrões definitivos, ou exatos, que possibilitem atribuir valores de medidas a essa qualidade. De qualquer forma, sabe-se que as soluções de alta qualidade precisam (i) aproximar-se o melhor possível do conjunto Pareto-ótimo real, (ii) cobrir toda a sua extensão, incluindo as regiões extremas da fronteira, e (iii) descrevê-lo com riqueza de detalhes, apresentando exemplares bem distribuídos ao longo da sua extensão.

Com o objetivo de avaliar a qualidade das soluções estimadas, bem como a velocidade de convergência desenvolvida pelo algoritmo, são empregadas três métricas, sendo estas “Nondominated Combined Set Ratio” (NDCSR), “Hierarchical Cluster Counting” (HCC) e “S-Metric or hypervolume”.

“Nondominated Combined Set Ratio” (NDCSR)

A “Nondominated Combined Set Ratio” (NDCSR), proposta em (Zitzler & Thiele 1998), representa uma métrica de qualidade binária que especifica a *razão de cobertura* entre dois conjuntos. Assim sendo, dados dois conjuntos não-dominados, $\mathcal{A}$ e $\mathcal{B}$, produzidos por algoritmos diferentes, e $\mathcal{C}$ a fronteira não-dominada dos pontos em $\mathcal{A} \cup \mathcal{B}$, então, a cobertura desempenhada pelo conjunto $\mathcal{A}$ é definida como o número de elementos em

$\mathcal{A} \cap \mathcal{C}$ dividido pelo número de elementos em $\mathcal{C}$. De forma similar, a cobertura obtida pelo conjunto $\mathcal{B}$ é dada pelo número de elementos em $\mathcal{B} \cap \mathcal{C}$ dividido pelo número de elementos em $\mathcal{C}$. Assim sendo, espera-se que o algoritmo cujo operador de busca local seja mais eficiente, apresente uma maior razão de cobertura.

A formulação matemática para esta métrica é definida na equação (6.8):

$$NDCSR^{\mathcal{A}} = \frac{|\mathcal{A} \cap \mathcal{C}|}{|\mathcal{C}|}, \quad NDCSR^{\mathcal{B}} = \frac{|\mathcal{B} \cap \mathcal{C}|}{|\mathcal{C}|} \quad (6.8)$$

em que a função $|\cdot|$ representa a cardinalidade do conjunto em seu argumento. A Fig. 6.7 mostra um exemplo ilustrativo desta métrica, em que a razão de cobertura de $\mathcal{A}$ é, visivelmente, melhor do que a expressa por $\mathcal{B}$.

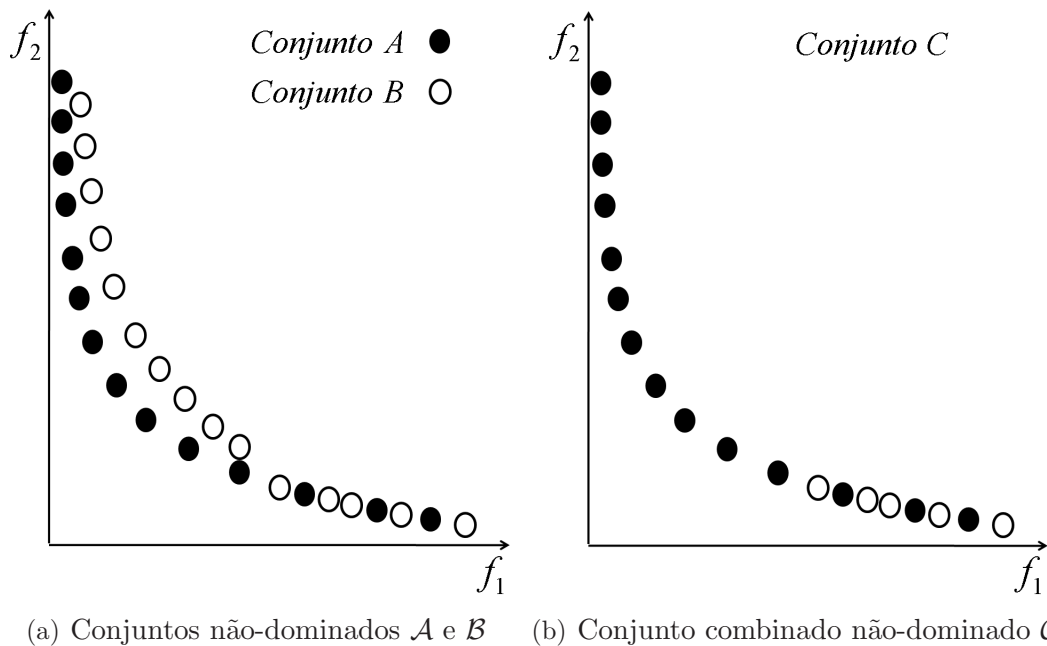


Figura 6.7: Razão de cobertura combinada entre os conjuntos não-dominados $\mathcal{A}$ e $\mathcal{B}$.

“Hierarchical Cluster Counting” (HCC)

A “Hierarchical Cluster Counting” (HCC) (Guimarães et al. 2009) pode ser interpretada como uma medida de uniformidade e extensão das soluções estimadas, sendo possível a partir desta identificar a fronteira que melhor descreve o conjunto Pareto-ótimo real. De certo modo, a HCC corresponde a uma versão aperfeiçoada da métrica de contagem de esferas (“sphere counting - SC”), proposta em (Wanner et al. 2008). Na HCC,

os pontos pertencentes ao conjunto $\mathcal{A}$ são submetidos a um método de agrupamento hierárquico baseado em processos aglomerativos. Esse mecanismo considera inicialmente que cada solução representa um agrupamento e, a cada iteração, identifica-se os dois agrupamentos cuja distância de *fusão* entre eles seja mínima, os reunindo então em um novo agrupamento. Esse sistema se repete até que todos os pontos estejam reunidos em um único agrupamento. Finalmente, as distâncias de fusão são somadas, e o valor resultante é atribuído a HCC. Após realizar esse mesmo processo para o conjunto $\mathcal{B}$, tem-se que a fronteira que melhor descreve o Pareto-ótimo real corresponde ao conjunto cujo valor da HCC associada seja maior.

A formulação matemática empregada para a obtenção da HCC é muito simples (equação (6.9)):

$$HCC = \sum_{i=1}^{\eta-1} Rf_i \quad (6.9)$$

em que η representa o número de soluções pertencente a fronteira considerada, e Rf_i é o raio de fusão da iteração i entre dois agrupamentos.

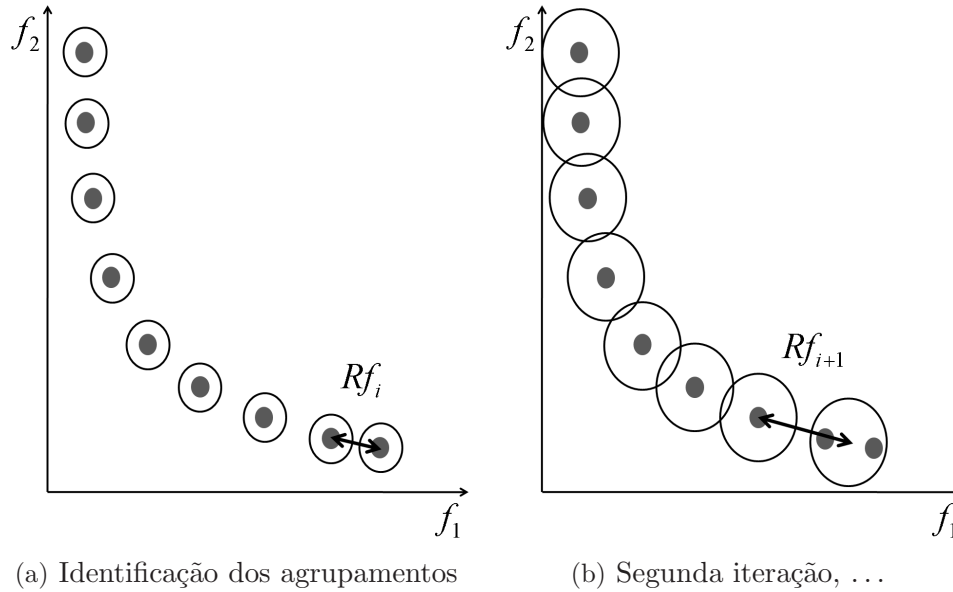


Figura 6.8: Esquema para a obtenção da métrica HCC.

A Fig. 6.8 mostra um exemplo ilustrativo para o cálculo da HCC correspondente a um dado conjunto fronteira. Nesse exemplo, apenas as duas primeiras iterações estão evidentes, em que o raio de fusão é medido considerando-se o centróide de cada agrupamento.

“S-Metric” ou “Hypervolume”

A “S-metric” é formalmente apresentada em (Zitzler 1999). Sendo breve, esta métrica calcula o hipervolume contido entre um conjunto fronteira e um ponto de referência, o qual deve ser dominado por todas as soluções desta fronteira. Assim sendo, determina-se a região dominada pelo conjunto Pareto-ótimo estimado, e quanto maior for esse volume, melhor é a representatividade da fronteira. Note que a especificação do ponto de referência depende somente do conhecimento, pelo usuário, dos limites superiores da região factível. Nesta dissertação de mestrado, a “S-metric” é utilizada como uma métrica de convergência, em que calcula-se o hipervolume dominado ao longo das gerações. Logo, o algoritmo que mais rapidamente domina o maior hipervolume, corresponde àquele que desempenha a maior velocidade de convergência.

Um exemplo ilustrativo da região dominada pelo conjunto fronteira é evidenciado na Fig. 6.9. Observe que o hipervolume é crescente com o passar das gerações. Considera-se aqui que ambos os objetivos são de minimização.

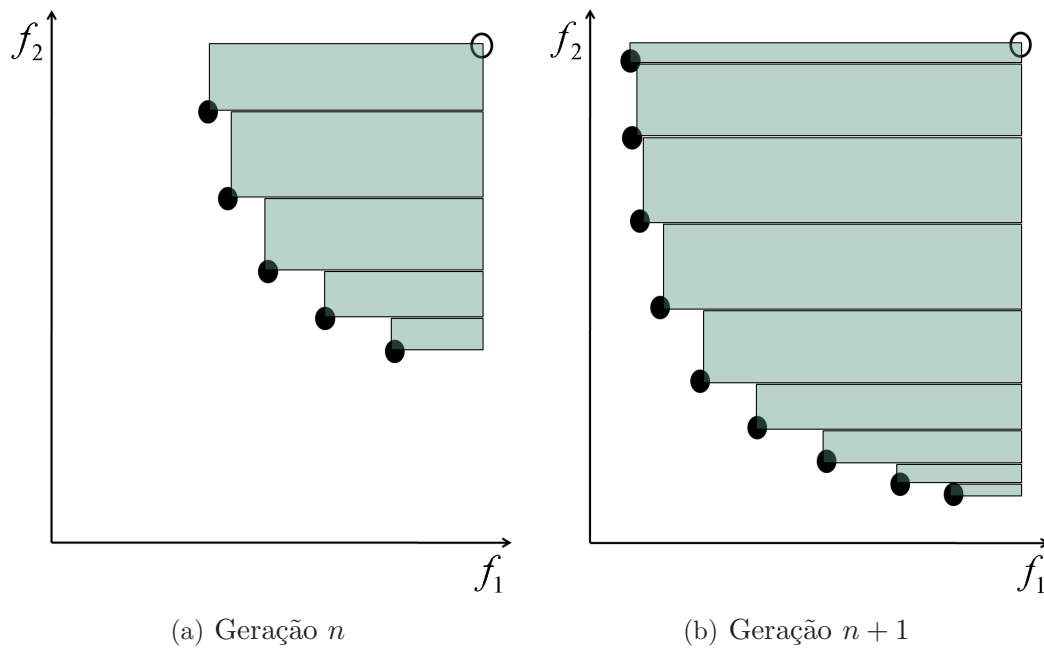


Figura 6.9: Cálculo da “S-metric” ao longo das gerações (o ponto de referência está representado pelo símbolo ○, enquanto as soluções estimadas correspondem ao símbolo ●).

6.3.2 Problemas Analíticos

Embora vários sistemas multi-objetivo tenham sido abordados no capítulo 4, a comparação de desempenho é realizada somente entre o MCSA e o NSGA-II. Observe que esse estudo comparativo é razoável, pois tanto o sistema de classificação, quanto o mecanismo de avaliação empregados no MCSA, são idênticos ao implementado pelo NSGA-II (ver seção 5.3).

Os testes baseiam-se no processo de otimização de seis problemas bi-objetivo muito utilizados na literatura, os quais são obtidos a partir dos estudos de Poloni (POL) (Poloni 1995), Fonseca e Fleming (FON) (Fonseca & Fleming 1998a), Kursawe (KUR) (Kursawe 1990), e Zitzler (ZDT1, ZDT2 e ZDT3) (Zitzler et al. 2000). Maiores detalhes sobre estes problemas são mostrados na tabela 6.10.

Os resultados medianos a serem apresentados consideram 10 execuções dos métodos MCSA e NSGA-II sobre cada um dos problemas mencionados. Além disso, o mesmo critério de parada é empregado, em que estipula-se o limite de 20000 avaliações da função objetivo. Os parâmetros de otimização utilizados pelos algoritmos são mostrados na tabela 6.11.

Comparação entre MCSA e NSGA-II

A realização de parte da análise baseia-se no emprego de “box plots”, o que possibilita ilustrar a distribuição dos conjuntos de dados resultantes, facilitando assim a comparação entre os algoritmos. Essa técnica, proposta em 1977 pelo estatístico britânico N. A. Sheldon, é brevemente descrita nas próximas linhas.

O diagrama de Sheldon representa a distribuição das amostras por meio de uma caixa com dois braços, podendo estar na vertical ou horizontal. De forma geral, tem-se que a caixa central compreende 50% da variabilidade dos dados, e o extremo de cada braço demarca os quantis de 25 e 75%. No interior da caixa indica-se ainda o valor mediano das amostras, o qual se localiza no quantil 50%. Assim sendo, os exemplares localizados entre os quantis de 25 e 75% representam uma estimativa robusta da variabilidade mediana entre as amostras, e qualquer ponto fora deste intervalo é considerado um valor discrepante, sendo caracterizado pelo sinal “+”. De forma a melhor entender o diagrama

Tabela 6.10: Formulação dos problemas analíticos multi-objetivo.

<i>Problema</i>	<i>n</i>	<i>Limites</i>	<i>Objetivos</i>	<i>Comentários</i>
<i>POL</i>	2	$[-\pi, \pi]$	$f_1(\vec{x}) = 1 + (A_1 - B_1)^2 + (A_2 - B_2)^2$ $f_2(\vec{x}) = (x_1 + 3)^2 + (x_2 + 1)^2$ $A_1 = 0.5 \sin 1 - 2 \cos 1 + \sin 2 - 1.5 \cos 2$ $A_2 = 1.5 \sin 1 - \cos 1 + 2 \sin 2 - 0.5 \cos 2$ $B_1 = 0.5 \sin x_1 - 2 \cos x_1 + \sin x_2 - 1.5 \cos x_2$ $B_2 = 1.5 \sin x_1 - \cos x_1 + 2 \sin x_2 - 0.5 \cos x_2$	<i>não convexo,</i> <i>desconexo</i>
<i>FON</i>	3	$[-4, 4]$	$f_1(\vec{x}) = 1 - \exp \left(- \sum_{i=1}^n (x_i - 1/\sqrt{3})^2 \right)$ $f_2(\vec{x}) = 1 - \exp \left(- \sum_{i=1}^n (x_i + 1/\sqrt{3})^2 \right)$	<i>não convexo</i>
<i>KUR</i>	10	$[-5, 5]$	$f_1(\vec{x}) = \sum_{i=1}^{n-1} \left(-10 \exp \left(-0.2 \sqrt{x_i^2 + x_{i+1}^2} \right) \right)$ $f_2(\vec{x}) = \sum_{i=1}^n (x_i^{0.8} + 5 \sin^3 x_i)$	<i>não convexo</i>
<i>ZDT1</i>	30	$[0, 1]$	$f_1(\vec{x}) = x_1$ $f_2(\vec{x}) = g(\vec{x}) \left[1 - \sqrt{x_1/g(\vec{x})} \right]$ $g(\vec{x}) = 1 + 9 \left(\sum_{i=2}^n x_i \right) / (n - 1)$	<i>convexo</i>
<i>ZDT2</i>	30	$[0, 1]$	$f_1(\vec{x}) = x_1$ $f_2(\vec{x}) = g(\vec{x}) \left[1 - (x_1/g(\vec{x}))^2 \right]$ $g(\vec{x}) = 1 + 9 \left(\sum_{i=2}^n x_i \right) / (n - 1)$	<i>não convexo</i>
<i>ZDT3</i>	30	$[0, 1]$	$f_1(\vec{x}) = x_1$ $f_2(\vec{x}) = g(\vec{x}) \left[1 - \sqrt{x_1/g(\vec{x})} - \frac{x_1}{g(\vec{x})} \sin(10\pi x_1) \right]$ $g(\vec{x}) = 1 + 9 \left(\sum_{i=2}^n x_i \right) / (n - 1)$	<i>convexo,</i> <i>desconexo</i>

de Sheldon, a Fig. 6.10 mostra uma ilustração comparativa deste com a função densidade de probabilidade normal ($N(0, 1)$).

É pertinente mencionar que os resultados de desempenho obtidos pelas métricas NDCSR e HCC se baseiam somente nas amostras da última fronteira estimada pelos algoritmos. E como mencionado anteriormente, o progresso dos valores medianos determinados pela “S-metric” fornecem uma figura ilustrativa da velocidade de convergência desempenhada por cada método.

Tabela 6.11: Parâmetros de otimização empregados na solução dos problemas analíticos multi-objetivo.

(a) MCSA		(b) NSGA-II	
N_{pop}	30	N_{pop}	100
N_{sel}^1	0.90	ρ_c	0.80
N_{dN}	0.50	ρ_m^2	$1/n$
N_{dU}	0.30	η_c	05
N_{dC}	0.20	η_m	15
β	0.10	L	100
$\gamma_{N,U,C}$	0.10		
L	100		

¹Como $N_{sel} = 0.90$, logo, $N_{rep} = 0.10$.

² n representa o número de variáveis de otimização.

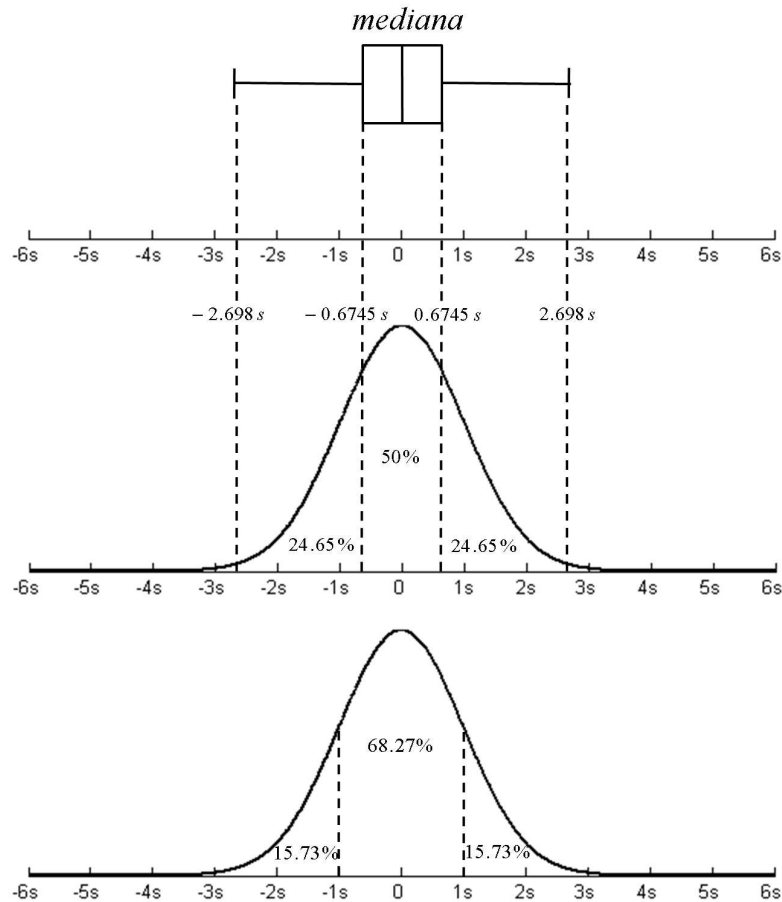


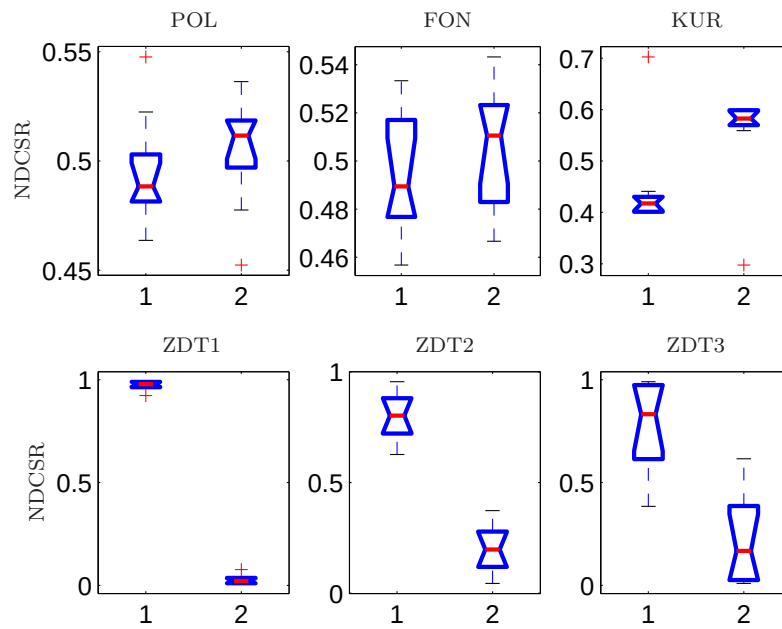
Figura 6.10: Diagrama de Sheldon (“box plot”).

Os valores estatísticos medianos calculados pela NDCSR são apresentados na Fig. 6.11(a). Observa-se claramente que o MCSA desempenhou uma melhor razão de cobertura em todos os três problemas de dimensões elevadas (ZDT1, ZDT2 e ZDT3), enquanto que o NSGA-II mostrou um melhor desempenho somente em KUR, sendo suavemente superior em POL e FON. Logo, esse resultado preliminar vem confirmar a capacidade de busca local desenvolvida pelo MCSA, principalmente em problemas de muitas dimensões.

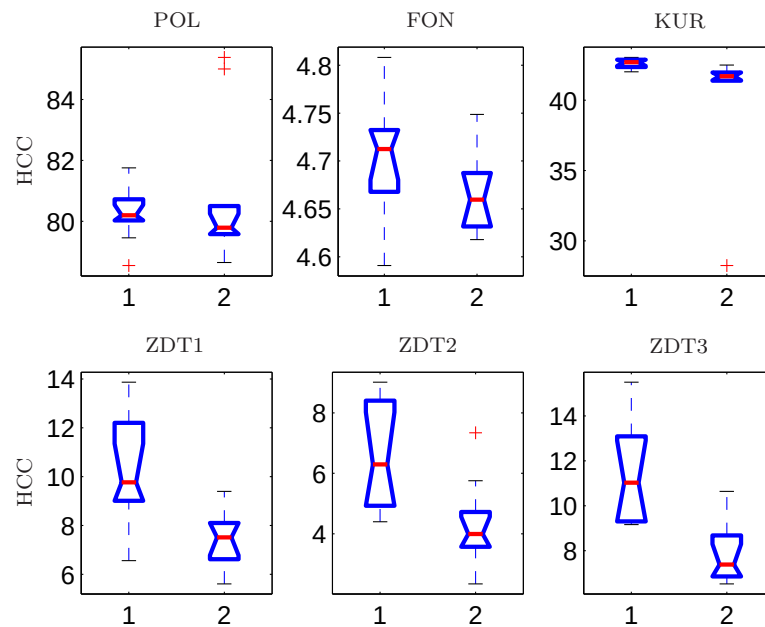
O estudo de sensibilidade seguinte se refere à análise dos dados estatísticos medidos pela HCC, Fig. 6.11(b). Neste caso, o desempenho apresentado pelo MCSA foi melhor em todos os testes realizados, o que mostra a maior habilidade deste quanto a distribuição uniforme das soluções ótimas ao longo da fronteira Pareto. Note que é importante que as informações fornecidas pelas métricas NDCSR e HCC sejam avaliadas em conjunto, pois um método é considerado eficiente na solução de algum problema somente se, além de desempenhar uma razão de cobertura eficaz, também apresentar uma melhor distribuição das soluções ao longo do Pareto-ótimo.

