



INTELIGÊNCIA DE ENXAMES APLICADA AO PROBLEMA DE OTIMIZAÇÃO
DE RECARGAS DE REATORES NUCLEARES A ÁGUA PRESSURIZADA

Ioná Maghali Santos de Oliveira

Tese de Doutorado apresentada ao Programa de Pós-graduação em Engenharia Nuclear, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Doutor em Engenharia Nuclear.

Orientador: Roberto Schirru

Rio de Janeiro

Julho de 2013

INTELIGÊNCIA DE ENXAMES APLICADA AO PROBLEMA DE OTIMIZAÇÃO
DE RECARGAS DE REATORES NUCLEARES A ÁGUA PRESSURIZADA

Ioná Maghali Santos de Oliveira

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

Examinada por:

Prof. Roberto Schirru, D.Sc.

Prof. Fernando Carvalho da Silva, D.Sc.

Prof. Antonio Carlos Marques Alvim, Ph.D

Prof. Cláudio Márcio do Nascimento Abreu Pereira, D.Sc.

Dr. Celso Marcelo Franklin Lapa, D.Sc.

Dr. Jorge Luiz Cachoeira Chapot, D.Sc.

RIO DE JANEIRO, RJ - BRASIL

JULHO DE 2013

Oliveira, Ioná Maghali Santos de

Inteligência de Enxames Aplicada ao Problema de Otimização de Recargas de Reatores Nucleares a Água Pressurizada / Ioná Maghali Santos de Oliveira. – Rio de Janeiro: UFRJ/COPPE, 2013.

XIII, 87 p.: il.; 29,7 cm.

Orientador: Roberto Schirru

Tese (doutorado) – UFRJ/ COPPE/ Programa de Engenharia Nuclear, 2013.

Referências Bibliográficas: p. 81-87.

1. Inteligência de Enxames. 2. Colônias de Abelhas Artificiais. 3. Otimização de Recargas de Reatores Nucleares. I. Schirru, Roberto. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Nuclear. III. Título.

*Dedico este trabalho
ao meu querido amigo
Marcos Oliveira de Pinho.*

Meus Agradecimentos

Gostaria de agradecer o suporte, a estrutura, os ensinamentos e o apoio incondicional de minha mãe, Dona Enecir, a melhor pessoa que eu conheço. O mais precioso ser humano com quem convivo desde sempre. Devo a ela tudo o que sou, em todos os aspectos. Não teria chegado a este momento tão especial e importante de minha vida, se ela não fosse a pessoa forte, corajosa e determinada que ela é. Minha mãe é a minha principal referência, o exemplo a ser seguido e o meu maior orgulho.

Estendo minha gratidão ao meu amado Eduardo, por todo o apoio e carinho a mim dedicados durante esta longa jornada que foi o doutorado. Seu amor, sua dedicação e sua infinita paciência, foram fundamentais para que eu mantivesse a calma necessária para concluir mais este trabalho. Agradeço também por toda a compreensão nos meus momentos mais difíceis, especialmente aqueles em que minhas obrigações muitas vezes se sobrepunham aos tão desejados momentos de lazer e de distração.

Mãe, “Duca”:

Amo vocês profundamente.

Particolare Ringraziamento

Desidero esprimere il mio più sentito ringraziamento al mio tutor, Prof. Roberto Schirru, che ha seriamente e pazientemente seguito il mio lavoro, guidandomi con competenza durante tutti questi anni di dottorato, sempre facendomi vivere il lavoro di ricerca in modo sereno.

“... Perhaps this final act was meant, to clinch a lifetime’s argument; that nothing comes from violence, and nothing ever could; for all those born beneath an angry star, lest we forget how fragile we are...”

*Fragile
Sting*

“Sete coisas distinguem o sábio do ignorante: o sábio não fala em presença de quem o ultrapasse em ciência ou em idade; nunca interrompe quem fala; não responde com precipitação; interroga com método e responde com precisão; discute as questões segundo a ordem em que foram colocadas; quando não compreende, confessa com franqueza; e curva-se diante da verdade. O ignorante faz precisamente o contrário.”

Tratado da Ética dos Pais.

Capítulo 5. Mishná 9.

Agradecimentos

A Deus, “... *Porque Dele, por Ele, e para Ele são todas as coisas; glória, pois, a Ele eternamente...*” (Rm. 11:36).

Ao meu orientador, Prof. Roberto Schirru, por sua aceitação em me orientar e por ter acreditado na realização deste trabalho. Foi uma grande honra estar ao seu lado, aprender com sua vasta experiência e sabedoria e ainda poder desfrutar de toda a sua amizade e simpatia.

Ao Prof. Paulo Fernando Ferreira Frutuoso e Melo, por sua amizade e por sua generosidade ao permitir meu ingresso na área de Engenharia de Fatores Humanos. Agradeço também por todo o apoio, por todos os ensinamentos e por toda a contribuição em todos os momentos.

Aos ilustres Professores: Fernando Carvalho da Silva, Antonio Carlos Marques Alvim, Cláudio Márcio do Nascimento Abreu Pereira, Celso Marcelo Franklin Lapa e Jorge Luiz Cachoeira Chapot, membros da banca examinadora, por terem dedicado parte de seus preciosos momentos na leitura de meu trabalho, e pelas valiosas contribuições para a melhoria desta tese.

Ao Programa de Engenharia Nuclear (PEN/COPPE/UFRJ) e a todos os seus funcionários, em especial a Jô, a Tânia, a Suzana, a Lili e ao Reginaldo, e também ao Professor Su Jian, em nome de todo o corpo docente, pela atenção e cordialidade.

Ao Laboratório de Monitoração de Processos (LMP) e aos grandes amigos que fiz por lá. Em especial: ao Márcio Henrique, ao Professor Canedo, ao Toni Anderson, a Andressa Nicolau, ao Marcos César, ao Eduardo Valdetaro, ao Anderson Meneses, e ao meu queridíssimo amigo Alan Miranda, por todo apoio e carinho, e pelas profícuas trocas de informações técnicas sobre o assunto abordado nesta tese.

Aos amigos que fiz no Instituto de Matemática Pura e Aplicada (IMPA). Em especial, agradeço ao Professor Mikhail Solodov, por toda atenção dispensada.

Aos meus queridos amigos da Universidade do Colorado (*University of Colorado Boulder*), pelas longas discussões sobre otimização e entomologia nos intervalos de nossas partidas de xadrez pelo *Yahoo! Games*.

Aos meus melhores amigos: “Nêga” e “Tigrão”, e ao meu mais novo amigo “Aquiles”, pela companhia nos muitos momentos de enclausuramento.

Ao CNPq, pelo apoio financeiro.

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

INTELIGÊNCIA DE ENXAMES APLICADA AO PROBLEMA DE OTIMIZAÇÃO DE RECARGAS DE REATORES NUCLEARES A ÁGUA PRESSURIZADA

Ioná Maghali Santos de Oliveira

Julho/2013

Orientador: Roberto Schirru

Programa: Engenharia Nuclear

O Algoritmo de Otimização por Colônia de Abelhas Artificiais, do inglês, *Artificial Bee Colony (ABC)* (KARABOGA, 2005), é um dos mais recentes membros da Inteligência de Enxames. Trata-se de uma Metaheurística Baseada em Populações (MBP) de indivíduos, na qual os candidatos à solução evoluem através da simulação de um modelo simplificado do comportamento forrageiro de uma colônia de abelhas melíferas reais. Este trabalho tem como objetivo demonstrar a viabilidade de aplicação do algoritmo ABC para solucionar o Problema de Otimização de Recargas (POR) de um Reator Nuclear a Água Pressurizada (RNAP). Associado com a operação periódica de substituição de parte do combustível de uma usina nuclear, o POR é um extraordinário problema de Análise Combinatória que há décadas vem desafiando especialistas da área nuclear devido ao seu alto grau de complexidade. Para realizar esta demonstração, a recarga do ciclo 7 de operação da Central Nuclear de Angra 1 é tomada como referência neste estudo, tendo-se como objetivo principal, a maximização da Concentração de Boro ao final do ciclo. Duas metodologias foram desenvolvidas a fim de exemplificar o desempenho do algoritmo ABC como otimizador para o POR. A primeira delas utiliza o modelo de Chaves Aleatórias (*Random Keys*) (BEAN, 1994) e a segunda é uma versão binária discreta do algoritmo ABC original. Aplicado de forma pioneira para a solução do POR, o ABC foi capaz de fornecer resultados superiores aos obtidos por técnicas consideradas estado da arte para este problema, ficando assim demonstrado o potencial promissor desta nova metaheurística para a solução de problemas com alto grau de complexidade, como é o caso do POR.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

SWARM INTELLIGENCE APPLIED TO THE RELOAD OPTIMIZATION
PROBLEM OF PRESSURIZED WATER NUCLEAR REACTORS

Ioná Maghali Santos de Oliveira

July/2013

Advisor: Roberto Schirru

Department: Nuclear Engineering

The *Artificial Bee Colony (ABC)* optimization algorithm (KARABOGA, 2005), is one of the most recent members of Swarm Intelligence. It is a Population-Based Metaheuristic (PBM) of individuals, in which the candidates to the solution evolve by the simulation of a simplified model of the foraging behavior of a colony of real honeybees. This work aims to demonstrate the viability of applying the ABC algorithm to solve the Reload Optimization Problem (ROP) of a Pressurized Water Nuclear Reactor (PWR). Associated with the periodic operation of replacing part of the fuel of a Nuclear Power Plant (NPP), the ROP is an extraordinary problem of Combinatorial Analysis which during decades has challenged experts of the nuclear area due to its high degree of complexity. In order to perform such a demonstration, the reload of the 7th operating cycle of Angra 1 NPP is taken as a reference in this study, whose main objective was considered to be the maximization of the Boron Concentration at the end of the cycle (EOC). Two methodologies were developed in order to exemplify the performance of the ABC algorithm as an optimizer to the ROP. The first method uses the *Random Keys* model proposed by BEAN (1994) and the second one is a binary discrete version of the original ABC algorithm. Applied in a pioneering way to solve the ROP, the ABC algorithm was able to provide superior results than those obtained by techniques considered state-of-the-art for this problem, thus demonstrating the promising potential of this new metaheuristic to solve problems with high degree of complexity, such as the ROP.

Índice

Capítulo 1 – Introdução	1
1.1. Considerações Iniciais	1
1.2. Revisão Bibliográfica	8
1.3. Objetivo, Relevância e Originalidade da Tese	11
1.4. Estrutura da Tese	13
Capítulo 2 – Conceitos Básicos sobre Otimização	14
2.1. Introdução	14
2.2. Otimização	14
2.3. Problemas de Otimização	16
2.4. Formulação dos Problemas de Otimização	18
2.5. Definições	19
2.6. Otimização Multiobjetivo	22
2.7. Sumário do Capítulo	26
Capítulo 3 – Algoritmos Genéticos e Evolução Diferencial	27
3.1. Introdução	27
3.2. Considerações Iniciais	27
3.3. Algoritmos Genéticos	29
3.4. Algoritmo de Evolução Diferencial	32
3.5. Sumário do Capítulo	37
Capítulo 4 – Apresentação das Metodologias Propostas	38
4.1. Introdução	38
4.2. O comportamento de Abelhas Reais	38
4.3. O Algoritmo ABC Original Básico	40
4.4. O Algoritmo ABCRK	52
4.5. O Algoritmo DBABC	53
4.6. Sumário do Capítulo	56

Capítulo 5 – O Problema de Otimização de Recargas	57
5.1. Introdução.....	57
5.2. O Problema da Otimização de Recargas	57
5.3. As Regras Aprendidas ao Longo dos Anos.....	60
5.4. O Problema de Otimização de Recargas no Contexto da CNA1	61
5.5. O Algoritmo ABCRK Aplicado ao POR	65
5.6. O Algoritmo DBABC Aplicado ao POR	67
5.7. Sumário do Capítulo.....	69
Capítulo 6 – Resultados	70
6.1. Introdução.....	70
6.2. Resultados da Aplicação do ABCRK ao POR	70
6.3. Resultados da Aplicação do DBABC ao POR	73
6.4. Sumário do Capítulo.....	76
Capítulo 7 – Conclusões	78
Referências Bibliográficas	81

Lista de Siglas

EC – Elemento Combustível
POR – Problema de Otimização de Recargas
PC – Padrão de Carregamento
CN – Configuração de Núcleo
RNAP – Reator Nuclear a Água Pressurizada
CNA1 – Central Nuclear de Angra 1
DE – *Differential Evolution*
EOC – *End of Cycle*
DEPP – Dia Efetivo à Plena Potência
GA – *Genetic Algorithm*
MBP – Metaheurística Baseada em Populações
NBTV – Número de bits trocados para vizinhança
NPP – *Nuclear Power Plant*
PSO – *Particle Swarm Optimization*
SA – *Simulated Annealing*
ABC – *Artificial Bee Colony*
ABCRK – *Artificial Bee Colony with Random Keys*
DBABC – *Discrete Binary Artificial Bee Colony*
TS – *Tabu Search*
VQ – Veneno Queimável
LVQ – *Learning Vector Quantization*
PBIL – *Population-Based Incremental Learning*
PCU – Piscina de Combustível Usado
 C_B – Concentração Crítica de Boro
 P_{rm} – Máxima Potência Média Relativa
ppm – Partes por Milhão
SN – *Source Number*

Capítulo 1

– Introdução –

1.1. Considerações Iniciais

Quando um Reator Nuclear a Água Pressurizada (RNAP) precisa ser reabastecido, seu desligamento é requerido e uma fração de seu inventário total de combustível é substituída. Tradicionalmente, esse procedimento ocorre após um período de operação, o qual é chamado ciclo, ou sempre que não for mais possível manter o reator crítico, operando à potência nominal.

Ao final de um ciclo de operação, dá-se o início do procedimento de recarga do novo ciclo. Para continuar a produção de energia, elementos combustíveis (EC) novos são requeridos para substituir os mais queimados. Isto se deve ao fato de a queima dos EC depender de suas posições no núcleo do reator em ciclo (s) anterior (es), o que faz com que os mesmos apresentem características neutrônicas distintas entre si. Neste contexto, considera-se que os EC mais queimados sejam, portanto, aqueles que apresentem as mais baixas concentrações do isótopo físsil, sendo por isso, substituídos por EC novos. Além disso, é comum existirem EC que tenham sido apenas parcialmente queimados, os quais, sendo ainda úteis, poderão ser reaproveitados.

Deste modo, o procedimento de recarga do novo ciclo, deverá incluir, dentre outras decisões, a determinação dos atributos dos EC novos a serem inseridos, bem como a identificação dos EC parcialmente queimados a serem reinseridos no núcleo do reator, compondo assim, o núcleo para o próximo ciclo.

Uma vez decididos quais os EC novos e parcialmente queimados a serem carregados no núcleo, a pergunta imediata, torna-se, então: “Como combinar esses diferentes EC no núcleo do reator de modo que o próximo ciclo de operação seja otimizado?”

O chamado Problema de Otimização de Recargas (POR), que será apresentado em detalhes mais adiante, no Capítulo 5, preocupa-se em obter respostas para esta pergunta. Neste problema, o principal objetivo é determinar onde deverão ser posicionados os EC novos e usados no núcleo do reator, de modo que, levando-se em consideração as regras de simetria a que os EC deverão obedecer no núcleo, e,

satisfazendo-se restrições quanto à máxima queima de descarga do combustível e quanto ao máximo fator de pico de potência radial (F_{XY}) (CHAPOT, 2000), algum objetivo preestabelecido relacionado com o desempenho do reator, seja então, otimizado.

Objetivos tipicamente comuns são: a minimização do fator de pico de potência radial do núcleo (F_{XY}) e a maximização do comprimento do ciclo (o que implica a maximização da Concentração de Boro (C_B) ao final do ciclo).

Se o objetivo do POR for a minimização do fator de pico de potência radial do núcleo (F_{XY}), a tendência é que os EC novos sejam posicionados na periferia do núcleo. Esse esquema de carregamento é conhecido como *out-in*. Ao longo dos anos, foi-se percebendo que os padrões de carregamento (PC) que utilizam o esquema *out-in*, fornecem uma distribuição de potência mais “achatada” no núcleo, porém, à custa de uma alta fuga de nêutrons. Embora uma das metas básicas da utilização do combustível nuclear seja mesmo esse perfil mais “achatado” a fim de maximizar a capacidade de geração de energia do núcleo, a alta fuga de nêutrons é um aspecto negativo, uma vez que o fluxo de nêutrons rápidos provoca danos por radiação na estrutura do vaso do reator, limitando fortemente a vida útil de uma usina nuclear (CHAPOT, 2000).

Se o objetivo do POR for a maximização do comprimento do ciclo operacional, busca-se inserir os EC novos nas posições mais internas do núcleo, enquanto os EC parcialmente queimados são posicionados na periferia. Esse esquema de carregamento é conhecido como baixa-fuga, pois, como o próprio nome sugere, reduz a fuga de nêutrons do núcleo para o vaso do reator. Embora esse seja um aspecto positivo desse tipo de carregamento, o aspecto negativo fica por conta do aparecimento de picos de potência, que devem ser limitados.

Do ponto de vista da otimização, o POR é um extraordinário problema de otimização combinatória (GOLDBARG e LUNA, 2005), envolvendo a consideração de restrições e de múltiplos objetivos (os quais são muitas vezes, conflitantes) relacionados com aspectos econômicos, operacionais e de segurança. Embora possua formulação relativamente simples, o POR é um problema mais do que intratável¹ (TOSCANI e VELOSO, 2001). Ele pertence ao conjunto dos problemas NP-Completo (TOSCANI e VELOSO, 2001 e CORMEN et al., 2002), que consiste nos casos em que a

¹ Um problema é tratável se o seu limite superior de complexidade é polinomial, e é intratável se o limite superior de sua complexidade é exponencial, isto é, se seu tempo de execução é da ordem de uma função exponencial (como 2^n , por exemplo) ou fatorial ($n!$). Em outras palavras, um problema é intratável se o tempo necessário para resolvê-lo é considerado inaceitável para os requerimentos do usuário da solução.

complexidade é não polinomial, mas verificáveis em tempo polinomial. Isto é, dada uma solução, pode-se verificá-la em um tempo razoável.

Em termos práticos, pode-se afirmar que a dificuldade do POR cresce exponencialmente de acordo com o número de EC a serem combinados no núcleo do reator. Por exemplo, para um típico RNAP como o da Central Nuclear de Angra 1 (usina utilizada como referência neste trabalho e doravante referenciada como CNA1), situada em Itaorna, no Município de Angra dos Reis, no Estado do Rio de Janeiro, se fosse considerada a substituição de todos os EC do núcleo do reator, iriam existir, aproximadamente, $8,09 \times 10^{200}$ possíveis PC distintos², já que o reator desta usina possui 121 EC. Basicamente, isso significa que, para um número n de EC, o número total de PC seria $n!$

Para se ter uma idéia do quão inconveniente é o tempo de execução associado a uma função exponencial ou fatorial, basta observar o comportamento assintótico de algumas dessas funções, como mostra a Tabela 1.1.

Tabela 1.1. Comportamento assintótico de algumas funções usuais.

n	n^2	n^3	2^n	$n!$
10	10^2	10^3	$\approx 10^3$	$\approx 10^6$
100	10^4	10^6	$\approx 10^{30}$	$\approx 10^{158}$
1000	10^6	10^9	$\approx 10^{300}$	$\gg 10^{1500}$
10000	10^8	10^{12}	$> 10^{3000}$	$\gg 10^{10000}$

De acordo com a Tabela 1.1, é possível inferir que, levando-se em consideração o caso hipotético de uma máquina moderna ser capaz de realizar algo em torno de 10^9 operações em um segundo, e que fosse necessário realizar 10^{30} operações (o suficiente para resolver um problema cujo tempo seria proporcional a 2^n com $n = 100$ elementos), seria necessário um tempo da ordem de 10^{21} para terminá-las. Considerando-se que um dia tem pouco menos de 10^5 segundos, isso significaria um total de 10^{16} dias ou aproximadamente 10^{13} anos (algumas ordens de grandeza a mais do que o que é correntemente aceito como a idade do universo, estimada em cerca de 10^{10} anos), caracterizando assim, a intratabilidade do problema (LINDEN, 2008). De maneira

² Evidentemente, nem todos esses PC representariam soluções válidas ou possíveis para o problema. Como será visto em momento oportuno, existem restrições operacionais e relacionadas com aspectos de segurança que fazem com que algumas configurações não possam ser consideradas válidas, por não satisfazerem a algumas dessas restrições.

análoga a este raciocínio, depreende-se que se fosse necessário realizar as $\approx 10^{200}$ operações relacionadas com a substituição dos 121 EC que compõem o núcleo do reator da CNA1, isto é, realizar $n!$ operações, com $n = 121$ EC, seria necessário um tempo da ordem de 10^{191} para terminá-las³. Isto significaria um total de 10^{186} dias, ou, aproximadamente, 10^{183} anos. Há de se ter em referência que, mesmo que a velocidade das máquinas modernas fosse elevada ao quadrado, ou até mesmo ao cubo, um problema com este tempo de execução continuaria inaceitavelmente lento.

No entanto, vale ressaltar que, em raras ocasiões substitui-se todo o combustível de um reator nuclear de potência. Geralmente, como foi mencionado no primeiro parágrafo desta seção, apenas uma fração do combustível presente no núcleo é substituída por EC novos ao término de um ciclo de operação. Por exemplo, se cada lote de combustível⁴ permanecer no reator por um mínimo de x ciclos, então, a fração do núcleo substituída em cada recarga será igual a $1/x$. Na prática, considerando-se o núcleo do reator da CNA1 e seus 121 EC, cada lote permanece em média no núcleo durante 3 ciclos. Então, a fração de recarga é de $1/3$. Logo, o número de lotes de combustível no núcleo é igual a 3 e o número de EC substituídos é dado por $121/3$, o que equivale na prática, a aproximadamente, 40 EC. Essa recarga parcial introduz um efeito de acoplamento entre os ciclos, uma vez que cada ciclo contém uma parte do combustível que estava presente no ciclo anterior, assim como uma parte que estará no reator nos ciclos subsequentes.

Dentre a pluralidade de fatores que corroboram a complexidade e a natureza multiobjetivo do POR e das questões a ele relacionadas, vale também ressaltar a influência da seleção da fração de recarga, como um dos muitos aspectos legitimadores da dificuldade associada aos processos decisórios envolvidos no procedimento de recarga.

Para se ter uma idéia, um aumento da fração de recarga implica o aumento do fator de disponibilidade da planta, a redução da queima de descarga por elemento combustível, bem como o aumento do inventário de combustível. Por outro lado, uma diminuição da fração de recarga implica a redução do fator de disponibilidade da planta, o aumento da queima de descarga por elemento combustível, bem como a redução do inventário de combustível. Diante de aspectos conflitantemente positivos e negativos,

³ Caso fosse utilizada a mesma máquina do exemplo anterior.

⁴ Grupo de EC idênticos carregados no reator ao mesmo tempo e (geralmente) removidos do núcleo ao mesmo tempo.

simultaneamente, é necessário que se estabeleça um ponto ótimo, a fim de que se possa atender a requisitos não apenas econômicos e operacionais, mas, principalmente, aos requisitos relacionados com a segurança.

Basicamente, os requisitos econômicos estão associados aos custos envolvidos na geração de energia por parte da central nuclear, sendo a minimização destes custos, a principal meta a ser atingida pelo POR. Embora a quantificação de tais custos não seja uma tarefa simples - haja vista a enorme complexidade envolvida na operação de uma usina nuclear - diversos estudos, tais como: (GALPERIN, 1989), (KROPACKZEK e TURINSKY, 1990), (POON e PARKS, 1992), (DECHAIINE e FELTUS, 1995), (VAN DER SCHAAR et al., 1996), (LIN et al., 1998), (CHAPOT, 2000), (MACHADO, 2001), (DE LIMA, 2005), (MACHADO, 2005), (CALDAS e SCHIRRU, 2008), (WAINTRAUB, 2009) e (MENESES, 2010), têm demonstrado que, com a otimização operacional, especialmente no uso do combustível nuclear, pode-se chegar a consideráveis reduções de custos.

Em parte, isto se deve ao fato que, ao contrário dos combustíveis fósseis utilizados nas usinas termelétricas convencionais, o combustível nuclear requer lenta e sofisticada preparação, a qual inclui diversas etapas, tais como: enriquecimento, conversão em material cerâmico, compactação em forma de pastilhas, sinterização, acabamento, além de criterioso transporte (CHAPOT, 2000). Por esta razão, os EC são de estruturas elaboradas e caras, havendo, portanto, motivos econômicos para queimar o combustível até o máximo possível.

Além disso, os RNAP são caracterizados pela necessidade de desligamento para o recarregamento. Logo, a parada para a substituição dos EC significa a indisponibilidade da planta e, por isso, são desejados os ciclos mais longos possíveis.

Com base nestas pontuações, há de se ter em referência dois importantes aspectos relacionados com a operação de um reator nuclear de potência, quais sejam: a taxa máxima de queima do combustível e a duração do ciclo operacional.

A taxa máxima de queima do combustível é uma importante restrição tanto operacional, quanto de segurança, a que o POR deve observar. Isso porque os combustíveis nucleares não são queimados totalmente, assim como o carvão, o óleo, ou o gás, por exemplo. Na prática, apenas uma pequena parte do combustível (isto é, átomos isolados dentro da matriz de urânio) sofre fissões, e a maioria dos átomos permanece no núcleo do reator, sem sofrer alterações.

Basicamente, a taxa de queima é indicada por duas maneiras: a fração dos átomos que sofreram a fissão, ou a energia produzida por uma dada quantidade de combustível inicial. Teoricamente, é possível atingir altas taxas de queima colocando-se combustível em excesso no núcleo e controlando-se a reatividade⁵ por meio de absorvedores adicionais, os chamados “venenos”. Todavia, na prática, existem limites envolvendo aspectos econômicos e de segurança.