A Fig. 6.12(a) mostra o progresso dos valores estatísticos medianos obtidos pela “S-metric” em função do número de avaliações realizadas. Como pode ser observado, o MCSA desempenhou uma melhor velocidade de convergência em todos os problemas, exceto em POL, onde o NSGA-II foi suavemente melhor. Novamente, tem-se que os resultados mais relevantes foram obtidos durante o processo de otimização dos problemas de dimensões elevadas.

Com base nos problemas testes empregados, e considerando-se as métricas adotadas, observa-se que o MCSA é hábil quanto a determinação de ótimos espalhados ao longo da extensão do Pareto, e também capaz de desempenhar uma velocidade de convergência elevada. Visto que este algoritmo possui dois operadores principais, hipermutação somática e edição de receptores, os mesmos são estudados na próxima subseção, com o intuito de esclarecer o papel que cada um desenvolve durante o processo de otimização.



(a) Métrica de razão de cobertura - NDCSR



(b) Métrica de uniformidade e extensão - HCC

Figura 6.11: Análise de desempenho frente aos probs. analíticos bi-objetivo - parte 1 (MCSA: “box plot” à esquerda; NSGA-II: “box plot” à direita).

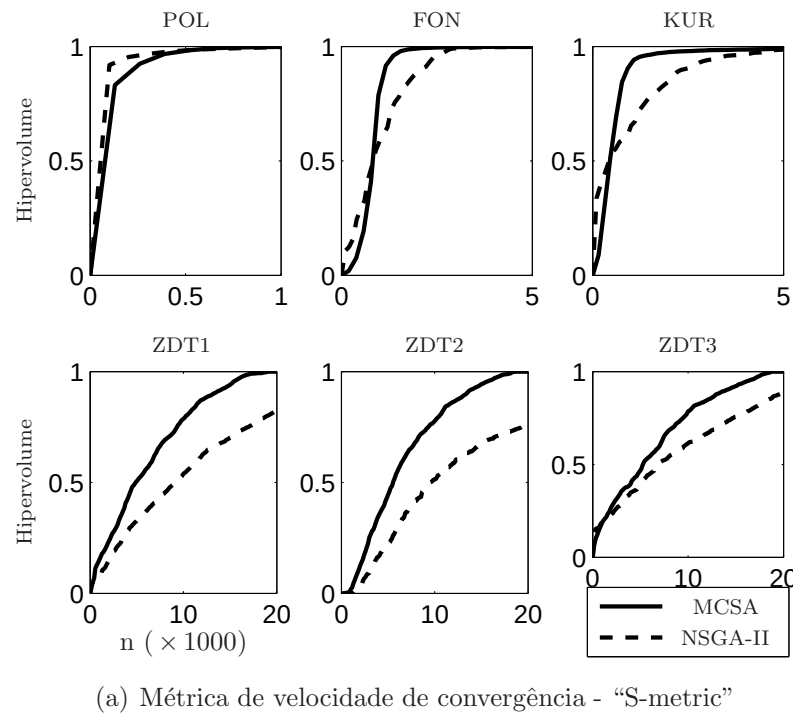
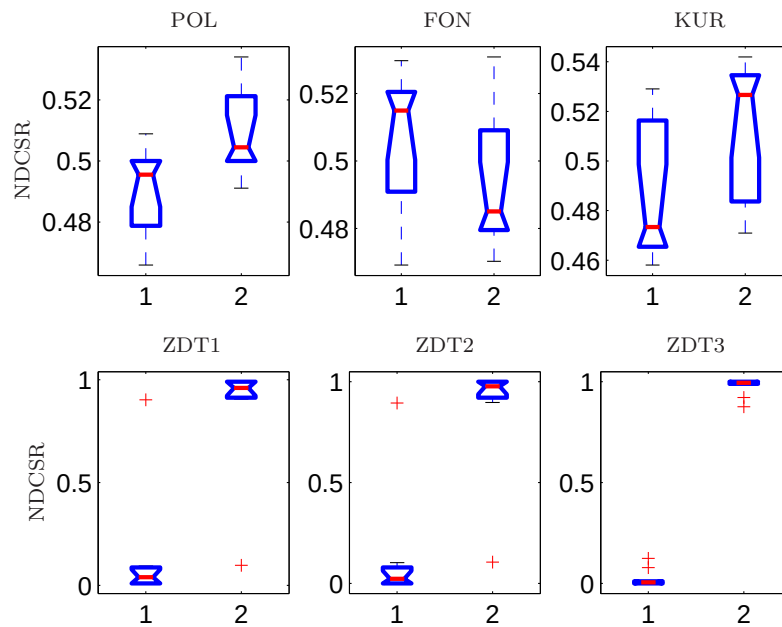


Figura 6.12: Análise de desempenho frente aos probs. analíticos bi-objetivo - parte 2.

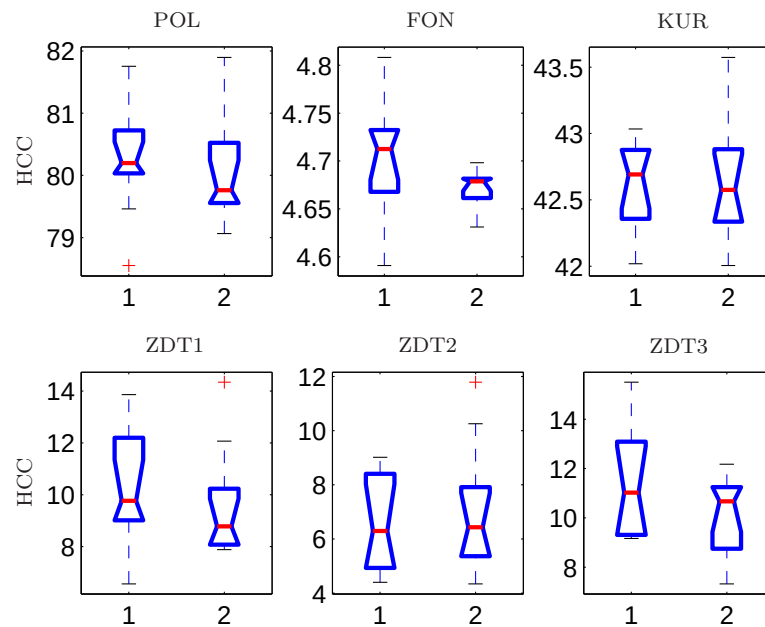
Atuação dos Operadores *Hipermutação Somática* e *Edição de Receptores*

Propôs-se neste trabalho um sistema de hipermutação somática, composto por diferentes funções densidade de probabilidade, com o objetivo de prover uma busca equilibrada sobre a região factível. Além disso, implementou-se um operador de edição de receptores de forma a efetuar o refinamento das soluções de alta afinidade. Entretanto, visando avaliar o verdadeiro benefício destes operadores no desempenho do algoritmo, o MCSA é inicialmente comparado com uma versão modificada de si próprio, em que considera-se somente a distribuição de Gauss, ao invés das três *pdf*'s normalmente utilizadas. Esse algoritmo é chamado IA_{gauss} . Já a segunda comparação é realizada entre o MCSA e uma outra versão de si próprio, a qual não emprega o operador de edição, sendo nomeada IA_{edit}^- .

Como está ilustrado nas Figs. 6.13 e 6.14, a versão modificada IA_{gauss} apresenta uma melhor razão de convergência na maior parte dos problemas, exceto no FON, no entanto, o MCSA foi capaz de encontrar um conjunto Pareto-ótimo com soluções melhores distribuídas, exceto no ZDT2, onde os resultados são similares. Dessa forma, observa-se

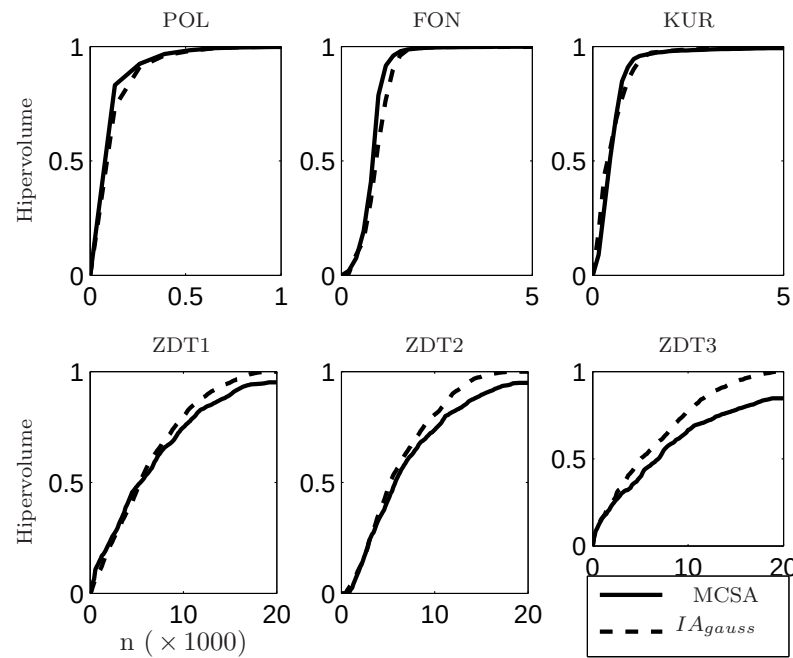


(a) Métrica de razão de cobertura - NDCSR



(b) Métrica de uniformidade e extensão - HCC

Figura 6.13: Efeito do operador *hipermutação somática* no desempenho do MCSA - parte 1 (MCSA: “box plot” à esquerda; IA_{gauss} : “box plot” à direita).



(a) Métrica de velocidade de convergência - “S-metric”

Figura 6.14: Efeito do operador *hipermutação somática* no desempenho do MCSA - parte 2.

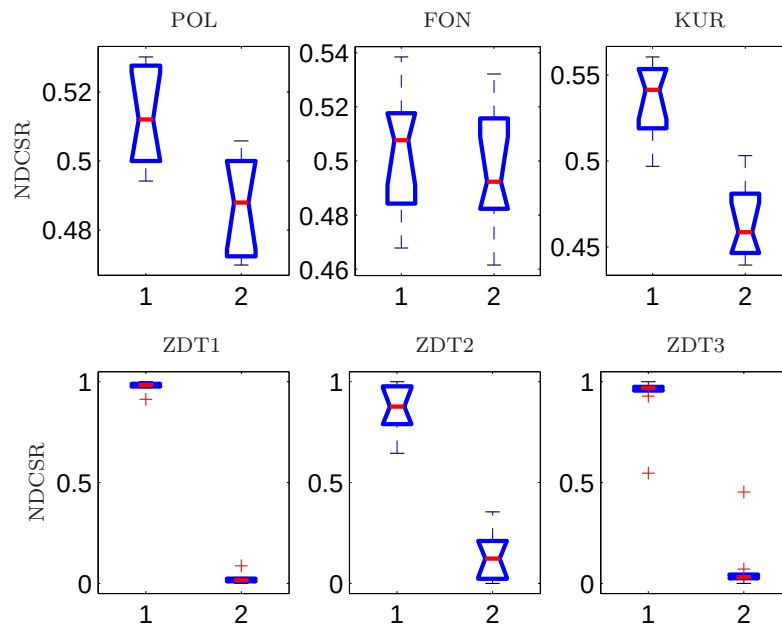
que o papel da distribuição normal é fundamental no refinamento local das soluções, e, além disso, tem-se que a hipermutação distribuída, principalmente em função da distribuição caótica, é que torna possível ao método melhorar a extensão e uniformidade das soluções do conjunto fronteira estimado.

Ambos os algoritmos apresentaram velocidades de convergência similares, exceto no ZDT3, onde a diferença em favor do IA_{gauss} foi mais significativa. Assim sendo, tomando-se por base os testes realizados, nota-se que hipermutação somática distribuída não influi sensivelmente na velocidade de convergência desempenhada pelo MCSA. Por fim, vale citar que a versão modificada IA_{gauss} foi obtida a partir do MCSA, onde fez-se simplesmente $N_{dN} = 1.0$ e $N_{dU} = N_{dC} = 0$.

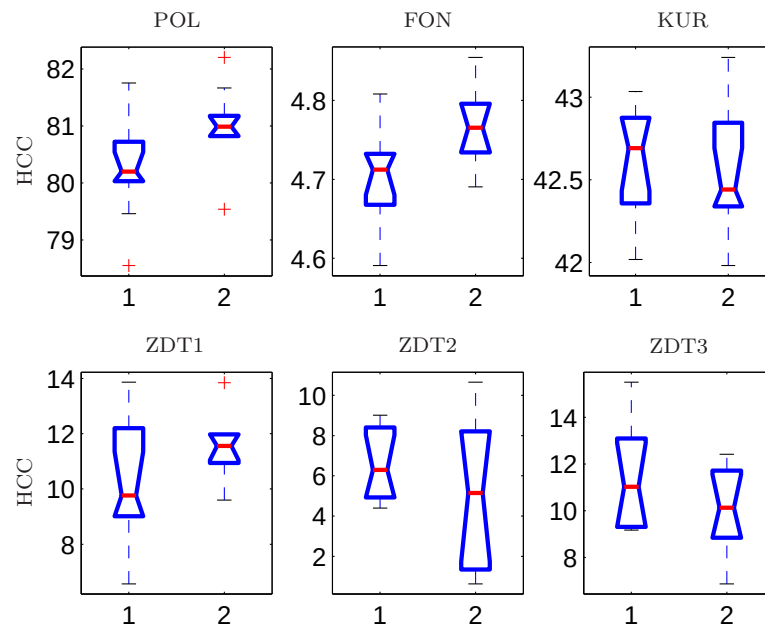
A influência do operador de edição é apresentada nas Figs. 6.15 e 6.16. De certa maneira, pode-se observar que este operador possui um grande efeito em relação a razão de cobertura alcançada pelas soluções ótimas estimadas. Além disso, conforme está evidente, a razão de cobertura obtida nos problemas ZDT's se aproxima de 100%, indicando a eficiência deste operador na solução de problemas de dimensões elevadas. Em contraposição, a influência positiva do sistema de edição na distribuição das soluções não é sensível, chegando até mesmo a proporcionar um efeito negativo, o que é observado nos problemas POL, FON e ZDT1, onde o IA_{edit}^- apresenta um desempenho superior ao do MCSA. Esse baixo desempenho observado nos problemas citados, sugere que a escolha incoerente do parâmetro que controla a amplitude do vetor diferencial, calculado durante a etapa de edição, pode incentivar uma convergência prematura do método, prejudicando assim o processo de distribuição das soluções ao longo da extensão da fronteira Pareto.

O operador de edição revela também uma influência positiva quanto ao aumento da velocidade de convergência do algoritmo. Como está ilustrado na Fig. 6.16(a), o desempenho do MCSA foi sensivelmente superior ao do IA_{edit}^- em todos os ZDT's, encontrando curvas praticamente idênticas para os problemas POL, FON e KUR.

Baseando-se nos testes realizados, e nas métricas utilizadas, as evidências indicam que a edição de receptores possui um papel muito importante no desempenho do algoritmo, o qual, além de favorecer uma maior velocidade de convergência, garante a realização do refinamento local das soluções de alta qualidade. De certo modo, tem-se que os danos causados à uniformidade e à extensão das soluções são recuperados por meio do processo de hipermutação distribuída, revelando assim funções complementares destes operadores, cada um influenciando de diferente forma no desempenho do MCSA.



(a) Métrica de razão de cobertura - NDCSR



(b) Métrica de uniformidade e extensão - HCC

Figura 6.15: Efeito do operador *edição de receptores* no desempenho do MCSA - parte 1 (MCSA: “box plot” à esquerda; IA_{edit}^- : “box plot” à direita).

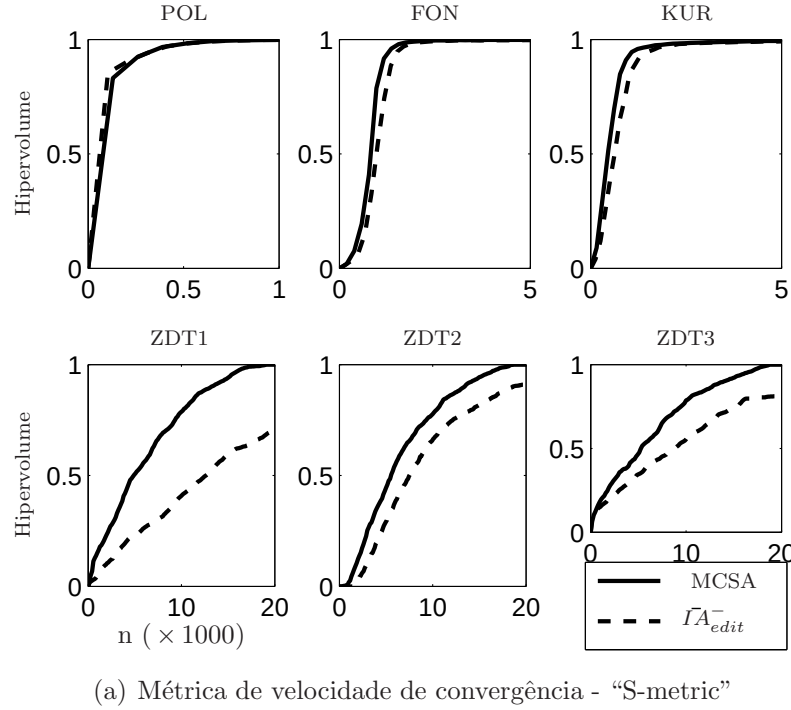


Figura 6.16: Efeito do operador *edição de receptores* no desempenho do MCSA - parte 2.

6.3.3 Problema Eletromagnético

A versão multi-objetivo do dispositivo eletromagnético “TEAM Problem 22” consiste da minimização da densidade do fluxo magnético de dispersão e do erro em relação a quantidade de energia a ser armazenada no SMES. De forma geral, tem-se que o primeiro objetivo é definido exatamente como na versão mono-objetivo deste problema, já o segundo objetivo corresponde ao que antes representava uma restrição de igualdade.

A formulação matemática deste problema pode ser encontrada em (Dias & Vasconcelos 2002), a qual é mostrada na equação (6.10):

$$\begin{aligned}
 \mathcal{X}^* &= \arg \min_x f(\vec{x}) \triangleq \{f_1(\vec{x}), f_2(\vec{x})\} \\
 f_1(\vec{x}) &= (B_{Stray}/B_{norm})^2 \\
 f_2(\vec{x}) &= |Energy - E_{ref}|/E_{ref} \\
 \text{sujeito a: } &\begin{cases} g_1(\vec{x}) = |J_i| + 6.4|B_{max_i}| - 54 \leq 0 \\ g_2(\vec{x}) = (R_1 - R_2) + \frac{1}{2}(d_1 + d_2) < 0 \end{cases}
 \end{aligned} \tag{6.10}$$

em que $B_{norm} = 3mT$, $E_{ref} = 180MJ$, e B_{Stray} é definido conforme evidenciado na equação (6.5).

Tomando-se por base a análise realizada na seção anterior, alguns dos parâmetros do MCSA são alterados. Dessa forma, visando intensificar a busca local sobre a região factível, adota-se $N_{dN} = 0.60$, $N_{dU} = 0.20$, $N_{dC} = 0.20$ e $\gamma_{N,U,C} = 0.05$. Além disso, para que o processo de otimização se alongue por mais gerações, o índice de clonagem é reduzido para $\beta = 0.05$, garantindo um menor número de avaliações a cada iteração. O parâmetro que controla a amplitude do vetor diferencial, durante a etapa de edição, é também modificado, o qual deixa de ser escolhido aleatoriamente no intervalo $[0.5, 1.0]$, e passa a receber um valor constante $\omega = 0.5$. Os demais parâmetros são mantidos exatamente como mostrado na tabela 6.11 ($N_{pop} = 30$ e $L = 100$).

A configuração empregada no NSGA-II não é alterada, e mantém-se conforme a tabela 6.11. As variáveis do dispositivo são as mesmas definidas para o problema mono-objetivo 3D e 8D (tabelas 6.6 e 6.8).

Durante a realização dos testes, executou-se cada algoritmo 10 vezes, considerando-se como critério de parada o limite de 3000 avaliações da função objetivo. A Fig. 6.17 apresenta os resultados estatísticos encontrados para o SMES multi-objetivo 3D. De certo modo, observa-se que embora o MCSA tenha desempenhado um melhor refinamento local das soluções, o NSGA-II alcançou uma melhor distribuição das soluções ao longo da extensão do Pareto-ótimo. Já as diferenças em relação a velocidade de convergência são insignificantes. Entretanto, quando avalia-se os resultados apresentados para o SMES 8D (Fig. 6.18), verifica-se que o MCSA apresenta um melhor desempenho em ambas as métricas, ilustrando assim sua habilidade na solução de problemas eletromagnéticos de dimensões elevadas.

Apoiando-se nos dados estatísticos retornados pelas métricas, e considerando-se a natureza multi-dimensional em relação a comparação de algoritmos multi-objetivo, pode-se supor que o MCSA não foi dominado pelo NSGA-II no SMES 3D, e claramente dominou o NSGA-II no SMES 8D.

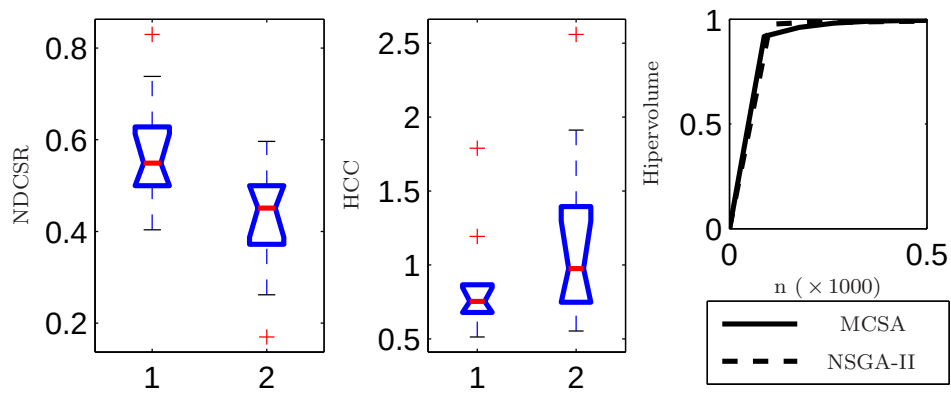


Figura 6.17: Análise de desempenho obtida a partir do problema bi-objetivo SMES 3D (MCSA: “box plot” à esquerda; NSGA-II: “box plot” à direita).

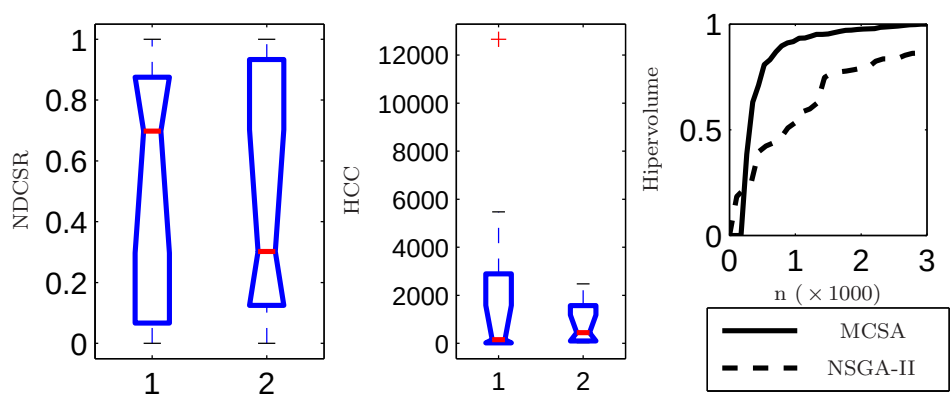


Figura 6.18: Análise de desempenho obtida a partir do problema bi-objetivo SMES 8D (MCSA: “box plot” à esquerda; NSGA-II: “box plot” à direita).

O processo de tomada de decisão é realizado com base nos dados contidos em um conjunto Pareto-ótimo, escolhido aleatoriamente dentre as 10 fronteiras obtidas (Fig. 6.19). Dessa forma, elimina-se todas as soluções em que $B_{Stray3D} > 1mT$, $B_{Stray8D} > 0.4mT$ e $\Delta E > 5\%$, e seleciona-se, dentre os ótimos restantes, a solução cujo volume de material supercondutor seja mínimo ($V = 2\pi \sum_{i=1}^2 r_i h_i d_i$).

Usando este critério, escolhe-se as seguintes soluções:

$$3D : \begin{cases} R_2 = 3.0424 \text{ m} \\ h_2 = 0.6269 \text{ m} \\ d_2 = 0.3108 \text{ m} \end{cases}, 8D : \begin{cases} R_1 = 1.5011 \text{ m} \\ h_1 = 1.5181 \text{ m} \\ d_1 = 0.5431 \text{ m} \\ R_2 = 2.5718 \text{ m} \\ h_2 = 1.8042 \text{ m} \\ d_2 = 0.1410 \text{ m} \\ J_1 = 16.2932 \text{ MA/m}^2 \\ J_2 = 18.7948 \text{ MA/m}^2 \end{cases} \quad (6.11)$$

as quais são comparadas na tabela 6.12.

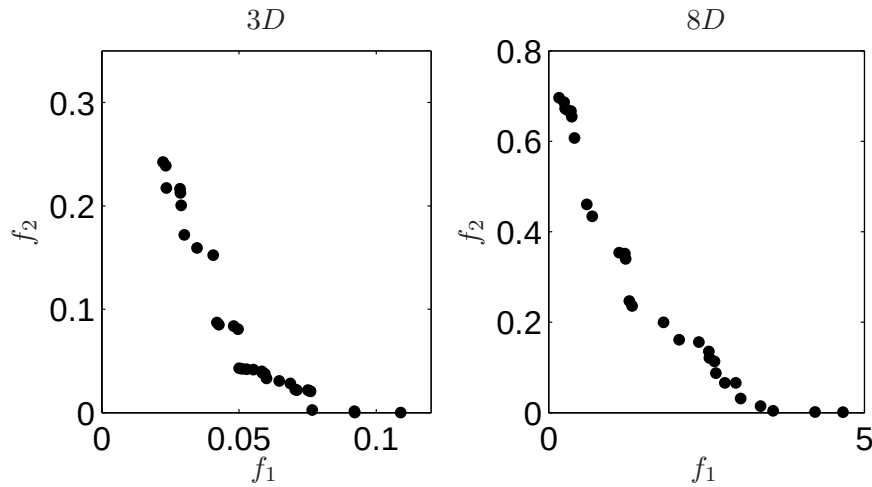


Figura 6.19: Fronteira Pareto-ótimo obtida para o dispositivo SMES multi-objetivo 3D e 8D.

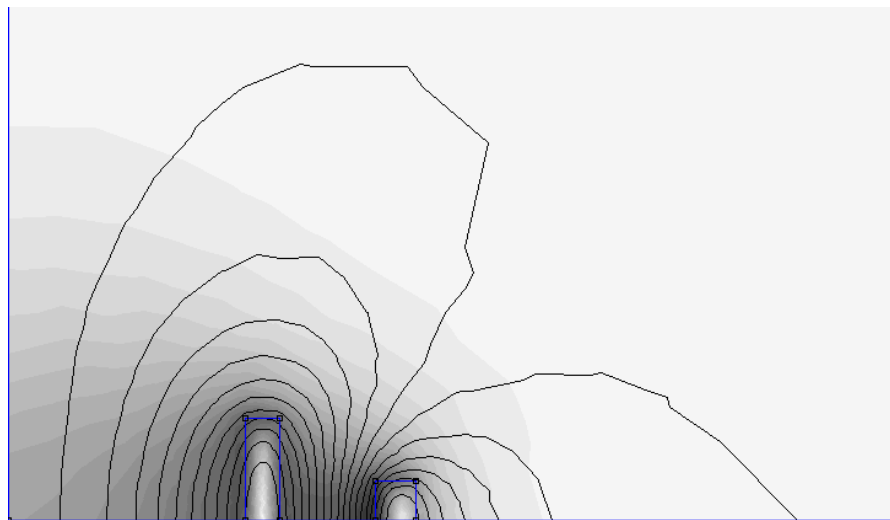
Tabela 6.12: Comparação com os melhores resultados multi-objetivo SMES 3D e 8D conhecidos na literatura.

Variável	B_{stray}	$Energy$	B_{max}	$Volume$
Unidade	mT	MJ	T	m^3
$MCSA_{3D}$	0.7864	174.90	4.2772	3.7247
$TEAM_{3D}^1$	0.8896	179.80	3.6300	3.6446
$MCSA_{8D}$	0.3663	182.62	5.4906	11.8885
$TEAM_{8D}^1$	0.0148	179.99	—	18.7905

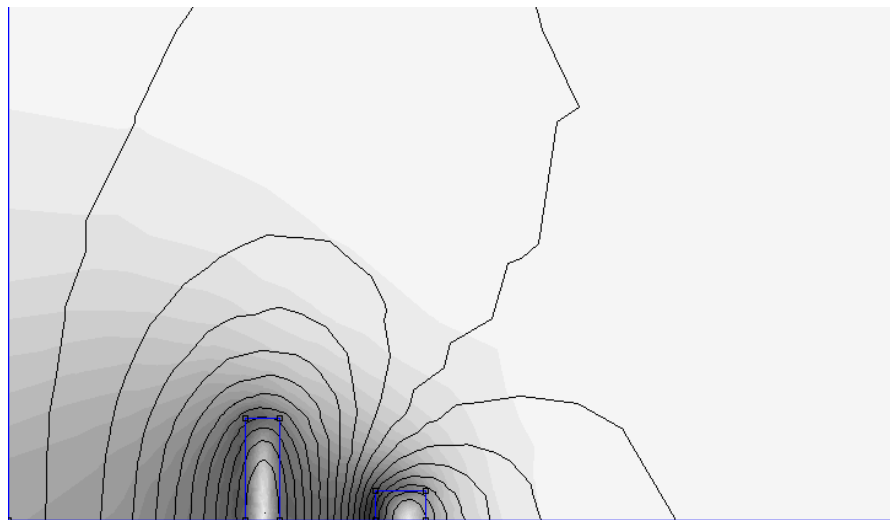
¹Resultado extraído de (Alotto et al. 1996) - solução ótima conhecida na literatura.

Como pode ser averiguado, o resultado encontrado para o SMES 3D apresenta valores similares ao ótimo conhecido pela literatura, com um erro em relação a energia desejada de 2.83%. Já a solução em 8D, apesar de não ter encontrado um fluxo de dispersão tão baixo quanto ao ótimo conhecido, o erro em relação a energia é de apenas 1.45%. Além disso, o baixo volume de material supercondutor supõe um custo financeiro mais econômico.

As Figs. 6.20 e 6.21 apresentam as configurações obtidas considerando-se as soluções, 3D e 8D, indicadas na equação (6.11). Como forma de comparação são apresentadas também as melhores configurações conhecidas na literatura. Nestas figuras são ilustradas também as linhas de fluxo magnético e a densidade de fluxo magnético, a qual é expressa em função das diversas tonalidades mostradas.

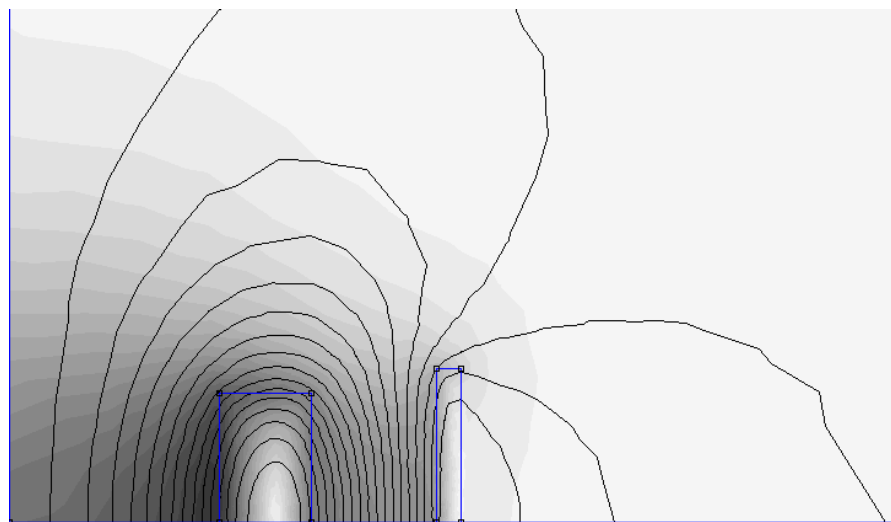


(a) MCSA

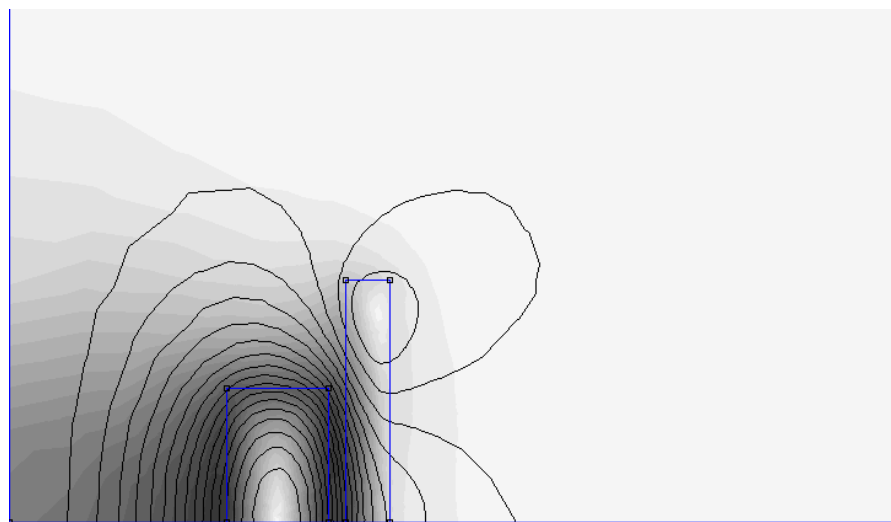


(b) TEAM

Figura 6.20: Configuração otimizada obtida para o SMES 3D multi-objetivo.



(a) MCSA



(b) TEAM

Figura 6.21: Configuração otimizada obtida para o SMES 8D multi-objetivo.

6.3.4 Discussão

O método MCSA, proposto neste trabalho, utiliza algumas técnicas comumente empregadas por algoritmos evolucionários multi-objetivo, dentre as quais se destacam a manutenção de um arquivo de memória, usado para armazenar as soluções não-dominadas da geração corrente, e um mecanismo de nicho baseado em informações acerca da densidade das soluções no espaço factível (“crowding distance”).

O diferencial deste método relaciona-se a etapa de maturação por afinidade, a qual implementa dois operadores principais, em que a hipermutação somática distribuída promove o melhor espalhamento das soluções ao longo da fronteira Pareto-ótimo, e a edição de receptores garante o refinamento local das soluções de alta afinidade, além de favorecer o aumento da velocidade de convergência do algoritmo.

Considerando-se os resultados analíticos, bem como o processo de otimização referente ao dispositivo SMES, parece razoável supor o MCSA como uma ferramenta eficiente para a solução de problemas eletromagnéticos reais de alto custo computacional.

6.4 Conclusão

Neste capítulo analisou-se o desempenho de ambas as abordagens imunes propostas nesta dissertação. Para tanto, vários problemas analíticos e dispositivos eletromagnéticos foram sugeridos, com base nos quais realizou-se experimentos comparativos considerando-se cada um dos algoritmos.

A metodologia empregada para a avaliação de desempenho do DCSA baseou-se, principalmente, na análise de curvas medianas de velocidade de convergência. Esse método foi comparado a inúmeros outros da literatura, e mostrou-se eficiente na solução de problemas com diferentes características de otimização, tais como, alta suavidade, multimodalidade, múltiplas variáveis e restrições. Os resultados encontrados para os dispositivos de microonda (WIT) e eletromagnético (SMES) evidenciaram a maior capacidade do DCSA quanto ao refinamento das melhores soluções encontradas.

Para a avaliação de desempenho do MCSA frente ao NSGA-II utilizou-se três métricas. De maneira geral, essas métricas serviram de ferramenta para obter curvas medianas de velocidade de convergência, e também para caracterizar o quanto um dado Pareto-ótimo é mais representativo do que outro, em que considerou-se a distribuição e extensão das

soluções ao longo da fronteira ótima estimada. Os testes realizados mostraram que o MCSA não foi dominado em nenhum momento, mas, principalmente, apresentou-se superior ao NSGA-II em problemas com elevadas dimensões, tanto nos analíticos quanto no eletromagnético.

Uma conclusão mais detalhada sobre o trabalho realizado nesta dissertação é delineada no capítulo seguinte, onde são enumerados, por fim, alguns pontos a serem explorados em trabalhos futuros.

Capítulo 7

Conclusões

“Me transformei num tipo de máquina de observar fatos e formular conclusões.”

— Charles Darwin, 1809–1882 (naturalista britânico)

“Já que se há de escrever, que pelo menos não se esmaguem com palavras as entrelinhas.”

— Clarice Lispector, 1920–1977 (escritora brasileira)

Este capítulo conclui a presente dissertação de mestrado. Apresenta-se inicialmente uma discussão geral sobre o tema tratado, os testes realizados e os principais resultados alcançados. Finalmente, são enumeradas algumas sugestões a serem exploradas em trabalhos futuros.

Conclusões

O estudo de novas abordagens de otimização, ou mesmo o aperfeiçoamento das técnicas existentes, é de grande valia no que diz respeito a necessidade, cada vez mais comum, de solução de problemas reais de engenharia, ou áreas afins, com alto custo computacional associado. Partindo-se deste pressuposto, essa dissertação propôs dois novos algoritmos imunes artificiais, sendo um dedicado a otimização de problemas mono-objetivo (“Dis-

tributed Clonal Selection Algorithm” - DCSA), e o outro, a otimização multi-objetivo (“Multi-Objective Clonal Selection Algorithm” - MCSA).

No capítulo 1, apresentou-se o sistema imunológico artificial como um sistema bio-inspirado, sendo considerado um dos mais novos campos de pesquisas inseridos na *computação natural*. Um breve histórico destaca os principais trabalhos relacionados ao tema, os quais, além de explorarem características peculiares desta vertente bio-inspirada (e.g. mapeamento de subótimos), exemplificam inúmeras aplicações (e.g. aprendizagem, reconhecimento de padrões, segurança de redes de computadores, otimização em geral).

Segundo está indicado pelo nome dos algoritmos propostos, os mesmos baseiam-se no *princípio da seleção clonal*. Assim sendo, estes métodos são regidos por quatro processos principais, os quais são *clonagem celular*, *maturação por afinidade*, *seleção elitista* e *geração de diversidade*. Com o objetivo de facilitar a compreensão do leitor ao longo deste trabalho, apresentou-se uma visão geral sobre alguns desses conceitos e princípios dos sistemas imunes, biológico e artificial, no capítulo 2, onde ilustrou-se também o funcionamento iterativo básico de um algoritmo imunológico.

Os capítulos 3 e 4 se dedicaram, inicialmente, à descrição de algumas definições relacionadas a modelagem matemática dos problemas de otimização, e às condições de otimalidade intrínsecas aos mesmos. Posteriormente, foram então apresentados alguns métodos evolucionários mono e multi-objetivo, respectivamente. De certo modo, foram discutidos os principais algoritmos a serem empregados durante a etapa de análise de desempenho do DCSA e MCSA.

Por fim, apresentou-se a contribuição dessa dissertação ao longo dos capítulos 5 e 6, onde cada um dos métodos foram detalhadamente explicados e avaliados. Baseando-se nos vários testes realizados, e levando-se em conta as comparações realizadas durante a análise de desempenho, pôde-se perceber a eficiência dos algoritmos propostos, e o papel dos operadores imunes implementados, sendo os mais relevantes a *hipermutação somática distribuída*, presente nas versões mono e multi-objetivo, e a *edição de receptores*, presente apenas na versão multi-objetivo.

A hipermutação somática, composta por três funções densidade de probabilidade distintas, normal, uniforme e caótica, representou o diferencial do método DCSA, permitindo a este desenvolver uma busca local balanceada ao redor das melhores soluções estimadas. Assim sendo, devido a consideração de diferentes regiões de vizinhança, em função da *pdf* adotada, esse operador, além de possibilitar o refinamento local das melhores soluções, por meio de uma pesquisa normal, também forneceu ao método a

capacidade de escapar, com maior facilidade, de ótimos locais (pesquisa caótica), favorecendo então a determinação de áreas factíveis mais prósperas.

As características desse operador foram também herdadas pelo MCSA, produzindo neste método um efeito colateral muito interessante. Assim sendo, em função das diferentes *pdf*'s empregadas pela hipermutação somática, principalmente a caótica, as soluções ótimas estimadas tornaram-se melhores distribuídas ao longo da extensão da fronteira Pareto-ótimo, permitindo uma melhor representatividade desta.

Como ficou claro no capítulo 6, os operadores *hipermutação somática* e *edição de receptores* revelam papéis complementares no desempenho do MCSA. Deste modo, enquanto a hipermutação promove a melhor distribuição das soluções, a edição garante o refinamento local dos ótimos de alta afinidade, além de favorecer o aumento da velocidade de convergência do algoritmo. A edição de receptores é implementada com base em princípios da evolução diferencial, dessa forma, no início do processo de otimização, quando a população encontra-se bem espalhada, esse operador atua como um gerador de diversidade, entretanto, à medida que a população se concentra nas proximidades da região do Pareto, ele passa a atuar como um refinador local. Visto isso, pode-se concluir que a edição funciona como uma mutação segundo uma *pdf* dinâmica, ditada pela maneira como as melhores soluções se encontram distribuídas sobre a região factível.

Propostas de Continuidade

De maneira geral, considerando-se os testes realizados e as métricas adotadas, conclui-se que tanto o DCSA, quanto o MCSA, representam importantes ferramentas para a otimização de problemas de alto custo computacional, principalmente, dispositivos eletromagnéticos. Entretanto, em contraposição à grande eficiência desempenhada por estes métodos, os mesmos exigem o ajuste inicial de um elevado número de parâmetros, o que embora possa representar uma dificuldade ao usuário inexperiente, não credita um problema real, visto que os algoritmos são pouco sensíveis à variação dos seus parâmetros.

Contudo, uma alternativa para minimizar essas possíveis dificuldades consiste na redução do número de parâmetros desses métodos. De forma geral, observa-se que a distribuição caótica privilegia uma pesquisa em regiões afastadas da solução original, entretanto, desenvolve também uma busca mais ou menos uniforme ao redor da mesma (rever Fig. 5.4). Assim sendo, os parâmetros relativos às distribuições uniforme e caótica

podem ser fundidos, ou pode-se ainda eliminar a distribuição uniforme, uma vez que a distribuição caótica realiza o seu papel. Embora a adoção de qualquer dessas propostas reduza em dois o número de parâmetros, acredita-se que a segunda opção seja mais conveniente, pois além de facilitar a implementação dos métodos, favoreceria um controle mais eficiente sobre o operador de hipermutação somática.

Como foi discutido no capítulo 5, o emprego de sistemas caóticos em algoritmos de otimização é recente, no entanto, o seu uso tem se tornado muito comum nos últimos anos. Observa-se que a substituição de parâmetros aleatórios por uma sequência caótica, ou mesmo a consideração de uma distribuição caótica em vez de uma normal, ou uniforme, traz alguns benefícios ao processo de otimização, geralmente melhorando o desempenho do método. Algumas das possíveis causas desse aumento de desempenho foram citadas nessa dissertação, entretanto, existe ainda a necessidade de realização de novos experimentos que as caracterizem melhor. Assim sendo, testes futuros poderão considerar outras *pdf*'s caóticas (e.g. neurônio caótico), e, além disso, pressupõe-se que seja interessante comparar o DCSA com um operador de hipermutação somática modificado, composto apenas por distribuições normais de diferentes desvios padrões. Tendo em vista os excelentes resultados alcançados devido a aplicação de edição no MCSA, parece também razoável supor a inserção deste operador na versão mono-objetivo proposta.

Uma breve discussão realizada no capítulo 2, mostrou que uma das principais características dessa classe de algoritmos refere-se a capacidade que os mesmos possuem de mapear múltiplos ótimos ao longo de uma única execução (rever Fig. 2.3). Entretanto, visto que pouca ênfase tem sido dada a essa característica peculiar dos métodos imunes, simulações futuras poderão explorar, por exemplo, o trajeto de convergência de algumas subpopulações, verificando assim o refinamento local que cada uma proporciona sobre uma região específica do espaço de busca.

Essa habilidade dos métodos imunes pode ser também estendida para sistemas multi-objetivo. Dessa forma, enquanto nos métodos mono-objetivo apenas a melhor solução de cada subpopulação é escolhida para prosseguir para a geração seguinte, em sistemas multi-objetivo cada subpopulação seria capaz de gerar várias soluções não-dominadas, as quais comporiam um subPareto-ótimo associado. A determinação de um conjunto de subParetos a cada geração proporcionaria, por exemplo, a formulação de novos mecanismos para a escolha das soluções para a próxima geração, o que poderia trazer ganhos quanto a melhor distribuição das soluções ótimas ao longo da extensão da fronteira Pareto.

Conforme apresentado no capítulo 5, a técnica empregada para a análise de sensibilidade, e também calibração dos parâmetros dos algoritmos, baseou-se em um estudo desacoplado quanto a influência de cada parâmetro no desempenho do método, o que certamente reduz a confiança em relação aos valores escolhidos para os parâmetros avaliados. Assim sendo, trabalhos futuros deverão dedicar-se a um formalismo matemático, e também estatístico, visando estabelecer métricas eficientes para a realização do ajuste dos parâmetros iniciais dos métodos, bem como para a análise dos resultados alcançados.

Como ficou claro neste trabalho, os métodos foram implementados com base em conhecimentos da imunologia clássica, destacando principalmente a teoria da seleção clonal. Entretanto, uma vez que existem diferentes metáforas acerca do funcionamento do sistema imune natural, acredita-se que o estudo destes princípios seja de grande valia quanto a elaboração de novos mecanismos e operadores a serem aplicados em algoritmos imunes artificiais, possibilitando ganhos, por exemplo, quanto a velocidade de convergência e qualidade das soluções ótimas estimadas.

“Quando o aluno está pronto para a lição, surge o mestre.”
— Provérbio espanhol

Apêndice A

Literatura Especializada

A Distributed Clonal Selection Algorithm for Optimization in Electromagnetics

Lucas de S. Batista, Frederico G. Guimarães, and Jaime A. Ramírez

Departamento de Engenharia Elétrica, Universidade Federal de Minas Gerais, MG, 31270-901, Brazil

Abstract—This paper proposes the real-coded distributed clonal selection algorithm (DCSA) for use in electromagnetic design optimization. This algorithm employs different types of probability distributions for the mutation of the clones. In order to illustrate the efficiency of this algorithm in practical optimization problems, we compare the results obtained by DCSA with other immune and genetic algorithms over analytical problems and for the TEAM Workshop Problem 22 for the 3 and 8 variables versions. The results indicate that the DCSA is a suitable optimization tool in terms of accuracy and performance.

Index Terms—Artificial immune systems, electromagnetic design optimization.