Os aspectos econômicos já foram mencionados anteriormente, e no que diz respeito aos interesses do presente estudo, irão referir-se, exclusivamente, ao custo do combustível propriamente dito.

Já os aspectos de segurança, referem-se à integridade das varetas, uma vez que todos os materiais são modificados por irradiação intensa por nêutrons, sendo os efeitos resultantes, geralmente, bastante negativos (ISHIGURO, 2002).

Uma descrição detalhada do modo como são constituídas as 28.435 varetas combustíveis presentes nos 121 EC que compõem o núcleo da CNA1, pode ser encontrada em (CHAPOT, 2000). Neste caso, a matriz de combustível, inicialmente cerâmica, poderia ser danificada, ou ter suas propriedades (tais como condutividade térmica e resistência mecânica) modificadas caso fosse submetida à irradiação intensa por nêutrons além de certos limites. Agregue-se a isto o fato que, nestas condições, os produtos de fissão poderiam aumentar o volume da matriz e danificar as ligas utilizadas como revestimento. Por estas e outras razões, o tempo de queima permitido para o combustível é uma importante restrição a ser observada, não devendo, portanto, ser negligenciada.

Outra importante restrição relacionada com aspectos operacionais e de segurança, diz respeito ao máximo fator de pico de potência radial (F_{XY}) do núcleo. Este pode ser descrito como a razão entre a densidade de potência linear máxima e a densidade de potência linear média no plano horizontal do núcleo do reator onde ocorre o pico local de potência (MACHADO, 1998).

Conforme será visto mais adiante no Capítulo 5, de acordo com as Especificações Técnicas para a CNA1 contidas em (FSAR, 1980), o valor máximo de F_{XY} deverá sempre ser menor ou igual a 1,435. Isto significa que PC que apresentem valores de F_{XY} maiores que este, não serão considerados válidos, pelo fato de

⁵ A reatividade é um indicador da capacidade do sistema de manter as reações em cadeia e também de mudanças da taxa de fissão, sendo que uma reatividade positiva (negativa) significa aumento (diminuição) da taxa de fissão ou da potência do reator. A reatividade aumenta com adição de combustível e diminui com a introdução de absorvedores.

representarem uma violação a esta importante restrição de segurança imposta pelas Especificações Técnicas.

Outra restrição que deve ser observada está relacionada com as regras de simetria a que os EC devem obedecer no núcleo. Isso ocorre porque os RNAP são caracterizados por apresentarem regiões simétricas em seus núcleos, delimitadas por eixos de simetria que dividem o núcleo em quatro, ou em oito “fatias”. Desta forma, para que haja uma distribuição de potência simétrica no núcleo do reator, utilizam-se modelagens que representam 1/4 ou 1/8 do formato original do núcleo. Ainda no que diz respeito à complexidade do POR, a existência desses eixos de simetria no núcleo do reator da CNA1 é de extrema importância, uma vez que a consideração dessas simetrias reduz de forma significativa o número de possíveis combinações para o problema. Desta forma, é possível modelar o POR considerando-se apenas um dos quadrantes do núcleo (simetria de 1/4) ou um dos octantes do núcleo (simetria de 1/8) e depois, utilizando-se as regras de rebatimento do núcleo, obter-se a configuração completa do mesmo.

Neste caso, como será visto adiante no Capítulo 5, o espaço de busca é reduzido, uma vez que o número de possíveis soluções passa de $121! \approx 8,09 \times 10^{200}$ (caso fosse considerado o núcleo inteiro), para $36! \approx 3,7 \times 10^{41}$ (caso fosse considerada a simetria de 1/4 de núcleo) ou para $20! \approx 2,4 \times 10^{18}$ (caso fosse considerada a simetria de 1/8 de núcleo). Em outras palavras, a consideração dessas simetrias simplifica o problema, pois, quanto menor for o número de EC participantes do processo de otimização, menor será o espaço de busca das soluções do problema, ou seja, menor será a complexidade.

Pode-se perceber que mesmo com as considerações de simetria, o número de possíveis soluções é ainda extremamente alto, inviabilizando a avaliação de todos os possíveis PC. Nesse momento, o papel da otimização é de fundamental importância, uma vez que, do ponto de vista prático, a otimização tem como objetivo principal, a determinação da melhor configuração de projeto possível para um dado sistema, sem ter que testar todas as possibilidades.

A solução do POR também é extremamente dificultada por sua natureza não linear, pela existência de regiões desconexas de possíveis soluções, pelo alto custo computacional envolvido na avaliação de um possível PC, bem como pelas restrições anteriormente mencionadas. Todas essas características interagem entre si, tornando o POR um verdadeiro desafio para muitas técnicas de otimização.

Durante décadas, o POR tem sido o cerne de várias pesquisas na área nuclear e, atualmente, a publicação de um grande número de trabalhos dedicados a este problema,

corroborar a importância e a atualidade do tema, provando que a busca pelo aperfeiçoamento e pelo desenvolvimento de técnicas de otimização cada vez mais eficientes, ainda estão em urgente demanda.

O presente estudo busca fornecer mais uma contribuição nesta área, através do desenvolvimento de metodologias para a solução eficiente do POR. Tomando-se como referência a recarga do ciclo 7 de operação da CNA1, de propriedade da Eletrobrás Termonuclear S. A. – ELETRONUCLEAR, o algoritmo de otimização por Colônia de Abelhas Artificiais, do inglês, *Artificial Bee Colony (ABC)* (KARABOGA, 2005), é utilizado como otimizador para o POR, com o objetivo de maximizar a Concentração de Boro (C_B) ao final do ciclo. Este objetivo também pode ser entendido como a maximização do comprimento do ciclo operacional, uma vez que a C_B possui uma relação direta com o número de Dias Efetivos à Plena Potência (DEPP) do ciclo de uma usina nuclear. Em outras palavras, isto significa que configurações de núcleo (CN)⁶ que maximizem a C_B , permitem ciclos maiores, ou seja, permitem que a usina opere uma quantidade maior de dias com tais configurações. Além das benesses econômicas advindas dessas configurações (vale ressaltar que 1 DEPP equivale a um retorno financeiro da ordem de centenas de milhares de dólares) (MENESES, 2010), existe também a vantagem relacionada com o aumento da disponibilidade da planta, considerando-se que o intervalo entre as recargas aumenta.

O algoritmo ABC, cujos princípios biológicos que motivaram seu desenvolvimento, bem como seu respectivo algoritmo computacional, serão apresentados em detalhes no Capítulo 4, tem se mostrado bastante eficaz na solução de problemas de alta dimensionalidade (SAMANTA e CHAKRABORTY, 2011), (HORNG, 2011), (ZHANG et al., 2011), (IRANI e NASIMI, 2011), (GAO et al., 2011), (RASHIDI et al., 2011) e (KARABOGA e OZTURK, 2011), sem a necessidade de conhecimento prévio sobre a complexidade dos espaços de busca envolvidos.

1.2. Revisão Bibliográfica

A otimização manual, foi, durante várias décadas, a única forma pela qual a indústria nuclear solucionava o complexo POR. Ainda hoje, a experiência operacional

⁶ Ou padrões de carregamento (PC).

aliada à intuição de especialistas é utilizada para o projeto de CN que, uma vez projetados, são testados para a verificação de atendimento às necessidades operacionais da planta, bem como ao atendimento das restrições de segurança da mesma.

Tal prática, no entanto, além de dispendiosa (por adotar o método da tentativa e erro) tem dado lugar a técnicas computacionais cada vez mais sofisticadas, graças aos avanços da informática ao longo de vários anos.

Pesquisadores em todo o mundo passaram a buscar soluções através de técnicas de otimização, que foram gradativamente sendo introduzidas, possibilitando o desenvolvimento de sistemas cada vez mais eficientes, capazes de operar com um número bem menor de insucessos.

Métodos baseados em conhecimento (GALPERIN et al., 1989), por exemplo, mostraram-se bastante eficientes em uma época em que se começava a falar sobre conhecimento heurístico aplicado à solução de problemas complexos.

(KROPAZECK e TURINSKY, 1990) desenvolveram o FORMOSA (*Fuel Optimization for Reloads: Multiple Objectives by Simulated Annealing*) utilizando a técnica de Recozimento Simulado (*Simulated Annealing, SA*) (KIRKPATRICK et al., 1983). Devido ao bom desempenho apresentado pelo FORMOSA para a solução do POR, o mesmo passou a ser comercializado por seus autores.

(POON e PARKS, 1992) fizeram modificações no sistema FORMOSA e criaram o FORMOGA por meio da substituição da técnica de (*Simulated Annealing, SA*) pela técnica dos Algoritmos Genéticos (*Genetic Algorithms, GA*) (HOLLAND, 1975) e (GOLDBERG, 1989). Com este trabalho, os autores lançaram a expectativa da superioridade dos GAs em relação ao SA, ainda que não comprovada, devido à necessidade de maior aprofundamento nas pesquisas.

Ainda na linha dos GAs, outro sistema de grande destaque foi o código CIGARO (*Code Independent Genetic Algorithm Reactor Optimization System*), proposto por (DECHAINE e FELTUS, 1995).

(VAN DER SCHAAR, et al., 1997) propuseram o ROSA (*Reloading Optimization by Simulated Annealing*), para a otimização simultânea da composição de EC novos e de PC. Isto porque o ROSA combina em um mesmo “pacote”, um simulador de núcleo de alta velocidade, um gerador de PC e um algoritmo de otimização. Assim como o FORMOSA, o ROSA também utiliza a técnica de SA, e devido ao bom desempenho apresentado, também passou a ser comercializado por seus desenvolvedores.

Outra abordagem de grande destaque foi proposta por (LIN et al., 1998). Neste trabalho, os autores utilizaram a técnica de Busca Tabu (*Tabu Search, TS*) (GLOVER e LAGUNA, 1997) para a solução do POR.

Em sua dissertação de mestrado, (MACHADO, 1999) propôs um novo algoritmo evolucionário com aprendizado LVQ (*Learning Vector Quantization*) para a solução de problemas combinatórios e o aplicou ao POR da CNA1.

(CHAPOT, 2000), em sua tese de doutorado, apresentou um sistema automático para a solução do POR. Com base na teoria dos GAs, o sistema proposto neste trabalho acopla os códigos GENESIS 5.0 e ANC por intermédio da interface ALGER. O relevante Modelo de Listas, que evita o uso de variantes do operador recombinação dos GAs e gera apenas CN válidas para o POR, também foi apresentado. Ainda neste trabalho, onde os ciclos 7 e 9 de operação da CNA1 foram otimizados, também foi desenvolvido o código nodal bidimensional RECNOD, o qual é utilizado tanto no presente trabalho, como em muitos outros que, de igual modo, utilizam a CNA1 como referência (ciclo 7).

Em (MACHADO e SCHIRRU, 2002), uma nova abordagem para o POR utilizando o algoritmo Ant-Q, baseado na técnica de Otimização por Colônia de Formigas (*Ant Colony Optimization, ACO*) (DORIGO e GAMBARDELLA, 1997) foi apresentado. Os autores também aplicaram o algoritmo na otimização da recarga do ciclo 7 de operação da CNA1 e obtiveram muito bons resultados.

Em sua tese de doutorado, (DE LIMA, 2005) desenvolveu um sistema computacional paralelo chamado de Redes Conectivas de Colônias Artificiais (RCCA), o qual também é baseado no ACO de (DORIGO e GAMBARDELLA, 1997). Assim como o algoritmo Ant-Q de (MACHADO e SCHIRRU, 2002) o RCCA foi de igual modo aplicado na otimização da recarga do ciclo 7 de operação da CNA1, obtendo resultados bastante satisfatórios.

Em (CALDAS e SCHIRRU, 2008), um algoritmo evolucionário sem parâmetros chamado FPBIL, baseado no PBIL (*Population-Based Incremental Learning*) (BALUJA, 1994) foi proposto para a solução do POR. Também aplicado na otimização da recarga do ciclo 7 de operação da CNA1, o FPBIL, assim como os métodos anteriores, forneceu resultados bastante significativos.

(WAINTRAUB, 2009) utilizou o algoritmo de Otimização por Enxame de Partículas (*Particle Swarm Optimization, PSO*) (KENNEDY e EBERHART, 1995) para a otimização do POR e também para a otimização do projeto de reatores. Como ambos

os problemas são conhecidamente complexos e grandes consumidores de tempo de processamento, a computação paralela foi utilizada com o objetivo de reduzir o custo computacional.

Em (MENESES, 2010) nova aplicação do PSO como otimizador para o POR é reportada. Neste trabalho, também são apresentadas: a Heurística de Aceitação de Vizinhaça Reativa para uso com metaheurísticas, e também o desenvolvimento da metaheurística Busca Baseada em Classes, duas novas abordagens para a solução do POR.

Em (OLIVEIRA e SCHIRRU, 2011), o algoritmo de otimização por Colônia de Abelhas Artificiais (*Artificial Bee Colony, ABC*) (KARABOGA, 2005) é aplicado pela primeira vez como otimizador para o POR. Tal publicação é fruto de parte das pesquisas que resultaram neste trabalho de tese.

1.3. Objetivo, Relevância e Originalidade da Tese

O algoritmo de otimização por Colônia de Abelhas Artificiais, do inglês, *Artificial Bee Colony (ABC)* (KARABOGA, 2005), é um dos mais recentes membros da Inteligência de Enxames. Trata-se de uma Metaheurística Baseada em Populações (MBP) de indivíduos, na qual os candidatos à solução evoluem por meio da simulação de um modelo simplificado do comportamento forrageiro observado em uma colônia de abelhas melíferas reais.

Originalmente proposto para a otimização de funções numéricas multimodais e multidimensionais em espaços de busca contínuos, o ABC destaca-se por sua simplicidade, fácil implementação, eficiência, robustez na descoberta de soluções, e também pelo fato de empregar um número menor de parâmetros de controle que outras metaheurísticas baseadas em populações.

Desde sua criação em 2005, o ABC vem demonstrando muito bom desempenho na otimização de problemas em espaços de busca contínuos, porém, não se aplica a problemas em espaços de busca discretos.

Transpondo esta limitação, o presente estudo tem como objetivo demonstrar a viabilidade de utilização do algoritmo ABC como uma alternativa eficiente para a solução do Problema de Otimização de Recargas (POR) de um Reator Nuclear a Água Pressurizada (RNAP).

Para cumprir com este objetivo, foram realizadas algumas modificações no algoritmo ABC original, a fim de torná-lo adequado à solução de problemas combinatórios, como é o caso do POR.

Em duas aplicações distintas para a otimização da recarga do ciclo 7 de operação da CNA1 objetivando a maximização da Concentração de Boro (C_B) ao final do ciclo, o algoritmo ABC foi capaz de encontrar resultados superiores aos obtidos por outras metaheurísticas baseadas em populações já bem conhecidas, demonstrando assim, o seu potencial como otimizador tanto para o POR (OLIVEIRA e SCHIRRU, 2011), como também para outros problemas de similar complexidade, como pode ser visto em (OLIVEIRA et al., 2009).

Para a primeira aplicação do ABC ao POR, foi desenvolvido um modelo no qual se utilizou a versão original do algoritmo, combinada com a metodologia de Chaves Aleatórias (*Random Keys, RK*) (BEAN, 1994). Deste modo, foi possível mapear o espaço de busca em um espaço de soluções combinatórias que são, de fato, possíveis para a solução do POR.

Para possibilitar uma segunda aplicação do ABC ao POR, também foi desenvolvida uma versão binária discreta do algoritmo ABC original (OLIVEIRA e SCHIRRU, 2010). Esta versão binária discreta estende as capacidades da versão original do ABC e ainda apresenta a vantagem de ser aplicável à otimização de qualquer problema, seja em espaços de busca contínuos, ou mesmo em espaços de busca discretos.

A relevância do presente estudo reside no fato do mesmo apresentar novas metodologias para a solução eficiente do POR, o qual é um problema de grande relevância para a área nuclear e que há décadas vem desafiando especialistas devido ao seu alto grau de complexidade. Como será visto mais adiante, no Capítulo 6, os resultados obtidos corroboram não apenas a viabilidade de uso das técnicas desenvolvidas, mas também sua eficiência com relação aos resultados obtidos por outros métodos de otimização.

Esta tese também possui conteúdo original, uma vez que apresenta os resultados da primeira aplicação do ABC ao POR, bem como a primeira utilização do ABC em conjunto com a metodologia de Chaves Aleatórias (*Random Keys, RK*) (BEAN, 1994) para aplicação em problemas combinatórios. Adicionalmente, apresenta uma nova abordagem binária discreta, a qual se mostrou capaz de sobrepujar a versão original do

algoritmo ABC, além de produzir resultados superiores também a outras metaheurísticas já bem conhecidas.

1.4. Estrutura da Tese

O Capítulo 2 é dedicado à definição dos principais conceitos básicos sobre otimização que serão utilizados no decorrer do presente estudo.

No Capítulo 3, faz-se uma breve revisão da fundamentação teórica dos Algoritmos Genéticos (*Genetic Algorithms, GA*) (HOLLAND, 1975) e (GOLDBERG, 1989) e do Algoritmo de Evolução Diferencial (*Differential Evolution, DE*) (STORN e PRICE, 1995). O objetivo deste capítulo é apresentar, de forma sucinta, os conhecimentos teóricos necessários à compreensão das metodologias propostas no capítulo subsequente.

O Capítulo 4 destina-se à apresentação do algoritmo ABC. São descritos os princípios biológicos que motivaram o seu desenvolvimento, bem como seu respectivo algoritmo computacional. Em seguida, são apresentadas as duas versões modificadas do mesmo, as quais são propostas neste estudo como alternativas para a solução do POR da CNA1. São elas: o algoritmo ABCRK (*Artificial Bee Colony with Random Keys*) (OLIVEIRA e SCHIRRU, 2011) e o algoritmo DBABC (*Discrete Binary Artificial Bee Colony*) (OLIVEIRA e SCHIRRU (2010).

O Capítulo 5 descreve o POR no contexto da CNA1, cuja recarga do ciclo 7 de operação é utilizada como referência neste estudo. As particularidades deste ciclo, bem como sua modelagem utilizando-se as duas versões do algoritmo ABC propostas no Capítulo 4, são também apresentadas.

O Capítulo 6 é dedicado à exposição dos resultados obtidos a partir da modelagem apresentada no Capítulo 5.

No Capítulo 7 encontram-se as conclusões do trabalho, bem como as recomendações para trabalhos futuros.

Capítulo 2

– Conceitos Básicos sobre Otimização –

2.1. Introdução

Este capítulo é dedicado à introdução dos conceitos básicos sobre otimização que serão utilizados no decorrer do presente estudo. A literatura sobre esse assunto é bastante abrangente e, por isso, buscou-se definir apenas os principais conceitos de interesse. Caso haja a necessidade de maior aprofundamento no estudo das seções seguintes, existe uma vasta bibliografia sobre o assunto apresentado. Dentre as muitas referências disponíveis, destacam-se: (FOX, 1971), (REKLAITIS et al., 1983), (DEB, 1995), (VANDERPLAATS, 1999), (BOYD e VANDENBERGUE, 2004) e (IZMAILOV e SOLODOV, 2009).

2.2. Otimização

Do ponto de vista prático, a otimização trata do conjunto de métodos capazes de determinar as melhores configurações possíveis para a construção ou para o funcionamento de sistemas de interesse para o ser humano. Neste contexto, fala-se da aplicação de uma mesma teoria, com um mesmo conjunto de métodos e ferramentas, quando, por exemplo:

- Um engenheiro mecânico busca a melhor configuração possível para o projeto de um motor, de modo a minimizar seus custos de fabricação;
- Um analista do mercado financeiro busca o melhor *portfolio* de investimentos, a fim de maximizar a expectativa de lucros de sua empresa;
- Um motorista de caminhão de entregas busca a melhor rota para a distribuição de mercadorias por todos os seus clientes da forma mais eficiente possível, de modo a minimizar tempo e custo associados com a tarefa de entrega;

- Um engenheiro nuclear busca o melhor padrão de carregamento de elementos combustíveis para o núcleo de um reator, de modo a maximizar o aproveitamento desse combustível ao longo dos ciclos em que ele é utilizado; dentre muitos outros exemplos.

Apesar dos contextos completamente distintos, todos estes problemas, assim como muitos outros, uma vez formulados matematicamente, possuem basicamente a mesma estrutura e suas soluções são obtidas, essencialmente, através da utilização de um mesmo conjunto de técnicas: a otimização.

Entende-se por otimização, a busca pela melhor solução para um determinado problema dentro de um conjunto finito ou infinito de possíveis soluções para este problema. O modo como essa busca é realizada é o que caracteriza os diferentes métodos de otimização existentes.

A escolha do método depende de uma série de características⁷ do problema a ser otimizado, principalmente do comportamento da função que o representa.

Por exemplo: a função ser diferenciável implica na possibilidade de se tentar sua otimização fazendo-se uso do cálculo, pelo menos aproximado, de seu gradiente.

Se a função for unimodal, ou seja, se tiver um único ponto de mínimo, as estratégias para a determinação desse mínimo serão bem diferentes daquelas que seriam empregadas caso a função fosse multimodal, ou seja, caso tivesse muitos mínimos locais⁸.

Na prática, um método pode ser adequado para a solução de um problema e não ser adequado para a solução de outros. Isto requer do usuário, o conhecimento de uma grande variedade de métodos de otimização, a fim de que se possa escolher, de forma eficiente, a ferramenta mais indicada às características do problema que se deseja solucionar.

Os problemas de otimização apresentam alguns conceitos e definições largamente empregados na literatura e de conhecimento imprescindível. Com o intuito de tornar mais fácil o entendimento dos próximos capítulos, são descritos os principais conceitos de interesse para o presente estudo, os quais são apresentados a seguir, na forma de exemplos básicos do problema geral de otimização.

⁷ Por exemplo: modalidade (unimodal, multimodal), diferenciabilidade (diferenciável, não diferenciável), convexidade (convexo, não convexo) e linearidade (linear, não linear).

⁸ A expressão: “mínimos locais” serve para designar pontos que são de mínimo para uma vizinhança ao seu redor, enquanto a expressão: “mínimos globais” serve para designar o ponto em que a função objetivo atinge seu mínimo valor em todo o domínio considerado.

Adicionalmente, porém, sem a intenção de se fazer um estudo detalhado sobre o assunto, um breve tópico sobre os conceitos básicos da otimização multiobjetivo (ou multicritério), encerra o presente capítulo.

2.3. Problemas de Otimização

De acordo com (IZMAILOV e SOLODOV, 2009), dados um conjunto $\Omega \subset \mathfrak{R}^n$ e uma função $f : \Omega \rightarrow \mathfrak{R}$, um problema de otimização visa determinar, por exemplo, um minimizador de f no conjunto Ω .

Tal problema pode ser escrito da seguinte forma:

$$\min f(X) \text{ sujeito a } X \in \Omega \tag{2.1}$$

O conjunto Ω é chamado de conjunto viável do problema e é delimitado por um conjunto de restrições, igualdades e desigualdades a que os elementos de Ω devem satisfazer.

Os elementos de Ω são chamados de soluções viáveis, e a função $f(X)$ é chamada de função objetivo (ou função custo), sendo esta a responsável por determinar o quão bem um determinado elemento de Ω soluciona o problema formulado.

A solução que maximiza ou minimiza (de acordo com o problema) a função objetivo, é chamada de solução ótima.

No exemplo representado pela Equação (2.1), a solução ótima é aquela que minimiza a função objetivo $f(X)$, pelo fato de se tratar de um problema de minimização.

Um ponto $X^* \in \Omega$ é um minimizador global de (2.1) se $f(X^*) \leq f(X)$, para todo $X \in \Omega$ e é minimizador local do problema se existe uma vizinhança U de X^* tal

que $f(X^*) \leq f(X)$, para todo $X \in \Omega \cap U$, como mostra a Figura 2.1, extraída de (IZMAILOV e SOLODOV, 2009), com modificações.

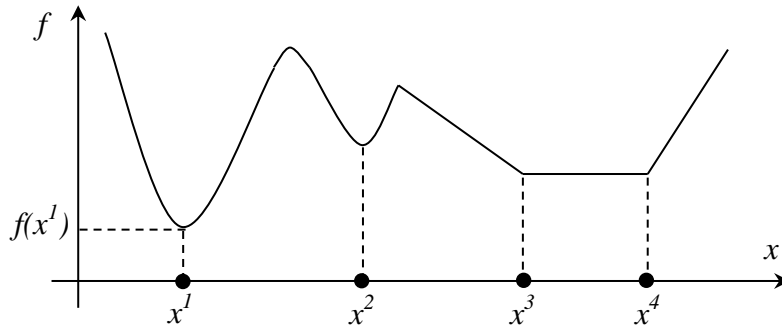


Figura 2.1. x^1 é o minimizador global, $f(x^1)$ é o valor ótimo, x^2 é um minimizador local estrito e $[x^3, x^4]$ é um conjunto de minimizadores locais não estritos.

Do ponto de vista matemático, não há nenhuma diferença relevante entre os problemas de minimização e de maximização: todos os resultados obtidos para uma classe de problemas podem ser estendidos para a outra classe, sem dificuldade.

Por exemplo, caso o problema fosse maximizar a função objetivo, bastaria transformar o problema através da multiplicação da função por -1 , ou seja, minimiza-se a função $-f(X)$, pois as soluções locais e globais de ambos os problemas são as mesmas, apenas com sinais opostos para os valores ótimos, como mostra a Figura 2.2, extraída de (IZMAILOV e SOLODOV, 2009), com modificações.

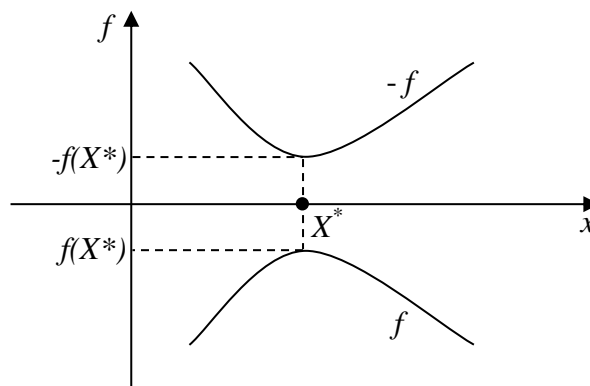


Figura 2.2. $\max f(X) = \min (-f(X))$.

2.4. Formulação dos Problemas de Otimização

De modo geral, um problema de otimização consiste na maximização ou na minimização de uma função objetivo de uma ou mais variáveis (também chamadas de variáveis de otimização, de decisão ou de projeto), sujeita, ou não, a restrições de comportamento (igualdade e desigualdade) e restrições laterais (efetuadas diretamente sobre as variáveis de otimização, limitando seus valores). Estas restrições determinam o espaço de busca viável (ou região viável de busca, ou ainda, espaço de soluções) onde se espera encontrar a melhor solução, também chamada de solução ótima.

A função objetivo (também referenciada como função de adaptação, de adequação, de aptidão, de avaliação, custo ou *fitness*) é geralmente expressa por uma função matemática que representa as características do problema a ser resolvido. Tanto a função objetivo quanto as funções de restrições do problema, podem ser funções lineares ou não lineares em relação às variáveis de otimização, podendo ser calculadas por meio de métodos analíticos ou numéricos.

As Equações (2.2) e (2.3) apresentam a formulação característica para um problema de otimização cujo objetivo é encontrar um vetor de n variáveis de decisão $X = (x_1, x_2, \dots, x_n)^T$ (onde T representa a transposição do vetor), que minimize uma função objetivo $f(X)$ e que satisfaça as restrições de comportamento e laterais.

$$\text{Minimizar } \begin{cases} \text{função objetivo :} \\ f(X), \quad X \in \mathfrak{R}^n \end{cases} \quad (2.2)$$

$$\text{Sujeito a } \begin{cases} \text{Restrições de comportamento :} \\ \begin{cases} g_j(X) \leq 0, & j = 1, 2, \dots, J \\ h_k(X) = 0, & k = 1, 2, \dots, K \end{cases} \\ \text{Restrições laterais :} \\ \begin{cases} x_i^{inf} \leq x_i \leq x_i^{sup}, & i = 1, 2, \dots, n \end{cases} \end{cases} \quad (2.3)$$

A partir desta formulação, são consideradas as seguintes definições:

2.5. Definições

✓ Variáveis de Otimização

As variáveis de otimização, de decisão ou de projeto, são aquelas que se modificam durante o processo de otimização, alterando o valor da função objetivo $f(X)$. São representadas por x_i , para $i = 1, 2, \dots, n$, formando o vetor de n variáveis dado por: $X = (x_1, x_2, \dots, x_n)^T$, (vide Equação (2.2)). De modo geral, essas variáveis podem ser contínuas (reais), inteiras ou discretas (valores compreendidos dentro de certo conjunto fixo). Uma das diferenças mais importantes entre os problemas de otimização, que conduz a técnicas de resolução com fundamentações bastante distintas, é aquela que separa os problemas em que as variáveis de otimização são reais dos problemas que apresentam variáveis de otimização discretas (binárias ou inteiras).

✓ Restrições

As restrições são funções que delimitam o espaço de busca. Dividem-se em dois grupos: restrições de comportamento e restrições laterais. As restrições de comportamento são expressas através de igualdades e/ou desigualdades matemáticas que estabelecem exigências para a solução ótima, representando o conjunto dos requisitos que o resultado do problema deve atender para ser admissível enquanto solução. Já as restrições laterais, são funções que servem para limitar os valores das variáveis de otimização, sendo definidas como: limite inferior: $X^{inf} = (x_1^{inf}, x_2^{inf}, \dots, x_n^{inf})^T$ e limite superior: $X^{sup} = (x_1^{sup}, x_2^{sup}, \dots, x_n^{sup})^T$.

A seguir, são definidas algumas nomenclaturas relacionadas com as restrições:

○ Região Factível

Conjunto dos pontos do espaço de busca que satisfazem, simultaneamente, a todas as restrições. A região factível é também chamada de conjunto factível ou conjunto viável.

- **Região Inactivável**

Conjunto dos pontos do espaço de busca que deixam de satisfazer (ou seja, violam) pelo menos uma das restrições do problema.

- **Ponto Factível e Ponto Inactivável**

Um ponto factível é um ponto pertencente à região factível e um ponto inactivável é um ponto pertencente à região inactivável.

- **Restrição Violada**

Cada uma das componentes do vetor $g_j(X)$ que apresentar valor positivo, ou cada uma das componentes do vetor $h_k(X)$ que apresentar valor não nulo, será chamada de restrição violada (vide Equação (2.3)).

- **Restrição Activa**

Restrições activas são as restrições de desigualdade que são satisfeitas por X^* (onde X^* é o ponto ótimo) na igualdade, ou seja, X^* pertence ao conjunto viável tal que $g_j(X^*) = 0$, onde $j = 1, 2, \dots, J$, caso contrário, a restrição é dita inactiva. A Figura 2.3, adaptada de (IZMAILOV e SOLODOV, 2009), mostra um exemplo onde, para $j = 1$, e para $j = 2$, as restrições são activas, e para $j = 3$, a restrição é inactiva.

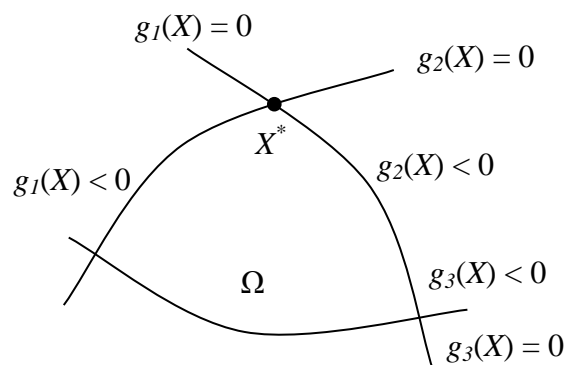


Figura 2.3. A restrição g^3 é inactiva no ponto ótimo, pois ela não define a estrutura do conjunto viável Ω em torno do ponto ótimo X^* .

✓ Espaço de Busca ou Região Viável

É o conjunto, espaço ou região que compreende as soluções possíveis ou viáveis do problema a ser otimizado. É determinado por um conjunto de restrições a que as soluções viáveis e a função objetivo devem satisfazer.

✓ Função Objetivo ou Função de Avaliação

É a função que se quer otimizar, e portanto, é aquela que atinge um valor extremo. Representada por $f(X)$ na Equação (2.2), é a função que permite avaliar o grau de adequação de cada elemento do espaço de busca. É também, muitas vezes referenciada como função de aptidão, de adequação, de adaptação, de avaliação, custo ou *fitness*. Pode ser unidimensional, ou seja, possuir apenas uma variável de otimização, ou multidimensional, isto é, possuir mais de uma variável de otimização. A função objetivo e as funções de restrições também podem ser funções lineares ou não lineares em relação às variáveis de otimização. Tais funções também podem ser implícitas ou explícitas, e calculadas por técnicas analíticas ou numéricas.

✓ Ponto Ótimo

É o ponto caracterizado pelo vetor $X^* = (x_1, x_2, \dots, x_n)^T$, formado pelas variáveis de otimização que otimizam a função objetivo e satisfazem as restrições.

✓ Valor Ótimo

É o valor da função objetivo $f(X^*)$ no ponto ótimo.

✓ Solução Ótima

É o par formado pelo ponto ótimo e pelo valor ótimo $[X^*, f(X^*)]$, podendo ser, local: quando o valor ótimo é localizado; global: quando o valor ótimo é global na região viável; restringida: quando há restrições satisfeitas como igualdade (ativas); e não restringidas: quando não há restrições ativas no ponto ótimo.

2.6. Otimização Multiobjetivo

Muitos problemas do mundo real apresentam mais de um objetivo a ser otimizado, ou seja, em vez de uma única função objetivo, várias delas são consideradas, precisando ser minimizadas ou maximizadas simultaneamente. Na maioria das vezes, os objetivos em questão são conflitantes entre si, o que significa que, ao se buscar a melhoria de um ou mais objetivos, existe conseqüentemente, a deterioração de um ou mais dos outros objetivos. Esses problemas são chamados de problemas de otimização multiobjetivo ou multicritério.

Um exemplo de problema com objetivos conflitantes seria a tarefa de comprar um automóvel. Dentre os possíveis objetivos, o que poderia ser considerado como a aquisição ótima, seria aquela que fornece o custo mínimo enquanto maximiza o desempenho e o conforto do veículo. Esses objetivos são conflitantes entre si, uma vez que à medida que se reduz o custo do veículo, também se diminui a quantidade de opcionais relacionados ao conforto e ao desempenho do mesmo. Um automóvel com maior conforto e maior desempenho pelo menor custo, embora ideal, não existe no mundo real. Portanto, não existe uma solução ótima única, e sim, um conjunto de soluções.

Um conjunto de soluções será considerado como o conjunto de soluções ótimas, quando não existirem outras soluções no espaço de busca que sejam melhores do que as soluções deste conjunto, quando todos os objetivos forem considerados simultaneamente. Tal conjunto de soluções é conhecido como o conjunto das soluções ótimas de Pareto (MACHADO, 2005).

Assim, nenhuma solução que tenha menor custo, menor conforto e menor desempenho pode ser considerada como superior a outra com maior custo, maior conforto e maior desempenho. Contudo, dentre todas as opções de automóveis, existem algumas que são superiores a outras, isto é, apresentam desempenho e conforto maiores ou equivalentes por um custo menor ou igual. Estas opções (soluções) que superam outras são as chamadas soluções não dominadas, enquanto que as opções que são superadas por pelo menos uma das outras opções, são as soluções dominadas.

Deste modo, é muito interessante um método que encontre o conjunto das soluções não dominadas, para que o indivíduo que realiza o processo de otimização possa escolher, dentre estas, aquela que melhor atenda as suas necessidades. Esta é a tarefa da otimização multiobjetivo.

No problema anteriormente descrito, como em muitos outros casos, não existe uma solução única para o problema, e dificilmente os diferentes objetivos serão otimizados levando-se em consideração uma única escolha. Assim, algum tipo de tomada de decisão torna-se necessária para que se possa atingir a solução ótima. Dentro deste contexto, existem dois modos para o tratamento do problema:

- No primeiro modo, definem-se prioridades e/ou pesos entre os vários objetivos de interesse. Tendo-se feito isto, encontra-se a solução ótima segundo tais informações fornecidas a priori. Neste caso, por exemplo, podem-se utilizar métodos que combinem as diversas funções objetivo numa única função, obtendo-se como resultado da otimização, uma solução única.

- No segundo modo, sem nenhuma informação adicional, encontra-se o conjunto das soluções ótimas de Pareto, para que, dentre estas, possa-se escolher uma a posteriori. Neste caso, métodos que possibilitem a obtenção de um grande número de soluções ótimas de Pareto podem ser utilizados, a fim de possibilitar uma melhor escolha pessoal posterior.

Estas considerações sugerem que os métodos baseados em populações sejam preferidos em relação aos métodos clássicos, uma vez que os métodos clássicos, ao contrário dos métodos baseados em populações, não foram projetados para trabalhar com múltiplas soluções.

Um problema de otimização multiobjetivo pode ser matematicamente descrito conforme as Equações (2.1) e (2.2), sendo que $f(X)$ é dada por um vetor cujas componentes são k funções objetivo, como mostra a Equação (2.4) a seguir:

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

Com $X \in E$:

$X = (x_1, x_2, \dots, x_n)$, corresponde aos n parâmetros de otimização, onde $E \in \mathfrak{R}^n$ é o espaço de soluções.

Denotando-se por y os vetores objetivos obtidos por $\{f(X) \mid X \in E\}$, tem-se:

$f: E \rightarrow y$, ou seja, E é mapeado por f em y .

$y \in \mathfrak{R}^k$ é usualmente denominado como o espaço de atributos, sendo δy definido como a fronteira de y .

Definindo-se $f_1^*(X), f_2^*(X), \dots, f_k^*(X)$ como, por exemplo, o valor máximo individual de cada função objetivo, uma solução $f^*(X)$ pode ser escrita da seguinte forma: $f^*(X) = [f_1^*(X), f_2^*(X), \dots, f_k^*(X)]$. Como $f^*(X)$ maximiza simultaneamente todos os objetivos, ela é uma solução ideal. No entanto, este tipo de solução raramente é factível, principalmente para os problemas reais, onde na maioria das vezes os objetivos são conflitantes entre si. A Figura 2.4 ilustra a representação bidimensional da solução de um problema de otimização multiobjetivo com dois objetivos (MACHADO, 2005).

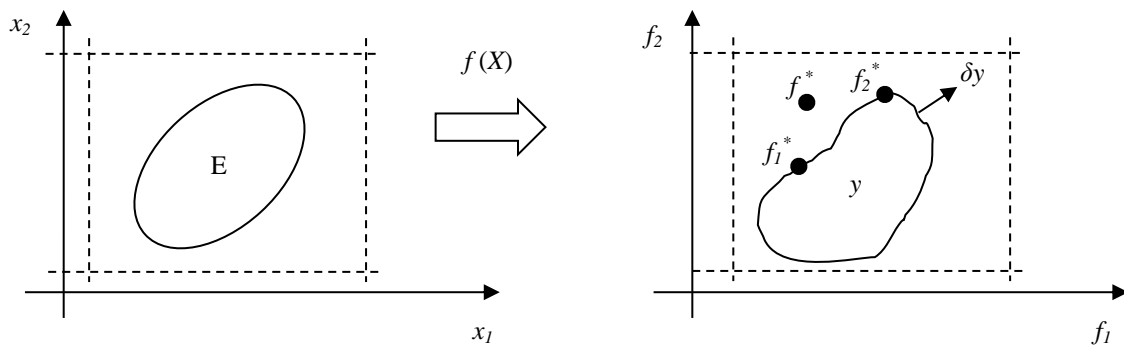


Figura 2.4. Representação bidimensional da solução de um problema de otimização multiobjetivo com dois objetivos.

O conjunto de soluções do problema representado pela Equação (2.4), consiste de todos os vetores de otimização que não podem melhorar o valor de qualquer objetivo, sem a degradação de pelo menos um dos outros objetivos. Estes vetores são os chamados ótimos de Pareto.

Sem perda de generalidade, podem-se descrever as soluções ótimas de Pareto formalmente para um problema de maximização puro, isto é, onde todas as funções objetivo devem ser maximizadas, pelas seguintes definições:

✓ **Dominância**

Um ponto $X = (x_1, x_2, \dots, x_n)$ domina outro ponto $Y = (y_1, y_2, \dots, y_n)$ se:

$$\forall i \in \{1, 2, \dots, k\}: f_i(X) \geq f_i(Y) \quad (2.5)$$

$$\exists j \in \{1, 2, \dots, k\}: f_j(X) > f_j(Y) \quad (2.6)$$

✓ Solução Não Dominada ou Solução Ótima de Pareto

Um ponto X é uma solução não dominada e pertence à fronteira de Pareto se: não existe nenhum outro ponto Y , tal que Y domine X . Em outras palavras, o conjunto formado por soluções do problema que não são dominadas por nenhuma outra solução é denominado conjunto de soluções não dominadas ou conjunto ótimo de Pareto.

A região em \mathcal{R}^k formada pelos vetores objetivos das soluções ótimas de Pareto, é chamada de superfície de Pareto.

A Figura 2.5 mostra a superfície de Pareto para um problema bidimensional com dois objetivos (MACHADO, 2005).

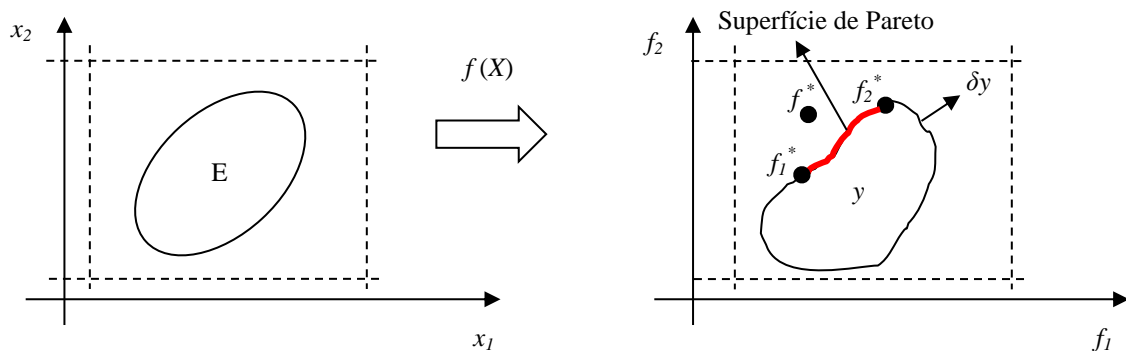


Figura 2.5. Representação bidimensional da superfície de Pareto para um problema de otimização multiobjetivo com dois objetivos.

Vale ressaltar que existe uma importante diferença entre um conjunto de soluções não dominadas e um conjunto ótimo de Pareto. Um conjunto de soluções não dominadas é definido no contexto de uma amostra do espaço de busca, enquanto que o conjunto ótimo de Pareto é definido em relação a todo o espaço de busca.

Existem diversas técnicas para a solução de problemas multiobjetivo, tais como: o método dos objetivos ponderados (ou soma ponderada), a formulação *Min-Max*, Programação de Metas, entre outras. Uma visão geral deste e de outros métodos pode ser encontrada em (MACHADO, 2005).

Um exemplo simples de problema de otimização multiobjetivo conhecido como f_2 , foi proposto por (SCHAFFER, 1984) e é descrito a seguir:

O problema consiste na minimização simultânea de duas funções objetivo $g(x)$ e $h(x)$, sendo $g(x) = x^2$ e $h(x) = (x-2)^2$.

A Figura 2.6 mostra os gráficos das funções $g(x)$ e $h(x)$, bem como um gráfico paramétrico no espaço das funções objetivo, identificando a fronteira ótima de Pareto.

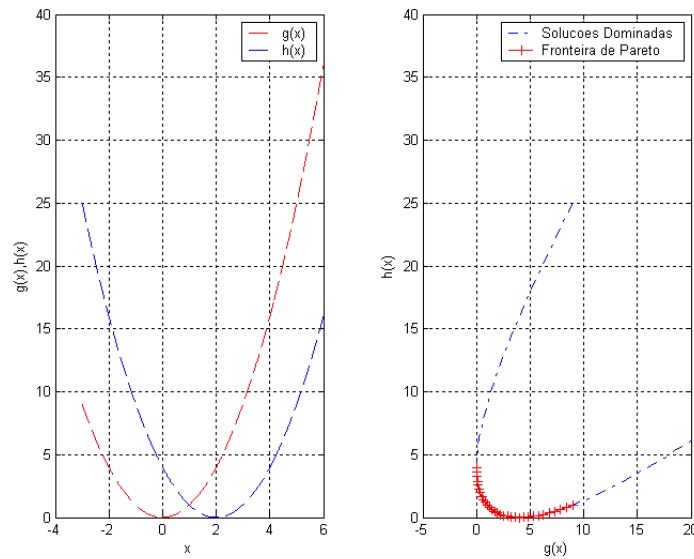


Figura 2.6. À esquerda: Gráfico das funções objetivo do problema f2 (SCHAFFER, 1984).
À direita: Identificação da Fronteira de Pareto ótima para o problema.

As evidências disponíveis na Figura 2.6 sugerem que as soluções ótimas de Pareto estejam compreendidas no intervalo $[0,2]$, tendo em vista que fora desta região, tanto a função $g(x)$ quanto a função $h(x)$ crescem. Dentro do intervalo, ocorre uma compensação entre as funções objetivo, isto é, enquanto uma cresce, a outra diminui de valor, não sendo tão fácil identificar as soluções ótimas de Pareto, embora as soluções não dominadas que compõem a fronteira de Pareto estejam compreendidas neste intervalo. O gráfico paramétrico serve de auxílio para a visualização da fronteira de Pareto para este problema. Para problemas maiores pode tornar-se necessário o uso de métodos computacionais para tal finalidade.

2.7. Sumário do Capítulo

Neste capítulo, fez-se uma breve revisão dos conceitos básicos sobre otimização que serão utilizados ao longo do presente estudo. Sem a pretensão de se fazer um estudo prolixo sobre os assuntos tratados, buscou-se de forma clara, direta e objetiva, a introdução dos principais conceitos de interesse de modo a tornar mais fácil o entendimento dos próximos capítulos.

Capítulo 3

– Algoritmos Genéticos e Evolução Diferencial –

3.1. Introdução

Neste capítulo, apresenta-se uma breve revisão dos fundamentos teóricos básicos de dois importantes Algoritmos Evolucionários (AE). São eles: os Algoritmos Genéticos (*Genetic Algorithms, GA*) (HOLLAND, 1975) e (GOLDBERG, 1989) e o algoritmo de Evolução Diferencial (*Differential Evolution, DE*) (STORN e PRICE, 1995). A introdução desses algoritmos neste capítulo é justificada pelo fato que, tanto os GAs quanto o DE, apresentam, em suas estratégias de busca, certas características de similaridade com o algoritmo ABC original, o qual é apresentado em detalhes no capítulo subsequente. Acredita-se, portanto, que a abordagem prévia destes algoritmos neste capítulo, permitirá ao leitor uma melhor interpretação geométrica do algoritmo ABC, facilitando assim, o correto entendimento do mesmo e das metodologias propostas neste trabalho de tese.

3.2. Considerações Iniciais

Conforme mencionado no capítulo anterior, existem muitos métodos de otimização e cada um desses métodos apresenta melhor ou pior desempenho dependendo do tipo de problema considerado. Sabe-se também, que a escolha do método para a otimização de um determinado problema depende basicamente das características do referido problema, ou, mais precisamente, das características da função que o representa.

Assim, de acordo com as características dos problemas (comportamento da função objetivo e/ou das funções de restrições), pode-se classificar os métodos de otimização em dois grandes grupos: Programação Linear e Programação Não Linear.

Os problemas que apresentam a função objetivo e as funções de restrições como funções lineares das variáveis de otimização, são tratados através de técnicas da Programação Linear.

Já os problemas que apresentam algum tipo de não linearidade (seja na função objetivo ou em qualquer função de restrição em geral), são tratados por meio das técnicas da Programação Não Linear, ou então, são linearizados em torno de um ponto de operação para que possam ser resolvidos através de alguma técnica da Programação Linear.

Os métodos da Programação Não Linear são classificados em outros dois grandes grupos: os métodos determinísticos, também denominados de métodos clássicos, e os métodos não determinísticos, também conhecidos como métodos naturais (ou bioinspirados ou probabilísticos).

A grande maioria dos métodos clássicos faz uso do cálculo de derivadas da função objetivo para determinar a direção de busca do ponto de solução. Pelo fato de utilizarem um único ponto do espaço de busca (ou uma única solução atualizada a cada iteração), e de necessitarem de informações do vetor gradiente (seja procurando o ponto onde ele se anula ou utilizando a direção para a qual aponta), tais métodos, ao se depararem com mínimos locais, por exemplo, não conseguem avançar na busca, convergindo prematuramente, sem atingir o mínimo global. Devido a esta característica, o ponto de solução encontrado quase sempre é um ótimo local. Outra grande desvantagem destes métodos é a dependência do ponto inicial para o processo iterativo, ou seja, para encontrar um ponto de solução de boa qualidade é necessário o conhecimento prévio de um ponto inicial.

Ainda assim, os métodos clássicos de otimização costumam ser bastante confiáveis e produzem excelentes resultados quando as funções são contínuas, convexas e unimodais. Por outro lado, tais métodos não podem ser considerados bons candidatos como algoritmos de otimização eficientes quando aplicados a uma vasta gama de problemas práticos da engenharia, seja por causa da dificuldade de formulação da modelagem ou do esforço matemático exigido na solução, principalmente quando estão envolvidas funções não lineares, não convexas, multimodais, não diferenciáveis ou descontínuas.

Deste modo, os métodos não determinísticos (naturais) têm sido muito utilizados para a solução de problemas de busca e otimização em diversos domínios para os quais soluções robustas⁹ são difíceis ou até mesmo impossíveis de serem encontradas por

⁹ O termo precisão designa a capacidade de um algoritmo de, estando próximo da solução exata do problema, aproximar ainda mais tal solução exata. Já o termo robustez, por sua vez, designa a capacidade do algoritmo de, estando distante da solução exata do problema, atingir as proximidades desta solução.

meio das abordagens clássicas, justificando-se o fato de que cada vez mais estão sendo desenvolvidas pesquisas com estes métodos.

Em alguns métodos de otimização naturais, a função objetivo pode ser avaliada diversas vezes, sendo possível trabalhar com vários pontos de busca ao mesmo tempo em uma iteração (população). Estes métodos ganharam bastante destaque com o avanço dos recursos computacionais, uma vez que requerem um número elevado de avaliações do problema a fim de que se dê chance aos mesmos de explorarem devidamente toda a região viável de busca em que está contida a solução ótima. O alto custo computacional, no entanto, é compensado pela diminuição das chances de que estes métodos fiquem presos em mínimos locais.

Em síntese, pode-se dizer que os chamados métodos naturais, dos quais os Algoritmos Evolucionários (AE) fazem parte, exigem maior esforço computacional quando comparados aos métodos clássicos. Por outro lado, a grande maioria dos métodos naturais trabalha com mais de uma solução atualizada a cada iteração e não com apenas uma, como é o caso dos métodos clássicos em geral. Os métodos naturais são também mais adaptáveis a diferentes tipos de problemas e apresentam vantagens bastante atrativas, tais como: fácil implementação e maior robustez na descoberta de soluções, além de não necessitarem de garantias da continuidade da função objetivo e das restrições, dispensando também, o conhecimento prévio sobre a complexidade dos espaços de busca envolvidos.

A seguir, são revistos dois importantes e bem conhecidos AE. São eles: os Algoritmos Genéticos (*Genetic Algorithms, GA*) (HOLLAND, 1975) e GOLDBERG, 1989) e o Algoritmo de Evolução Diferencial (*Differential Evolution, DE*) (STORN e PRICE, 1995).