I. INTRODUCTION

THE recent development in the area of *artificial immune systems* (AIS) [1] – [3] has given rise to new bio-inspired stochastic optimization techniques. Most of these techniques are based on the *clonal selection principle* (CSP) [4], which is one of the models used to explain the behavior of the adaptive immune system. CSP-based algorithms are stochastic methods capable of optimizing multimodal problems and maintaining some local solutions along a single run. In order to achieve better performance and reduce the number of objective function evaluations other algorithms were proposed using real-coded variables, e.g. the real-coded clonal selection algorithm (RCSA) [5] and a modified AINet algorithm [6].

In this paper, we present an improved version of the RCSA, called the *distributed clonal selection algorithm* (DCSA) for mono-objective problems in electromagnetics. While RCSA works only with the Gaussian distribution, the DCSA employs different probability distributions in the population, with the aim of balancing local and global search in the algorithm. We compare DCSA with other immune and genetic algorithms on analytical and numerical problems. The results show that the proposed DCSA performs better on these test problems.

II. THE DISTRIBUTED CLONAL SELECTION ALGORITHM

Suppose the general unconstrained optimization mono-objective problem of the type:

$$x^* = \arg \min_x f(\vec{x}) \quad lb_i^- \leq x_i \leq ub_i^+ \quad (1)$$

where $f(\cdot) : \mathbb{R}^n \mapsto \mathbb{R}$ is the objective function, $\vec{x} \in \mathbb{R}^n$ is the variable vector, and lb_i^- and ub_i^+ are, respectively, the lower and the upper limits of the corresponding variable x_i .

The DCSA starts with the generation of an initial population, usually by spreading N_{pop} random points in the search space. These points are evaluated over a fitness function, which can be $f(\vec{x})$ or $-f(\vec{x})$ for minimization and maximization problems, respectively. So this vector of points is ranked in decreasing order of affinity. After that, this vector is separated

in four main groups: the first, $N_{dN}\%$ points are selected for cloning and mutation using the Gaussian distribution; the second, $N_{dU}\%$ points are selected for cloning and mutation using the uniform distribution; the third, $N_{dC}\%$ points are selected for cloning and mutation using the chaotic distribution; and the last group (the remaining $N_{REP}\%$ points not selected for cloning) is replaced by new randomly generated points. This replacement is an important characteristic of this algorithm, because the diversity is maintained and new areas of the search space can be potentially explored.

Each one of these generated clones receives a number of copies proportional to its position in the ranking, given by

$$N_C^i = \text{round} \left(\frac{\beta \cdot N_{pop}}{i} \right) \quad (2)$$

Then the clones, not the original individual, undergo the maturation process: each clone is submitted to a noise, such that

$$x^{mut} = x + \alpha \cdot S_k \cdot P \quad (3)$$

where $\alpha \in (0, 1)$ represent the size of the perturbation and can be called γ_N , γ_U or γ_C depending on the type of the noise (Gaussian for a local search, uniform for a uniform search and chaotic for an enlarged search); S_k is the difference between the upper and lower limits on the respective ordinate, $(ub_k - lb_k)$; and P represent the kind of perturbation. In this way, the use of the Gaussian mutation allows a local exploration around the original individual while the use of the chaotic mutation allows a global exploration around the individual. The use of the uniform mutation presents intermediate characteristics.

A given individual and its matured clones forms a subpopulation of points (antibodies - Ab). So the matured clones are evaluated over the affinity function and only the best of each subpopulation is allowed to pass to the next generation, maintaining the same size of the population.

Finally, the basic structure of the DCSA is described next.

III. SENSITIVITY ANALYSIS

In this section we study the effect of some parameters on the performance of the algorithm over a sample test function.

Manuscript received October 07, 2008. Current version published February 19, 2009. Corresponding author: J. A. Ramírez (e-mail: jramirez@ufmg.br).
Digital Object Identifier 10.1109/TMAG.2009.2012752

Algorithm 1: Distributed Clonal Selection Algorithm

Data:

- 1 $N_{pop} \leftarrow$ population size;
- 2 $N_{dN} \leftarrow$ rate of the population submitted to a Gaussian noise;
- 3 $N_{dU} \leftarrow$ rate of the population submitted to a Uniform noise;
- 4 $N_{dC} \leftarrow$ rate of the population submitted to a Chaotic noise;
- 5 $\gamma_N \leftarrow$ representing the size of the normal perturbation;
- 6 $\gamma_U \leftarrow$ representing the size of the uniform perturbation;
- 7 $\gamma_C \leftarrow$ representing the size of the chaotic perturbation;
- 8 $\beta \leftarrow$ multiplying factor for cloning;
- 9 $nfe_{max} \leftarrow$ maximum number of function evaluations;
- 10 **begin**
- 11 Generate and evaluate the initial population;
- 12 **while** ($nfe < nfe_{max}$) **do**
- 13 Rank the individuals in decreasing order of affinity;
- 14 Select the points N_{dN} , N_{dU} , N_{dC} and clone them;
- 15 Maturate the clones and evaluate them;
- 16 Select the best Ab from each subpopulation to survive;
- 17 Replace the N_{rep} individuals not cloned by new ones;
- 18 Evaluate the new N_{rep} individuals generated;
- 19 $nfe = nfe +$ function evaluations realized;
- 20 **end**
- 21 **end**

As seen in the previous section, the DCSA has eight main parameters for adjusting: the size of the population, N_{pop} ; the rates of the population submitted to a normal, uniform and chaotic noise, N_{dN} , N_{dU} , N_{dC} , respectively; the multiplying factor for cloning, β ; and the factors that represent the sizes of the normal, uniform and chaotic perturbations, γ_N , γ_U , γ_C , respectively.

Then, for evaluating the sensitivity of the algorithm to its parameters, the algorithm was executed 100 times over a test function, and each parameter was varied over a wide range while the other parameters were kept constant. The minimum, maximum and fixed values for each parameter are shown according to Table I.

TABLE I

VALUES OF THE PARAMETERS FOR THE SENSITIVITY ANALYSIS

	N_{pop}	N_{dN}	N_{dU}	N_{dC}	β	γ_N	γ_U	γ_C
min	5.00	0.10	0.10	0.10	0.50	0.05	0.05	0.05
max	50.0	0.60	0.60	0.60	5.00	1.00	1.00	1.00
fixed	10.0	0.20	0.20	0.20	1.00	0.10	0.10	0.10

The unconstrained test function (Rastrigin) is given by

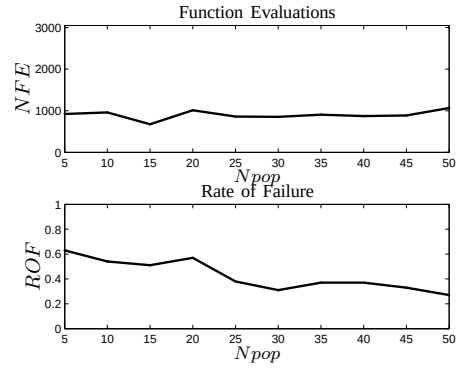
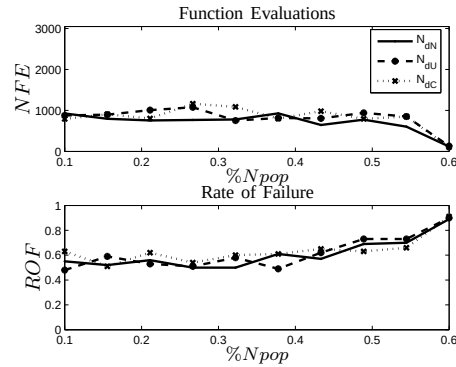
$$\min f(\vec{x}) = 10n + \sum_{i=1}^n x_i^2 - 10 \cos(2\pi x_i)$$

$$x_i \in [-5.12, 5.12], i = 1, \dots, n \quad (4)$$

where $\vec{x}$ is the variable vector and $n = 2$. This is a multimodal function characterized by 10^n local minima and a global minimum at $x_i = 0, \forall i \in [1, 2]$, where $f(x^*) = 0$.

As suggested in [7], the convergence criterium used is $\|\vec{x}\|_2 < \sqrt{0.02}$. Moreover, the influence of the DCSA parameters in the performance of the algorithm will be examined according to two different measures: the number of function evaluations until convergence (NFE) and the rate of failure to converge (ROF). So the DCSA is considered good if it presents low values for both cases.

As shown in Fig. 1, the algorithm presents low computational cost for a population size near 30, where the convergence rate increases up to 70%. Fig. 2 shows that the rate of failure presents an increasing tendency as the value of the parameters increases. A lower number of function evaluations is obtained for approximately $N_{dN} \approx 30\%$, $N_{dU} \approx 20\%$ and $N_{dC} \approx 20\%$, respectively. In Fig. 3 the rate of failure falls to low values at $\gamma_N \approx 30\%$, $\gamma_U \approx 25\%$ and $\gamma_C \approx 25\%$. Finally, Fig. 4 presents better convergence and lower computation cost for $\beta \approx 0.5$.

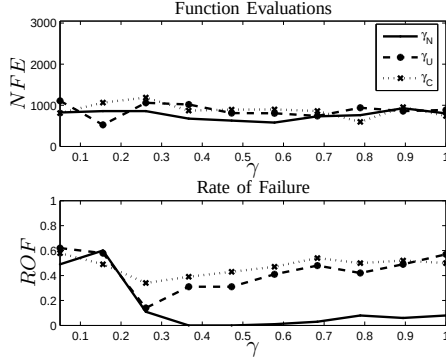
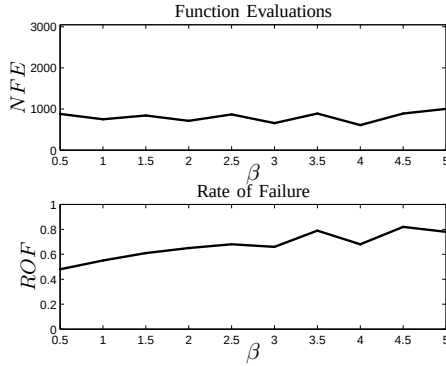
Fig. 1. Sensitivity of the DCSA to the parameter N_{pop} Fig. 2. Sensitivity of the DCSA to the parameters N_{dN} , N_{dU} and N_{dC}

IV. RESULTS

In this section we test the DCSA over two optimization problems. Based on the analysis of the previous section, we have decided to use the parameter values shown in Table II.

TABLE II
VALUES OF THE PARAMETERS USED

N_{pop}	N_{dN}	N_{dU}	N_{dC}	β	γ_N	γ_U	γ_C
15	0.30	0.20	0.20	0.50	0.10	0.25	0.25

Fig. 3. Sensitivity of the DCSA to the parameters γ_N , γ_U and γ_C Fig. 4. Sensitivity of the DCSA to the parameter β

A. Analytical Problems

For testing the ability of the DCSA, the following minimization problem was considered:

$$\min f(\vec{x}) = \sum_{i=1}^{n-1} \left[100(x_i^2 - x_{i+1})^2 + (1 - x_i)^2 \right] \\ x_i \in [-2.48, 2.48], i = 1, \dots, n \quad (5)$$

with $n = 2$. The two-dimensional Rosenbrock function present a global minimum at $\vec{x} = [1, 1]$, where $f(x^*) = 0$.

Another analytical test function is given by

$$\min f(\vec{x}) = 2.6164 + \frac{1}{n} \sum_{i=1}^n 0.01 \left[(x_i + 0.5)^4 - 30x_i^2 - 20x_i \right] \\ x_i \in [-6, 6], i = 1, \dots, n \quad (6)$$

with $n = 3$. This three-dimensional function present a global minimum at $x_i = -4.4538, \forall i \in [1, 3]$, where $f(x^*) = 0$.

The convergence speed of the DCSA is compared with those obtained for the clonal algorithm (CLONALG) [4], the real-coded clonal selection algorithm (RCSA) [5], the simple genetic algorithm (SGA) [7] and the b-cell algorithm (BCA) [8]. The results are shown in Figs. 5 and 6. Each algorithm was executed 50 times and the maximum number of function evaluations was kept to 3000 and 10000, respectively.

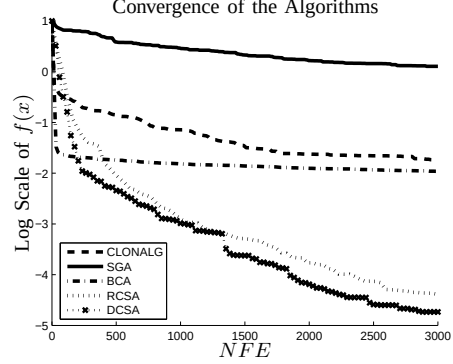


Fig. 5. Average convergence speed of the 2D function

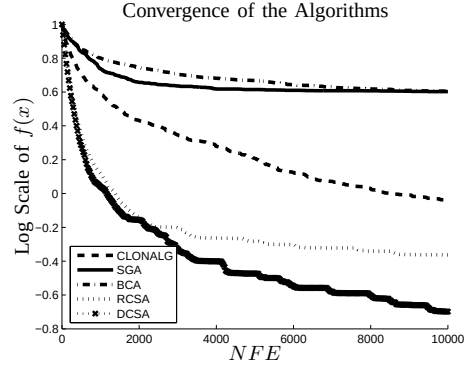


Fig. 6. Average convergence speed of the 3D function

These results show that the DCSA presents a convergence speed better than the other algorithms. Although the RCSA [5] presents a similar performance at the beginning of the minimization process for the Rosenbrock and 3D functions, the DCSA reaches best solutions after 1300 and 2000 function evaluations, respectively. In both cases the DCSA presented better performance.

B. Electromagnetic Problem

The proposed algorithm was also tested on the design of an electromagnetic device. The TEAM Benchmark Problem 22 [10] consists on the minimization of the stray magnetic flux density at a certain distance from a superconducting magnetic energy storage (SMES) device, shown in Fig. 7.

The problem is given by

$$\min f(\vec{x}) = B_{Stray} = \sqrt{\frac{\sum_{i=1}^{21} |B_{Si}|^2}{21}} \quad (7)$$

subject to

$$h_1(\vec{x}) = \frac{Energy - E_{ref}}{E_{ref}} = 0 \quad (8)$$

$$g_1(\vec{x}) = |J_i| + 6.4 |B_{max_i}| - 54 \leq 0 \quad (9)$$

$$g_2(\vec{x}) = (R_1 - R_2) + \frac{1}{2} (d_1 + d_2) < 0 \quad (10)$$

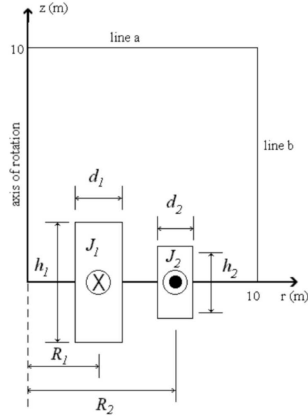


Fig. 7. SMES device configuration

where $E_{ref} = 180 MJ$ and the third constraint guarantees the non-superposition of the inner and outer coils.

We have used the 3 variables and 8 variables versions of the problem 22, as defined in [10]. The variable ranges are shown in Tables III and IV. The penalized objective function is given by (11) and the parameter values are shown in Table II, considering $N_{pop} = 30$. Tables V and VI shows the solutions, which are compared to the others available in the literature.

As seen in the Tables V and VI, the DCSA was able to find a set of optimal solutions for the problem at a single run, which is an interesting feature of this algorithm as it provides a range of options for the designer. These solutions consumed 1025 and 1350 objective function evaluations for the 3D and 8D versions. All solutions respected the energy constraint with a maximum error of 0.1% and 2.3%, respectively.

$$F(\vec{x}) = f(\vec{x}) + 100 \left\{ \sum_{i=1}^p |h_i(\vec{x})| + \sum_{j=1}^q \max[g_j(\vec{x}), 0] \right\} \quad (11)$$

V. CONCLUSION

We have proposed an improved version of the RCSA in which the main characteristic is that the cloned antibodies are submitted to different kind of probability distribution functions. Another interesting feature is that this method allows the

TABLE III
VARIABLE RANGES AND FIXED VALUES FOR THE 3D SMES DESIGN

Var Unit	r_1 m	h_1 m	d_1 m	r_2 m	h_2 m	d_2 m	J_1 MA/m ²	J_2 MA/m ²
min	—	—	—	2.6	0.408	0.1	—	—
max	—	—	—	3.4	2.2	0.4	—	—
fixed	2.0	1.6	0.27	—	—	—	22.5	22.5

TABLE IV
VARIABLE RANGES FOR THE 8D SMES DESIGN

Var Unit	r_1 m	h_1 m	d_1 m	r_2 m	h_2 m	d_2 m	J_1 MA/m ²	J_2 MA/m ²
min	1.0	0.2	0.1	1.8	0.2	0.1	10	10
max	4.0	3.6	0.8	5.0	3.6	0.8	30	30

TABLE V
RESULTS FOR THE 3D SMES PROBLEM

Var Unit	R_2 m	h_2 m	d_2 m	B_{Stray} mT	B_{max} T	Energy MJ
DCSA	3.1417	0.7031	0.2665	0.8997	3.9510	180.13
—	3.0959	0.5219	0.3670	1.0398	4.4355	179.94
—	3.0440	0.5330	0.3860	1.1090	4.7369	179.87
—	3.0577	0.5154	0.3928	1.1575	4.6822	180.19
—	3.0197	0.6161	0.3496	1.4625	4.4956	180.07
—	2.9785	0.6452	0.3557	2.1186	4.7982	180.01
RCSA [5]	3.1176	0.6017	0.3152	0.8890	4.1348	179.91
GA [9]	3.0500	0.4920	0.4000	0.9642	4.8550	175.43
TEAM [10]	3.0800	0.4780	0.3940	0.8896	4.6230	180.03

TABLE VI
RESULTS FOR THE 8D SMES PROBLEM

Parameters	DCSA	—	—	—	TEAM [10]
R_1 [m]	2.5037	1.8226	1.9902	1.9901	1.5703
h_1 [m]	2.2761	1.8625	1.7140	1.6794	1.5692
d_1 [m]	0.5226	0.3154	0.3084	0.3819	0.5943
R_2 [m]	3.1185	2.3961	2.6136	2.6359	2.0999
h_2 [m]	3.7406	3.1901	3.1250	2.6728	2.8368
d_2 [m]	0.0975	0.1956	0.1832	0.1935	0.2562
J_1 [MA/m ²]	10.3674	23.5680	23.1993	19.5660	17.3367
J_2 [MA/m ²]	22.0921	12.7373	12.2607	13.7385	12.5738
B_{Stray} [mT]	0.2008	0.2537	0.5029	0.5944	0.0148
B_{max} [T]	3.3732	4.0817	3.6893	4.0007	—
Energy [MJ]	176.47	180.63	175.91	180.77	179.99

determination of multiple optimal solutions, at an acceptable computational cost. This makes the algorithm a good tool for solving real electromagnetic problems. Furthermore, as seen in the SMES device optimization process, the DCSA was able to find a solution comparable to the others available in the literature.

REFERENCES

- [1] L. N. de Castro and F. J. Von Zuben, *Artificial Immune Systems: Part I - Basic Theory and Applications*, Technical Report, TR - DCA 01/99, Dec. 1999.
- [2] L. N. de Castro and F. J. Von Zuben, *Artificial Immune Systems: Part II - A Survey of Applications*, Technical Report, TR - DCA 02/00, Feb. 2000.
- [3] L. N. de Castro and J. Timmis, *Artificial Immune Systems: A New Computational Intelligence Approach*, Berlin, Germany: Springer-Verlag, 2002.
- [4] L. N. de Castro and F. J. Von Zuben, *Learning and Optimization using the Clonal Selection Principle*, IEEE Trans. Evol. Comput., vol. 6, no. 3, pp. 239-251, Jun. 2002.
- [5] F. Campelo, F. G. Guimarães, H. Igarashi, and J. A. Ramírez, *A Clonal Selection Algorithm for Optimization in Electromagnetics*, IEEE Trans. Magn., vol. 41, no. 5, pp. 1736-1739, May 2005.
- [6] F. Campelo, F. G. Guimarães, H. Igarashi, J. A. Ramírez, and So Noguchi, *A Modified Immune Network Algorithm for Multimodal Electromagnetic Problems*, IEEE Trans. Magn., vol. 42, no. 4, pp. 1111-1114, Apr. 2006.
- [7] J. A. Vasconcelos, J. A. Ramírez, R. H. C. Takahashi, and R. R. Saldanha, *Improvements in Genetic Algorithms*, IEEE Trans. Magn., vol. 37, no. 5, pp. 3414-3417, Sep. 2001.
- [8] J. Kelsey and J. Timmis, *Immune Inspired Somatic Contiguous Hypermutation for Function Optimization*, Proceedings of the on Genetic and Evolutionary Computation Conference (GECCO 2003), Springer, Lecture Notes in Computer Science, vol. 2723, pp. 207-218, 2003.
- [9] R. H. C. Takahashi, J. A. Vasconcelos, J. A. Ramírez, and L. Krahenbuhl, *A Multiobjective Methodology for Evaluating Genetic Operators*, IEEE Trans. Magn., vol. 39, no. 3, pp. 1321-1324, May 2003.
- [10] P. Alotto, A. V. Kuntsevitch, Ch. Magele, G. Molinari, C. Paul, K. Preis, M. Repetto, and K. R. Richter, *Multiobjective Optimization in Magnetostatics: A Proposal for Benchmark Problems*, IEEE Trans. Magn., vol. 32, no. 3, pp. 1238-1241, May 1996. [Online]. Available: http://www.igte.tugraz.at/archive/team_new/description.php.

Optimization of Electromagnetic Devices Using Artificial Immune Systems

Lucas de S. Batista¹, Frederico G. Guimarães², Prakash Paul³

¹Universidade Federal de Minas Gerais, Departamento de Engenharia Elétrica, lusoba@ufmg.br

²Universidade Federal de Ouro Preto, Departamento de Ciência da Computação, frederico.gadelha.guimaraes@gmail.com

³McGill University, Electrical and Computer Engineering Department, prakash.paul@mcgill.ca

Jaime A. Ramírez¹

¹Universidade Federal de Minas Gerais, Departamento de Engenharia Elétrica,

31270-010, Belo Horizonte, Minas Gerais, Brazil, jramirez@ufmg.br

Abstract— Optimization algorithms based on principles inspired from the immune system are capable of achieving an arbitrary set of optima, including the global solution. These algorithms differ in the way they implement the encoding, cloning, maturation and replacement steps, which are the basic ingredients of optimization algorithms based on artificial immune systems. This paper presents the Distributed Clonal Selection Algorithm (DCSA), which employs different probability distributions for the maturation step. The performance of the DCSA is compared with the Real-Coded Clonal Selection Algorithm (RCSA) and the B-Cell Algorithm (BCA) in the design of a waveguide and in the TEAM benchmark problem 22. The DCSA presents better convergence speed, in terms of number of evaluations, being 8% faster than the RCSA and 78% faster than the BCA, for the minimization of the return loss of a 3D waveguide impedance transformer. In the 8D TEAM problem, the DCSA and RCSA respect the energy constraint with a maximum error of 2.2% while the BCA presents high violations. Regarding these methods, the DCSA achieves better values for the stray magnetic flux density.

Index Terms— Artificial immune systems, electromagnetic design optimization.

I. INTRODUCTION

The *natural immune system* protects our bodies from infection by a complex interaction of white blood cells, called B cells and T cells. This complex system and its beddings and processes has inspired optimization techniques for solving engineering problems, which will be discussed next.

The recent development in the area of *artificial immune systems* (AIS), [1]- [3], has given rise to new bio-inspired

stochastic optimization techniques. Most of these techniques are based on the *clonal selection principle* (CSP) [4], which is one of the models used to explain the behavior of the adaptive immune system. CSP-based algorithms are stochastic methods capable of optimizing multimodal problems and maintaining some local solutions in a single run, which enables the designer to choose the most suitable solution from a set of optimal points.

In order to achieve better performance and reduce the number of objective function evaluations, other algorithms were proposed using real-coded variables, e.g. the real-coded clonal selection algorithm (RCSA) [6] and the distributed clonal selection algorithm (DCSA) [7].

In this paper we study the behavior of different AIS, namely, the *B-Cell Algorithm* (BCA) [5], and the *Real-Coded Clonal Selection Algorithm* (RCSA) [6]. Further, we present the *Distributed Clonal Selection Algorithm* (DCSA) [7], which is an improvement on the RCSA for solving mono-objective, multivariate, nonlinear, multimodal and computationally expensive electromagnetic problems.