3.3. Algoritmos Genéticos

Os Algoritmos Genéticos (*Genetic Algorithms, GA*) (HOLLAND, 1975) e (GOLDBERG, 1989) são algoritmos de busca que empregam conceitos de seleção natural e genética. A execução do algoritmo genético começa com a criação de uma população inicial de indivíduos. Esta população é avaliada de forma que os indivíduos que representam uma solução melhor para o problema, ou seja, os mais aptos terão mais

chances de se reproduzirem. O processo termina quando a condição de parada é atingida, como por exemplo, número máximo de populações geradas sem mudanças.

Segundo (GOLDBERG, 1989), os GAs diferem dos métodos tradicionais para a otimização de problemas deste tipo nos seguintes pontos:

- (i) Trabalham com a codificação do conjunto de variáveis, e não com as variáveis propriamente ditas;
- (ii) Trabalham com uma população de possíveis soluções, e não com um único ponto;
- (iii) Trabalham com a informação da função objetivo, e não com derivadas ou outras informações auxiliares;
- (iv) Utilizam regras de transição probabilísticas, e não regras determinísticas;
- (v) Necessitam de pouca ou nenhuma informação sobre a relação entre as variáveis do problema.

Para utilizar o GA, inicialmente é necessário atribuir a cada um dos possíveis valores de uma variável, uma determinada cadeia de caracteres binários, ou seja, codificar as variáveis do problema como uma cadeia de caracteres binários de tamanho finito (*string*). Cada *string* é referenciada como genótipo ou cromossomo, em analogia ao sistema biológico. O genótipo é a codificação de uma possível solução para o problema, solução esta que recebe o nome de fenótipo.

Uma vez definida a codificação dos indivíduos, a próxima etapa é definir um modo de avaliar o quão boa é a solução que eles representam para o problema, ou seja, definir uma função objetivo.

A codificação e a definição da função objetivo do problema são pontos críticos no desempenho do GA, e, se inadequadamente escolhidos, comprometem os resultados.

A primeira etapa na implementação de um GA é a criação da população inicial de cromossomos, normalmente aleatória.

Na criação da população inicial, um ponto importante é definir a dimensão desta população, que será a mesma nas gerações futuras. Não existe um número ideal de

indivíduos de uma população. Uma população maior leva a um aumento do esforço computacional.

Em uma segunda etapa, cada indivíduo da população é avaliado através da função objetivo. Aplica-se, então, a esta população, os operadores genéticos. Um GA básico que leva a bons resultados é composto por três operadores:

- ⇒ Seleção (ou Reprodução);
- ⇒ Cruzamento (ou Recombinação);
- ⇒ Mutação.

A Seleção (ou Reprodução) é um processo em que uma nova população é formada a partir de uma população já existente de acordo com o valor de sua função objetivo. Este processo determina que os indivíduos mais aptos passem para a próxima geração e aqueles menos aptos sejam eliminados.

Um dos métodos mais comuns de seleção é conhecido como método da roleta (*Roulette Wheel Selection*) (GOLDBERG, 1989). Considere uma roleta com divisões onde cada indivíduo da população ocupa um espaço nesta roleta e este espaço é proporcional ao valor da função objetivo. A roleta será girada e aqueles elementos com uma proporção de espaço maior na roleta, ou seja, aqueles que são mais aptos terão uma maior probabilidade de serem selecionados. A Figura 3.1 mostra duas roletas para seleção de indivíduos com base em suas *Fitness* (MACHADO, 2005).

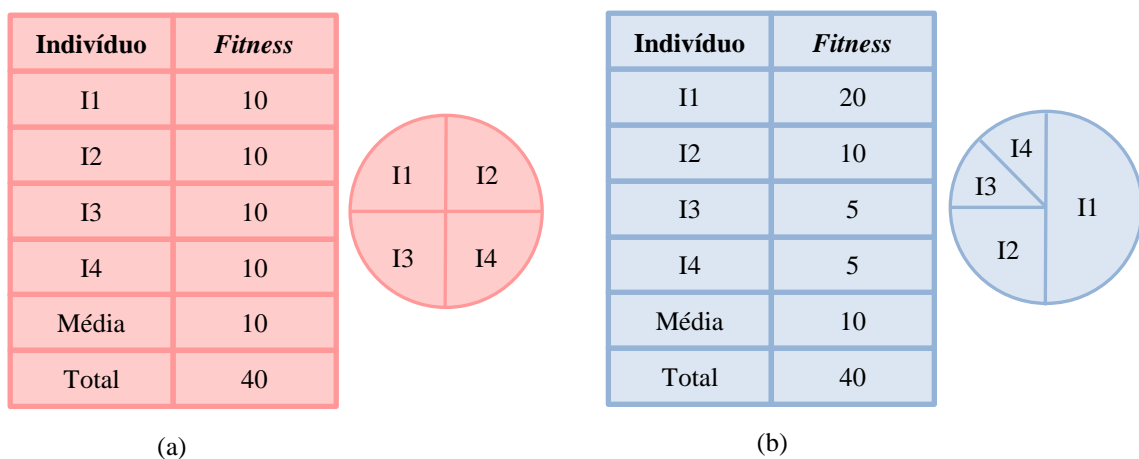


Figura 3.1. Representação do processo de seleção por roleta proporcional à *fitness*.

- (a) Roleta para indivíduos de mesma *fitness*.
- (b) Roleta para indivíduos de *fitness* diferentes.

Para garantir que os melhores indivíduos sobrevivam e sejam copiados para a próxima geração, utiliza-se um recurso denominado elitismo. Esta estratégia permite que o cromossomo melhor adaptado de uma geração permaneça na população até que um indivíduo de melhor desempenho surja, tomando o seu lugar.

Uma vez que a seleção dos indivíduos esteja completa, inicia-se a fase da evolução da população que costuma ser processada de duas formas: cruzamento e mutação.

O cruzamento é um dos operadores mais utilizados. Como na natureza, atributos do pai e da mãe são misturados na criação dos seus descendentes. Nesta etapa do GA é realizado o cruzamento do material genético de diferentes cromossomos, com filhos recebendo informações de seus progenitores, conforme mostra a Figura 3.2. Define-se, ainda, uma taxa de cruzamento, a qual representa a probabilidade de que o operador cruzamento seja aplicado a um cromossomo na evolução da geração.

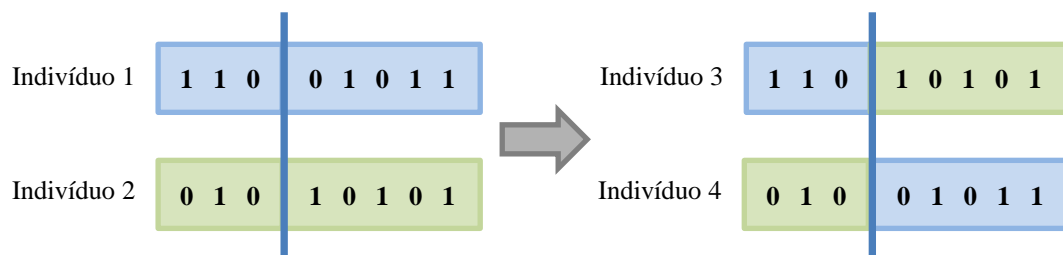


Figura 3.2. Representação do processo de cruzamento em um GA clássico.

O operador mutação modifica aleatoriamente um ou mais genes de um cromossomo. Na implementação do GA, é definida uma taxa de mutação que representa a probabilidade de que o operador mutação seja aplicado a um cromossomo.

Uma vez que os processos de seleção, cruzamento e mutação são aplicados por completo a uma população, tem-se a criação de uma nova população que novamente deverá ser avaliada através da função objetivo. Este processo se repete até que o critério de parada seja atingido.

3.4. Algoritmo de Evolução Diferencial

O Algoritmo de Evolução Diferencial (*Differential Evolution, DE*) (STORN e PRICE, 1995) é um método de busca estocástica que surgiu de tentativas de resolver o

problema de ajuste polinomial de *Chebyshev* (PRICE e STORN, 2005). A idéia básica é usar a diferença de vetores para perturbar a população de vetores (indivíduos) resultando em um método que requer poucas variáveis de controle, é de rápida convergência, fácil de ser utilizado e robusto.

Um algoritmo DE simples pode ser descrito da seguinte forma: Seja uma população inicial escolhida aleatoriamente, composta por Np vetores (indivíduos), cobrindo todo o espaço de busca. Para um problema com n variáveis de otimização, cada vetor possui n componentes. Normalmente, essa população é criada por uma distribuição de probabilidade uniforme, quando não há nenhum conhecimento sobre o problema. A população de vetores segue uma evolução natural conforme descrita a seguir, porém, o número de indivíduos permanece fixo durante o processo de otimização (minimização ou maximização).

A idéia básica do DE é gerar novos indivíduos, chamados vetores “base”, pela adição da diferença vetorial ponderada entre dois indivíduos aleatórios da população a um terceiro indivíduo. Esta operação é chamada mutação.

As componentes deste novo indivíduo (vetor base) são misturadas com as componentes de um indivíduo escolhido aleatoriamente, o qual é chamado de vetor “alvo”, para resultar no chamado vetor “experimental”. O processo de misturar os parâmetros é referenciado normalmente como cruzamento no campo dos Algoritmos Evolucionários (AE). Se o custo (ou *fitness*) do vetor experimental for menor (no caso de um problema de otimização) que o custo do vetor alvo, então o vetor experimental será o vetor alvo da geração seguinte. Esta operação é chamada de seleção. O procedimento é finalizado através de algum critério de parada.

Os operadores do DE são baseados no princípio da evolução natural, cujos objetivos são manter a diversidade da população, evitar convergências prematuras e obter a melhor solução. Esses operadores são descritos a seguir:

- Mutação:

Para a obtenção do vetor base $V^{(q+1)}$, considere os vetores $X_\alpha^{(q)}$, $X_\beta^{(q)}$, $X_\gamma^{(q)}$, distintos entre si e escolhidos aleatoriamente em uma população com Np indivíduos. Np deve ser maior ou igual a 4 para garantir uma quantidade suficiente de indivíduos para a execução do algoritmo. Os índices aleatórios $\alpha, \beta, \gamma, \in \{1, \dots, Np\}$ são inteiros distintos entre si. Utilizando o par de vetores (X_β, X_γ) da q -ésima geração define-se o vetor

diferença $(X_\beta - X_\gamma)$. Esta diferença é multiplicada por F_p , sendo denotada diferença vetorial ponderada ou apenas diferença ponderada, e será utilizada para “perturbar” o terceiro vetor $X_\alpha^{(q)}$. O fator de perturbação F_p é um número real, positivo e pertencente ao intervalo $[0,2]$ (STORN e PRICE, 1995) e controla a amplitude do vetor diferença.

O processo de mutação pode ser escrito como:

$$V^{(q+1)} = X_\alpha^q + F_p(X_\beta^q - X_\gamma^q) \quad (3.1)$$

A Figura 3.3 mostra um exemplo bidimensional que ilustra os diferentes vetores que geram o vetor base $V^{(q+1)}$.

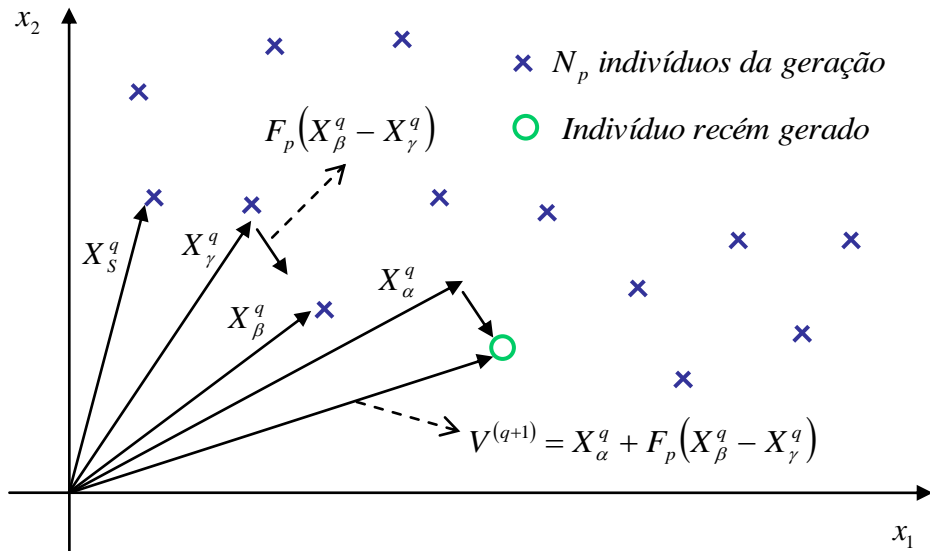


Figura 3.3. Processo para formar o vetor base $V^{(q+1)}$ de uma função bidimensional.

- **Cruzamento:**

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

$$u(i)^{(q+1)} = \begin{cases} v(i)^{(q+1)}, & \text{se } rand_i \leq Pc \\ x_S(i)^q, & \text{se } rand_i > Pc, \quad i = 1, \dots, n \end{cases} \quad (3.2)$$

Sendo $v(i)^{(q+1)}$ a $(q+1)$ -ésima componente do vetor base $V^{(q+1)}$, $x_s(i)$ a do vetor alvo $X_s^{(q)}$ e $rand_i$ um número gerado aleatoriamente no intervalo $[0,1]$. $Pc \in [0,1]$ é a probabilidade de cruzamento e representa a probabilidade do vetor experimental herdar os valores das variáveis do vetor base, devendo ser fornecida pelo usuário. Quando $Pc = 1$, por exemplo, todas as componentes do vetor experimental virão do vetor base $V^{(q+1)}$. Por outro lado, se $Pc = 0$, todas as componentes do vetor experimental virão do vetor alvo $X_s^{(q)}$. Este tipo de cruzamento, apresentado por (STORN e PRICE, 1995) é denominado operador cruzamento binomial, sendo executado em cada variável sempre que um número aleatório $rand \in [0,1]$ for menor que a probabilidade de cruzamento denominada Pc . A Figura 3.4 mostra o processo de cruzamento binomial para uma função de 7 variáveis.

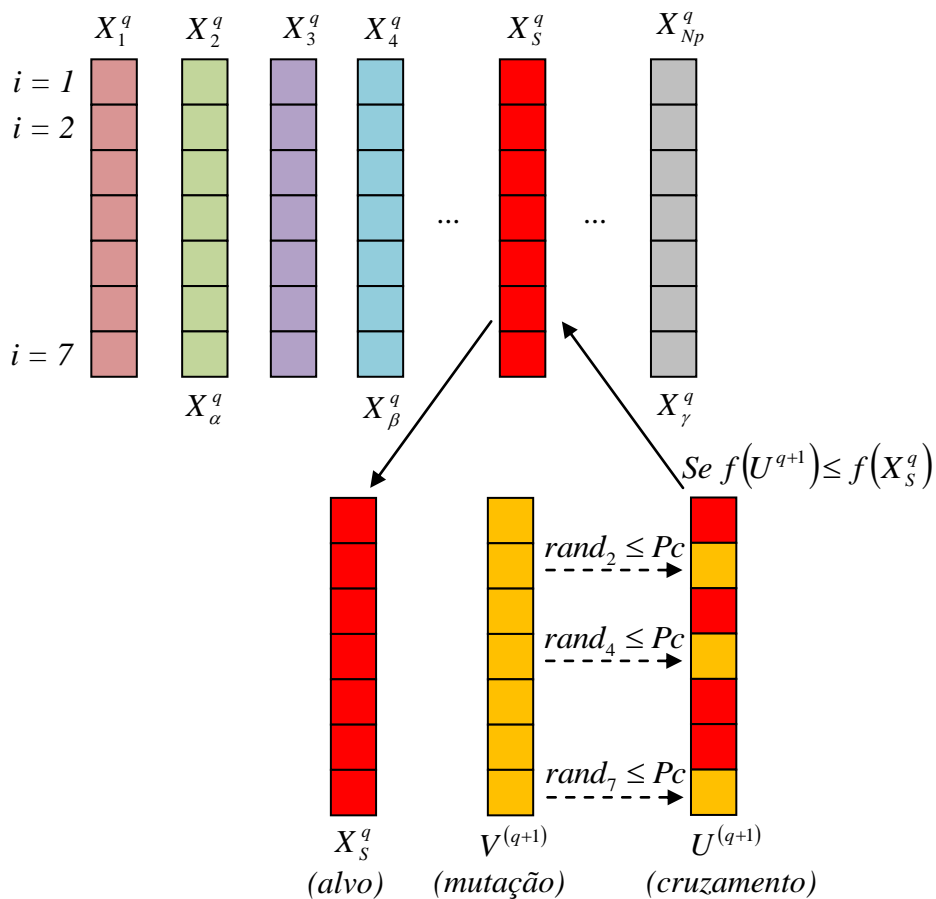


Figura 3.4. Processo de cruzamento binomial ($\alpha = 2$, $\beta = 4$ e $\gamma = Np$).

Anos mais tarde, (STORN e PRICE, 1997) desenvolveram o operador cruzamento exponencial, em que o cruzamento é executado nas variáveis enquanto o número aleatório $rand \in [0,1]$ for menor que a probabilidade de cruzamento Pc . Na primeira vez em que este número aleatório ultrapassar o valor de Pc , nenhum cruzamento é executado e as variáveis restantes são deixadas intactas, de acordo com Equação (3.3), ou seja:

$$\begin{aligned} \text{Enquanto } rand_i \leq Pc, \quad u(i)^{(q+1)} &= v(i)^{(q+1)}, \\ \text{Se } rand_i > Pc, \quad u(j)^{(q+1)} &= x_s(j)^q, \quad j = (i + 1), \dots, n \end{aligned} \quad (3.3)$$

A Figura 3.5 mostra o processo de cruzamento exponencial para uma função de 7 variáveis.

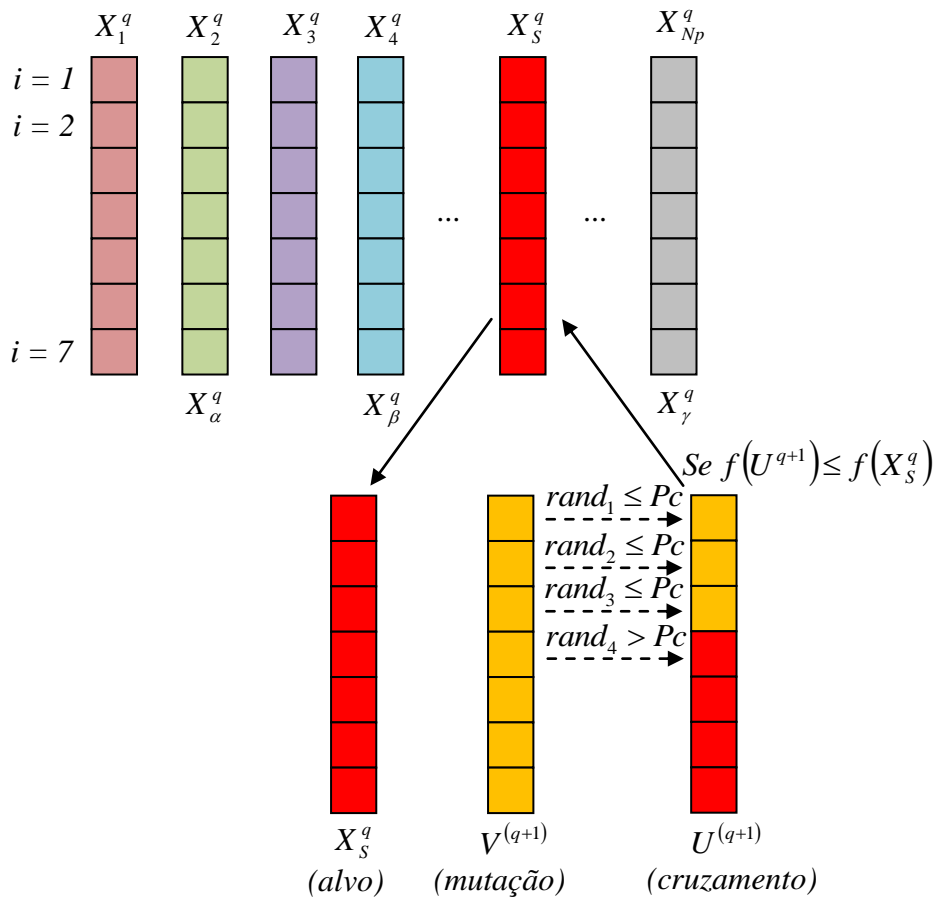


Figura 3.5. Processo de cruzamento exponencial ($\alpha = 2, \beta = 4$ e $\gamma = Np$).

Se após o cruzamento uma ou mais componentes do vetor experimental estiver fora da região viável de busca definida pelas restrições laterais das variáveis de otimização, as correções mostradas na Equação (3.4) devem ser feitas:

$$\begin{aligned} \text{Se } u(i) < x(i)^{inf}, \quad \text{então } u(i) &= x(i)^{inf} \\ i &= 1, \dots, n \\ \text{Se } u(i) > x(i)^{sup}, \quad \text{então } u(i) &= x(i)^{sup} \end{aligned} \quad (3.4)$$

- Seleção:

A seleção é o processo de produzir filhos melhores. Diferentemente de outros algoritmos evolutivos, o DE não usa elitismo nem seleção proporcional. Em vez disso, o custo do vetor experimental $U^{(q+1)}$ é calculado e comparado com o custo do vetor alvo $X_S^{(q)}$. Se o custo do vetor experimental for menor que o custo do vetor alvo, o vetor alvo da próxima geração será o vetor experimental. Caso contrário, o vetor alvo da próxima geração será o vetor alvo da geração atual, conforme Equação (3.5).

$$\begin{aligned} \text{Se } f(U^{(q+1)}) \leq f(X_S^q), \quad \text{então } X_S^{(q+1)} &= U^{(q+1)} \\ \text{Se } f(U^{(q+1)}) > f(X_S^q), \quad \text{então } X_S^{(q+1)} &= X_S^q \end{aligned} \quad (3.5)$$

O procedimento é finalizado através de algum critério de parada, sendo que o número máximo de gerações deve ser estabelecido. Em geral, o desempenho do DE depende basicamente do tamanho da população Np , da taxa de cruzamento e também do fator de perturbação Fp .

3.5. Sumário do Capítulo

Neste capítulo, fez-se uma breve revisão da teoria básica dos GAs e do algoritmo DE. Foram descritas as principais características desses algoritmos com o objetivo de introduzir os principais operadores de uso comum entre eles e o algoritmo ABC original. Tais operadores serão detalhados no capítulo seguinte, juntamente com a descrição das metodologias propostas.

Capítulo 4

– Apresentação das Metodologias Propostas –

4.1. Introdução

Este capítulo é dedicado à apresentação das metodologias propostas neste trabalho de tese. São descritos os princípios biológicos que motivaram o desenvolvimento do algoritmo ABC (*Artificial Bee Colony, ABC*) (KARABOGA, 2005), bem como seu respectivo algoritmo computacional. Em seguida, são apresentadas as duas versões propostas no presente estudo, as quais são versões modificadas do algoritmo ABC original e que serão utilizadas nas simulações de recarga do ciclo 7 de operação da CNA1, cuja modelagem encontra-se descrita em detalhes no próximo capítulo.

4.2. O Comportamento de Abelhas Reais

O modelo que representa a inteligência coletiva de um enxame de abelhas melíferas consiste de quatro componentes essenciais: fontes de alimento, abelhas operárias, abelhas observadoras e abelhas exploradoras. E define dois modos básicos de comportamento: recrutamento para uma fonte de alimento e abandono de uma fonte de alimento. Os principais componentes deste modelo são assim explicados:

- Fontes de alimento: Para escolher uma fonte de alimento, uma abelha avalia diversas propriedades relacionadas com esta fonte, tais como: proximidade com a colméia, quantidade e qualidade do alimento (néctar) e facilidade ou dificuldade para extração deste alimento. Por simplificação, a todas essas propriedades de uma fonte de alimento, dá-se o nome de proveitabilidade da fonte.

- Abelhas exploradoras: São abelhas que estão procurando por fontes proveitosas de alimento para serem exploradas. Elas buscam por essas fontes ao redor da colméia de forma aleatória e, eventualmente, são capazes de encontrar boas fontes de alimento. Ao encontrar uma boa fonte de alimento, essas abelhas deixam de ser exploradoras e passam a ser operárias dessa fonte.

- Abelhas operárias: São abelhas que antes eram exploradoras e que por já terem encontrado uma fonte proveitosa de alimento, passaram a ser operárias desta fonte. Carregam consigo informações a respeito desta fonte em particular e compartilham essas informações com as abelhas observadoras com certa probabilidade.

- Abelhas observadoras: São abelhas que ficam na colméia à espera de informações sobre as fontes mais proveitosas trazidas pelas abelhas operárias. Elas tentam descobrir novas fontes com base nas informações compartilhadas pelas abelhas operárias, as quais compartilham essas informações com certa probabilidade através de uma espécie de “dança”. De posse das informações sobre as fontes mais proveitosas, uma abelha observadora poderá decidir-se por uma fonte de alimento, a qual passará a explorá-la, tornando-se de igual modo uma abelha operária com relação a essa fonte.

A troca de informações entre as abelhas é a ocorrência mais importante na formação do conhecimento coletivo. Logo, a parte mais importante da colméia no que diz respeito à troca de informações é a chamada “área de dança”, pois é lá que ocorre a comunicação entre as abelhas sobre a qualidade das fontes de alimento já descobertas. Essa dança realizada pelas abelhas operárias dentro da colméia a fim de recrutar as abelhas observadoras para as fontes de alimento descobertas é chamada de “*waggle dance*” (KARABOGA, 2005). Essa dança é maior em duração quanto melhor for uma fonte de alimento.