II. PREVIOUS WORK

In this section we present three mono-objective optimization methods based on the clonal selection theory. The main characteristics of these algorithms are described next.

A. The B-Cell Algorithm

The B-Cell algorithm (BCA) [5] is coded as vectors of 64-bit long binary strings, each string representing a bit-encoded double-precision number. This codification allows the use of the *metadynamics* and *somatic contiguous hypermutation* (SCH) operators, which shall be discussed below.

The B-Cell algorithm starts with the generation of a random initial population of N_{pop} B-Cells. The main iterative loop of the algorithm starts with the fitness evaluation of the population. After the evaluation, each B-Cell receives a fixed number of clones, usually equal to the size of the population, although not necessarily. A B-Cell and its clones form a subpopulation of points in the search space.

The metadynamics operator is applied to only one clone from each subpopulation. This operator randomly changes the values of all bits in the clone, subject to a certain probability p_{meta} . This operator aims at maintaining the population diversity and the exploration of new areas of the search space.

All the clones are then subject to the SCH operator, that draws its inspiration from the fact that, in immune cells, the mutations usually occur in adjacent sections of the DNA molecule, instead of randomly spread over it. So, in the SCH, a random site (also known as hotspot) is chosen within a vector, along with a random length $L \in [1, L_{max}]$. The clone is subject to mutation from the hotspot onwards, until the length of the contiguous region has been reached. The clones are evaluated over the fitness function, and the best point from each subpopulation is selected for joining the B-Cell population in the next generation, while all the others are excluded. The iterative cycle continues until a given stop criterion is met (e.g., maximum number of function evaluations).

B. The Real Coded Clonal Selection Algorithm

The Real-Coded Clonal Selection Algorithm (RCSA) [6] starts with the generation of an initial population,

usually by spreading N_{pop} random points in the search space. These points are evaluated over a fitness function, which can be $f(\vec{x})$ or $-f(\vec{x})$ for minimization and maximization problems, respectively. These points are ranked in decreasing order of affinity and are separated in two main groups: the first, $N_{sel}\%$ points of the ordered population are selected for cloning and mutation, using the Gaussian distribution with zero mean and unitary standard deviation; and the remaining, $N_{rep}\%$ points are replaced by new randomly generated points, keeping the diversity in the population.

Each one of these generated clones receives a number of copies proportional to its position in the ranking, given by

$$N_C^i = \text{round}\left(\frac{\beta \cdot N_{pop}}{i}\right) \quad (1)$$

where β is a multiplying factor for cloning.

The clones undergo the maturation process: each clone is submitted to a noise, such that

$$\begin{aligned} x^{mut} &= x + \nu \\ \nu_k &= \alpha \cdot S_k \cdot P \end{aligned} \quad (2)$$

where ν_k is the noise added to the k^{th} variable; $\alpha \in (0, 1)$ represents the size of the perturbation; S_k is the difference between the upper and lower limits on the respective ordinate, $(ub_k - lb_k)$; and P represents the Gaussian distribution ($N(0, 1)$).

A given individual and its matured clones form a subpopulation of points (antibodies - Ab). Finally, the matured clones are evaluated over the affinity function and only the best of each subpopulation is allowed to pass to the next generation, maintaining the population size constant.

III. THE DISTRIBUTED CLONAL SELECTION ALGORITHM

The Distributed Clonal Selection Algorithm (DCSA) [7] represents an improved version of the RCSA. While RCSA works only with the Gaussian distribution for the maturation process, the DCSA employs different probability density functions in the population with the aim of balancing local and global search in the algorithm. These algorithms present the same steps, but in the DCSA the ordered vector of points is separated in four main groups: the first, $N_{dN}\%$ points are selected for cloning and mutation using the Gaussian distribution; the second, $N_{dU}\%$ points are selected for cloning and mutation using the Uniform distribution; the third, $N_{dC}\%$ points are selected for cloning and mutation using the Chaotic distribution; and the last group (the remaining $N_{rep}\%$ points not selected for cloning) is replaced by new randomly generated points. This replacement is an important characteristic of these algorithms, because the diversity is maintained and new areas of the search space can be potentially explored. The distribution scheme of the ordered population is shown in Fig. 1.

The number of copies for the clone is given by (1) and the maturation process is similar to that in (2), where $\alpha \in (0, 1)$ can be called γ_N , γ_U or γ_C depending on the type of the noise (Gaussian for a local search, Uniform for an uniform search and Chaotic for an enlarged search); S_k has the same formulation; and P represents the kind of perturbation, with $P = N(0, 1)$ for the Gaussian distribution, $P = 2U(0, 1) - 1$ for the Uniform distribution and $P = C(\zeta)$ for the Chaotic distribution, where $C(\zeta) : z(t+1) = \zeta z(t)[1 - z(t)]$ is a nonlinear system, known

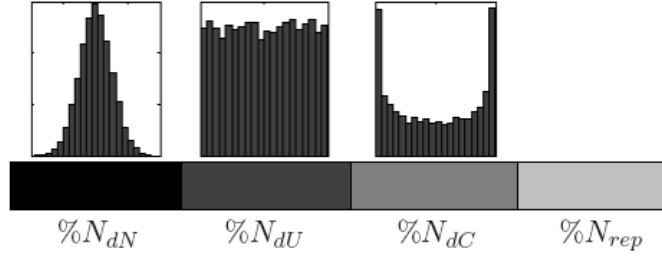


Fig. 1. Distribution rank for the DCSA supposing an ordered population.

as *logistic map*, with $\zeta = 4$.

In this way, the use of the Gaussian mutation allows a local exploration around the original individual while the use of the Chaotic mutation allows a global exploration around the individual. The use of the Uniform mutation presents intermediate characteristics.

IV. RESULTS

In this section we compare the performance of the DCSA, RCSA and BCA algorithms in the design of electromagnetic devices. Following [5]- [7], we decided to use the parameter values shown in Table 1.

We can observe that the size of the population for the BCA is very small compared to the other algorithms, but according to Kelsey [5] this method converges more rapidly to the solution with a smaller size for the population, using less evaluations of the function. This characteristic is due to its encoding.

Table 1. VALUES OF THE OPTIMIZATION PARAMETERS

(a) DCSA								
<i>Prob.</i>	N_{pop}	N_{dN}	N_{dU}	N_{dC}	β	γ_N	γ_U	γ_C
SMES	30	0.30	0.20	0.20	0.50	0.30	0.20	0.20
MWave	25	0.30	0.20	0.20	0.50	0.10	0.25	0.25

(b) RCSA					
<i>Prob.</i>	N_{pop}	N_{sel}	β	α	
SMES	30	0.60	0.50	0.10	
MWave	15	0.60	0.80	0.10	

(c) BCA					
<i>Prob.</i>	N_{pop}	L	p_{meta}	L_{max}	$Nc(i)$
SMES	06	64	0.50	48	N_{pop}
MWave	04	64	0.50	48	N_{pop}

A. Rectangular Waveguide Impedance Transformer

Fig. 2 shows a rectangular waveguide impedance transformer, which is an intermediate section between a waveguide of bigger cross-section and another one with a smaller cross-section. The feed component has a cross-

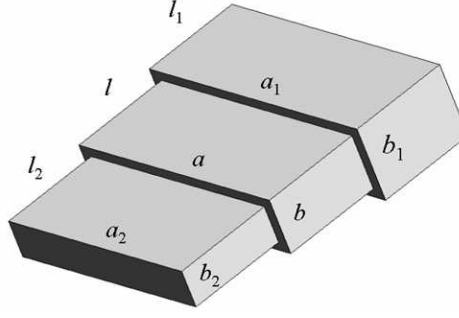


Fig. 2. Waveguide transformer configuration.

section of $a_1 \times b_1$ and length L_1 . The receiver component has a cross-section of $a_2 \times b_2$ and length L_2 .

The goal here is to find the values a , b , and L of the intermediate section that minimize the reflection coefficient. The reflection coefficient is calculated from the scattering parameters obtained via 3D finite element analyses. The transformer is designed to operate between 9.75 and 10.25GHz.

The optimization problem is stated as:

$$\begin{aligned} \min f(\vec{x}) &= 20 \log_{10} |S_{11}| \\ \text{such that } \begin{cases} a_2 = 20mm < a < a_1 = 24mm \\ b_2 = 4mm < b < b_1 = 8mm \\ 5mm < L < 20mm \end{cases} \end{aligned} \quad (3)$$

in which the optimization variables are $x = [a; b; L]$, and S_{11} is the scattering parameter. To obtain the S_{11} over a frequency range, the asymptotic waveform evaluation (AWE) is used [8].

Table 2 shows a set of optimal solutions for this device obtained in a single run. The stop criterion used was the *maximum number of function evaluations*, set as $nfe_{max} = 1000$. As seen in this Table, the three methods achieve good solutions, however the average number of evaluations that each method spent to reach a return loss of $-29.80dB$ was 389, 421 and 692 for the DCSA, RCSA and BCA, respectively. This indicates the better performance of the DCSA. In this test each algorithm was executed 30 times.

B. TEAM Benchmark Problem 22

The three considered algorithms were also tested on the TEAM Benchmark Problem 22 [9], which consists of the minimization of the stray magnetic flux density at a certain distance from a superconducting magnetic energy storage (SMES) device, shown in Fig. 3(a).

The objective function is defined as the minimization of the value of the B_{Stray}

$$\min f(\vec{x}) = B_{Stray} = \sqrt{\frac{\sum_{i=1}^{21} |B_{Si}|^2}{21}} \quad (4)$$

where B_{Si} is the magnetic flux density evaluated at each one of the 21 points, which are uniformly distributed

Table 2. RESULTS FOR THE MICROWAVE DEVICE

(a) DCSA				
Var Unit	$f(x)$ dB	a mm	b mm	L mm
DCSA	-29.9502	21.6125	5.6321	10.1186
	-29.9500	21.6123	5.6346	10.1261
	-29.9498	21.6113	5.6339	10.1099
	-29.9490	21.6119	5.6379	10.1109

(b) RCSA				
Var Unit	$f(x)$ dB	a mm	b mm	L mm
RCSA	-29.9495	21.6149	5.6358	10.1031
	-29.9493	21.6179	5.6302	10.1209
	-29.9482	21.6176	5.6412	10.1178
	-29.9478	21.6148	5.6396	10.0980

(c) BCA				
Var Unit	$f(x)$ dB	a mm	b mm	L mm
BCA	-29.9456	21.6252	5.6306	10.1326
	-29.9408	21.6300	5.6421	10.1447
	-29.9400	21.6260	5.6321	10.1643
	-29.9337	21.5913	5.6241	10.1316

along the straight lines a and b .

This problem has three main constraints. The equality constraint accounts for the amount of energy stored at the SMES device:

$$h(\vec{x}) = \frac{Energy - E_{ref}}{E_{ref}} = 0 \quad (5)$$

where $E_{ref} = 180 MJ$; Additionally, we have two inequality constraints:

$$g_1(\vec{x}) = |J_i| + 6.4 |B_{max_i}| - 54 \leq 0 \quad (6)$$

$$g_2(\vec{x}) = (R_1 - R_2) + \frac{1}{2} (d_1 + d_2) < 0 \quad (7)$$

where the first one concerns with the quench condition, Fig. 3(b), that guarantees the superconductivity state; and the second one guarantees the non-superposition of the inner and outer coils.

We have used the 8D version of the problem 22, as defined in [9], where the variable ranges are stated as in the Table 3. With the goal of considering the constraints, we have used the penalized objective function given by (8). Table 4 shows the results achieved in a single run for these methods, in which the stop criterion has been set as $nfe_{max} = 2000$.

$$F(\vec{x}) = f(\vec{x}) + 100 \left\{ \sum_{i=1}^p |h_i(\vec{x})| + \sum_{j=1}^q \max[g_j(\vec{x}), 0] \right\} \quad (8)$$

Table 3. VARIABLE RANGES FOR THE 8D SMES DESIGN

Var Unit	r_1 m	h_1 m	d_1 m	r_2 m	h_2 m	d_2 m	J_1 MA/m^2	J_2 MA/m^2
min	1.0	0.2	0.1	1.8	0.2	0.1	10	10
max	4.0	3.6	0.8	5.0	3.6	0.8	30	30

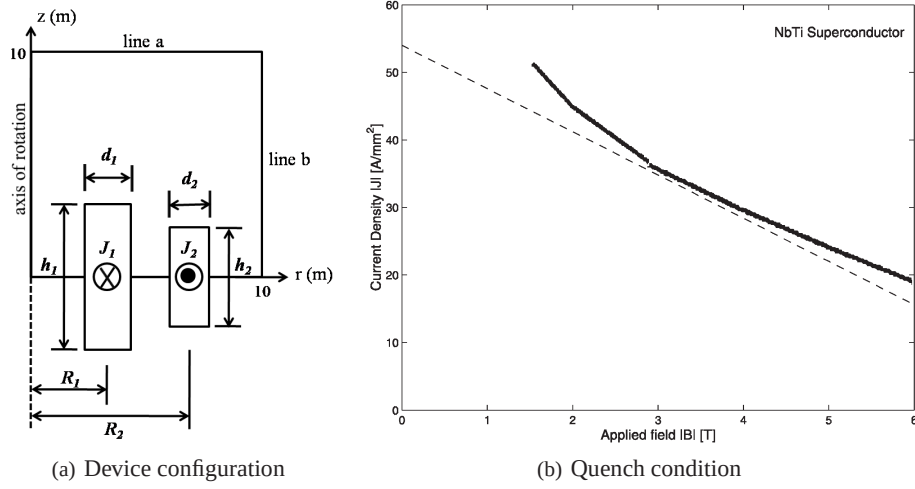


Fig. 3. SMES device.

The solutions achieved by the methods show that the TEAM problem has a multimodal feature, which can be observed by the standard deviation (S.D.) of the values of the optimization variables concerning the best known solution. Although the BCA has found smaller values of σ , the DCSA and RCSA performed a more efficient local refinement, providing not large violations of the energy constraint, with a maximum error of 2.2% for both.

We have adopted a rigorous stop criterion and a different objective function of that used by [9]. These features contribute to the deviation between the B_{Stray} calculated with the proposed methods when compared to the best results available for the TEAM problem. Nonetheless, regarding the simulations, the DCSA has acquired volume of coils ($V = 2\pi \sum_{i=1}^2 R_i h_i d_i$) similar to the best results available; furthermore, it presented better values for the stray magnetic flux density. The BCA reached acceptable values for the B_{Stray} , but the energy constraint was severely violated, with a maximum error of 10%.

As seen in these Tables, these methods were able to find a set of optimal solutions for the problem, providing a range of options to the designer, who can select the final solution. This ability of these CSP-based algorithms represents a valuable characteristic for electromagnetic and microwave design optimization, because it allows the designer to compare the solutions and, then, to decide the most adequate.

Overall, considering the simulation results, we can point out that the application of different probability density functions in the maturation step of the DCSA performs a better local search around the solutions. This search with different degrees of neighborhood allowed the method to acquire the good solutions reported.

Table 4. RESULTS FOR THE 8D SMES PROBLEM

(a) DCSA

Parameters	<i>DCSA</i>				TEAM [9]
$R_1[m]$	2.5037	1.8226	2.5006	1.9902	1.5703
$h_1[m]$	2.2761	1.8625	2.1108	1.7140	1.5692
$d_1[m]$	0.5226	0.3154	0.5338	0.3084	0.5943
$R_2[m]$	3.1185	2.3961	3.1077	2.6136	2.0999
$h_2[m]$	3.7406	3.1901	3.5995	3.1250	2.8368
$d_2[m]$	0.0975	0.1956	0.1025	0.1832	0.2562
$J_1[MA/m^2]$	10.3674	23.5680	10.9063	23.1993	17.3367
$J_2[MA/m^2]$	22.0921	12.7373	21.9398	12.2607	12.5738
$Volume[m^3]$	25.8583	16.1213	24.9073	16.0114	18.7905
$S.D.(\sigma)$	4.5107	2.3694	4.3401	2.2392	—
$B_{max}[T]$	3.3732	4.0817	3.5021	3.6893	—
$B_{Stray}[mT]$	0.2008	0.2537	0.2809	0.5029	0.0148
$Energy[MJ]$	176.47	180.63	181.53	175.91	179.99

(b) RCSA

Parameters	<i>RCSA</i>				TEAM [9]
$R_1[m]$	1.7392	1.7034	1.7401	1.9628	1.5703
$h_1[m]$	1.5922	1.5314	1.0693	1.0385	1.5692
$d_1[m]$	0.3872	0.4111	0.5780	0.5340	0.5943
$R_2[m]$	2.5574	2.5109	3.0195	3.3036	2.0999
$h_2[m]$	1.1541	1.2042	0.6018	1.2753	2.8368
$d_2[m]$	0.2113	0.2088	0.2336	0.1672	0.2562
$J_1[MA/m^2]$	19.5459	19.6241	16.6529	17.1195	17.3367
$J_2[MA/m^2]$	24.0252	23.8510	25.9853	16.7766	12.5738
$Volume[m^3]$	10.6554	10.7048	9.4245	11.2652	18.7905
$S.D.(\sigma)$	4.4582	4.3964	5.1611	1.7746	—
$B_{max}[T]$	4.4390	4.2899	5.1607	4.7598	—
$B_{Stray}[mT]$	0.4862	0.5519	0.6757	0.8344	0.0148
$Energy[MJ]$	179.42	181.88	178.86	177.54	179.99

(c) BCA

Parameters	<i>BCA</i>				TEAM [9]
$R_1[m]$	1.5272	2.0738	1.3137	2.0599	1.5703
$h_1[m]$	1.8302	0.9275	1.4567	1.1552	1.5692
$d_1[m]$	0.3078	0.4802	0.7024	0.3646	0.5943
$R_2[m]$	2.6475	3.1049	3.2239	3.3416	2.0999
$h_2[m]$	0.7248	1.1621	0.5208	1.7992	2.8368
$d_2[m]$	0.4784	0.1941	0.4301	0.1813	0.2562
$J_1[MA/m^2]$	23.6075	19.7495	12.9713	22.0599	17.3367
$J_2[MA/m^2]$	13.3768	18.4074	10.7866	11.2575	12.5738
$Volume[m^3]$	11.1736	10.2038	12.9829	12.3001	18.7905
$S.D.(\sigma)$	2.5335	2.5171	2.0353	1.9687	—
$B_{max}[T]$	4.4800	4.4680	6.3793	4.4897	—
$B_{Stray}[mT]$	0.5547	0.7121	0.8097	0.8852	0.0148
$Energy[MJ]$	195.87	161.79	166.77	177.33	179.99

V. CONCLUSION

We have presented a mono-objective clonal selection algorithm in which different kind of probability density functions have been applied in the maturation step. This characteristic provides a balancing local investigation in the landscape, performing searches with different degrees of neighborhood around the solutions.

The proposed algorithm has been compared with two other known methods in the design of a microwave and electromagnetic problems. The DCSA and RCSA reached good solutions at an acceptable computational cost, which makes these algorithms a good tool for solving real optimization problems. As seen in the SMES device optimization process, these methods were able to find a solution comparable to the best available in the literature. The B-Cell Algorithm did not achieve efficient solutions, spending a lot of time for the computing process, which is probably due to its binary-coding of the variables.

In general, the results show that the DCSA has performed better on these test problems, providing better convergence speed in the waveguide transformer, and good solutions in the electromagnetic problem.

VI. ACKNOWLEDGEMENT

This work was supported by the National Council of Scientific and Technologic Development - CNPq, Brazil, under grant: 306910/2006-3.

REFERENCES

- [1] L. N. de Castro and F. J. Von Zuben, *Artificial Immune Systems: Part I - Basic Theory and Applications*, Technical Report, TR - DCA 01/99, Dec. 1999.
- [2] L. N. de Castro and F. J. Von Zuben, *Artificial Immune Systems: Part II - A Survey of Applications*, Technical Report, TR - DCA 02/00, Feb. 2000.
- [3] L. N. de Castro and J. Timmis, *Artificial Immune Systems: A New Computational Intelligence Approach*, Berlin, Germany: Springer-Verlag, 2002.
- [4] L. N. de Castro and F. J. Von Zuben, *Learning and Optimization using the Clonal Selection Principle*, IEEE Trans. Evol. Comput., vol. 6, no. 3, pp. 239-251, Jun. 2002.
- [5] J. Kelsey and J. Timmis, *Immune Inspired Somatic Contiguous Hypermutation for Function Optimization*, Proceedings of the on Genetic and Evolutionary Computation Conference (GECCO 2003), Springer, Lecture Notes in Computer Science, vol. 2723, pp. 207-218, 2003.
- [6] F. Campelo, F. G. Guimarães, H. Igarashi, and J. A. Ramírez, *A Clonal Selection Algorithm for Optimization in Electromagnetics*, IEEE Trans. Magn., vol. 41, no. 5, pp. 1736-1739, May 2005.
- [7] L. S. Batista, F. G. Guimarães and J. A. Ramírez, *A Distributed Clonal Selection Algorithm for Optimization in Electromagnetics*, to appear in IEEE Transactions on Magnetics, vol. 45, 2009.
- [8] D. Jiao, X.-Y. Zhu, J.-M. Jin, *Fast and accurate frequency-sweep calculations using asymptotic waveform evaluation and the combined-field integral equation*, Radio Science, 34:5 (1999), 1055-1063.
- [9] P. Alotto, A.V. Kuntsevitch, Ch. Magele, G. Molinari, C. Paul, K. Preis, M. Repetto, K.R. Richter, *Multiobjective Optimization in Magnetostatics: A Proposal for Benchmark Problems*, IEEE Trans. Magn., vol. 32, no. 3, pp. 1238-1241, May 1996. [Online]. Available: <http://www.igte.tugraz.at/archive/team.new/description.php>.

New Operators for Multi-Objective Clonal Selection Algorithms

Lucas de S. Batista and Jaime A. Ramírez

*Departamento de Engenharia Elétrica, Universidade Federal de Minas Gerais,
Minas Gerais, Brazil*

Frederico G. Guimarães

*Departamento de Ciência da Computação, Universidade Federal de Ouro Preto,
Minas Gerais, Brazil*

lusoba@ufmg.br; fgg@ufmg.br; jramirez@ufmg.br

Abstract

Purpose – This paper aims to present a new multi-objective clonal selection algorithm (MCSA) for the solution of electromagnetic optimization problems.

Design/methodology/approach – The method performs the somatic hypermutation step using different probability distributions, balancing the local search in the algorithm. Furthermore, it includes a receptor editing operator that implicitly realizes a dynamic search over the landscape.

Findings – In order to illustrate the efficiency of MCSA, we compare its performance with the nondominated sorting genetic algorithm II (NSGA-II) in some analytical problems and in the well-known TEAM benchmark Problem 22. Three performance evaluation techniques are used in the comparison, and the effect of each operator of the MCSA in its accomplishment is estimated.

Research limitations/implications – In the analytical problems, the MCSA enhanced both the extension and uniformity in its solutions, providing better Pareto-optimal sets than the NSGA-II. In the Problem 22, the MCSA also outperformed the NSGA-II. The MCSA was not dominated by the NSGA-II in the 3 variables case, and clearly presented a better convergence speed in the 8 variables problem.

Practical implications – This paper could be useful for researchers that deal with multi-objective optimization problems involving high computational cost.

Originality/value – The MCSA improved both the extension, uniformity and the convergence speed of the solutions, in terms of the number of function evaluations, then representing a robust tool for real-world optimization problems.

Keywords Multi-objective optimization, Artificial immune systems, Electromagnetic design optimization.

Paper type Research paper.

1 Introduction

In the last years, the principles of artificial immune systems (AIS) have motivated many authors to present new stochastic population-based algorithms for mono-objective problems, see for instance (de Castro and Von Zuben 2000, Campelo et al. 2005, Campelo et al. 2006). These algorithms are capable of optimizing multimodal functions and maintaining local solutions, two valuable characteristics for the design of electromagnetic devices. In (Campelo et al. 2005) a Gaussian mutation was employed to improve the performance of AIS-based algorithms. In (Batista et al. 2009), we extended that work by introducing mutation operators based on different probability distributions, such as the chaotic distribution. The use of the chaotic distribution in optimization algorithms first appeared in the chaotic simulated algorithm (Chen and Aihara 1995, Mingjun and Huanwen 2004), accelerating the search performed by the standard simulated annealing algorithm. The chaotic distribution has been proposed for the mutation by (Cheng et al. 2008) and the crossover operators of genetic algorithms by (Coelho and Alotto 2008). Evolutionary algorithms that rely solely on a perturbation operator to explore the search space, as is the case with the simulated annealing, may benefit from the modification of the probability distribution function (PDF) used in the perturbation generator in a more significant way. This is the main motivation for exploring the use of different PDF's in AIS-based algorithms, as the mutation is the main mechanism in their search capability.

The solution of real-world electromagnetic design problems can be, in some cases, translated into optimization problems that involve conflicting goals. For these multi-objective optimization problems, there is not a single optimal solution, but a set of *nondominated solutions*, also known as *Pareto-optimal* solutions, in which each point represents different degrees of importance for each objective. In this paper, we extend the ideas of our previous work (Batista et al. 2009) to multi-objective optimization problems, introducing a multi-objective clonal selection algorithm (MCSA) that uses different PDF's for producing perturbations. The distribution is selected depending on the quality of the solution. In addition to the PDF based mutation, the algorithm also includes a mutation operator based on

the *receptor editing* performed during the process of affinity maturation, see (Nussenzweig 1998) and (George and Gray 1999). This additional operator does not employ any explicit probability distribution, but it implicitly creates one based on the difference vectors within the population. We perform experiments to assess the effect of each operator within the MCSA. The algorithm is compared with a nondominated sorting genetic algorithm (NSGA-II) proposed by (Deb et al. 2000).

2 Description of the Algorithm

As the name indicates, the multi-objective clonal selection algorithm (MCSA) is inspired from the clonal selection theory, which aims to model the behavior of the adaptive immune system. The algorithm is essentially defined by four main steps, namely *affinity evaluation*, *cloning*, *affinity maturation* and *replacement*. The equivalence between the terminology adopted in AIS and that used in optimization is presented in Table 1. Each step is explained next.

Take in Table (1)

Table 1: Equivalence between the terminology from immune systems and the terminology of optimization

Immune Systems	Optimization
affinity	fitness of a solution
antibodies, cells	candidate solutions
clones	copies of an antibody
lymphocyte pool	population of points

2.1 Affinity Evaluation

The candidate solutions tested by the algorithm are evaluated over all objective functions, penalizing any violation of the constraints, which is the basic approach to deal with constrained optimization problems. After that, each solution receives a scalar value - its quality value, usually called affinity value in AIS-based algorithms - according to the fast nondominated sorting with crowding distance (Deb et al. 2000). The population is sorted based on these affinity values.

2.2 Cloning

In this step, the N_{sel} best solutions in the original population of size N_{pop} are selected for cloning according to the front they belong and their crowding

distance values. Each one of the N_{sel} points selected receives a number of clones given by:

$$N_C^i = \text{round}(\beta N_{pop} \exp(\bar{\xi})) \quad (1)$$

in which $\bar{\xi} \in [0, 1]$ is the normalized crowding distance and $\beta \in [0, 1]$ is a parameter of the algorithm.

In the original nondominated sorting with crowding distance of the NSGA-II, the crowding distance values for the solutions at the boundaries of each front are positive infinity. In our algorithm we set their values as 1.10 times the maximum crowding distance value excluding the boundary individuals. In this way, these boundary points can be normalized to $\bar{\xi} = 1$. According to (1), the minimum and maximum number of clones is respectively βN_{pop} and $e\beta N_{pop}$, with a smooth variation between these extrema. Also notice that more clones are generated for the isolated and boundary points in the front, while less copies are produced for the solutions in the crowded regions of the front.

2.3 Affinity Maturation

As suggested by (Nussenzweig 1998, George and Gray 1999), the maturation of the immune response has been imposed basically by two mechanisms: somatic hypermutation and receptor editing. These two mechanisms are implemented as two independent operators in the MCSA. The population of clones undergo the hypermutation process, allowing the exploration of a local area by making perturbations to them. The perturbation generator employs different PDF's depending on the quality of the original point. The best selected antibodies undergo the receptor editing operator, providing an additional searching resource in the affinity maturation step. These operators are described next.

2.3.1 Somatic Hypermutation

Given the population of clones, the hypermutation operator consists of adding different kinds of perturbations to them (Batista et al. 2009). The clones of the first N_{dN} best solutions are perturbed using a Gaussian distribution, the clones of the succeeding N_{dU} best solutions are perturbed using an uniform distribution, and the clones of the remaining N_{dC} solutions are perturbed using a Chaotic distribution, such that $N_{dN} + N_{dU} + N_{dC} = N_{pop}$. The use of different PDF's have the effect of balancing the local refinement around the solutions due to different degrees of neighborhood within the local search.

2.3.2 Receptor Editing

The receptor editing operator can be viewed as an special kind of mutation, similar to the procedure employed by a simple differential evolution (DE) algorithm (Storn and Price 1997) to generate mutant vectors. The search capability of the DE algorithm relies on the differential mutation, whence a mutant vector v_i is produced from three randomly selected individuals in the population:

$$v_i = x_{r1} + \omega (x_{r2} - x_{r3}) \quad (2)$$

Finally, a new offspring is given by the recombination of each individual and its mutant vector.

Here, we adopt a similar procedure to the best N_{sel} antibodies, producing new solutions v_i , $i \in [1, N_{sel}]$, according to (2) with random and different indexes $r1, r2, r3 \in [1, N_{sel}]$ and $\omega > 0$ being a random value within $[0.5, 1.0]$.

Note that the receptor editing operator implicitly creates an adaptive PDF based on the difference vectors produced with the best solutions. In the beginning of the optimization, the best solutions are spread in the search domain, and the difference vectors become fairly distributed in all directions and with different amplitudes. Therefore, this operator introduces more diversity in the population. When the best solutions are getting close to the Pareto-optimal set, the difference vectors become distributed along the Pareto-optimal region. Therefore, this operator works as a local search performed around the high-quality solutions.

In brief, the somatic hypermutation plays an important role in exploring local regions of the search space, while the receptor editing may improve high affinity solutions using these solutions to produce difference vectors.

2.4 Replacement

In addition to somatic hypermutation and receptor editing, a fraction of new random cells is added to the lymphocyte pool in order to maintain the diversity of the solutions. According to (Jerne 1984), from 5 – 8% of the worst lymphocytes are replaced by new ones. In the MCSA, the replacement operator generates N_{rep} new random points within the search region, thus working as a diversity generation mechanism. The receptor editing and the replacement operators play similar roles in the beginning of the searching process, but after some iterations the receptor editing becomes a specific local refinement of high affinity solutions, while replacement preserves its global search characteristic.

All new solutions produced by the hypermutation, receptor editing and replacement operators are classified into nondominated fronts, and the first one is stored in an external memory population whose maximum size is $\mathcal{L} = 100$. If the size of the first front is greater than $\mathcal{L}$, then only the solutions with higher crowding distance are preserved.

Note that, when possible, the best N_{sel} antibodies are selected from the first front $\mathcal{F}_1$, however in the case where the number of points in $\mathcal{F}_1$ is smaller than the required value N_{sel} , the next fronts are considered, fixing $\bar{\xi} = 0$ for the points that are not in $\mathcal{F}_1$. The best N_{pop} solutions, i.e., the less dominated ones with greater crowding distance, are selected for the next iteration, hence maintaining the same population size.

2.5 Algorithm

Finally, we summarize the previous steps in the basic structure described in the Alg. 1.

Input: Adjustment parameters
Output: Estimatives of the Pareto-optimal set
begin
 $\mathcal{P}(n) \leftarrow$ Generate the initial N_{pop} solutions;
 $\Phi(n) \leftarrow$ Evaluate the points in $\mathcal{P}(n)$;
 while *Not stop criterion* **do**
 $\mathcal{F}(n) \leftarrow$ Classify the solutions $\mathcal{P}(n)$ into nondominated fronts;
 $\mathcal{I}(n) \leftarrow$ Perform the crowding distance over the fronts in $\mathcal{F}(n)$;
 $\mathcal{S}(n) \leftarrow$ Select the best N_{sel} solutions according to $\mathcal{I}(n)$;
 $\mathcal{C}(n) \leftarrow$ Perform the cloning process over $\mathcal{S}(n)$;
 $\mathcal{Q}(n) \leftarrow$ Apply the somatic hypermutation over $\mathcal{C}(n)$;
 $\mathcal{E}(n) \leftarrow$ Apply the receptor editing over $\mathcal{S}(n)$;
 $\mathcal{R}(n) \leftarrow$ Replacement and diversity generation;
 $\mathcal{P}(n+1) \leftarrow (\mathcal{F}_1(n) \cup \mathcal{Q}(n) \cup \mathcal{E}(n) \cup \mathcal{R}(n))$;
 $\Phi(n+1) \leftarrow$ Evaluate the points in $\mathcal{P}(n+1)$;
 $n \leftarrow n + 1$;
 end
end

Algorithm 1: The Multi-objective Clonal Selection Algorithm

3 Performance Evaluation

Two important concerns are involved in the design of a multi-objective algorithm: the “quality” of the estimates of the Pareto-optimal set that are produced by the algorithm and the time required by the algorithm to converge. The assessment of the quality of the estimates is also a multi-criteria problem, and there are no definitive standards for measuring such quality. Anyhow, a high-quality solution should (i) approach the true Pareto set as close as possible, (ii) cover its extension, including the regions near the extrema of such Pareto set, and (iii) describe the set in some detail, i.e., have samples that are well spread along its extension. Due to this multi-dimensional nature in the evaluation of multi-objective algorithms, we have used three different metrics in our analysis.

3.1 Nondominated Combined Set Ratio (NDCSR)

The NDCSR is a binary quality metric of coverage ratio among two sets, it has been proposed in (Zitzler and Thiele 1998). Given two nondominated sets, $\mathcal{A}$ and $\mathcal{B}$, produced by different algorithms, and $\mathcal{C}$ as the nondominated points of $\mathcal{A} \cup \mathcal{B}$, then the coverage ratio accomplished by the set $\mathcal{A}$ is defined as the number of elements in $\mathcal{A} \cap \mathcal{C}$ divided by the number of elements in $\mathcal{C}$. In a similar manner, the coverage ratio obtained by the set $\mathcal{B}$ will be the number of elements in $\mathcal{B} \cap \mathcal{C}$ divided by the number of elements in $\mathcal{C}$. So, it is expected that the algorithm with the best local search operator present a higher coverage ratio.

3.2 Hierarchical Cluster Counting (HCC)

The HCC metric (Guimarões et al. 2009) can be interpreted as a measure of both uniformity and extension of the estimates, identifying the front with the best description of the Pareto set shape. It is an improved version of the sphere counting (SC) metric proposed in (Wanner et al. 2008). In the HCC, the points of $\mathcal{A}$ are provided to an hierarchical clustering method using agglomerative procedures. The agglomerative procedures start with each point being a cluster. The two clusters that present the least distance of “fusion” in a given iteration are grouped into a new cluster, until all points have been grouped into a single cluster. The distances of fusion are summed to get the value of the HCC for that set. The procedure is repeated for the set $\mathcal{B}$. The set with the best description of the Pareto front is the one with the higher value for the HCC metric.

3.3 S-Metric or hypervolume

A definition of the S-metric is given in Zitzler (Zitzler 1999). This metric calculates the hypervolume enclosed by a front set and a reference point dominated by all the solutions of this front, hence computing the size of the region dominated by the set. The larger the volume dominated the better the front is. Note that the reference point depends only on the upper boundaries of the feasible region. By considering that the hypervolume calculated should increase as generations evolve, the S-metric can be used as a convergence metric. The progress of the value of this metric along the generations can be used to represent the convergence speed of the algorithm.

4 Results

In this section, we compare the performance of the MCSA and the NSGA-II (Deb et al. 2000) over some analytical problems and for the SMES benchmark Problem 22. Since the MCSA employs the same method for ranking

the solutions that is used in the NSGA-II, it is reasonable to compare the MCSA against the NSGA-II.

Based on the sensitivity analysis presented in a previous mono-objective work, see (Batista et al. 2009), the parameters of the multi-objective extension, MCSA, have been set as $N_{pop} = 30$, $N_{sel} = 0.90$, $N_{rep} = 0.10$, $\gamma_{N,U,C} = 0.10$, $N_{dN} = 0.50$, $N_{dU} = 0.30$, $N_{dC} = 0.20$ and $\beta = 0.10$. As suggested in (Deb et al. 2000), and recently proposed in (Sharma et al. 2007), the NSGA-II parameters for a bi-objective problem have been chosen often as $N_{pop} = 100$, $p_{xover} = 0.80$ and $p_{mut} = 1/n$, where n represents the number of optimization variables. Moreover, we have taken $\eta_{xover} = 5$ for the *simulated binary crossover* (SBX), $\eta_{mut} = 15$ for the *polynomial mutation*, and a population archive of same length $\mathcal{L} = 100$.

We have selected six analytical benchmark problems: Poloni (POL) (Poloni 1995), Fonseca and Fleming (FON) (Fonseca and Fleming 1998), Kursawe (KUR) (Kursawe 1990), and the three first Zitzler’s problems (ZDT1, ZDT2 and ZDT3) (Zitzler et al. 2000). These problems have been described in the Table 2, and none of them have any constraint. This Table also shows the number of variables, their bounds, and the nature of the Pareto-optimal front for the problems. Each algorithm was executed 10 times with a stop criterion of 20,000 function evaluations.

Take in Table (2)

4.1 Comparison between the MCSA and the NSGA-II

For the analysis, we have used box plots to illustrate the distribution of the results. The box plot produces a box and whisker plot for the samples, in which the notches represent a robust estimate of the variability of the median between samples and the outliers are displayed with a “+” sign. The central box represents 50% of the data, and its lower and upper boundary lines are at the 25% and 75% quantile of the data. The central line indicates the median of the data, staying in the 50% quantile. For all the following results, in each plot the box on the left represents the distribution of the MCSA. The coverage ratio metric and the HCC metric have been based only in the samples of the last front. The progress of the median of the S-metric gives a picture of the convergence speed of each algorithm.

The results for the NDCSR are shown in Fig. 1. Note that the MCSA performed better than the NSGA-II in all three high-dimensional problems (ZDT1, ZDT2 and ZDT3), while the NSGA-II performed better in KUR, and slightly better in POL and FON. The results for the HCC metric show the ability of the MCSA to widely spread solutions in the nondominated front, Fig. 2. In this case, the MCSA outperformed the NSGA-II in all problems. It can be inferred from these problems that the MCSA shows good results for the high-dimensional problems. Fig. 3 shows the progress

Table 2: Analytical bi-objective test problems

<i>Prob</i>	<i>n</i>	<i>Bounds</i>	<i>Objective functions</i>	<i>Comments</i>
<i>POL</i>	2	$[-\pi, \pi]$	$f_1(x) = 1 + (A_1 - B_1)^2 + (A_2 - B_2)^2$ $f_2(x) = (x_1 + 3)^2 + (x_2 + 1)^2$ $A_1 = 0.5 \sin 1 - 2 \cos 1 + \sin 2 - 1.5 \cos 2$ $A_2 = 1.5 \sin 1 - \cos 1 + 2 \sin 2 - 0.5 \cos 2$ $B_1 = 0.5 \sin x_1 - 2 \cos x_1 + \sin x_2 - 1.5 \cos x_2$ $B_2 = 1.5 \sin x_1 - \cos x_1 + 2 \sin x_2 - 0.5 \cos x_2$	<i>nonconvex, disconnected</i>
<i>FON</i>	3	$[-4, 4]$	$f_1(x) = 1 - \exp \left(- \sum_{i=1}^n (x_i - 1/\sqrt{3})^2 \right)$ $f_2(x) = 1 - \exp \left(- \sum_{i=1}^n (x_i + 1/\sqrt{3})^2 \right)$	<i>nonconvex</i>
<i>KUR</i>	3	$[-5, 5]$	$f_1(x) = \sum_{i=1}^{n-1} \left(-10 \exp \left(-0.2 \sqrt{x_i^2 + x_{i+1}^2} \right) \right)$ $f_2(x) = \sum_{i=1}^n (x_i^{0.8} + 5 \sin^3 x_i)$	<i>nonconvex</i>
<i>ZDT1</i>	30	$[0, 1]$	$f_1(x) = x_1$ $f_2(x) = g(x) \left[1 - \sqrt{x_1/g(x)} \right]$ $g(x) = 1 + 9 \left(\sum_{i=2}^n x_i \right) / (n-1)$	<i>convex</i>
<i>ZDT2</i>	30	$[0, 1]$	$f_1(x) = x_1$ $f_2(x) = g(x) \left[1 - (x_1/g(x))^2 \right]$ $g(x) = 1 + 9 \left(\sum_{i=2}^n x_i \right) / (n-1)$	<i>nonconvex</i>
<i>ZDT3</i>	30	$[0, 1]$	$f_1(x) = x_1$ $f_2(x) = g(x) \left[1 - \sqrt{x_1/g(x)} - \frac{x_1}{g(x)} \sin(10\pi x_1) \right]$ $g(x) = 1 + 9 \left(\sum_{i=2}^n x_i \right) / (n-1)$	<i>convex, disconnected</i>

of the median of the S-metric versus the number of evaluations. In this experiment, the MCSA presented a better convergence speed in all problems except POL, when the NSGA-II performed better, but the difference is small.

At this point, for these test problems, we can observe that the MCSA presented a better extension and uniformity in its solutions, which is probably due to its distributed somatic hypermutation. The next experiments try to clarify the role of each operator in the performance of the MCSA.

Take in Figure (1)

Take in Figure (2)

Take in Figure (3)

4.2 The Effect of the Distributed Somatic Hypermutation and Receptor Editing Operators

In order to assess the benefit of using hypermutation with different PDF's and vector difference for the refinement of high affinity solutions in the

algorithm, we first compare the MCSA against a version of itself without using different PDF's, but considering only the Gaussian distribution, which is termed IA_{gauss} . Secondly, we compare the MCSA against a version of itself without the receptor editing operator, which is termed IA_{edit}^- . Again the MCSA is represented by the box on the left in all box plots.

As illustrated in Figs. 4-6, although the IA_{gauss} presented a better coverage ratio, except in FON, the MCSA achieved a better distributed Pareto set, except in ZDT2, where the results are similar. Both algorithms present similar convergence speeds regarding the S-metric, except in ZDT3, where the difference in favor of the IA_{gauss} was more significant.

Considering these results, we can conclude that the distributed hypermutation has the ability to improve the extension and the uniformity of the solutions on the Pareto set, but the capacity of achieving nondominated solutions is decreased. Note that the IA_{gauss} can be obtained from the MCSA by simply making $N_{dN} = 1$ and $N_{dU} = N_{dC} = 0$.

Take in Figure (4)

Take in Figure (5)

Take in Figure (6)

The influence of the receptor editing operator is shown in Figs. 7-9. In a way, we can observe that this operator has a great effect on the coverage ratio of the solutions. Moreover, the coverage ratio obtained for the three ZDT problems came close to 100%, indicating the efficiency of this operator in solving high-dimensional problems. On the other hand, the positive influence of the receptor editing over the distribution of the solutions is not sensible. In a sense, it suggests that if the amplification of the differential variation in editing is incorrectly selected, then the method could present a premature convergence. In this experiment the MCSA performed better only in KUR, ZDT2 and ZDT3. Regarding the convergence speed, the MCSA performed better than the IA_{edit}^- in the ZDT problems, and achieved similar results in POL, FON and KUR.

These tests seem to imply that the editing is very important in improving the NDCSR metric for the algorithm, thus, showing the characteristic of local refinement of high quality solutions that this operator has. However, the uniformity of the solutions in the front is decreased. Therefore, the receptor editing and somatic hypermutation are complementary operators, each one improving different characteristics in the MCSA.

Take in Figure (7)

Take in Figure (8)

Take in Figure (9)

4.3 Electromagnetic Problem

In a SMES system, a large amount of current is kept circulating in a superconducting coil or magnet, which can be supplied to the system when needed. Since there are only insignificant losses in the superconducting coil, the transfer of energy in and out of storage is highly efficient and very fast. However, to remain in its superconducting state, the coil must be cooled to cryogenic temperatures, which requires a sophisticated refrigeration subsystem (Schoenung et al. 1996). A mathematical definition for the electromagnetic device, including its physical constraints, is shown in (Alotto et al. 1996), in which the authors propose a multi-objective benchmark problem. As stated next, the multi-objective version of the TEAM Problem 22 consists of the minimization of the stray magnetic flux density at a certain distance from the device (Fig. 10(a)), and the minimization of the deviation of the stored energy (Dias and Vasconcelos 2002):

$$\begin{aligned} \min F &= \{f_1, f_2\} \\ f_1(\vec{x}) &= (B_{Stray}/B_{norm})^2 \\ f_2(\vec{x}) &= \frac{|Energy - E_{ref}|}{E_{ref}} \\ \text{subject to } \begin{cases} g_1(\vec{x}) = |J_i| + 6.4|B_{max_i}| - 54 \leq 0 \\ g_2(\vec{x}) = (R_1 - R_2) + \frac{1}{2}(d_1 + d_2) < 0 \end{cases} \end{aligned} \quad (3)$$

where the first constraint concerns with the quench condition that guarantees superconductivity (Fig. 10(b)), and the second one guarantees the non-superposition of the inner and outer coils. The stray flux equation is exhibit in (4); $B_{norm} = 3mT$ and $E_{ref} = 180MJ$.

$$B_{Stray} = \sqrt{\frac{\sum_{i=1}^{21} |B_{Si}|^2}{21}} \quad (4)$$

Take in Figure (10)

The variable ranges for the 3D and 8D SMES design are shown in Tables 3 and 4.

Take in Table (3)

Take in Table (4)

Small changes of the control variables can significantly affect the behavior of an electromagnetic device. In the same manner, slight variations of the parameters in the MCSA can significantly alter the design. In this way, a best local refinement may be attained by decreasing the mutation radius ($\gamma_{N,U,C} = 0.05$), and increasing the number of points subjected to the Gaussian mutation ($N_{dN} = 0.70$, $N_{dU} = 0.20$, $N_{dC} = 0.10$). In order to spend a smaller number of function evaluations per generation, the cloning

Table 3: Variable ranges for the 3D SMES design

Var	r_1	h_1	d_1	r_2	h_2	d_2	J_1	J_2
Unit	m	m	m	m	m	m	$\frac{MA}{m^2}$	$\frac{MA}{m^2}$
min	—	—	—	2.6	0.408	0.1	—	—
max	—	—	—	3.4	2.2	0.4	—	—
fixed	2.0	1.6	0.27	—	—	—	22.5	22.5

Table 4: Variable ranges for the 8D SMES design

Var	r_1	h_1	d_1	r_2	h_2	d_2	J_1	J_2
Unit	m	m	m	m	m	m	$\frac{MA}{m^2}$	$\frac{MA}{m^2}$
min	1.0	0.2	0.1	1.8	0.2	0.1	10	10
max	4.0	3.6	0.8	5.0	3.6	0.8	30	30

factor can be reduced as well ($\beta = 0.05$), which ensures a larger number of iterations until achieving the maximum number of evaluations (stop criterion). As the SMES formulation represents a bi-objective problem, the configuration of the NSGA-II has not been modified (Sharma et al. 2007).

Fig. 11 presents the statistical results achieved for the problem with three variables. Each algorithm was executed 10 times with a stop criterion of 3,000 function evaluations. The MCSA achieved a better coverage ratio, but the NSGA-II achieved a better spreading of their solutions. There is no significant difference in the convergence speed of both methods. Fig. 12 presents the results for the problem with eight variables. In this case, the MCSA achieves a better coverage ratio than the NSGA-II and presented a better spreading of solutions. The MCSA also presented a higher convergence speed, illustrating its suitability to solve high-dimensional problems in electromagnetic design.