Uma vez que as informações sobre as melhores fontes estiverem disponíveis na área de dança para uma abelha observadora (através da dança realizada pelas abelhas operárias) ela poderá assistir a inúmeras danças e então se decidir pela fonte que achar mais proveitosa. Logo, quanto mais informações a respeito das melhores fontes, maior a probabilidade de que as abelhas observadoras escolham as melhores fontes. Vale ressaltar que as abelhas operárias compartilham suas informações sobre as fontes descobertas com uma probabilidade que é proporcional à proveitabilidade da fonte e o compartilhamento destas informações por meio da *waggle dance* é maior em duração para as melhores fontes. Deste modo, o recrutamento para uma fonte de alimento é proporcional à proveitabilidade da fonte.

Após encontrar uma fonte de alimento, uma abelha utiliza sua própria capacidade de memorizar a localização desta fonte e então imediatamente começa a explorá-la. Deste modo, ela se torna uma operária. Em seguida, essa abelha operária colhe uma carga do néctar da fonte e retorna à colméia. Lá, ela descarrega o néctar em um depósito de alimento.

Em seguida, ela tem as seguintes opções:

- ⇒ Tornar-se novamente uma abelha exploradora após abandonar a fonte de alimento;
- ⇒ Ela pode dançar e recrutar companheiras da colônia (abelhas observadoras) antes de retornar para a mesma fonte;
- ⇒ Ela pode continuar explorando a mesma fonte, sem recrutar outras abelhas.

É importante notar que nem todas as abelhas começam a forragear simultaneamente.

4.3. O Algoritmo ABC Original Básico

O algoritmo ABC foi inicialmente proposto por Dervis Karaboga (KARABOGA, 2005) como uma metaheurística baseada em populações para a otimização de funções multimodais e de múltiplas variáveis em espaços de busca contínuos.

O algoritmo ABC tenta simular o comportamento forrageiro “inteligente” observado em uma colônia de abelhas reais, propiciando um procedimento de busca baseada em populações na qual os indivíduos, também chamados posições de fonte de alimento, são modificadas pelas abelhas artificiais com o tempo. A escala de tempo é medida em intervalos discretos, denominados ciclos e o objetivo das abelhas artificiais é descobrir os locais de fontes de alimento com grandes quantidades de néctar e, finalmente, aquela com a maior quantidade de néctar.

De maneira resumida, no algoritmo ABC, uma colônia de abelhas artificiais busca por ricas fontes de alimento (boas soluções para um dado problema).

As abelhas artificiais “voam” em um espaço de busca multidimensional escolhendo fontes de alimento de acordo com suas próprias experiências (operárias) ou de acordo com a experiência de suas congêneres (observadoras) e, então, ajustam suas posições. Já as exploradoras escolhem as fontes de alimento de forma aleatória, sem usar experiência.

Se a quantidade de néctar de uma determinada fonte for maior que a quantidade de néctar existente na fonte que uma abelha já possui em sua memória, ela memoriza a nova fonte e abandona a fonte anterior.

Em outras palavras o ABC combina métodos de busca local (realizada pelas operárias) com métodos de busca global (realizada pelas exploradoras) utilizando, ainda a informação compartilhada (observadoras), ou seja, o conhecimento e a experiência das operárias são transmitidos para as observadoras.

Para aplicar o algoritmo ABC, o problema de otimização é primeiro transformado no problema de se encontrar o melhor vetor de parâmetros que minimiza ou maximiza uma função objetivo. Assim, as abelhas artificiais aleatoriamente descobrem uma população inicial de vetores-solução e então, iterativamente os “melhoram” movendo-os em direção das melhores soluções por meio de um mecanismo de busca estocástica, combinada com busca por vizinhança, enquanto abandonam soluções ruins.

No algoritmo ABC a posição de uma fonte representa uma possível solução para o problema de otimização e a quantidade de néctar corresponde à qualidade ou medida de aptidão da solução associada (equivalente ao valor da função objetivo) (KARABOGA, 2005).

No primeiro passo do algoritmo, o ABC gera uma população inicial aleatoriamente distribuída de SN soluções (do inglês, *Source Number*), onde SN é a quantidade de abelhas operárias. Em outras palavras, o número de abelhas operárias é igual ao número de candidatos à solução na população, que, por sua vez, também é igual ao número de posições de fontes de alimento, já que cada abelha operária ocupa uma posição de fonte de alimento.

Cada candidato à solução x_i ($i = 1, 2, \dots, SN$) é um vetor de dimensão D . Neste caso, D é o número de parâmetros de otimização.

Após a inicialização, a população de posições (soluções) é submetida a repetidos ciclos, $C = 1, 2, \dots, MCN$, do processo de busca das operárias, observadoras e exploradoras.

Uma abelha operária produz uma modificação na posição da fonte (solução) em sua memória dependendo de informação local (informação visual) e testa a quantidade de néctar (valor da função objetivo) da nova fonte (nova solução). Caso a quantidade de néctar da nova fonte seja maior que a da fonte anterior, a abelha memoriza a posição da

nova fonte e esquece a anterior. Caso contrário, ela mantém a posição da fonte anterior em sua memória.

Após todas as abelhas operárias completarem seu processo de busca, elas compartilham a informação sobre a posição e a quantidade de néctar das fontes visitadas com as abelhas observadoras. Uma abelha observadora avalia as informações sobre o néctar de todas as operárias e escolhe uma fonte de alimento com uma probabilidade relacionada com a quantidade de néctar. (Seleção das melhores fontes. Equivale ao processo de Seleção por roleta proporcional à *fitness*).

Assim como no caso das operárias, as observadoras também produzem uma modificação na posição das fontes escolhidas em suas memórias e avaliam a quantidade de néctar da fonte candidata. Caso a quantidade de néctar da nova fonte seja maior que a quantidade de néctar da fonte anterior, a observadora memoriza a posição da nova fonte e esquece a posição da fonte anterior.

Os principais passos do algoritmo são:

1. Inicialize a população;
2. Repita os seguintes passos:
 - a. Posicione as abelhas operárias em suas fontes de alimento;
 - b. Posicione as abelhas observadoras nas fontes de alimento dependendo da quantidade de néctar das fontes;
 - c. Envie as exploradoras para a região de busca para descobrirem novas fontes de alimento;
 - d. Memorize a melhor posição de fonte já encontrada¹⁰;
 - e. Até que critérios de parada sejam alcançados.

No estágio de inicialização do algoritmo, um conjunto de fontes de alimento é aleatoriamente selecionado pelas abelhas e as quantidades de néctar das fontes são determinadas.

No primeiro passo do ciclo, essas abelhas vão até a colméia e compartilham as informações sobre o néctar das fontes com as abelhas que ficam esperando na colméia

¹⁰ Fonte com maior quantidade de néctar, caso seja um problema de maximização.

(observadoras). Isso ocorre na área de dança através da “*waggle dance*” (KARABOGA, 2005).

Depois de compartilharem as informações com as observadoras, cada abelha operária retorna à fonte já visitada por ela em ciclos anteriores, uma vez que essas fontes ainda existem em suas memórias, e então, através de informações visuais, ela escolhe uma nova fonte na vizinhança da fonte em sua memória e avalia a quantidade de néctar dessa nova fonte.

No segundo passo, uma observadora vai preferir uma região de fontes de alimento de acordo com as informações sobre o néctar que foi distribuído pelas abelhas operárias na área de dança. Quanto maior for a quantidade de néctar de uma fonte, maior será a probabilidade de que essa fonte seja escolhida por uma observadora.

Na região selecionada, ela escolhe uma nova fonte na vizinhança da fonte em sua memória dependendo de informações visuais como no caso das abelhas operárias. A determinação da nova fonte de alimento é feita pelas abelhas com base no processo de comparação das posições das fontes visualmente.

No terceiro passo do ciclo, quando o néctar de uma fonte é abandonado pelas abelhas, uma nova fonte de alimento é aleatoriamente determinada por uma exploradora, substituindo a fonte que foi abandonada. Nos modelos considerados no presente estudo, a cada ciclo, no máximo uma abelha exploradora sai em busca de novas fontes de alimento para substituir uma fonte abandonada.

Estes três passos são repetidos por um número pré-determinado de ciclos chamado de MCN (do inglês, *Maximum Cycle Number*) ou até que um critério seja satisfeito.

Uma abelha observadora escolhe uma fonte de alimento dependendo do valor de probabilidade (P_i) associado com essa fonte, calculada de acordo com a Equação (4.1):

$$P_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_n} \quad (4.1)$$

Onde fit_i é o valor da função de aptidão da solução i que é proporcional à quantidade de néctar da fonte na posição i e SN é o número de fontes de alimento, que é igual ao número de operárias ou observadoras.

Para produzir uma nova posição de fonte candidata a partir de uma fonte já existente na memória de uma abelha, o ABC utiliza a Equação (4.2). Esse procedimento serve para explorar regiões na vizinhança de uma determinada posição de fonte.

$$v_{ij} = x_{ij} + \varphi_{ij} (x_{ij} - x_{kj}) \quad (4.2)$$

Onde $k \in \{1, 2, \dots, SN\}$ e $j \in \{1, 2, \dots, D\}$ são índices aleatoriamente escolhidos. Embora o valor de k seja determinado aleatoriamente, ele deve ser diferente de i .

$\varphi_{i,j}$ é um número aleatório entre $[-1, 1]$ ¹¹. Ele controla a produção de fontes vizinhas a x_{ij} e representa a comparação entre duas posições de fontes visualmente por uma abelha. A Figura 4.1 mostra a interpretação geométrica da criação de uma nova posição de fonte quando o parâmetro $\varphi_{i,j}$ varia entre $[-1, 1]$:

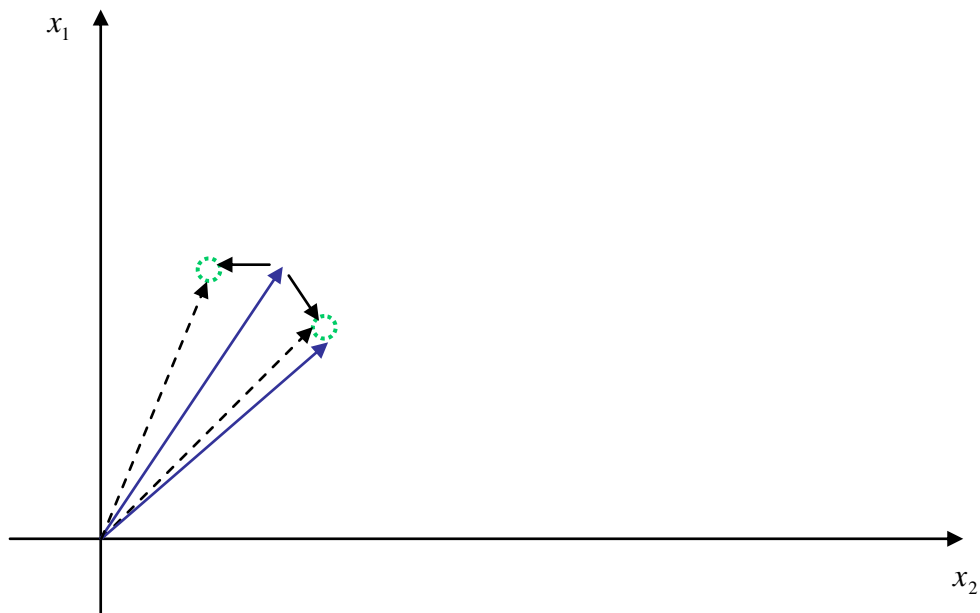


Figura 4.1. Exemplos de possíveis novas posições de fontes (setas tracejadas), considerando-se $\varphi_{i,j}$ entre $[-1, 1]$.

Como pode ser visto na Equação (4.2), conforme a diferença entre os parâmetros de x_{ij} e x_{kj} diminui, a distância entre as posições de fontes também diminui. Desta

¹¹ Valor previamente estabelecido pelo criador do algoritmo.

forma, conforme a busca se aproxima da solução ótima no espaço de busca, o comprimento do passo é adaptativamente reduzido.

A Figura 4.2 mostra a interpretação geométrica da criação de uma nova posição de fonte (novo candidato à solução) quando o parâmetro $\varphi_{i,j}$ varia entre $[0,1]$:

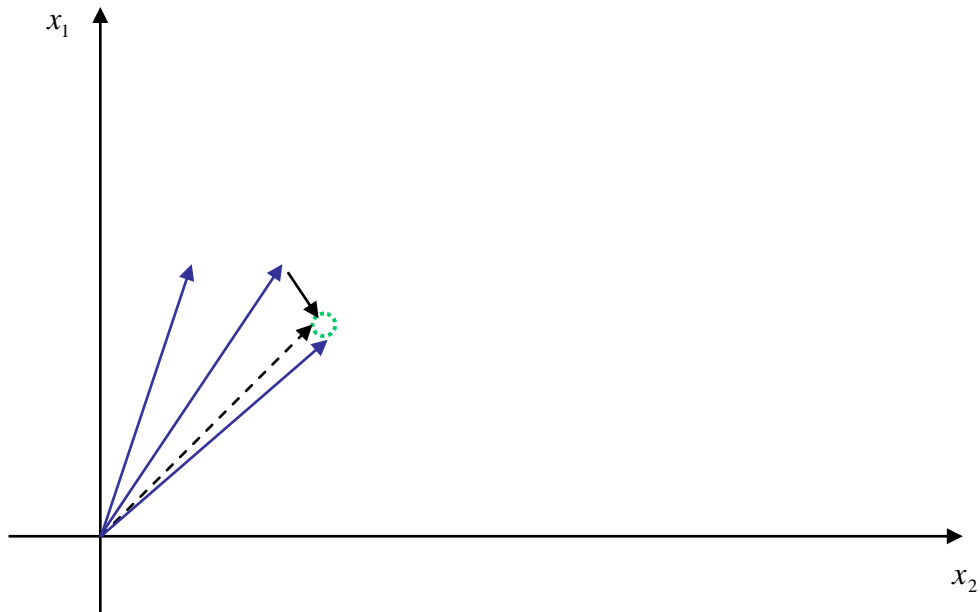


Figura 4.2. Exemplo de possível nova posição de fonte (seta tracejada)
(Considerando-se $\varphi_{i,j}$ entre $[0,1]$).

Se após a produção de uma nova posição de fonte candidata, esta estiver fora da região viável de busca, é possível efetuar a correção de modo aceitável. No presente estudo, impõe-se como limite as próprias restrições laterais das variáveis de otimização.

A fonte de alimento cujo néctar foi abandonado pelas abelhas é substituída por uma nova fonte de alimento descoberta pelas abelhas exploradoras. No algoritmo ABC, isto é simulado através da produção aleatória de uma posição de fonte e substituindo a fonte abandonada por essa nova fonte. No algoritmo ABC, caso uma fonte (possível solução) não melhore (valor da fitness) através de um número pré-determinado de ciclos, então, essa fonte é abandonada. Este valor pré-determinado de ciclos é um importante parâmetro de controle do algoritmo ABC, o qual é chamado “*limite*” para abandono.

Assumindo-se que a fonte abandonada seja x_i e $j \in \{1, 2, \dots, D\}$, então a abelha exploradora descobre uma nova fonte de alimento para ser substituída por x_j . Ou seja, a abelha exploradora busca uma nova posição de fonte de forma aleatória que,

eventualmente, possa representar uma melhor solução para o problema, evitando assim, a estagnação do algoritmo. Esta operação é definida de acordo com a Equação (4.3):

$$x_{i,j} = \min_j + \text{rand}(0,1) * (\max_j - \min_j) \quad (4.3)$$

Depois que cada posição de fonte candidata v_{ij} é produzida e avaliada pela abelha, seu néctar é comparado com o néctar da fonte anterior. Se o néctar da nova fonte for igual ou maior que o néctar da fonte anterior, então a nova fonte substitui a anterior na memória da abelha. Caso contrário, a fonte anterior é retida na memória da abelha.

Em outras palavras, um mecanismo de seleção gulosa é empregado como operação de seleção entre a fonte antiga e a fonte candidata.

No total, o ABC emprega quatro processos de seleção diferentes:

1. Um processo de seleção probabilístico global, no qual o valor de probabilidade é dado pela equação (4.1), usado pelas observadoras para a descoberta de regiões promissoras;
2. Um processo de seleção probabilístico local realizado em uma região pelas abelhas operárias e observadoras dependendo de informações visuais, tais como: cor, forma e perfume das flores¹² para a determinação de uma fonte na vizinhança da fonte em sua memória como descrito pela Equação (4.2);
3. Uma seleção local chamada de processo de seleção gulosa realizado pelas abelhas operárias e observadoras, no qual: se a quantidade de néctar da fonte candidata seja maior que o da fonte atual, a abelha esquece a fonte atual e memoriza a posição da fonte candidata produzida pela Equação (4.2). Caso contrário, a abelha mantém a posição da fonte atual na memória;
4. Um processo de seleção aleatória realizado pelas exploradoras definido pela Equação (4.3).

¹² Abelhas não serão capazes de identificar o tipo de fonte de néctar se não chegarem à localização exata da fonte e discriminarem seu cheiro.

No caso de abelhas melíferas, a taxa de recrutamento representa uma medida de quão rapidamente a colônia de abelhas encontra e explora uma nova fonte descoberta. O recrutamento artificial pode similarmente representar a medida da velocidade com que soluções possíveis ou soluções de boa qualidade para os difíceis problemas de otimização possam ser descobertas.

A sobrevivência e progresso da colônia de abelhas dependem da descoberta rápida e da eficiente utilização dos recursos alimentares. Similarmente, a descoberta de boas soluções para os difíceis problemas de otimização está diretamente ligada à descoberta relativamente rápida de boas soluções, principalmente para os problemas que precisam ser solucionados em tempo real.

O pseudocódigo do algoritmo ABC é apresentado a seguir:

Fase das Abelhas Operárias:

1. Inicialize a população de soluções ($x_{i,j}$ (*operárias*)), utilizando a Equação (4.3):

$$x_{i,j} = \min_j + \text{rand}(0,1) * (\max_j - \min_j) \quad (4.3)$$

onde ($i = 1, \dots, SN$) e ($j = 1, \dots, D$)

2. Avalie a população de soluções ($x_{i,j}$ (*operárias*))
3. Ciclo = 1
4. Repetir os seguintes passos:
 5. Produza novas soluções (posições de fontes de alimento) ($v_{i,j}$ (*operárias*)) na

vizinhança de ($x_{i,j}$ (*operárias*)), utilizando a Equação (4.2) e as avalie.

$$v_{i,j} = (x_{i,j} (\text{operárias})) + \varphi_{i,j} ((x_{i,j} (\text{operárias})) - x_{k,j}) \quad (4.2)$$

onde k é uma solução na vizinhança de i , e φ é um número aleatório no intervalo $[-1,1]$.

6. Aplique o processo de seleção gulosa entre $(x_{i,j}(\text{operárias}))$ e $(v_{i,j}(\text{operárias}))$. A melhor delas vence a competição e toma o lugar como a solução $(x_{i,j}(\text{operárias}))$ que servirá como base para o processo de seleção por roleta (*Roulette Wheel Selection*) (GOLDBERG, 1989), a fim de produzir cópias dos melhores indivíduos para a fase das abelhas observadoras.

Fase das Abelhas Observadoras:

7. Calcule os valores das probabilidades P_i para as soluções $(x_{i,j}(\text{operárias}))$ por meio dos valores de suas *fitnesses* a fim de produzir cópias dos melhores indivíduos de $(x_{i,j}(\text{operárias}))$ utilizando a Equação (4.1):

$$P_i = \frac{fit_i}{\sum_{i=1}^{SN} fit_i} \quad (4.1)$$

As cópias dos melhores indivíduos obtidos a partir de $(x_{i,j}(\text{operárias}))$ irão formar a população transitória $(x_{i,j}(\text{observadoras}))$.

8. Do mesmo modo como foi feito para $(x_{i,j}(\text{operárias}))$ no passo 5, produza novas soluções (posições de fontes de alimentos) $(v_{i,j}(\text{observadoras}))$ na vizinhança de $(x_{i,j}(\text{observadoras}))$ utilizando a Equação (4.2) e as avalie.
9. Aplique o processo de seleção gulosa entre as soluções $(v_{i,j}(\text{observadoras}))$ e as soluções $(x_{i,j}(\text{operárias}))$ que foram obtidas no passo 6. A melhor delas

vence a competição e toma o lugar como a população de soluções $(x_{i,j}(\text{operárias}))$.

Fase das Abelhas Exploradoras:

10. Determine as soluções abandonadas (fontes de alimento), caso existam, e as substituam por novas fontes produzidas aleatoriamente $(x_{i,j}(\text{exploradoras}))$ para as abelhas exploradoras utilizando a Equação (4.3). (Uma vez criadas, $(x_{i,j}(\text{exploradoras}))$ tornar-se-ão outras soluções (posições de fontes) $(x_{i,j}(\text{operárias}))$).
11. Memorize a melhor posição de fonte de alimento (possível solução) já encontrada.
12. Ciclo = Ciclo + 1.
13. Até Ciclo = MCN (*Maximum Cycle Number*).

A Figura 4.3 mostra a interpretação geométrica das fases do algoritmo ABC que foram descritas no pseudocódigo.

Fase das Abelhas Operárias:

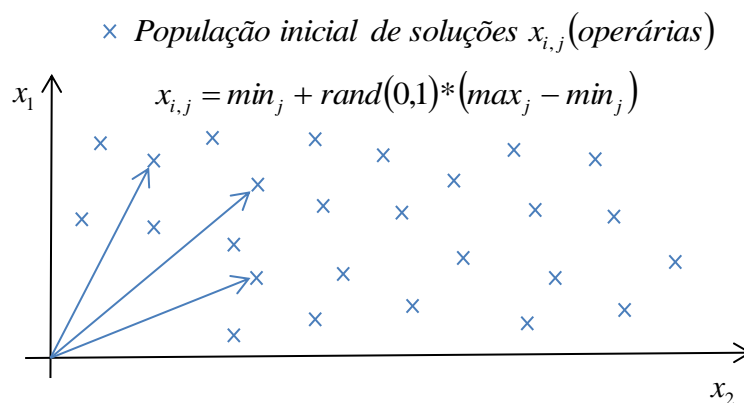


Figura 4.3 (a). Inicialização da população inicial de soluções $x_{i,j}(\text{operárias})$.

Fase das Operárias (continuação):

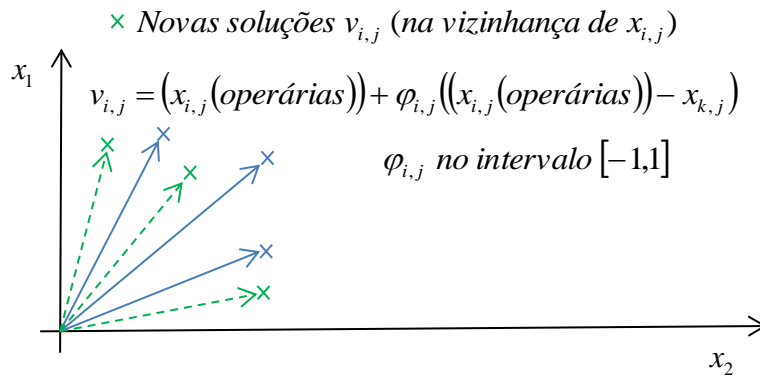


Figura 4.3 (b). Produção de novas soluções $v_{i,j}$ (operárias).

Processo de Seleção Gulosa (*Greedy Selection*) entre $x_{i,j}$ (operárias) e $v_{i,j}$ (operárias): Se: $\text{fit}_{(x_{i,j} (\text{operárias}))} > \text{fit}_{(v_{i,j} (\text{operárias}))}$, então: $x_{i,j} (\text{operárias}) = x_{i,j} (\text{operárias})$. Caso contrário, $v_{i,j} (\text{operárias}) = x_{i,j} (\text{operárias})$.

Supondo que: $\text{fit}_{(x_{i,j} (\text{operárias}))} > \text{fit}_{(v_{i,j} (\text{operárias}))}$, tem-se:

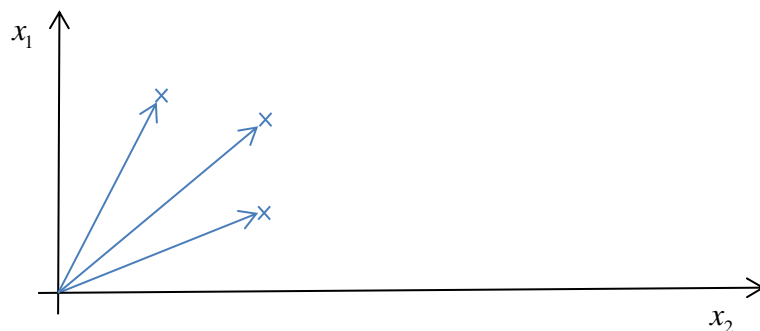


Figura 4.3 (c). Soluções $x_{i,j}$ (operárias) vencem a competição e continuam como as soluções $x_{i,j}$ (operárias).

Fase das Abelhas Observadoras:

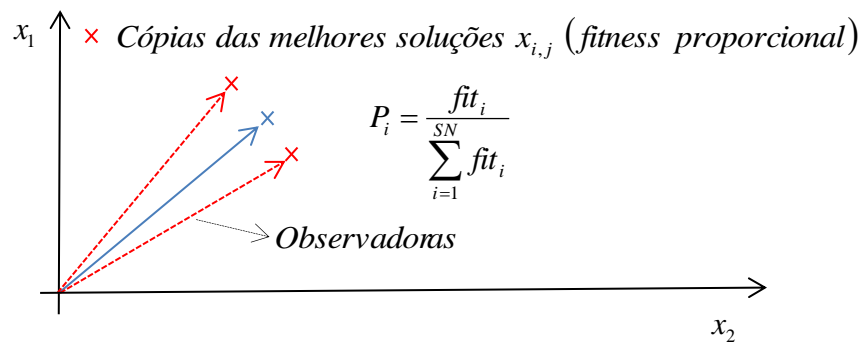


Figura 4.3 (d). Formação da população transitória $x_{i,j}$ (observadoras).