Take in Figure (11)

Take in Figure (12)

Considering the multi-dimensional nature of the evaluation of multi-objective algorithms, these results show that the MCSA was not dominated by the NSGA-II in the 3D problem, and clearly dominated the NSGA-II in the 8D problem, regarding the metrics adopted.

For the decision-making process, we took one of the Pareto-optimal sets found for the 3D and 8D design (Fig. 13) and eliminated all the solutions in which $B_{Stray}^{3D} > 1mT$, $B_{Stray}^{8D} > 0.4mT$ and $\Delta E > 5\%$, selecting from the remaining solutions the one that corresponds to the smallest volume of coil ($V = 2\pi \sum_{i=1}^2 R_i h_i d_i$).

Take in Figure (13)

Using these criteria, we get the final selected solutions:

$$3D : \left\{ \begin{array}{l} R_2 = 3.0424 \text{ m} \\ h_2 = 0.6269 \text{ m} \\ d_2 = 0.3108 \text{ m} \end{array} \right., 8D : \left\{ \begin{array}{l} R_1 = 1.5011 \text{ m} \\ h_1 = 1.5181 \text{ m} \\ d_1 = 0.5431 \text{ m} \\ R_2 = 2.5718 \text{ m} \\ h_2 = 1.8042 \text{ m} \\ d_2 = 0.1410 \text{ m} \\ J_1 = 16.2932 \text{ MA/m}^2 \\ J_2 = 18.7948 \text{ MA/m}^2 \end{array} \right. \quad (5)$$

at which the objective function values are comparable to the best available in the literature (Alotto et al. 1996), see Table 5.

Take in Table (5)

Table 5: Comparison with the best known results

Variable Unit	B_{Stray} mT	$Energy$ MJ	B_{max} T	$Volume$ m^3
$MCSA^{3D}$	0.7864	174.90	4.2772	3.7247
$TEAM^{3D}$ (Alotto et al. 1996)	0.8896	179.80	3.6300	3.6446
$MCSA^{8D}$	0.3663	182.62	5.4906	11.8885
$TEAM^{8D}$ (Alotto et al. 1996)	0.0148	179.99	—	18.7905

We can note that the results for the 3D design are similar, and although the stray magnetic flux density for the 8D design presented a smaller value in (Alotto et al. 1996), the achieved solution with the MCSA presents a smaller volume of superconducting material, which is an important characteristic used to define the cost of the SMES device.

5 Conclusion

This paper has proposed two new operators which can be employed in the affinity maturation step of multi-objective immune algorithms. The first one performs the somatic hypermutation using different degrees of neighborhood that is based on the use of different probability distribution functions in the generation of perturbations; while the second improves high affinity solutions by editing itself and using difference vectors produced with the best solutions of the population. These operators present complementary roles in the performance of the MCSA, enhancing both the extension and uniformity of its solutions. The results show that the MCSA outperformed the NSGA-II, accomplishing better convergence speed in all high-dimensional problems, including the TEAM Problem 22. In addition, the results for the SMES design indicate that the MCSA represent an effective tool for solving multi-objective real electromagnetic optimization problems.

References

- Alotto, P., Kuntsevitch, A., Magele, C., Molinari, G., Paul, C., Preis, K., Repetto, M. and Richter, K.: 1996, Multiobjective optimization in magnetostatics: a proposal for benchmark problems, *IEEE Transactions On Magnetism* **32**(3), 1238–1241.
- Batista, L. S., Guimarães, F. G. and Ramírez, J. A.: 2009, A distributed clonal selection algorithm for optimization in electromagnetics, *to appear in IEEE Transactions on Magnetism* **45**.
- Campelo, F., Guimarães, F. G., Igarashi, H. and Ramírez, J. A.: 2005, A clonal selection algorithm for optimization in electromagnetics, *IEEE Transactions on Magnetism* **41**(5), 1736–1739.
- Campelo, F., Guimarães, F. G., Igarashi, H., Ramírez, J. A. and Noguchi, S.: 2006, A modified immune network algorithm for multimodal electromagnetic problems, *IEEE Transactions on Magnetism* **42**(4), 1111–1114.
- Chen, L. and Aihara, K.: 1995, Chaotic simulated annealing by a neural network model with transient chaos, *Neural Networks* **8**(6), 915–930.
- Cheng, C.-T., Wang, W.-C., Xu, D.-M. and Chau, K. W.: 2008, Optimizing hydropower reservoir operation using hybrid genetic algorithm and chaos, *Water Resources Management* **22**(7), 895–909.
- Coelho, L. and Alotto, P.: 2008, Multiobjective electromagnetic optimization based on a nondominated sorting genetic approach with a chaotic crossover operator, *IEEE Transactions on Magnetism* **44**(6), 1078–1081.
- de Castro, L. N. and Von Zuben, F. J.: 2000, The clonal selection algorithm with engineering applications, *In Workshop Proceedings of GECCO, Workshop on Artificial Immune Systems and Their Applications*, Las Vegas, USA, pp. 36–37.
- Deb, K., Agarwal, S., Pratab, A. and Meyarivan, T.: 2000, A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II, *Technical Report 200001*, Kanpur Genetic Algorithms Laboratory, Indian Institute of Technology, Kanpur, India.
- Dias, A. and Vasconcelos, J.: 2002, Multiobjective genetic algorithm applied to solve optimization problems, *IEEE Transactions On Magnetism* **38**(2), 1133–1136.
- Fonseca, C. and Fleming, P.: 1998, Multiobjective optimization and multiple constraint handling with evolutionary algorithms - part ii: application example, *IEEE Trans. System, Man, and Cybernetics A* **28**, 38–47.

- George, A. J. and Gray, D.: 1999, Receptor editing during affinity maturation, *Immunol Today* **20**(4), 196.
- Guimarães, F. G., Wanner, E. F. and Takahashi, R. H. C.: 2009, A quality metric for multi-objective optimization based on hierarchical clustering techniques, *Proc. of IEEE CEC 2009*, Trondheim, Norway.
- Jerne, N. K.: 1984, Idiotypic networks and other preconceived ideas, *Immunological Reviews* **79**, 5–24.
- Kursawe, F.: 1990, A variant of evolution strategies for vector optimization, *Proceedings of the 1st Workshop on Parallel Problem Solving from Nature*, Vol. 496, pp. 193–197.
- Mingjun, J. and Huanwen, T.: 2004, Application of chaos in simulated annealing, *Chaos, Solitons and Fractals* **21**(4), 933–941.
- Nussenzweig, M. C.: 1998, Immune receptor editing: revise and select, *Cell* **95**(7), 875–878.
- Poloni, C.: 1995, Hybrid ga for multiobjective aerodynamic shape optimization, *Genetic Algorithms in Engineering and Computer Science* pp. 397–416.
- Schoenung, S., Eyer, J., Iannucci, J. and Horgan, S.: 1996, Energy storage for a competitive power market, *Annual Review of Energy and the Environment* **21**, 347–370.
- Sharma, D., Kumar, A., Deb, K. and Sindhya, K.: 2007, Hybridization of sbx based nsga-ii and sequential quadratic programming for solving multi-objective optimization problems, *Congress on Evolutionary Computation, Special Session & Competition on Performance Assessment of Multi-Objective Optimization Algorithms* pp. 3003–3010.
- Storn, R. M. and Price, K. V.: 1997, Differential evolution: a simple and efficient adaptive scheme for global optimization over continuous spaces, *Journal of Global Optimization* **11**, 341–359.
- Wanner, E., Guimarães, F., Takahashi, R. and Fleming, P.: 2008, Local search with quadratic approximations into memetic algorithms for optimization with multiple criteria, *Evolutionary Computation* **16**(2), 185–224.
- Zitzler, E.: 1999, *Evolutionary algorithms for multiobjective optimization: methods and application*, Ph.d. thesis, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland.

Zitzler, E., Deb, K. and Thiele, L.: 2000, Comparison of multiobjective evolutionary algorithms: empirical results, *Evolutionary Computation* **8**(2), 173–195.

Zitzler, E. and Thiele, L.: 1998, *Multiobjective optimization using evolutionary algorithms - a comparative case study*, Vol. 1498, Springer Berlin / Heidelberg, chapter Parallel Problem Solving from Nature – PPSN V: Lecture Notes in Computer Science, pp. 292–301.

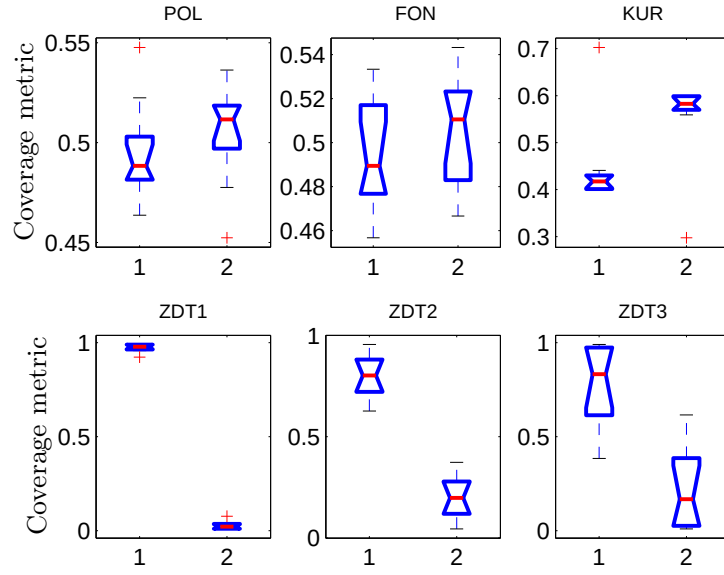


Figure 1: Statistic values of coverage - MCSA $\times$ NSGA-II

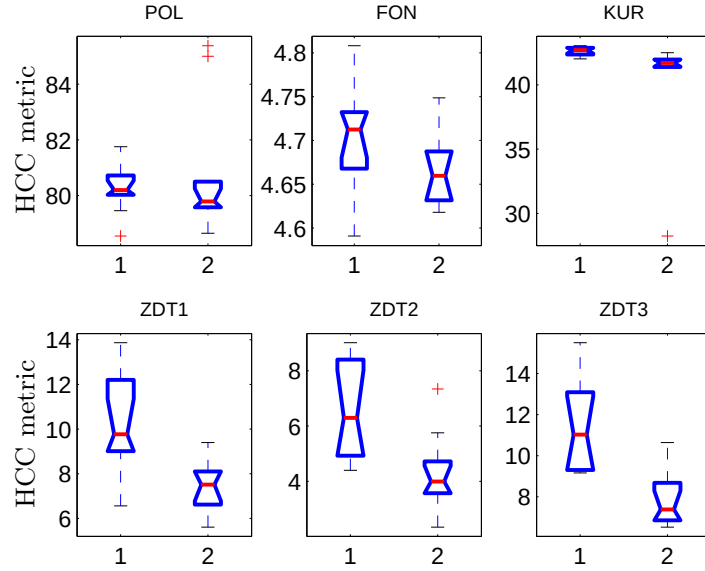


Figure 2: Statistic values of front uniformity - MCSA $\times$ NSGA-II

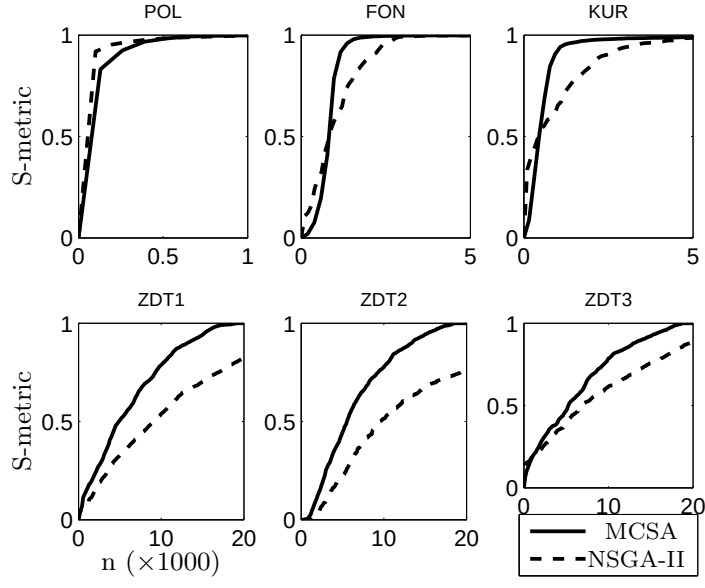


Figure 3: Statistic values of convergence speed - MCSA $\times$ NSGA-II

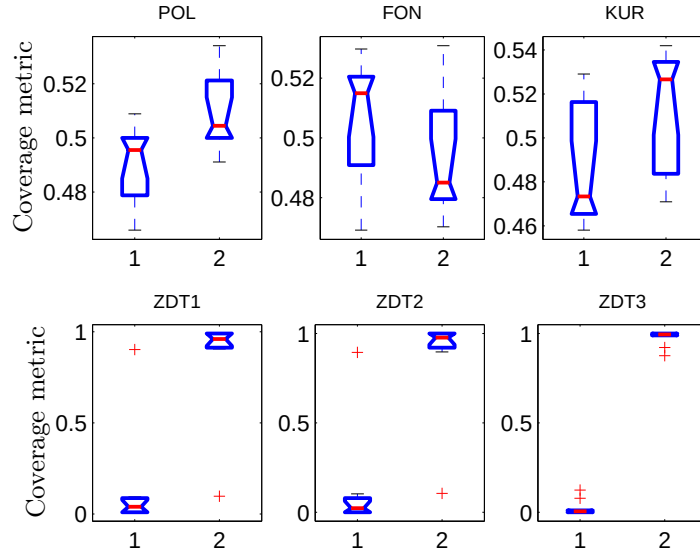


Figure 4: Statistic values of coverage - MCSA $\times$ IA_{gauss}

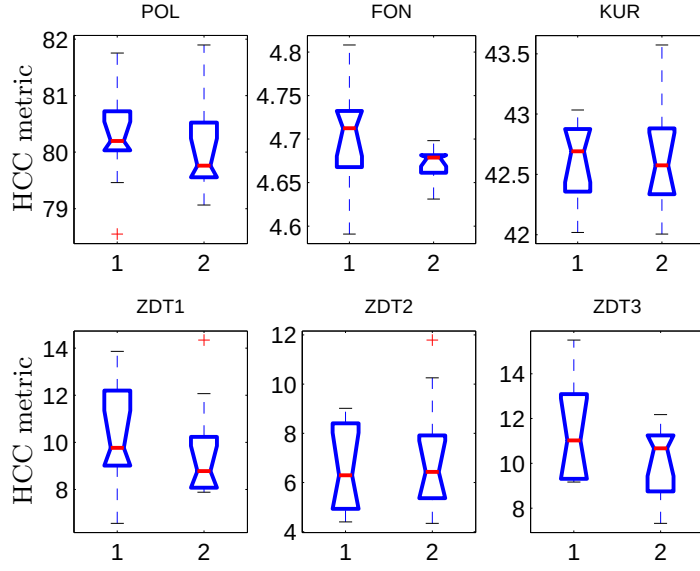


Figure 5: Statistic values of front uniformity - MCSA $\times$ IA_{gauss}

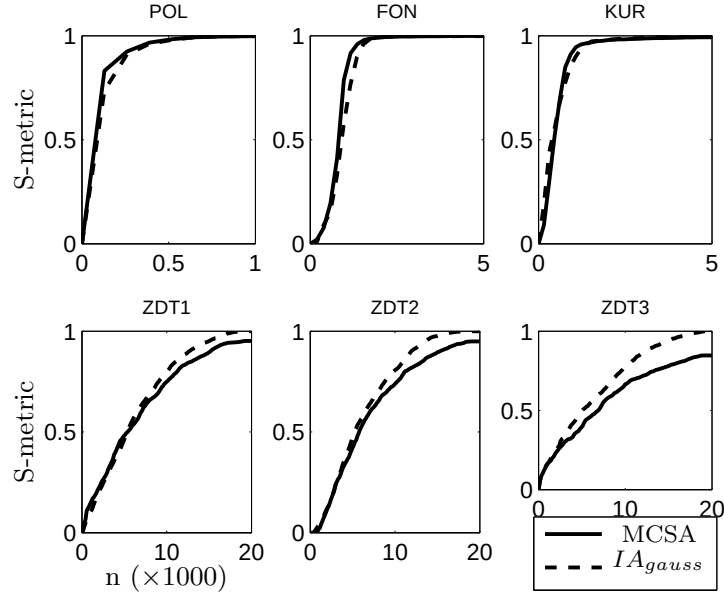


Figure 6: Statistic values of convergence speed - MCSA $\times$ IA_{gauss}

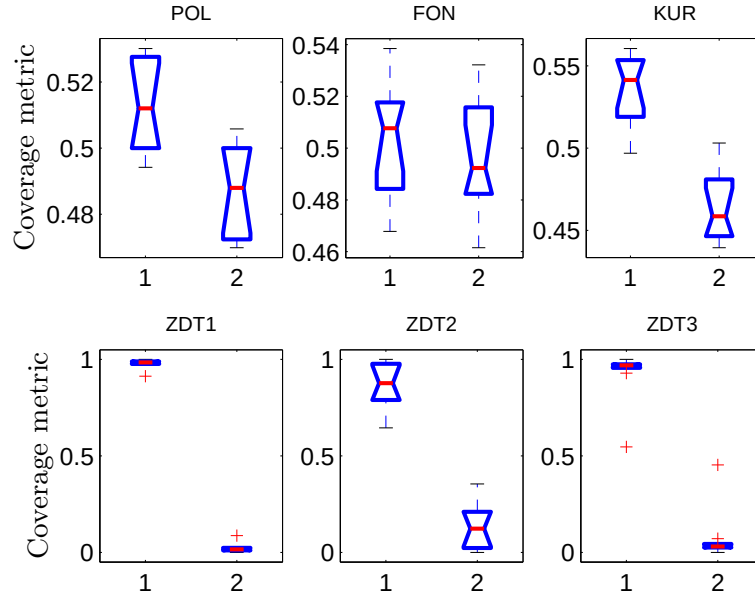


Figure 7: Statistic values of coverage - MCSA $\times$ IA_{edit}^-

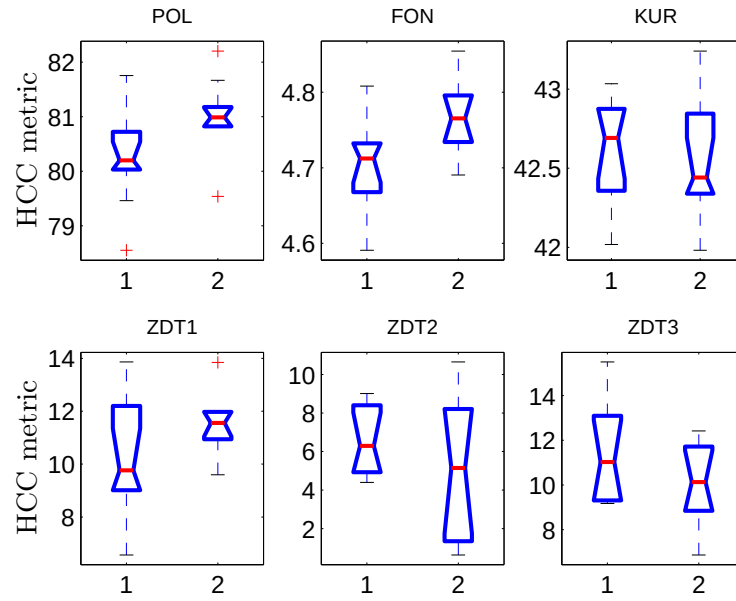


Figure 8: Statistic values of front uniformity - MCSA $\times$ IA_{edit}^-

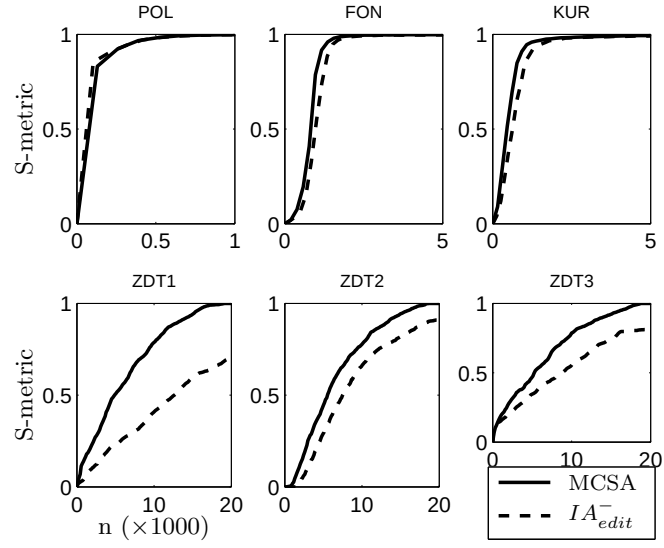


Figure 9: Statistic values of convergence speed - MCSA $\times$ IA_{edit}^-

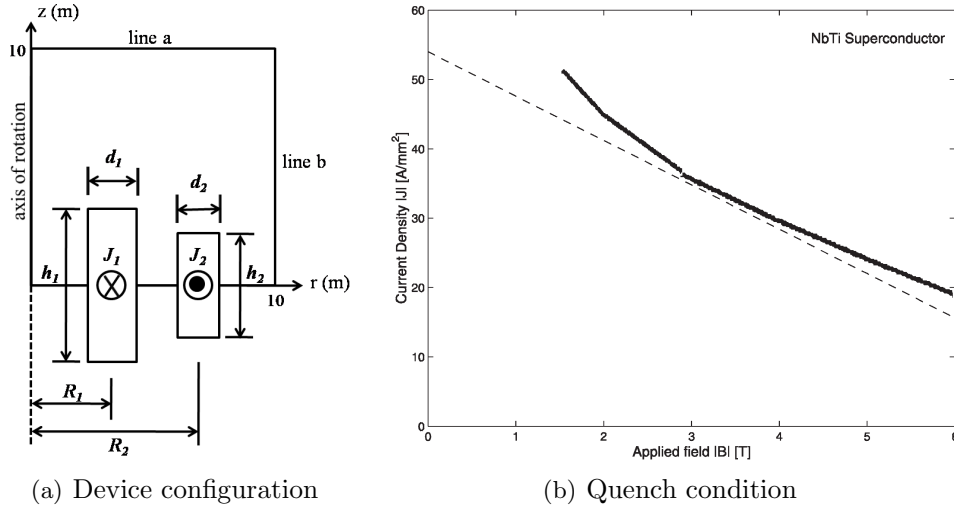


Figure 10: SMES device

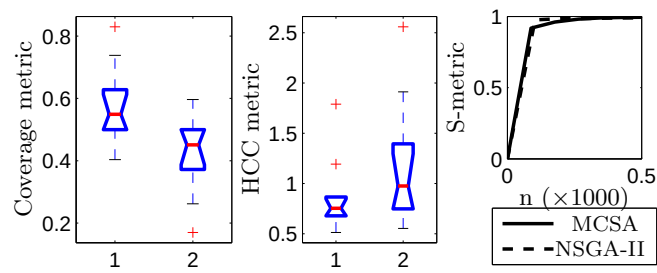


Figure 11: Performance analysis for the 3D SMES

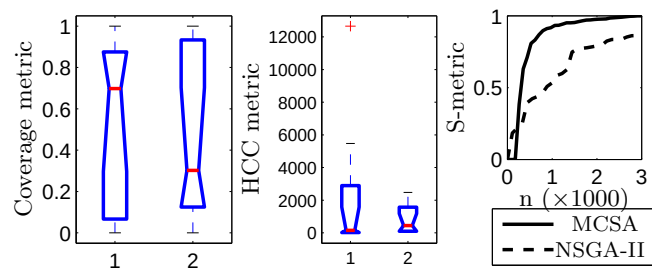


Figure 12: Performance analysis for the 8D SMES

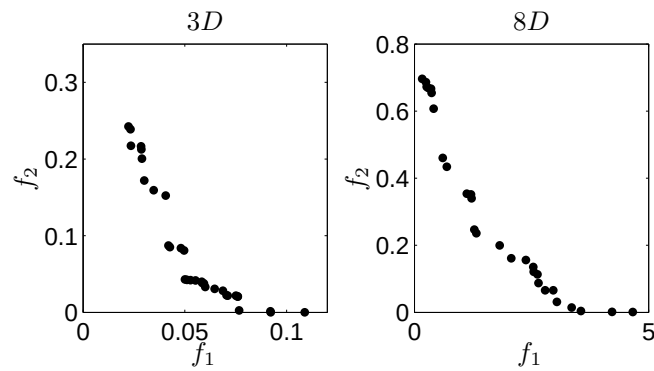


Figure 13: Pareto-optimal front for the SMES problem

Referências

- Abbas, A. K. & Lichtman, A. H. (March 2005). *Cellular and Molecular Immunology*.
- Abbass, H. A. (2002). The self-adaptive pareto differential evolution algorithm, *Proceedings of the IEEE Congress on Evolutionary Computation, CEC*, Vol. 1, IEEE Press, pp. 831–836.
- Alatas, B., Akin, E. & Karci, A. (2008). MODENAR: Multi-objective differential evolution algorithm for mining numeric association rules, *Applied Soft Computing* **8**: 646–656.
- Alotto, P., Kuntsevitch, A., Magele, C., Molinari, G., Paul, C., Preis, K., Repetto, M. & Richter, K. (1996). Multiobjective optimization in magnetostatics: a proposal for benchmark problems, *IEEE Transactions On Magnetics* **32**(3): 1238–1241.
- Amaral, J., Amaral, J., Tanscheit, R. & Pacheco, M. (2004). An immune inspired fault diagnosis system for analog circuits using wavelet signatures, *Proceedings of the 2004 NASA/DoD Conference on Evolution Hardware*, pp. 138–141.
- Amaral, J., Amaral, J., Tanscheit, R., Pacheco, M. & Mesquita, A. (2005). Tuning evolvable pid controllers through a clonal selection algorithm, *Proceedings of the 2005 NASA/DoD Conference of Evolution Hardware*, pp. 30–33.
- Babu, B. & Jehan, M. (2003). Differential evolution for multi-objective optimization, *Proceedings of the IEEE Congress on Evolutionary Computation, CEC*, Vol. 4, IEEE Press, pp. 2696–2703.
- Batista, L. S., Guimarães, F. G., Paul, P. & Ramírez, J. A. (2009). Optimization of electromagnetic devices using artificial immune systems, *to appear in Journal of Microwaves, Optoelectronics and Electromagnetic Applications*.
- Batista, L. S., Guimarães, F. G. & Ramírez, J. A. (2009a). A differential mutation operator for the archive population of multiobjective evolutionary algorithms, *to*