Fase das Observadoras (continuação):

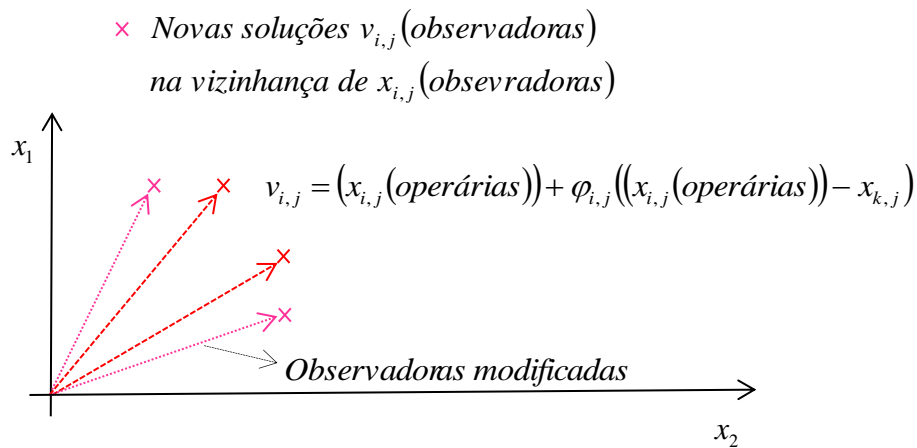


Figura 4.3 (e). Produção de novas soluções $v_{i,j}$ (*observadoras*).

Processo de Seleção Gulosa (*Greedy Selection*) entre $v_{i,j}$ (*observadoras*) e $x_{i,j}$ (*operárias*) obtidas na Figura 4.3 (c):

- Se: $\text{fit}(v_{i,j}(\text{observadoras})) > \text{fit}(x_{i,j}(\text{operárias}))$, então:

$v_{i,j}$ (*observadoras*) = $x_{i,j}$ (*operárias*).

Caso contrário, $x_{i,j}$ (*operárias*) = $x_{i,j}$ (*operárias*).

Supondo que: $\text{fit}(v_{i,j}(\text{observadoras})) > \text{fit}(x_{i,j}(\text{operárias}))$, tem-se:

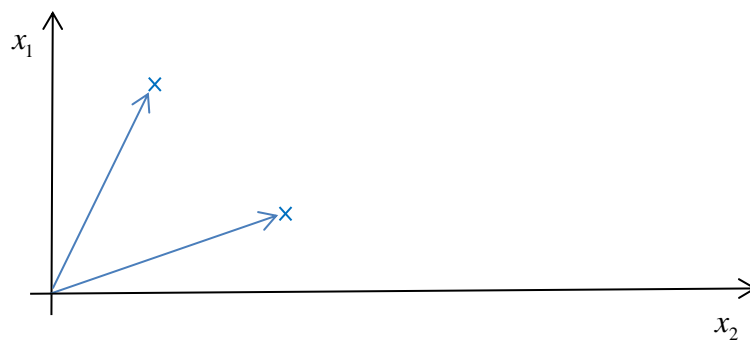


Figura 4.3 (f). Soluções $v_{i,j}$ (*observadoras*) vencem a competição e tomam o lugar como as soluções $x_{i,j}$ (*operárias*) definitivas.

Tal como anteriormente mencionado, exceto por parâmetros comuns (*tamanho da população e número máximo de ciclos*), o algoritmo ABC original utiliza um único parâmetro de controle, o qual é chamado *limite*.

Uma fonte de alimento terá sido exaurida e não mais será explorada quando o *limite* for excedido para essa fonte. Isto significa que a solução para a qual o número de avaliações excederem o valor *limite* não poderá mais ser melhorada.

Uma relação é definida para o valor do limite utilizando-se a dimensão do problema (D) e o número de fontes de alimento (SN), como mostra a Equação (4.4).

$$limite = (SN * D) \quad (4.4)$$

4.4. O Algoritmo ABCRK

Pelo fato do POR ser um problema combinatório, algumas modificações foram realizadas no algoritmo ABC original para torná-lo adequado a solução desse tipo de problema. A principal diferença entre a versão original do ABC e o modelo a ser aplicado ao POR, é que o modelo a ser aplicado ao POR utiliza o modelo de Chaves Aleatórias (*Random Keys, RK*) (BEAN, 1994).

O modelo de Chaves Aleatórias (*Random Keys, RK*) (BEAN, 1994), mapeia um vetor composto de números reais em uma solução contendo números inteiros não repetidos. Assim, soluções podem ser formadas em um problema de otimização, por exemplo.

Para exemplificar o princípio de funcionamento do modelo de Chaves Aleatórias, toma-se como exemplo um vetor $A = [0,39; 0,12; 0,54; 0,98; 0,41]$. O mapeamento desse vetor pelo modelo de Chaves Aleatórias resulta no vetor $A^* = [2, 1, 5, 3, 4]$.

Observe que o menor valor de A (0,12) aparece na posição 2. Isso significa que o primeiro valor do vetor A^* será 2. O segundo menor valor do vetor A (0,39) aparece na posição 1, logo, o segundo valor do vetor A^* será 1, e assim por diante.

Como será visto no próximo capítulo, repetições não são permitidas no POR, uma vez que um elemento combustível não pode ocupar duas posições no núcleo. Por

esta razão, foram utilizadas as posições das fontes $(x_{i,j})$ como chaves para gerar soluções possíveis, como mostra a Figura 4.4.

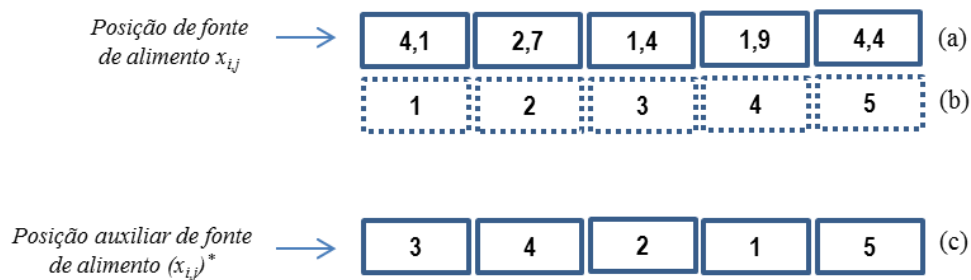


Figura 4.4. (a) Vetor de posições de fontes para um espaço de busca de 5 dimensões. (b) Novas posições como chaves reais. (c) Vetor auxiliar indicando uma solução válida, a qual será avaliada pela função objetivo do problema.

No algoritmo ABCRK o vetor de posições de fontes é decodificado pelas Chaves Aleatórias fornecendo uma solução válida para ser avaliada pela função objetivo. Desta forma, obtêm-se um espaço de busca com soluções válidas para o POR a partir de um espaço contínuo de busca de chaves. Deste modo, uma busca em espaço real contínuo pode gerar soluções possíveis para problemas combinatórios.

4.5. O Algoritmo DBABC

Na versão binária discreta, os vetores que representam as populações de candidatos à solução, são formados por zeros ou uns (*strings binárias*) e todos os pontos do espaço de busca (espaço de soluções) são possíveis de serem representados.

Esta particular característica da versão binária discreta permite que qualquer problema (combinatório, discreto ou até mesmo contínuo) possa ser tratado, sendo esta a principal vantagem do algoritmo DBABC proposto em relação ao algoritmo ABCRK e à versão original do algoritmo ABC.

Mudanças nas direções dos vetores da população podem ser representadas como probabilidades de que um bit assumirá o valor 0 ou o valor 1, e isto é feito por meio de uma transformação sigmoïdal da mesma forma como foi feito para a versão discreta do

algoritmo PSO proposto por KENNEDY e EBERHART (1995). A Figura 4.5 mostra o gráfico de uma função sigmoide.

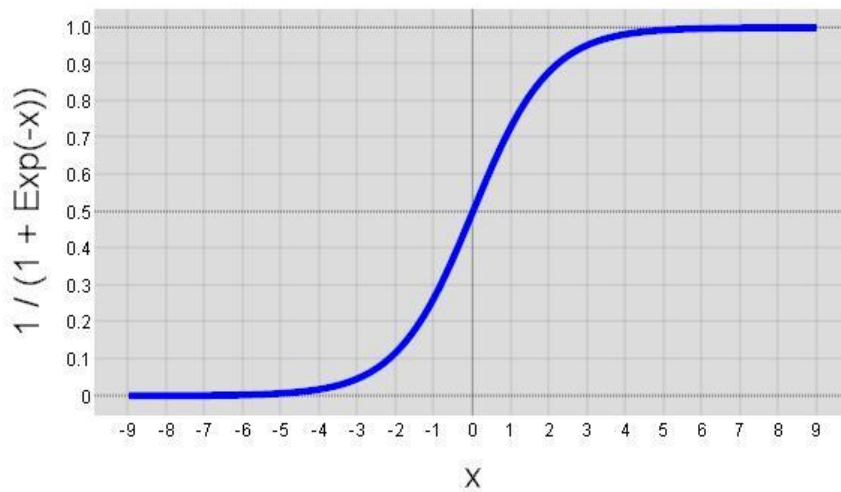


Figura 4.5. Gráfico de uma função Sigmoide.

O conceito de vizinhança passa a ser definido pelo número de bits trocados por iteração ou Distância de *Hamming* (GOLDBERG, 1989). Ambos os algoritmos são bastante similares, com algumas adaptações.

Essas afirmações podem ser melhor explicadas, conforme a seguir:

Em espaços de busca contínuos, as mudanças nas direções dos vetores da população são definidas como mudanças nas posições em algum número de dimensões.

No algoritmo ABC original, mudanças nas direções dos vetores da população são definidas por meio da alteração de apenas uma componente de x_i ($i = 1, 2, \dots, SN$), representada na Equação (4.2) pela variável k , onde $k \in \{1, 2, \dots, SN\}$.

Essas mudanças, na realidade, definem um conceito de vizinhança uma vez que definem um novo ponto (solução) na vizinhança.

Em espaços binários, mudanças nas direções dos vetores da população podem ser entendidas como movimentos em direção aos vértices de um hipercubo pela troca de vários números de bits. Deste modo, essas mudanças podem ser descritas pelo número de bits trocados por iteração, ou, em outras palavras, a Distância de *Hamming*. Assim,

um vetor da população com zero bits trocados não muda sua direção, ao passo que mudaria completamente sua posição se tivesse todos os seus bits trocados.

Na versão discreta proposta nesta tese, essas considerações foram feitas através da utilização de um parâmetro chamado NBTV (Número de Bits Trocados para Vizinhança), que representa o número de bits trocados para vizinhança a fim de controlar a taxa de troca dos bits em cada *string* dos vetores binários da população.

Em nosso modelo, os vetores binários mudam de direção em um espaço de busca restrito a zeros e uns em cada dimensão.

A pergunta imediata torna-se então: como representar a probabilidade de um bit assumir o valor 0 ou o valor 1?

A resposta para essa pergunta está relacionada com o fato de que ambos os algoritmos ABC (versão original) quanto esta versão binária discreta que está sendo proposta (DBABC), são bastante similares. Logo, existe a necessidade de que seja definido o que deverá resultar da diferença entre $(x_{i,j}(\text{operárias}))$ e $x_{k,j}$ expressa na Equação (4.2) (vide pseudocódigo).

De fato, essa diferença pode ser razoavelmente calculada como -1, 0 ou +1, tornando-se inadequados esses resultados considerando-se uma abordagem binária, onde apenas são permitidos zeros e uns como resultado.

Mais especificamente, no algoritmo DBABC, a fórmula (Equação (4.2) (vide pseudocódigo)):

$$v_{i,j} = (x_{i,j}(\text{operárias})) + \varphi_{i,j} ((x_{i,j}(\text{operárias})) - x_{k,j}),$$

Deve ser reescrita como:

$$\begin{aligned} \text{se } S(m) < 0.5, \quad \text{então, } v_{i,j} &= 0 \\ \text{senão, } v_{i,j} &= 1 \end{aligned}$$

Que é semelhante à expressão proposta por (KENNEDY e EBERHART, 1997) para a versão discreta do algoritmo PSO (KENNEDY e EBERHART, 1995).

Neste caso, $S(m)$ é uma transformação *sigmoidal* limitante e os limites assintóticos vão de 0,9975 a 0,0025 (vide Figura 4.5). Por esta razão, o valor de m deve

estar no intervalo entre $[-6, 6]$, onde, $m = \varphi_{i,j} ((x_{i,j}(\text{operárias})) - x_{k,j})$. Adicionalmente, como $S(m)$ pode ser escrita como uma probabilidade, deve estar então restrita ao intervalo $[0, 1]$. Uma vez que a diferença $((x_{i,j}(\text{operárias})) - x_{k,j})$ deve resultar em valores tais como $-1, 0$ e $+1$, φ deve ser definido como: $\varphi = \text{rand}() * 6$, a fim de ser capaz de definir uma vizinhança em torno de $(x_{k,j})$.

4.6. Sumário do Capítulo

Este capítulo foi dedicado à apresentação das metodologias propostas neste trabalho de tese. Foram introduzidos: o algoritmo ABC original, os princípios biológicos que motivaram o seu desenvolvimento, bem como suas principais características. Também foram apresentados: o algoritmo ABCRK e o algoritmo DBABC. Ambos serão aplicados na otimização da recarga do ciclo 7 de operação da CNA1, cujos detalhes da modelagem encontram-se descritos no próximo capítulo.

Capítulo 5

– O Problema de Otimização de Recargas –

5.1. Introdução

Este capítulo é destinado à apresentação dos aspectos básicos relacionados ao Problema de Otimização de Recargas (POR) de Reatores Nucleares a Água Pressurizada (RNAP). São apresentados seus aspectos teóricos, sua formulação matemática, bem como sua relação com os parâmetros de otimização fornecidos pelo código RECNOB (CHAPOT, 2000), o qual é o código de Física de Reatores utilizado neste trabalho. Ênfase é dada às particularidades do problema no que diz respeito ao ciclo 7 de operação da CNA1. Antes da descrição detalhada da modelagem proposta neste capítulo, são introduzidos alguns conceitos básicos relacionados com o funcionamento dos reatores nucleares com o objetivo de facilitar o entendimento dos principais aspectos relacionados ao problema.

5.2. O Problema da Otimização de Recargas

Em linhas gerais, as usinas nucleares são semelhantes às usinas termelétricas convencionais em ciclo simples. Basicamente, o processo envolve o aquecimento de vapor a alta pressão que, ao expandir-se, aciona uma turbina térmica, movimentando um gerador elétrico acoplado a seu eixo. O vapor que sai da turbina, em baixa pressão, é resfriado, condensado, e novamente aquecido, reiniciando o ciclo térmico de *Rankine*.

As principais diferenças entre usinas termelétricas convencionais e termonucleares estão no combustível utilizado e na forma de aquecimento e vaporização da água.

As termelétricas produzem energia elétrica a partir da queima de carvão, óleo combustível ou gás natural em uma caldeira projetada para essa finalidade específica. O funcionamento das centrais termelétricas é semelhante, independentemente do combustível usado. Na caldeira, o combustível é queimado, gerando vapor a partir da água que circula por uma extensa rede de tubos que revestem suas paredes. A função do

vapor é movimentar as pás de uma turbina, cujo rotor gira juntamente com o eixo de um gerador que produz a energia elétrica. Essa energia é transportada por linhas de alta tensão aos centros de consumo. O vapor é resfriado em um condensador e convertido outra vez em água, que volta aos tubos da caldeira, dando início a um novo ciclo.

Já as usinas termonucleares, utilizam material físsil como combustível. Em particular, a CNA1, utiliza como combustível o material físsil urânio-235 (^{235}U). O coração de uma usina nuclear é o núcleo do reator, gerando energia a partir das reações de fissão dos átomos de (^{235}U), iniciada com o bombardeio de nêutrons que irá produzir as reações em cadeia.

Há dois elementos essenciais em uma reação em cadeia controlada: o combustível e o refrigerante. O calor absorvido pelo refrigerante pode acionar diretamente uma turbina a vapor ou trocar calor com um circuito secundário, que fará este serviço.

No reator da CNA1, a água é utilizada tanto para o moderador quanto para o refrigerante. A partir, portanto, do circuito secundário de água, a usina não tem componente radioativos, sendo equivalente a uma usina térmica convencional.

Em um RNAP como o da CNA1, após o combustível ser carregado no núcleo, dá-se início às reações em cadeia por meio do bombardeio de nêutrons. O reator é então operado por um longo período de tempo (normalmente, um ano), denominado ciclo de operação. Durante esse período, o reator em condições normais, produz a energia requerida. Em contrapartida, a concentração do material físsil (^{235}U) começa a decrescer.

Após o período de operação chamado ciclo, ou sempre que não for mais possível manter o reator crítico, operando à potência nominal, o mesmo precisa ser reabastecido. Em ciclos anuais, cerca de um terço dos EC mais queimados são substituídos por EC novos.

Uma característica comum aos RNAP é a necessidade de desligamento para o recarregamento. Uma vez desligado o reator, os EC são então descarregados do núcleo e armazenados em uma piscina de combustível usado (PCU). Os elementos com as mais baixas concentrações de (^{235}U), ou seja, os mais queimados, são mantidos em definitivo na PCU, sendo substituídos por EC novos. Os EC novos, juntamente com os EC remanescentes de ciclos anteriores que foram parcialmente queimados, mas que ainda assim, permanecem úteis, irão compor o núcleo subsequente.

Neste contexto, o POR consiste na determinação do melhor posicionamento possível para os diferentes EC no núcleo do reator, de modo que, respeitando-se restrições operacionais e de segurança, além de regras de simetria existentes no núcleo do reator, o ciclo de operação seguinte possa ser otimizado. A meta principal é a máxima utilização do combustível dentro das margens de segurança, a fim de que se possa reduzir os custos com a geração de energia.

Considerado como um problema de otimização de ordem combinatória extremamente complexo, o POR pertence à classe dos problemas NP-Completo, isto é, não se conhece algoritmo polinomial que o resolva, muito embora isso não signifique que tal algoritmo não exista. Além disso, a complexidade desse problema cresce exponencialmente de acordo com o número de EC a serem combinados no núcleo do reator. Tal como mencionado no primeiro capítulo deste estudo, o núcleo do reator da CNA1, por exemplo, que é constituído por 121 EC, é capaz de gerar um número aproximado de $8,09 \times 10^{200}$ PC distintos. Todavia, como será visto mais adiante neste capítulo, existem regras de simetria de 1/4 e de 1/8 de núcleo, além de regras de colocação para os EC, que fazem com que este número caia para aproximadamente, 10^{25} PC possíveis. Há de se notar que este número é ainda extremamente alto e que, por conseguinte, uma busca exaustiva não seria uma solução razoável para este problema específico.

Outra característica importante do POR é que, por ser um problema multiobjetivo, sua solução é dependente do tipo de objetivo a ser alcançado. Dentre os objetivos mais comuns, destacam-se a maximização da duração do ciclo e a minimização do fator de pico de potência radial do núcleo (F_{XY}).

Quando o objetivo é a maximização da duração (ou comprimento) do ciclo, utiliza-se um esquema de carregamento conhecido como baixa fuga. Neste esquema, os EC novos são inseridos nas posições mais internas do núcleo, enquanto os parcialmente queimados são colocados na periferia. Desta forma, a fuga de nêutrons do núcleo para o vaso do reator é então reduzida, o que leva a um maior excesso de reatividade. Além de aumentar o comprimento do ciclo, esquemas do tipo baixa fuga também contribuem para aumentar a durabilidade do vaso do reator, cujo material é o aço inoxidável.

Quando o objetivo é a minimização do fator de pico de potência radial, utiliza-se um esquema de carregamento do tipo *out-in*, o que implica, no caso da CNA1, na introdução de 40 EC novos (cerca de um terço do núcleo), os quais deverão ser posicionados na periferia do núcleo do reator.

O comportamento altamente não linear dos reatores nucleares gera funções objetivos e restrições com características não lineares. Conseqüentemente, um grande número de pontos de mínimo ou de máximos locais irá existir (GALPERIN, 1995). Isso significa que qualquer que seja a técnica utilizada para a solução do POR, esta deverá apresentar bom desempenho em espaços de busca multimodais. Tais características, aliadas ao grande número de possíveis combinações e ao custo associado com a avaliação da função objetivo, tornam o POR um grande desafio para muitos métodos de otimização. Diversos estudos têm sido desenvolvidos nesta área com considerável sucesso. No entanto, soluções cada vez mais eficientes para o POR, ainda estão em urgente demanda.

5.3. As Regras aprendidas ao Longo dos Anos

Durante os mais de 50 anos em que os reatores nucleares operaram para a geração de energia, muitas regras para se organizar os EC no núcleo do reator foram aprendidas. Além da otimização manual realizada por especialistas, métodos baseados em conhecimento, algoritmos de otimização e metaheurísticas, têm sido amplamente utilizadas para a solução deste problema.

No entanto, seja qual for o método adotado, o especialista ou otimizador que se propõe a solucionar este problema poderá ter diversos (e muitas vezes conflitantes) objetivos para alcançar, como, por exemplo:

- i. Limitar a magnitude de todo e qualquer pico de potência local ao valor limite das Especificações Técnicas;
- ii. Limitar a fuga radial de nêutrons do núcleo para o vaso de pressão, de modo a reduzir os danos por radiação;
- iii. Maximizar a queima média acumulada por combustível;
- iv. Maximizar o excesso de reatividade do núcleo de modo geral (dependente do tipo de esquema de carregamento), a fim de reduzir a necessidade de EC novos.

Ao longo dos anos, algumas regras foram sendo observadas:

- i. Configurações de núcleo (ou padrões de carregamento – PC) nos quais elementos de um mesmo tipo são agrupados juntos causam diversos problemas de picos de potência, de modo que uma redistribuição do tipo “tabuleiro de xadrez” se torna essencial;
- ii. Configurações de núcleo com EC novos no interior do núcleo fornecem a mais baixa fuga de nêutrons, porém, à custa de problemas de picos de potência;
- iii. Configurações de núcleo com EC novos na periferia do núcleo fornecem um perfil de potência mais “achatado”, porém, à custa de uma alta fuga de nêutrons;
- iv. Configurações de núcleo que utilizam esquemas do tipo baixa-fuga tendem a reduzir a quantidade de EC novos requeridos, ou, de modo equivalente, aumentam a duração do ciclo de queima (comprimento do ciclo).

De acordo com as necessidades da planta, das características de cada ciclo e demanda planejada de energia, o especialista ou otimizador pode decidir pela melhor estratégia a ser utilizada.

5.4. O Problema de Otimização de Recargas no Contexto da CNA1

A fim de demonstrar o desempenho das metodologias propostas neste estudo, o ciclo 7 de operação da CNA1 é tomado como referência, embora a utilização das metodologias aqui propostas não estejam limitadas a este ciclo, nem tampouco unicamente à CNA1, sendo válidas para qualquer outra planta e ciclo, com algumas modificações pertinentes a cada caso em particular.

A opção pelo ciclo 7 de operação da CNA1 neste estudo, teve como motivação a possibilidade de comparação com trabalhos anteriores, tais como (MACHADO, 1999), (MACHADO, 2005) e (MENESES, 2010). Uma descrição detalhada das principais características deste ciclo pode ser encontrada em (CHAPOT, 2000).

A CNA1 dispõe de um RNAP cujo núcleo é constituído por 121 EC. Uma característica importante deste núcleo é a existência de eixos de simetria. A Figura 5.1 mostra a representação da vista superior do núcleo deste reator com seus referidos eixos de simetria.

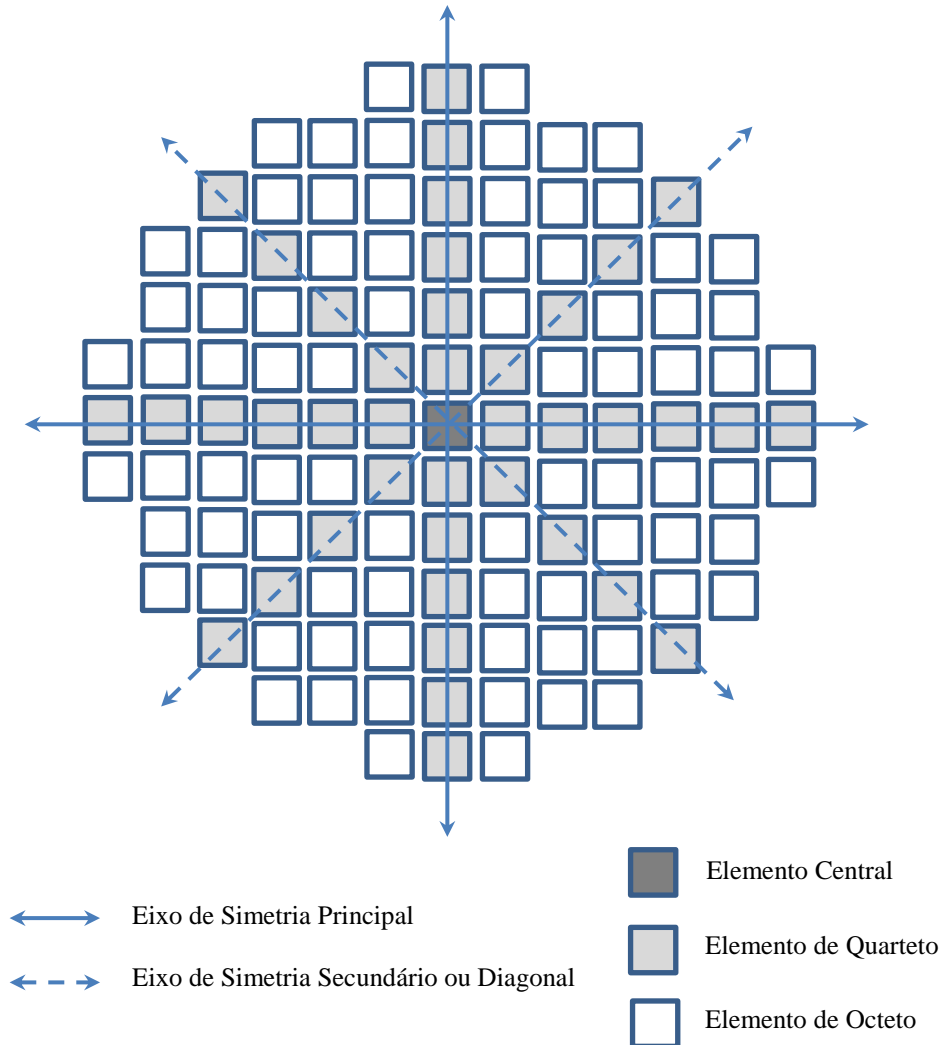


Figura 5.1. Vista superior do núcleo do RNAP da CNA1.