- appear in IEEE Congress on Evolutionary Computation* .
- Batista, L. S., Guimarães, F. G. & Ramírez, J. A. (2009b). A distributed clonal selection algorithm for optimization in electromagnetics, *IEEE Transactions on Magnetics* **45**(3): 1598–1601.
- Batista, L. S., Guimarães, F. G. & Ramírez, J. A. (2009c). New operators for multi-objective clonal selection algorithms, *to appear in The International Journal for Computation and Mathematics in Electrical Engineering (COMPEL)* **28**(4).
- Bazaraa, M. S., Sherali, H. D. & Shetty, C. M. (1979). *Nonlinear Programming: Theory and Algorithms*, 2 edn, John Wiley & Sons.
- Branco, P. J. C., Dente, J. A. & Mendes, R. V. (2003). Using immunology principles for fault detection, *IEEE Transactions On Industrial Electronics* **50**(2): 362–373.
- Burnet, F. M. (1959). The clonal selection theory of acquired immunity, Cambridge Univ. Press.
- Burnet, F. M. (1978). Clonal selection and after, *Theoretical Immunology* pp. 63–85.
- Cai, J., Ma, X., Li, L., Yang, Y., Peng, H. & Wang, X. (2007). Chaotic ant swarm optimization to economic dispatch, *Electric Power Systems Research* **77**(10): 1373–1380.
- Campelo, F. (2006). *Study on immune algorithms for the optimization of electromagnetic systems*, Master's thesis, Graduate School of Information Science and Technology, Hokkaido University.
- Campelo, F., Guimarães, F. G., Igarashi, H. & Ramírez, J. A. (2005). A clonal selection algorithm for optimization in electromagnetics, *IEEE Transactions on Magnetics* **41**(5): 1736–1739.
- Campelo, F., Guimarães, F. G., Igarashi, H., Ramírez, J. A. & Noguchi, S. (2006). A modified immune network algorithm for multimodal electromagnetic problems, *IEEE Transactions on Magnetics* **42**(4): 1111–1114.
- Castiglione, F., Motta, S. & Nicosia, G. (2001). Pattern recognition by primary and secondary response of an artificial immune system, *Theory Biosci* **120**: 93–106.
- Chen, G. & Dong, X. (1998). *From Chaos to Order: Methodologies, Perspectives and Applications*, Vol. 24 of A.

- Chen, L. & Aihara, K. (1995). Chaotic simulated annealing by a neural network model with transient chaos, *Neural Networks* **8**(6): 915–930.
- Chen, L. & Aihara, K. (1997). Chaos and asymptotical stability in discrete-time neural networks, *Physica D* **104**(3): 286–325.
- Coelho, L. & Alotto, P. (2008). Multiobjective electromagnetic optimization based on a nondominated sorting genetic approach with a chaotic crossover operator, *IEEE Transactions on Magnetics* **44**(6): 1078–1081.
- Coelho, L. & Mariani, V. (2006). Combining of chaotic differential evolution and quadratic programming for economic dispatch optimization with valve-point effect, *IEEE Transactions on Power Systems* **21**(2): 989–996.
- Coello, C. A. C. & Cortés, N. C. (2002). An approach to solve multiobjective optimization problems based on an artificial immune system, *First International Conference on Artificial Immune Systems* pp. 212–221.
- Coello, C. A. C. & Cortés, N. C. (2005). Solving multiobjective optimization problems using an artificial immune system, *Genetic Programming and Evolvable Machines* **6**(2): 163–190.
- Colorni, A., Dorigo, M. & Maniezzo, V. (1992). Distributed optimization by ant colonies, *Proceedings of the First European Conference on Artificial Life* .
- Corne, D. W., Knowles, J. D. & Oates, M. J. (2000). The pareto envelope-based selection algorithm for multiobjective optimization, *Proceedings of the Parallel Problem Solving from Nature VI Conference*, Vol. 1917, pp. 839–848.
- Cziko, G. (1995). *The Immune System: Selection by the Enemy*, In Without Miracles.
- Dasgupta, D. (1997). Artificial neural networks and artificial immune systems:similarities and differences, *IEEE International Conference on Systems, Man, and Cybernetics* **1**: 873–878.
- Dasgupta, D. (1999a). *Artificial Immune Systems and Their Applications*.
- Dasgupta, D. (1999b). Immunity-based intrusion detection system: A general framework, *In Proc. of the 22nd National Information Systems Security Conference*.
- Dasgupta, D. (2007). Artificial immune systems: A bibliography, *Technical report*, The University Of Memphis, USA.

- Dashora, Y., Kumar, S., Shukla, N. & Tiwari, M. (2008). Improved and generalized learning strategies for dynamically fast and statistically robust evolutionary algorithms, *Engineering Applications of Artificial Intelligence* **21**(4): 525–547.
- de Castro, L. N. (2002a). Immune engineering: A personal account, *II Workshop on Computational Intelligence and Semiotics*, São Paulo, Brazil.
- de Castro, L. N. (2002b). Immune, swarm, and evolutionary algorithms part i: Basic models, *Proc. of the International Conference on Neural Information Processing*.
- de Castro, L. N. (2002c). Immune, swarm, and evolutionary algorithms part ii: Philosophical comparisons, *Proc. of the International Conference on Neural Information Processing*.
- de Castro, L. N. & Timmis, J. (2002a). An artificial immune network for multimodal function optimization, *Proceedings of the IEEE Congress on Evolutionary Computation*, Vol. 1, pp. 669–674.
- de Castro, L. N. & Timmis, J. (2002b). Artificial immune system: A novel paradigm to pattern recognition, *In Artificial Neural Networks in Pattern Recognition*.
- de Castro, L. N. & Timmis, J. (2002c). *Artificial Immune Systems: A New Computational Intelligence Approach*, 1 edn, Springer-Verlag, Berlin.
- de Castro, L. N. & Von Zuben, F. J. (1999). Artificial immune systems: Part i - basic theory and applications, *Technical report*, FEEC/Univ. Campinas, Campinas, Brazil.
- de Castro, L. N. & Von Zuben, F. J. (2000a). Artificial immune systems: Part ii - a survey of applications, *Technical report*, FEEC/Univ. Campinas, Campinas, Brazil.
- de Castro, L. N. & Von Zuben, F. J. (2000b). The clonal selection algorithm with engineering applications, *In Workshop Proceedings of GECCO*.
- de Castro, L. N. & Von Zuben, F. J. (2002). Learning and optimization using the clonal selection principle, *IEEE Transactions on Evolutionary Computation* **6**(3): 239–251.
- Deb, K., Agarwal, S., Pratab, A. & Meyarivan, T. (2000). A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II, *Technical Report 200001*, KanGAL, Indian Institute of Technology, Kanpur, India.
- Deb, K. & Agrawal, R. B. (1995). Simulated binary crossover for continuous search

- space, *Complex System* **9**(2): 115–148.
- Deb, K. & Beyer, H. G. (2001). Self-adaptive genetic algorithms with simulated binary crossover, *Complex Systems* **9**(2): 197–221.
- Deb, K. & Goel, T. (2001). Controlled elitist non-dominated sorting genetic algorithms for better convergence, *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization*, Vol. 1993, pp. 67–81.
- Deb, K. & Goyal, M. (1996). A combined genetic adaptive search (geneas) for engineering design, *Computer Science and Informatics* **26**: 30–45.
- Deb, K., Karthik, S. & Okabe, T. (2007). Self-adaptive simulated binary crossover for real-parameter optimization, *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pp. 1187–1194.
- Deb, K., Pratap, A., Agarwal, S. & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii, *IEEE Transactions on Evolutionary Computation* **6**(2): 182–197.
- Dias, A. & Vasconcelos, J. (2002). Multiobjective genetic algorithm applied to solve optimization problems, *IEEE Transactions On Magnetics* **38**(2): 1133–1136.
- Farmer, J. D., Packard, N. H. & Perelson, A. S. (1986). The immune system, adaptation, and machine learning, *Physica D* **2**: 187–204.
- Fonseca, C. & Fleming, P. (1998a). Multiobjective optimization and multiple constraint handling with evolutionary algorithms - part ii: application example, *IEEE Trans. System, Man, and Cybernetics A* **28**: 38–47.
- Fonseca, C. M. & Fleming, P. J. (1993). Multiobjective genetic algorithms, *Proceedings of the IEE Colloquium on Genetic Algorithms for Control Systems Engineering*, pp. 6/1–5.
- Fonseca, C. M. & Fleming, P. J. (1998b). Multiobjective optimization and multiple constraint handling with evolutionary algorithms. i. a unified formulation, *IEEE Transactions on Systems, Man and Cybernetics, Part A* **28**(1): 26–37.
- Fonseca, C. M. & Fleming, P. J. (1998c). Multiobjective optimization and multiple constraint handling with evolutionary algorithms. ii. application example, *IEEE Transactions on Systems, Man and Cybernetics, Part A* **28**(1): 38–47.

- George, A. J. & Gray, D. (1999). Receptor editing during affinity maturation, *Immunol Today* **20**(4): 196.
- Goldsby, R. A., Kindt, T. J. & Osborne., B. A. (2000). *Kuby Immunology*.
- Gong, M., Zhang, L., Jiao, L. & Gou, S. (2007). Solving multiobjective clustering using an immune-inspired algorithm, *IEEE Congress on Evolutionary Computation* pp. 15–22.
- Gong, W. & Cai, Z. (2008). A multiobjective differential evolution algorithm for constrained optimization, *Proceedings of the IEEE Congress on Evolutionary Computation, CEC*, IEEE Press, pp. 181–188.
- Gu, J., Lee, D., Park, S. & Sim, K. (2000). An immunity-based security layer model, *Proc. of GECCO, Workshop on Artificial Immune Systems and Their Applications*, pp. 47–48.
- Guimarães, F. G., Palhares, R. M., Campelo, F. & Igarashi, H. (2007). Design of mixed h2/hinf control systems using algorithms inspired by the immune system, *Information Sciences: an International Journal* **177**(20): 4368–4386.
- Guimarães, F. G., Wanner, E. F. & Takahashi, R. H. C. (2009). A quality metric for multi-objective optimization based on hierarchical clustering techniques, *Proc. of IEEE CEC 2009*, Trondheim, Norway.
- Guzella, T. S. & Caminhas, W. M. (2009). A review of machine learning approaches to spam filtering, *Expert Systems with Applications* .
- Guzella, T. S., Santos, T. A. M. & Caminhas, W. M. (2007). A novel immune inspired approach to fault detection, *Lecture Notes in Computer Science* **4628**: 107–118.
- Guzella, T. S., Santos, T. A. M., Uchôa, J. Q. & Caminhas, W. M. (2008). Identification of spam messages using an approach inspired on the immune system, *Biosystems* **92**: 215–225.
- Hayakawa, Y., Marumoto, A. & Sawada, Y. (1995). Effects of the chaotic noise on the performance of a neural network model for optimization problems, *Phys. Rev. E* **51**(4): 2693–2696.
- He, Y. (2002). Chaotic simulated annealing with decaying chaotic noise, *IEEE Transactions on Neural Networks* **13**(6): 1526–1531.

- Hernandez-Diaz, A. G., Santana-Quintero, L. V., Coello, C. A. C., Caballero, R. & Molina, J. (2006). A new proposal for multi-objective optimization using differential evolution and rough sets theory, *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation, GECCO*, ACM, pp. 675–682.
- Hofmeyr, S. A. (2000). An interpretative introduction to the immune system.
- Hofmeyr, S. A. & Forrest, S. (1999). Immunity by design: An artificial immune system, *In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, pp. 1289–1296.
- Hofmeyr, S. A. & Forrest, S. A. (2000). Architecture for an artificial immune system, *Evolutionary Computation* **8**(4): 443–473.
- Horn, J., Nafpliotis, N. & Goldberg, D. E. (1993). Multiobjective optimization using the niched pareto genetic algorithm, *Technical Report 93005*, Illinois Genetic Algorithms Laboratory (IlliGAL).
- Horn, J., Nafpliotis, N. & Goldberg, D. E. (1994). A niched pareto genetic algorithm for multiobjective optimization, *In Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence*, Vol. 1, pp. 82–87.
- Hunt, J. E. & Cooke, D. E. (1996). Learning using an artificial immune system, *Journal of Network and Computer Applications* **19**(2): 189–212.
- Iorio, A. W. & Li, X. (2006). Incorporating directional information within a differential evolution algorithm for multi-objective optimization, *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation, GECCO*, ACM, pp. 691–697.
- Jerne, N. K. (1984). Idiotypic networks and other preconceived ideas, *Immunological Reviews* **79**: 5–24.
- Jiao, D., Zhu, X.-Y. & Jin, J.-M. (1999). Fast and accurate frequency-sweep calculations using asymptotic waveform evaluation and the combined-field integral equation, *Radio Science* **34**(5): 1055–1063.
- Kelsey, J. & Timmis, J. (2003). Immune inspired somatic contiguous hypermutation for function optimisation, *Genetic and Evolutionary Computation Conference* **2723**: 207–218.

- Kennedy, J. & Eberhart, R. (1995). Particle swarm optimization, *Proceedings., IEEE International Conference on Neural Networks*. **4**: 1942–1948.
- Kirkpatrick, S., Jr., C. D. G. & Vecchi, M. P. (1983). Optimization by simulated annealing, *Science* **220**(4598): 671–680.
- Krautmacher, M. & Dilger, W. (2004). Ais based robot navigation in a rescue scenario, *Artificial Immune Systems*, Vol. 3239, Springer Berlin / Heidelberg, pp. 106–118.
- Kursawe, F. (1990). A variant of evolution strategies for vector optimization, *Proceedings of the 1st Workshop on Parallel Problem Solving from Nature*, Vol. 496, pp. 193–197.
- Lorenz, E. N. (1963). Deterministic nonperiodic flow, *Journal of the Atmospheric Sciences* **20**(2): 130–141.
- Luh, G.-C. & Liu, W.-W. (2004). Reactive immune network based mobile robot navigation, *Lecture Notes In Computer Science - International Conference On Artificial Immune Systems* **3239**(3): 119–132.
- Luo, C. & Shao, H. (2000). Evolutionary algorithm with chaotic mutations, *Control and Decision* **15**(5): 557–560.
- Madavan, N. K. (2002). Multiobjective optimization using a pareto differential evolution approach, *Proceedings of the IEEE Congress on Evolutionary Computation, CEC*, Vol. 2, IEEE Press, pp. 1145–1150.
- Matzinger, P. (1994). Tolerance, danger, and the extended family, *Annual reviews of Immunology* **12**: 991–1045.
- Matzinger, P. (2001). The danger model in its historical context, *Scandinavian Journal of Immunology* **54**: 4–9.
- May, R. M. (1976). Simple mathematical models with very complicated dynamics, *Nature* **261**: 459–467.
- McCoy, D. F. & Devarajan, V. (1997). Artificial immune systems and aerial image segmentation, *IEEE International Conference on Systems, Man, and Cybernetics* **1**: 867–872.
- Meeker, D. (1994–2000). FEMM 4.0 software, Online.
- Mingjun, J. & Huanwen, T. (2004). Application of chaos in simulated annealing, *Chaos, Solitons and Fractals* **21**(4): 933–941.

- Nossal, G. J. V. (1994). Negative selection of lymphocytes, *Cell* **76**: 229–239.
- Nozawa, H. (1992). A neural network model as a globally coupled map and applications based on chaos, *Chaos* **2**(3): 377–386.
- Nussenzweig, M. C. (1998). Immune receptor editing: revise and select, *Cell* **95**(7): 875–878.
- Okamoto, T. & Ishida, Y. (1999). Multiagent approach against computer virus: an immunity-based system, *Proc. of the AROB*, pp. 69–72.
- Parsopoulos, K., Tasoulis, D., Pavlidis, N., Plagianakos, V. & Vrahatis, M. (2004). Vector evaluated differential evolution for multiobjective optimization, *Proceedings of the IEEE Congress on Evolutionary Computation, CEC*, Vol. 1, IEEE Press, pp. 204–211.
- Paul, P., Guimarães, F. G., Nair, D. & Lowther, D. A. (2008). A clonal selection algorithm with varying order finite elements for the optimization of microwave devices, *Microwave and Optical Technology Letters* **50**(5): 1392–1397.
- Poloni, C. (1995). Hybrid ga for multiobjective aerodynamic shape optimization, *Genetic Algorithms in Engineering and Computer Science* pp. 397–416.
- Price, K. V., Storn, R. M. & Lampinen, J. A. (2005). *Differential Evolution: A Practical Approach to Global Optimization*, Natural Computing Series, Springer.
- Qian, W. & Li, A. (2008). Adaptive differential evolution algorithm for multiobjective optimization problems, *Applied Mathematics and Computation* **201**: 431–440.
- Robic, T. & Filipic, B. (2005). DEMO: Differential evolution for multiobjective optimization, in C. A. C. Coello, A. H. Aguirre & E. Zitzler (eds), *Proceedings of the Third International Conference on Evolutionary Multi-Criterion Optimization, EMO, Lecture Notes in Computer Science*, Springer, pp. 520–533.
- Sarker, R. & Abbass, H. A. (2004). Differential evolution for solving multiobjective optimization problems, *Asia-Pacific Journal of Operational Research* **21**(2): 225–240.
- Schaffer, J. D. (1984). *Some Experiments in Machine Learning Using Vector Evaluated Genetic Algorithms*, PhD thesis, Vanderbilt University.
- Schoenung, S., Eyer, J., Iannucci, J. & Horgan, S. (1996). Energy storage for a compet-

- itive power market, *Annual Review of Energy and the Environment* **21**: 347–370.
- Silverman, B. (1986). Density estimation for statistics and data analysis, Published in Monographs on Statistics and Applied Probability, London: Chapman and Hall.
- Srinivas, N. & Deb, K. (1994). Multiobjective optimization using nondominated sorting in genetic algorithms, *Evolutionary Computation Journal* **2**(3): 221–248.
- Storn, R. M. & Price, K. V. (1997). Differential evolution: A simple and efficient adaptive scheme for global optimization over continuous spaces, *Journal of Global Optimization* **11**: 341–359.
- Takahashi, R. H. C. (2004). Otimização escalar e vetorial.
- Takahashi, R. H. C., Vasconcelos, J. A., Ramírez, J. A. & Krahenbuhl, L. (2003). A multiobjective methodology for evaluating genetic operators, *IEEE Transactions On Magnetism* **39**(3): 1321–1324.
- Talmage, D. W. (1957). Allergy and immunology, *Annual Review of Medicine*, Vol. 8, pp. 239–256.
- Timmis, J., Neal, M. & Hunt, J. (2000). An artificial immune system for data analysis, *BioSystems* **55**: 143–150.
- Tokuda, I., Aihara, K. & Nagashima, T. (1998). Adaptive annealing for chaotic optimization, *Physical Review E* **58**(4): 5157–5160.
- Vasconcelos, J. A., Ramírez, J. A., Takahashi, R. H. C. & Saldanha, R. R. (2001). Improvements in genetic algorithms, *IEEE Transactions On Magnetism* **37**(5): 3414–3417.
- Wang, L. (1996). Oscillatory and chaotic dynamics in neural networks under varying operating conditions, *IEEE Transactions on Neural Networks* **7**(6): 1382–1388.
- Wang, L. & Smith, K. (1998). On chaotic simulated annealing, *IEEE Transactions on Neural Networks* **9**(4): 716–718.
- Wanner, E., Guimarães, F., Takahashi, R. & Fleming, P. (2008). Local search with quadratic approximations into memetic algorithms for optimization with multiple criteria, *Evolutionary Computation* **16**(2): 185–224.
- Xue, F., Sanderson, A. C. & Graves, R. J. (2003a). Multi-objective differential evolution and its application to enterprise planning, *IEEE International Robotics and*

- Automation* **3**: 3535–3541.
- Xue, F., Sanderson, A. C. & Graves, R. J. (2003b). Pareto-based multi-objective differential evolution, *IEEE Congress on Evolutionary Computation* **2**: 862–869.
- Xue, F., Sanderson, A. C. & Graves, R. J. (2005). Multi-objective differential evolution - algorithm, convergence analysis, and applications, *IEEE Congress on Evolutionary Computation* **1**: 743–750.
- Yang, L. & Chen, T. (2002). Application of chaos in genetic algorithms, *Communications in Theoretical Physics* **38**(2): 168–172.
- Yuan, X., Yuan, Y. & Zhang, Y. (2002). A hybrid chaotic genetic algorithm for short-term hydro system scheduling, *Mathematics and Computers in Simulation* **59**(4): 319–327.
- Zhou, C. & Chen, T. (2000). Chaotic neural networks and chaotic annealing, *Neurocomputing* **30**(1): 293–300.
- Zilong, G., Sun'an, W. & Jian, Z. (2006). A novel immune evolutionary algorithm incorporating chaos optimization, *Pattern Recognition Letters* **27**(1): 2–8.
- Zitzler, E. (1999). *Evolutionary algorithms for multiobjective optimization: Methods and application*, Master's thesis, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland.
- Zitzler, E., Deb, K. & Thiele, L. (2000). Comparison of multiobjective evolutionary algorithms: Empirical results, *Evolutionary Computation* **8**(2): 173–195.
- Zitzler, E., Laumanns, M. & Thiele, L. (2001). SPEA2: improving the strength pareto evolutionary algorithm for multiobjective optimization, *Technical Report 103*, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Gloriastrasse 35, CH-8092 Zurich, Switzerland.
- Zitzler, E. & Thiele, L. (1998). *Multiobjective optimization using evolutionary algorithms - a comparative case study*, Vol. 1498, Springer Berlin / Heidelberg, chapter Parallel Problem Solving from Nature – PPSN V: Computer Science, pp. 292–301.
- Zitzler, E. & Thiele, L. (1999). Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach, *IEEE Transactions on Evolutionary Computation* **3**(4): 257–271.