Conforme mencionado no Capítulo 1, se fosse considerado na sua totalidade, ou seja, se todos os 121 EC fossem considerados, tal modelo forneceria um número aproximado de $121!$ ($\approx 8,09 \times 10^{200}$) possíveis soluções. No entanto, pelo fato da distribuição de potência no núcleo do reator apresentar uma relação de simetria nas regiões delimitadas pelos quadrantes demarcados pelos eixos mostrados na Figura 5.1, esta característica pode ser utilizada para reduzir a complexidade do POR. Com base no exposto, modelos mais simples do núcleo do reator podem ser considerados e o número de EC a serem combinados pode ser reduzido. Em outras palavras, isto significa que é

possível a utilização de apenas um dos quadrantes do núcleo (simetria de 1/4) ou de um dos octantes do núcleo (simetria de 1/8) e depois, utilizando as regras de rebatimento do núcleo, compor a configuração completa do mesmo. Nestes casos, o número de possíveis soluções pode ser reduzido, ainda que, mesmo com essas simplificações, o problema continue sendo considerado extremamente complexo.

Como pode ser visto na Figura 5.1, existem dois eixos de simetria principais (linhas contínuas) dividindo o núcleo do reator em quatro regiões. Este modelo representa o que é conhecido como simetria de 1/4 de núcleo ou simetria de quarteto. Esses dois eixos principais, juntamente com os dois eixos secundários ou diagonais (linha tracejada), dividem o núcleo em oito regiões, formando o que se conhece por simetria de 1/8 de núcleo ou simetria de octeto. A Figura 5.2 mostra em destaque (vista superior), a simetria de octeto (presente também na Figura 5.1), o qual é o modelo adotado no presente estudo.

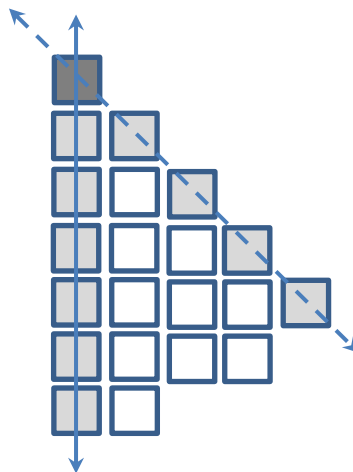


Figura 5.2. Vista superior da simetria de 1/8 de núcleo (ou simetria de octeto) do núcleo do RNAP da CNA1.

A simetria de octeto representada em destaque na Figura 5.2, conta com 21 EC:

- 10 em ambos os eixos de simetria, chamados de elementos de quarteto;
- 10 entre os eixos de simetria, chamados de elementos de octeto;
- 1 elemento central (fixo para fins de otimização), o qual não participa da permutação pelo fato de não haver para ele um correspondente simétrico.

Tal modelo reduz a complexidade do problema, uma vez que o número de EC a serem combinados cai de 121 (considerando-se o núcleo inteiro) para 20 (com o

elemento central fixo). Como mencionado, mesmo com essas simplificações, o problema continua apresentando alto grau de complexidade, fornecendo um número aproximado de $20! (\approx 2,4 \times 10^{18})$ possíveis soluções.

Tendo em vista que o objetivo considerado neste trabalho é a maximização do comprimento do ciclo operacional da planta de referência, ou, em outras palavras, a maximização da Concentração de Boro (C_B) ao final do ciclo 7 de operação da CNA1, as seguintes considerações foram feitas: a opção por um esquema de carregamento do tipo baixa-fuga; a não utilização de venenos queimáveis (fato justificado por ser uma característica particular do ciclo 7 de operação da CNA1) a fim de possibilitar a comparação com trabalhos anteriores; e a utilização do código de Física de Reatores RECNOd, o qual é um código nodal bidimensional, a dois grupos de energia, utilizado em (CHAPOT, 2000) e em muitos outros trabalhos que utilizam o ciclo 7 de operação da CNA1 como referência, tais como: (MACHADO e SCHIRRU, 2002), (CALDAS e SCHIRRU, 2008), (DE LIMA et al., 2008), (MENESES et al., 2009) e (WAINTRAUB, 2009).

Os parâmetros nucleares fornecidos pelo código RECNOd são, entre outros, a Máxima Potência Média Relativa (P_{rm}) e a Concentração de Boro (C_B). Ao contrário da maioria dos códigos comerciais de Física de Reatores, como por exemplo, o ANC (CHAPOT, 2000), o RECNOd não possui módulos de realimentação Termo Hidráulica e de reconstrução pino a pino, sendo esta, uma de suas maiores limitações. Por outro lado, o custo computacional e a avaliação de uma solução pelo RECNOd são menores, tendo como respaldo o fato de que a utilização da P_{rm} não viola as Especificações Técnicas da CNA1 (FSAR, 1980).

De acordo com CHAPOT (2000), para um Fator de Pico de Potência Radial $F_{XYmax} = 1,435$ para a CNA1, os cálculos correspondem a $P_{rm} = 1,395$. Deste modo, um PC que apresente $P_{rm} \leq 1,395$ estará dentro das exigências técnicas relativas à segurança. Caso contrário, o PC não poderá ser aceito como solução.

Vale ressaltar que o valor da C_B fornecido pela simulação através do RECNOd é dado no Equilíbrio do Xenônio, sendo esse outro aspecto importante para a redução do esforço computacional. (CHAPOT, 2000) demonstrou que é possível extrapolar e prever o ciclo de operação referente a um dado PC de modo que 4 ppm de Boro são equivalente a 1 Dia Efetivo à Plena Potência.

De acordo com o exposto nos parágrafos anteriores, a função objetivo utilizada neste trabalho (Equação 5.1) agrega dois objetivos para a solução do POR. São eles:

- A maximização da Concentração Crítica de Boro (C_B); e
- A minimização da Máxima Potência Média Relativa dos ECs (P_{rm}).

Formulada como a seguir:

$$\begin{aligned} & \text{Minimize} \quad \frac{1}{C_B} \\ & \text{Sujeito a} \quad P_{rm} \leq 1.395 \end{aligned}$$

Tem-se:

$$\text{Fitness} = \begin{cases} \frac{1}{C_B}, & \text{se } P_{rm} \leq 1.395 \\ P_{rm}, & \text{caso contrário} \end{cases} \quad (5.1)$$

5.5. O Algoritmo ABCRK Aplicado ao POR

O algoritmo ABCRK foi testado em um estudo de otimização da recarga do ciclo 7 de operação da CNA1, no qual foi adotado um esquema de carregamento do tipo baixa-fuga, mas, sem dispor de venenos queimáveis. Ao optar-se por uma estratégia do tipo baixa fuga, procura-se inserir os EC novos em posições mais internas do núcleo, enquanto os EC mais queimados são posicionados na periferia. Em decorrência disto, ocorrem elevados picos de potência no núcleo. Para manter os valores de F_{XY} abaixo do limite imposto pelas Especificações Técnicas (FSAR, 1980), são utilizados venenos queimáveis. Isto significa que a otimização de um ciclo sem a utilização de venenos queimáveis torna-se um desafio a mais para o otimizador.

A maximização do comprimento do ciclo implica na maximização da C_B de fim de ciclo. No entanto, para minimizar os custos computacionais, utilizou-se a C_B na segunda etapa de queima (xenônio em equilíbrio) (CHAPOT, 2000) como parâmetro a ser maximizado.

Neste estudo, o POR foi modelado para uma simetria de 1/8 de núcleo e somente o elemento central não foi incluído na codificação do problema. Isto porque, para fins de otimização, o elemento central é considerado fixo, ou seja, não participa do processo

da permutação, uma vez que não existe para ele um correspondente simétrico. Além disso, no modelo proposto neste estudo, elementos de quarteto só podem ser trocados com outros elementos de quarteto. Do mesmo modo, elementos de octeto só podem ser trocados com elementos de octeto. A Figura 5.3 mostra como esses EC são representados em separado no vetor de soluções.

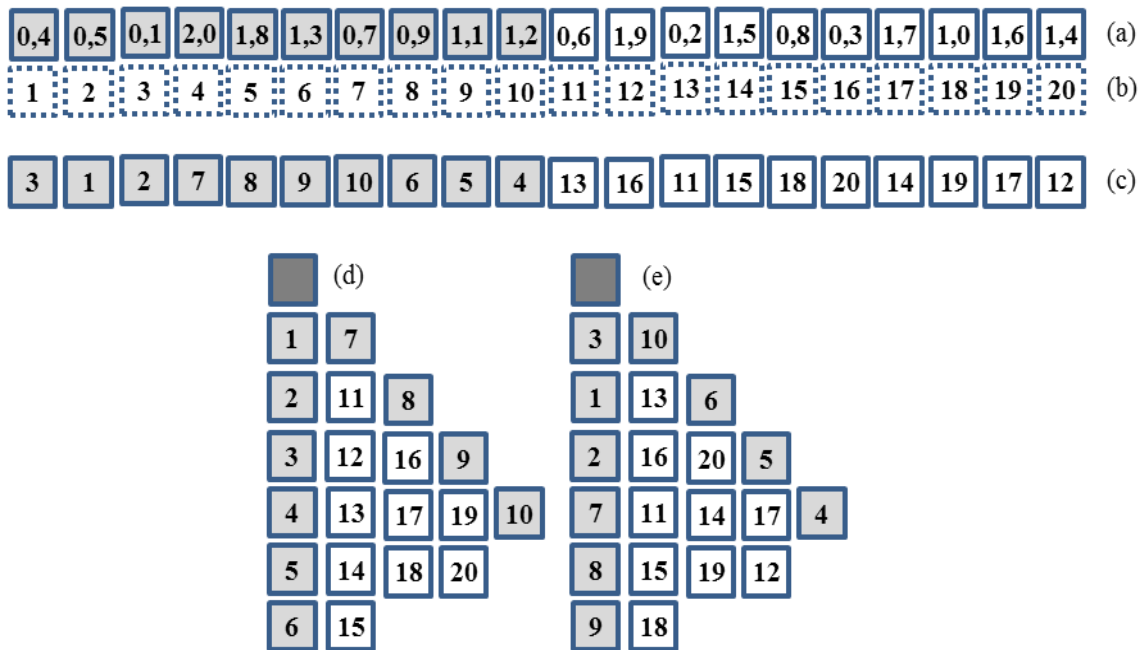


Figura 5.3. (a) Posição de uma fonte de alimento no ABCRK. (b) Codificação em chaves reais. (c) Codificação em Chaves Aleatórias. (d) Configuração base para a simetria de octeto. (e) O núcleo resultante a ser avaliado pelo código RECNOd.

Para o POR, o algoritmo ABCRK foi modelado com as abelhas tendo posições $x_{i,j}$ em um espaço de busca de 20 dimensões (simetria de octeto) como mostra a Figura 5.3. A população inicial é formada aleatoriamente, onde cada abelha artificial assume uma posição de fonte. A cada iteração do algoritmo, estas posições são modificadas de acordo com as estratégias de busca próprias do algoritmo, as quais foram descritas no Capítulo 4. As posições são utilizadas como chaves para o mapeamento das soluções candidatas, obtendo-se assim, um vetor auxiliar de soluções com 20 números de 1 a 20, sem repetições. Para se ter uma idéia da complexidade que vai ser tratada pelas metodologias propostas, vale ressaltar que em todas as simulações realizadas não serão permitidas permutações entre elementos de quarteto com elementos de octeto e vice

versa. Para a simetria de octeto adotada neste trabalho, isto significa um espaço de busca equivalente a $10! \times 8^{10} \times 10! \times 8^{10}$ para ambos os modelos propostos neste estudo, o que equivale a um número aproximado de $1,52 \times 10^{31}$ PCs distintos.

Cada posição de fonte de alimento corresponde a uma possível solução ou a um possível PC. A informação contida em uma posição de fonte de alimento configura um núcleo a ser avaliado pelo código de Física de Reatores RECNOB, o qual retorna C_B e P_{rm} para cada configuração de núcleo encontrada pelo algoritmo.

5.6. O Algoritmo DBABC Aplicado ao POR

No algoritmo DBABC, a população inicial é formada por uma *string* de bits e não mais por um vetor de números reais a serem codificados, como no caso do algoritmo ABCRk. A Figura 5.4 mostra a posição das abelhas em um espaço de busca de 20 dimensões equivalente à simetria de octeto.

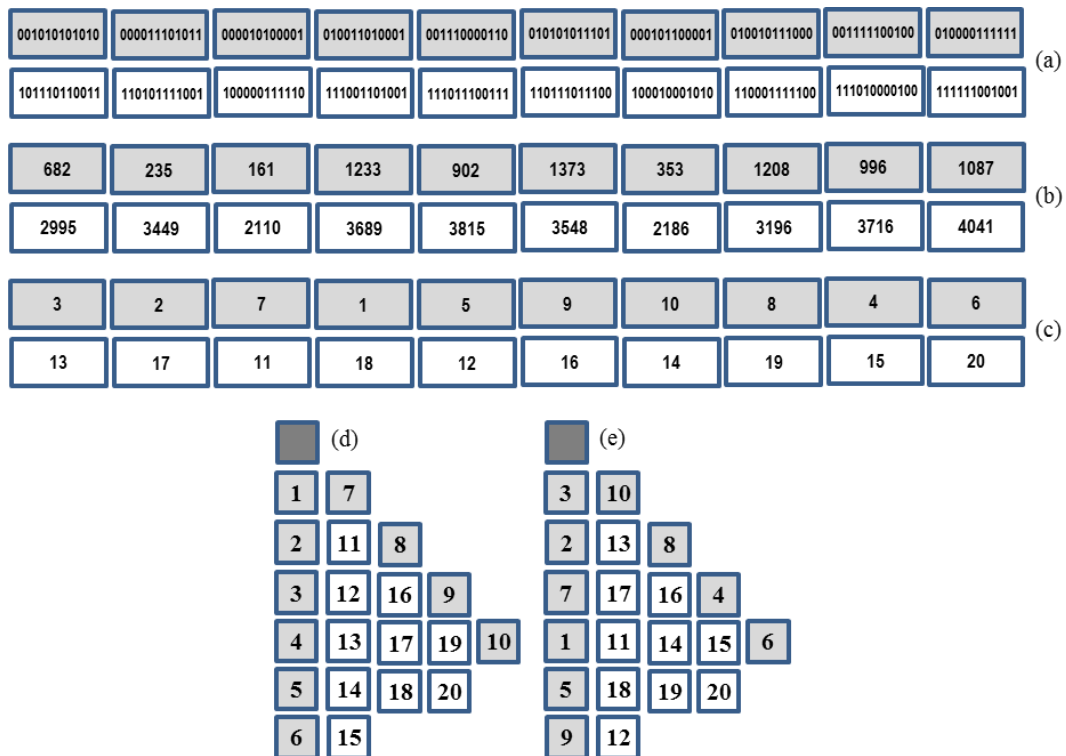


Figura 5.4. (a) Posição de uma fonte de alimento no DBABC. (b) Correspondente em decimal. (c) Codificação em Chaves Aleatórias. (d) Configuração base para a simetria de octeto. (e) O núcleo resultante a ser avaliado pelo código RECNOB.

No algoritmo DBABC, os vetores da população são formados por *strings* binárias (0s e 1s), sendo possível representar todos os pontos do espaço de busca. Esta particular característica do DBABC permite que qualquer problema (combinatório, contínuo ou discreto) possa ser tratado.

Mudanças nas direções dos vetores podem ser representadas como probabilidades de que um bit irá assumir o valor 0 ou o valor 1, e isto é feito por meio de uma função sigmoide (vide Figura 4.4).

O conceito de vizinhança pode ser definido pelo número de bits trocados por iteração, ou Distância de *Hamming* (GOLDBERG, 1989).

Tanto o ABC como o DBABC são muito semelhantes, apenas com algumas adaptações: Em espaços de busca contínuos, as mudanças nas direções dos vetores da população são definidas como mudanças nas posições em certo número de dimensões. Estas mudanças, na verdade, definem um conceito de vizinhança, uma vez que definem um novo ponto (ou solução) na vizinhança.

Em um espaço de busca binário, as mudanças nas direções dos vetores da população podem ser entendidas como movimentos de aproximação ou afastamento ao longo de um hipercubo por meio da troca de alguns bits. Deste modo, as mudanças podem ser descritas pelo número de bits trocados por iteração, ou, em outras palavras, a Distância de *Hamming* (GOLDBERG, 1989). Assim, um vetor da população com 0 bits trocados não muda sua direção, enquanto um vetor da população muda completamente sua direção por meio da troca de todas as suas coordenadas binárias.

No presente estudo, estas considerações foram levadas em conta por meio da utilização de um parâmetro chamado NBTV (Número de Bits Trocados para Vizinhança), a fim de controlar a taxa de troca de bits em cada string dos vetores binários da população.

No modelo, os vetores binários mudam de direção em um espaço de busca restrito a 0s e 1s em cada dimensão. O problema a ser resolvido, passa a ser, então: “Como representar a probabilidade de que um bit assumirá valor 0 ou valor 1?”

A solução está relacionada com o fato de que ambos os algoritmos ABC e DBABC, utilizam essencialmente as mesmas equações, apenas com algumas adaptações. A principal diferença entre esses algoritmos está no uso da seguinte equação (Equação 4.2):

$$v_{i,j} = (x_{i,j}(\text{operárias})) + \varphi_{i,j} ((x_{i,j}(\text{operárias})) - x_{k,j})$$

Uma vez que o resultado de $((x_{i,j}(\text{operárias})) - x_{k,j})$ pode resultar em valores tais como -1, 0 e 1, onde -1 seria um valor inadequado para uma codificação binária, faz-se necessário então que φ seja definido como: $\varphi = \text{rand}() * 6$, a fim de que se possa definir uma vizinha em torno de $x_{k,j}$.

Desse modo, $v_{i,j} = (x_{i,j}(\text{operárias})) + \varphi_{i,j}((x_{i,j}(\text{operárias})) - x_{k,j})$, passa a ser definida no algoritmo DBABC, como:

$$\begin{aligned} \text{se } S(m) < 0.5, \quad \text{então, } v_{i,j} &= 0 \\ \text{senão, } v_{i,j} &= 1 \end{aligned}$$

Sendo esta a principal diferença entre os dois algoritmos.

No algoritmo DBABC, $S(m)$ é uma transformação sigmoideal e os limites assintóticos de $S(m)$ são 0.9975 e 0.0025. Por esta razão, o valor de m deverá estar no intervalo entre $[-6,6]$, onde $m = \varphi_{i,j}((x_{i,j}(\text{operárias})) - x_{k,j})$. Adicionalmente, como neste caso, $S(m)$ representa agora uma probabilidade, seu valor deverá estar restrito ao intervalo $[0.0, 1.0]$.

Para o POR, o algoritmo DBABC foi modelado com o parâmetro NBTV igual a 12 em todos os experimentos computacionais realizados. Isto significa que todos os 12 bits das *strings* são trocados a cada iteração do algoritmo a fim de definir as mudanças nas posições das fontes. Esse valor foi escolhido por ter sido o que apresentou os melhores resultados em testes de análise de sensibilidade durante as fases de desenvolvimento das metodologias propostas.

5.7. Sumário do Capítulo

Neste capítulo foram apresentados os aspectos básicos relacionados ao Problema de Otimização de Recargas (POR) de Reatores Nucleares a Água Pressurizada (RNAP), especialmente contextualizados para o ciclo 7 de operação da CNA1. Também foram introduzidos alguns conceitos básicos relacionados com o funcionamento dos reatores nucleares. Finalmente, encerrou-se o capítulo com a apresentação da modelagem do problema para cada algoritmo proposto.

Capítulo 6

– Resultados –

6.1. Introdução

Neste capítulo, são apresentados os resultados obtidos com a aplicação das metodologias propostas para a otimização do POR da CNA1.

As simulações foram realizadas em ambiente MATLAB, e o tempo de processamento para o experimento de maior número de avaliações foi de pouco mais de 2 horas em notebook DELL – Windows 7 Home Premium, Modelo XPS L502X, Processador Intel ® Core™ i7-2670QM CPU 2.20GHz, com 8,00 GB de memória RAM e Sistema Operacional de 64 bits.

6.2. Resultados da Aplicação do algoritmo ABCRK ao POR

Para avaliar o desempenho do algoritmo ABCRK aplicado ao POR, a recarga do ciclo 7 de operação da CNA1 é tomada como referência no presente estudo.

As simulações computacionais utilizando o código de Física de Reatores RECNOB (CHAPOT, 2000), levaram em consideração uma estratégia de carregamento do tipo baixa-fuga, mas, sem dispor de venenos queimáveis (particular característica do ciclo 7), e tiveram como objetivo a maximização da concentração crítica de boro (C_B) de final do ciclo.

Além da C_B , a máxima potência média relativa P_{rm} foi o outro parâmetro de otimização a ser avaliado pelo código de Física de Reatores. Neste caso, P_{rm} substitui F_{XY} , uma vez que o mesmo não é fornecido pelo RECNOB. Segundo (CHAPOT, 2000), o uso da P_{rm} não viola as Especificações Técnicas da CNA1, uma vez que, para um Fator de Pico de Potência Radial $F_{XYmax} = 1,435$ para a CNA1, os cálculos correspondem a $P_{rm} = 1,395$. Assim, qualquer PC cuja simulação resulte em $P_{rm} \leq 1,395$ estará dentro dos requisitos técnicos de segurança. Caso contrário, o PC não pode ser aceito como solução.

A função objetivo (vide Equação 5.1) utiliza P_{rm} e C_B como parâmetros para a avaliação dos PC e é o inverso da concentração de Boro se $P_{rm} \leq 1,395$. Caso contrário, a função objetivo é igual a P_{rm} .

Para as simulações realizadas, o algoritmo ABCRK foi modelado com as abelhas operárias em posições de fontes $x_{i,j}$ do mesmo modo como foi representado na Figura 4.3, só que neste caso, em um espaço de 20 dimensões, considerando-se a simetria de octeto do núcleo. Estas posições são utilizadas como chaves a serem decodificadas, obtendo-se o vetor auxiliar $(x_{i,j})^*$ com números de 1 a 20 sem repetições, de acordo com o ilustrado na Figura 4.3. A informação contida nas posições de fontes de alimento configura um núcleo que é então avaliado pelo código RECNOD. Assim, cada posição de fonte de alimento representa uma possível solução ou PC. A decodificação permite a obtenção de soluções candidatas para o POR, a partir de um espaço de busca real de 20 dimensões, e o RECNOD avalia cada uma delas, fornecendo C_B e P_{rm} para a função objetivo.

Deste modo, a fim de verificar a capacidade do algoritmo ABCRK em maximizar o comprimento do ciclo 7 da CNA1, as seguintes simulações foram realizadas:

- Em uma primeira etapa, foram verificados 5 enxames com 50 abelhas cada. A cada um desses enxames, foi atribuída uma semente aleatória distinta. Inicialmente, foram considerados 40 ciclos, o que corresponde a 6.000 avaliações (vale ressaltar que o ABCRK, assim como o ABC original, realiza três avaliações da *fitness* a cada ciclo). Para acompanhar a evolução da *fitness*, foram considerados, na sequência, os seguintes números de ciclos e avaliações: 80 ciclos (12.000 avaliações), 160 ciclos (24.000 avaliações) e 320 ciclos (48.000 avaliações).

O valor do parâmetro *limite* utilizado em todas as simulações de recarga foi mantido fixo e igual a $(SN*D)*0,01$. Onde SN representa o número de posições de fontes de alimento e D corresponde à dimensão do problema. Tal como mencionado nos capítulos anteriores, o parâmetro *limite* é importante para evitar que certas soluções fiquem estagnadas em algum ponto por muito tempo. No algoritmo ABC original, após o valor do *limite* = $(SN*D)$ sem melhora para uma determinada solução, uma nova solução é produzida (abelha exploradora) aleatoriamente para explorar novos pontos do espaço de busca. No presente estudo. O valor do limite foi multiplicado por 0,01, pois, verificou-se que este limite era o que apresentava as melhores soluções para o problema em questão.

Os resultados das simulações realizadas nesta primeira etapa são apresentados na Tabela 6.1.

Tabela 6.1. ABCRK aplicado ao POR com 50 abelhas: C_B (ppm) e P_{rm} obtidos por 5 enxames (formato: C_B / P_{rm}). Resultados comparáveis com GA* e PBIL_N**.

Enxame	40 ciclos	80 ciclos	160 ciclos	320 ciclos
	6.000	12.000	24.000	48.000
	avaliações	avaliações	avaliações	avaliações
	C_B / P_{rm}	C_B / P_{rm}	C_B / P_{rm}	C_B / P_{rm}
#1	1173 / 1,388	1264 / 1,391	1264 / 1,391	1264 / 1,391
#2	1122 / 1,393	1122 / 1,393	1122 / 1,393	1122 / 1,393
#3	1122 / 1,393	1122 / 1,393	1122 / 1,393	1122 / 1,393
#4	1147 / 1,389	1148 / 1,387	1164 / 1,385	1164 / 1,385
#5	1291 / 1,383	1291 / 1,383	1291 / 1,383	1291 / 1,383
Média	1171 / 1,389	1189 / 1,389	1193 / 1,389	1193 / 1,389

*GA: 1026 ppm / 1,390 (CHAPOT, 2000) em 4.000 avaliações
 **PBIL_N: 1242 ppm / 1,361 (MACHADO, 2005) em 6.000 avaliações

- Na segunda etapa de testes com o ABCRK, buscou-se verificar o efeito causado pelo aumento do número de abelhas no enxame. Deste modo 30 abelhas foram incluídas no enxame da primeira etapa de testes, resultando em um enxame com 80 abelhas. As mesmas sementes foram mantidas, alterando-se apenas o número de ciclos para garantir o mesmo número de avaliações. Os resultados da segunda etapa de testes são apresentados na Tabela 6.2.

Tabela 6.2. ABCRK aplicado ao POR com 80 abelhas: CB (ppm) e Prm obtidos por 5 enxames (formato: CB / Prm). Resultados comparáveis: GA*, PBIL_N** e PBIL_MO***.

Enxame	25 ciclos	50 ciclos	100 ciclos	200 ciclos
	6.000	12.000	24.000	48.000
	avaliações	avaliações	avaliações	avaliações
	C_B / P_{rm}	C_B / P_{rm}	C_B / P_{rm}	C_B / P_{rm}
#1	1357 / 1,390	1357 / 1,390	1357 / 1,390	1357 / 1,390
#2	1317 / 1,392	1317 / 1,392	1317 / 1,392	1318 / 1,392
#3	1317 / 1,392	1317 / 1,392	1317 / 1,392	1318 / 1,392
#4	1121 / 1,387	1121 / 1,387	1121 / 1,387	1124 / 1,388
#5	1335 / 1,392	1335 / 1,392	1335 / 1,392	1335 / 1,392
Média	1289 / 1,391	1289 / 1,391	1289 / 1,391	1290 / 1,391

*GA: 1026 ppm / 1,390 (CHAPOT, 2000) em 4.000 avaliações
 **PBIL_N: 1242 ppm / 1,361 (MACHADO, 2005) em 6.000 avaliações
 ***PBIL_MO: 1305 ppm / 1,349 (MACHADO, 2005) em 10.000 avaliações

- Na terceira etapa de testes com o ABCRK, aumentou-se ainda mais o número de abelhas no enxame. Desta vez o número de abelhas no enxame passou de 80 para 100 abelhas no total. As mesmas sementes foram mantidas, alterando-se apenas o número de ciclos para garantir o mesmo número de avaliações. Os resultados da terceira etapa de testes são apresentados na Tabela 6.3.

Tabela 6.3. ABCRK aplicado ao POR com 100 abelhas: C_B (ppm) e P_{rm} obtidos por 5 enxames (formato: C_B / P_{rm}). Resultados comparáveis apenas ao GA*.

Enxame	20 ciclos	40 ciclos	80 ciclos	160 ciclos
	6.000	12.000	24.000	48.000
	avaliações	avaliações	avaliações	avaliações
	C_B / P_{rm}	C_B / P_{rm}	C_B / P_{rm}	C_B / P_{rm}
#1	1240 / 1,390	1240 / 1,390	1240 / 1,390	1243 / 1,350
#2	1079 / 1,325	1080 / 1,326	1080 / 1,326	1107 / 1,384
#3	1079 / 1,325	1079 / 1,325	1079 / 1,325	1079 / 1,325
#4	1236 / 1,374	1238 / 1,387	1238 / 1,387	1238 / 1,387
#5	1079 / 1,390	1079 / 1,390	1079 / 1,390	1079 / 1,390
Média	1143 / 1,361	1143 / 1,364	1143 / 1,364	1149 / 1,367

*GA: 1026 ppm / 1,390 (CHAPOT, 2000) em 4.000 avaliações

6.3. Resultados da Aplicação do algoritmo DBABC ao POR

Para avaliar o desempenho do algoritmo DBABC aplicado ao POR, as mesmas considerações feitas na Seção 6.2 para o algoritmo ABCRK aplicado ao POR foram mantidas. No entanto, em razão das diferenças próprias entre os dois algoritmos, destacam-se as seguintes considerações:

- Os indivíduos da população (posições de fonte de alimento ou abelhas operárias) são agora representados por *strings* de 12 bits em um espaço de busca de 20 dimensões equivalente à simetria de octeto.

- O parâmetro NBTV, o qual controla o número de bits trocados para vizinhança, foi mantido fixo e igual a 12 em todas as simulações de recarga. Isto significa que todos os 12 bits das *strings* são trocados a cada iteração do algoritmo a fim de definir as mudanças nas posições das fontes de alimento (abelhas operárias).

Com base nestas considerações, vale ressaltar que no DBABC, a população inicial de soluções aleatoriamente gerada pelas abelhas são *strings* binárias (0 e 1) e não mais um vetor de números reais como no caso do ABCRK.

Esta característica do DBABC torna possível a representação de todos os pontos do espaço de busca, permitindo assim, que qualquer problema (combinatório, contínuo ou discreto) possa ser tratado.

Também é curioso ressaltar que a opção pela representação binária deve levar em consideração o fato que, para efetuar uma mudança de valor unitário, torna-se necessário, em alguns casos, que seja feita a alteração de todos os bits de um número binário, efeito este que é chamado de Abismo de *Hamming* (LINDEN, 2008).

Por exemplo, para mudar do número 7 (0111) para o número 8 (1000), faz-se necessário alterar todos os bits de uma única vez. Já para mudar do número 8 para o número 9 (1001), basta alterar um único bit. Isto é, duas mudanças com o mesmo efeito final requerem alterações completamente díspares nas *strings*.

Para evitar este problema, o DBABC utiliza o código *Gray* (IDOETA e CAPUANO, 2001), cuja principal característica é que de um número a outro, apenas um bit varia. Por exemplo, no código Gray tem-se: 8 = 1100 e 9 = 1101.

Ainda assim, o parâmetro NBTV = 12 para as simulações de recarga foi o que apresentou os melhores resultados e por esta razão foi o valor utilizado em todos os experimentos.

Do mesmo modo como foi feito para o ABCRK, as simulações com o DBABC foram realizadas em três etapas e seguiram o mesmo procedimento padrão.

De maneira resumida, estas etapas são:

- Primeira etapa: 5 enxames, 50 abelhas, diferentes sementes aleatórias, 40 ciclos (6.000 avaliações), 80 ciclos (12.000 avaliações), 160 ciclos (24.000 avaliações) e 320 ciclos (48.000 avaliações).

- Segunda etapa: 5 enxames, 80 abelhas, mesmas sementes da primeira etapa de testes com o DBABC, 25, 50, 100 e 200 ciclos, correspondendo ao mesmo número de avaliações da primeira etapa.

- Terceira etapa de testes: 5 enxames, 100 abelhas, mesmas sementes da primeira e da segunda etapa de testes com o DBABC, 20, 40, 80 e 160 ciclos, correspondendo ao mesmo número de avaliações da primeira e da segunda etapa.

Os resultados da primeira etapa de testes com o DBABC são apresentados na Tabela 6.4.

Tabela 6.4. DBABC aplicado ao POR com 50 abelhas: C_B (ppm) e P_{rm} obtidos por 5 enxames (formato: C_B / P_{rm}). Resultados comparáveis aos seguintes algoritmos: GA^{*}, PBIL_N^{**}, PBIL_MO^{***}, PSORK^{****}.

Enxame	40 ciclos	80 ciclos	160 ciclos	320 ciclos
	6.000	12.000	24.000	48.000
	avaliações	avaliações	avaliações	avaliações
	C_B / P_{rm}	C_B / P_{rm}	C_B / P_{rm}	C_B / P_{rm}
#1	1332 / 1,385	1332 / 1,385	1333 / 1,384	1333 / 1,384
#2	1404 / 1,387	1404 / 1,387	1404 / 1,387	1404 / 1,387
#3	1337 / 1,394	1337 / 1,394	1337 / 1,394	1337 / 1,394
#4	1361 / 1,383	1389 / 1,390	1396 / 1,378	1398 / 1,385
#5	1307 / 1,393	1308 / 1,394	1308 / 1,394	1308 / 1,394
Média	1348 / 1388	1354 / 1,390	1356 / 1,387	1356 / 1,389

*GA: 1026 ppm / 1,390 (CHAPOT, 2000) em 4.000 avaliações
**PBIL_N: 1242 ppm / 1,361 (MACHADO, 2005) em 6.000 avaliações
***PBIL_MO: 1305 ppm / 1,349 (MACHADO, 2005) em 10.000 avaliações
****PSORK: 1394 ppm / 1,384 (MENESES, 2010) em 6.000 avaliações
*****PSORK: 1396 ppm / 1,378 (MENESES, 2010) em 10.000 avaliações

Os resultados da segunda etapa de testes com o DBABC são apresentados na Tabela 6.5.

Tabela 6.5. DBABC aplicado ao POR com 80 abelhas: C_B (ppm) e P_{rm} obtidos por 5 enxames (formato: C_B / P_{rm}). Resultados comparáveis apenas ao GA^{*}.

Enxame	25 ciclos	50 ciclos	100 ciclos	200 ciclos
	6.000	12.000	24.000	48.000
	avaliações	avaliações	avaliações	avaliações
	C_B / P_{rm}	C_B / P_{rm}	C_B / P_{rm}	C_B / P_{rm}
#1	1092 / 1,394	1103 / 1,391	1107 / 1,386	1135 / 1,391
#2	1092 / 1,394	1196 / 1,395	1196 / 1,395	1196 / 1,395
#3	1070 / 1,360	1070 / 1,360	1095 / 1,375	1203 / 1,382
#4	1108 / 1,368	1109 / 1,370	1180 / 1,382	1180 / 1,382
#5	1182 / 1,348	1182 / 1,348	1226 / 1,395	1226 / 1,395
Média	1109 / 1,373	1132 / 1,373	1161 / 1,387	1188 / 1,389

*GA: 1026 ppm / 1,390 (CHAPOT, 2000) em 4.000 avaliações

Os resultados da terceira e última etapa de testes com o DBABC são apresentados na Tabela 6.6.

Tabela 6.6. DBABC aplicado ao POR com 100 abelhas: C_B (ppm) e P_{rm} obtidos por 5 enxames (formato: C_B / P_{rm}). Resultados comparáveis com GA* e PBIL_N**.

Enxame	25 ciclos	50 ciclos	100 ciclos	200 ciclos
	6.000	12.000	24.000	48.000
	avaliações	avaliações	avaliações	avaliações
	C_B / P_{rm}	C_B / P_{rm}	C_B / P_{rm}	C_B / P_{rm}
#1	1243 / 1,387	1261 / 1,395	1408 / 1,389	1408 / 1,389
#2	1038 / 1,358	1169 / 1,373	1213 / 1,390	1213 / 1,390
#3	1132 / 1,353	1134 / 1,354	1140 / 1,393	1171 / 1,370
#4	1162 / 1,395	1162 / 1,395	1258 / 1,388	1262 / 1,389
#5	1107 / 1,372	1174 / 1,378	1174 / 1,378	1174 / 1,378
Média	1136 / 1,373	1180 / 1,379	1239 / 1,388	1246 / 1,383

*GA: 1026 ppm / 1,390 (CHAPOT, 2000) em 4.000 avaliações
 **PBIL_N: 1242 ppm / 1,361 (MACHADO, 2005) em 6.000 avaliações

6.4. Sumário do Capítulo

Neste capítulo foram apresentados os resultados obtidos com a aplicação das metodologias propostas à otimização do ciclo 7 de recarga da CNA1.

Com base nas informações contidas nas Tabelas 6.1 à 6.6, destacam-se os seguintes pontos:

- No que diz respeito ao desempenho do algoritmo ABCRK, percebe-se que os melhores resultados foram obtidos para enxames compostos por 80 abelhas. Em 6.000 avaliações, o algoritmo proposto foi capaz de produzir resultados superiores ao GA ($C_B = 1026$ ppm em 4.000 avaliações (CHAPOT, 2000)), ao PBIL_N ($C_B = 1242$ ppm em 6.000 avaliações (MACHADO, 2005)) e ao PBIL_MO ($C_B = 1305$ ppm em 10.000 avaliações (MACHADO, 2005)). Para enxames com apenas 50 abelhas os resultados foram apenas comparáveis aos obtidos pelo GA (CHAPOT, 2000) e, em alguns casos, ao PBIL_N (MACHADO, 2005). Os piores resultados obtidos pelo ABCRK ocorreram para enxames maiores (100 abelhas), o que indica a necessidade de um maior número de avaliações do que o que foi considerado.

- No que diz respeito ao desempenho do algoritmo DBABC, percebe-se que as melhores soluções foram obtidas com enxames compostos por apenas 50 abelhas. Em 6.000 avaliações, o DBABC foi capaz de superar os resultados obtidos pelo próprio ABCRK, bem como por todas as outras técnicas supracitadas.

Tal como mencionado no Capítulo 1, a Concentração de Boro (C_B) possui uma relação direta com o número de DEPP do ciclo de uma usina nuclear. Como 4 ppm de Boro correspondem a aproximadamente 1 DEPP, configurações de núcleo que maximizam a C_B permitem ciclos maiores, ou seja, indicam que a usina irá operar uma quantidade maior de dias com tais configurações.

Sendo assim, de acordo com a Tabela 6.4, pode-se afirmar que o DBABC obteve valores de C_B maiores que os obtidos por métodos tais como GA, PSO, PBIL_N e PBIL_MO, o que significa, em termos práticos, grande vantagem econômica.

É importante frisar que os parâmetros nucleares relevantes para o trabalho são a C_B (relacionada ao comprimento do ciclo) e P_{rm} (relacionada à critérios de segurança) agregados em uma única função objetivo (Equação 5.1).

Das Tabelas 6.1 à 6.6 pode-se inferir que os resultados obtidos por ambos os algoritmos propostos tendem à configurações em concordância com os requisitos técnicos de segurança, representados pela restrição imposta pela *fitness* dada na Equação 5.1.

Capítulo 7

– Conclusões –

O ABC é um algoritmo de otimização baseado na metáfora biológica da aprendizagem coletiva observada em uma colônia de abelhas reais, que se comunicam entre si na busca por melhores fontes de alimento.

Situado no paradigma da Inteligência de Enxames, o ABC realiza uma espécie de busca estocástica, combinada com busca por vizinhança, e pode ser aplicado na otimização de funções numéricas multimodais e multidimensionais em espaços de busca contínuos.

Com o objetivo de tornar o algoritmo ABC original adequado à solução de problemas em espaços de busca discretos e, de demonstrar a viabilidade de sua aplicação como otimizador para o POR da CNA1, foram propostas duas versões modificadas do mesmo, as quais foram utilizadas com sucesso na maximização do comprimento do ciclo 7 de operação da CNA1.

A primeira versão modificada do algoritmo ABC original proposta no presente estudo consistiu na utilização do ABC em sua versão original, combinada com a metodologia de Chaves Aleatórias (*Random Keys*) (BEAN, 1994). Por esta razão, o algoritmo proposto foi chamado de *Artificial Bee Colony with Random Keys* (ABCRK).

O modelo de Chaves Aleatórias mapeia um vetor composto por números reais em uma solução contendo números inteiros não repetidos. Deste modo, soluções possíveis podem ser obtidas em um problema de otimização. Embora não torne o ABC discreto, o modelo de Chaves Aleatórias o torna adequado para a solução de problemas combinatórios.

Já a segunda versão modificada do algoritmo ABC proposta no presente estudo foi chamada de *Discrete Binary Artificial Bee Colony* (DBABC). Trata-se de uma versão binária discreta do algoritmo ABC original, a qual apresenta a vantagem de ser aplicável à solução de qualquer problema, seja ele contínuo ou discreto.

No algoritmo DBABC, a população inicial é formada por uma string de bits e não mais por um vetor de números reais a serem codificados, como no caso do ABCRK. Por esta razão, todos os pontos do espaço de busca podem ser representados, sendo esta a principal vantagem do DBABC em relação ao ABCRK.

Como qualquer problema (contínuo ou discreto) pode ser escrito em uma notação binária, um algoritmo capaz de tratar problemas em ambos os espaços de busca torna-se extremamente vantajoso.

As conclusões do presente estudo estão baseadas nos resultados obtidos a partir das simulações realizadas no Capítulo 6, em função da metodologia aplicada à proposta inicial.

Os resultados obtidos confirmaram as expectativas iniciais do presente estudo com relação às potencialidades do algoritmo ABC. O bom desempenho do algoritmo ABC nas versões propostas neste estudo ficou evidente com o POR, avaliado e discutido no Capítulo 5.

A simplicidade de implementação dos modelos desenvolvidos, os poucos parâmetros de controle a serem utilizados, bem como o bom desempenho demonstrado nas simulações de maximização do comprimento do ciclo 7 da CNA1, são bons atrativos para a escolha das metodologias propostas.

De acordo com os resultados apresentados no Capítulo 6, pode-se afirmar que tanto o ABCRK quanto o DBABC, foram capazes de fornecer resultados comparáveis e, em muitas vezes, superiores a outras metaheurísticas baseadas em populações já bem conhecidas, demonstrando assim, a viabilidade de utilização destes modelos para a solução de problemas com alto grau de complexidade, assim como é o caso do POR.

O POR é um problema extremamente complexo, caracterizado por apresentar não linearidades, alta dimensionalidade, regiões desconexas de possíveis soluções, além de múltiplos mínimos e máximos locais no espaço de soluções.

Considerando que os exames não possuem informações prévias a respeito do problema e que nenhuma heurística foi utilizada, é possível afirmar também que o paradigma da Inteligência de Exames pode ser aplicado à solução do POR com considerável desempenho e que, em vista dos resultados obtidos, pode-se afirmar que os objetivos desta tese foram atingidos.

Outro grande atrativo para a utilização das metodologias propostas é o fato das mesmas empregarem um número menor de parâmetros de controle que outras metaheurísticas de otimização. Tanto o ABC original como o ABCRK utilizam um único parâmetro de controle, o qual é chamado limite. Já o DBABC, além do parâmetro limite, também utiliza o parâmetro NBTV (número de bits trocados para vizinhança). Ainda assim, o DBABC é extremamente simples de ser implementado.

É importante frisar que as metodologias aqui propostas podem ainda ser investigadas em maior profundidade, bem como podem ser aplicadas em conjunto com outras técnicas a fim de formarem modelos híbridos, o que pode constituir trabalhos futuros a serem desenvolvidos.

Referências Bibliográficas

BALUJA, S., 1994, “*Population-Based Incremental Learning: A Method for Integrating Genetic Search Based Function Optimizations and Competitive Learning*”. Technical Report CMU-CS-94-163, June.

BEAN, J. C., 1994, “*Genetic Algorithms and Random Keys for Sequencing and Optimization*”. ORSA Journal of Computing 6 (2).

BOYD, S., VANDENBERGHE, L., 2004, “*Convex Optimization*”. Cambridge University Press, ISBN 0521833787.

CALDAS, G. H. F., SCHIRRU, R., 2008, “*Parameterless Evolutionary Algorithm Applied to the Nuclear Reload Problem*”. Annals of Nuclear Energy 35, pp. 583-590.

CHAPOT, J. L. C., 2000, “*Otimização Automática de Recargas de Reatores a Água Pressurizada Utilizando Algoritmos Genéticos*”. Tese de Doutorado, COPPE/UFRJ, Rio de Janeiro, Brasil, Junho.

CORMEN, T. H., LEISERSON, C. E., et al., 2002, “*Algoritmos – Teoria e Prática*”. 2ª Edição, Ed. Campus, Rio de Janeiro.

DE LIMA, A. M. M., 2005, “*Recarga de Reatores Nucleares Utilizando Redes Conectivas de Colônias Artificiais*”. Tese de Doutorado, COPPE/UFRJ, Rio de Janeiro, Brasil, Junho.

DE LIMA, A. M. M., SCHIRRU, R., DA SILVA, F. C., MEDEIROS, J. A. C. C., 2008, “*A Nuclear Reactor Core Fuel Reload Optimization Using Ant Colony Connective Networks*”. Annals of Nuclear Energy, 35, pp. 1606-1612.

DEB, K., 1995, “*Optimization for Engineering Design: Algorithms and Examples*”. New Delhi: Prentice-Hall.

DECHAINED, M. D., FELTUS, M. A., 1995, “*Nuclear Fuel Management Optimization Using Genetic Algorithms*”. Nuclear Technology, v. 111, pp. 109-114, July.

DORIGO, M., GAMBARDELLA, L. M., 1997, “*Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem*”. IEEE Transactions on Evolutionary Computation 1 (1), pp. 53-66.

FOX, R. L., 1971, “*Optimization Methods for Engineering Design*”. Reading, MA: Addison-Wesley.

FSAR, 1980, “*Relatório Final de Análise de Segurança*”, Central Nuclear Almirante Álvaro Alberto, Unidade 1, Capítulo 16 (item 16.3.9.4, Revisão 7, 30/09/1980), Furnas Centrais Elétricas S. A.

GALPERIN, A., KIMHI, S., SEGEV, M., 1989, “*A Knowledge-Based System for Optimization of Fuel Reload Configurations*”. Nuclear Science and Engineering 102, pp. 43-53.

GALPERIN, A., 1995, “*Exploration of the Search Space of the In-Core Fuel Management Problem by Knowledge-Based Techniques*”. Nuclear Science and Engineering 119, pp. 144-152.

GAO, W., LIU, S., JIANG, F., 2011, “*An Improved Artificial Bee Colony Algorithm for Directing Orbits of Chaotic Systems*”. Applied Mathematics and Computation 218 (2011) 3868 - 3879.

GLOVER, F. W., LAGUNA, M., 1997, “*Tabu Search*”. Kluwer Academic Publishers.

GOLDBARG, M. C., LUNA, H. P. L., 2005, “*Otimização Combinatória e Programação Linear: Modelos e Algoritmos*”. Ed. Campus, 2ª Edição, Rio de Janeiro.

GOLDBERG, D. E., 1989, “*Genetic Algorithms in Search, Optimization and Machine Learning*”. EUA, Addison Wesley.

HOLLAND, J. H., 1975, “*Adaptation in Natural and Artificial Systems*”. EUA, University of Michigan Press.

HORNG, M., 2011, “*Multilevel Thresholding Selection Based on the Artificial Bee Colony Algorithm for Image Segmentation*”. *Expert Systems with Applications* 38 (2011) 13785 – 13791.

IDOETA, I. V., CAPUANO, F., G., 2001. “*Elementos de Eletrônica Digital*”. Editora Érica, 32ª Edição.

IRANI, R., NASIMI, R., 2011, “*Application of Artificial Bee Colony-Based Neural Network in Bottom Hole Pressure Prediction in Underbalanced Drilling*”. *Journal of Petroleum Science and Engineering* 78 (2011) 6-12.

ISHIGURO, Y., 2002, “*A Energia Nuclear para o Brasil*”. São Paulo, Makron Books.

IZMAILOV, A., SOLODOV, M., 2009, “*Otimização – Volume 1: Condições de Otimalidade, Elementos de Análise Convexa e de Dualidade*”. 2ª Edição. IMPA – Instituto Nacional de Matemática Pura e Aplicada.

KARABOGA, D., 2005, “*An Idea Based on Honey Bee Swarm for Numerical Optimization*”. Technical Report – TR06, October.

KARABOGA, D., OZTURK, C., 2011, “*A Novel Clustering Approach: Artificial Bee Colony (ABC) Algorithm*”. *Applied Soft Computing* 11 (2011) 652 - 657.

KENNEDY, J., EBERHART, R. C., 1995, “*A New Optimizer Using Particles Swarm Theory*”. In: *Proceedings of Sixth International Symposium on Micro Machine and Human Science*, EUA, IEEE Service Center, pp. 39-43, 1995.

KENNEDY, J., EBERHART, R. C., 1997, “*A Discrete Binary Version of the Particle Swarm Algorithm*”. In: *Proceedings of the IEEE Conference on Systems, Man and Cybernetics*, pp. 4104-4109.

KIRKPATRICK, S., GELLAT, C. D., VECCHI, M. P., 1983, “*Optimization by Simulated Annealing*”. *Science*, v. 220, pp. 671-680, May.

KROPACKZEK, D. J., TURINSKY, P. J., 1990, “*In-Core Nuclear Fuel Management Optimization for Pressurized Water Reactors Utilizing Simulated Annealing*”. *Nuclear Technology* v.95, n.9, pp. 9-31.

LIN, C., YANG, J.-I., LIN, K.-J., WANG, Z.-D., 1998, “*Pressurized Water Reactor Loading Pattern Design Using the Simple Tabu Search*”. *Nuclear Science and Engineering* 129, pp. 61-71.

LINDEN, R., 2008, “*Algoritmos Genéticos: Uma Importante Ferramenta da Inteligência Computacional*”. Ed. Brasport, 2ª Edição, Rio de Janeiro.

MACHADO, M. D., 1998, “*Aplicação de uma Nova Metodologia baseada na Propagação Estatística das Incertezas para Avaliar os Limites do DNBR*”. Dissertação de Mestrado, Programa de Engenharia Nuclear, COPPE/UFRJ, Rio de Janeiro, Brasil, Agosto.

MACHADO, M. D., 1999, “*Um Novo Algoritmo Evolucionário com Aprendizado LVQ para a Otimização de Problemas Combinatórios como a Recarga de Reatores Nucleares*”. Dissertação de Mestrado, Programa de Engenharia Nuclear, COPPE/UFRJ, Rio de Janeiro, Brasil, Abril.

MACHADO, L., 2001, “*Otimização de Recarga do Combustível Nuclear por Agentes Artificiais*”. Tese de Doutorado, Programa de Engenharia Nuclear, COPPE/UFRJ, Rio de Janeiro, Brasil, Março.

MACHADO, M. D., 2002, “*Aplicação de uma Nova Metodologia Baseada na Propagação Estatística das Incertezas para Avaliar os Limites do DNBR*”. Dissertação de Mestrado, Programa de Engenharia Nuclear, COPPE/UFRJ, Rio de Janeiro, Brasil, Agosto.

MACHADO, L., SCHIRRU, R., 2002, “*The Ant-Q Algorithm Applied to the Nuclear Reload Problem*”. *Annals of Nuclear Energy* 29, pp. 1455-1470.

MACHADO, M. D., 2005, “*Algoritmo Evolucionário PBIL Multiobjetivo aplicado ao Problema da Recarga de Reatores Nucleares*”. Tese de Doutorado, Programa de Engenharia Nuclear, COPPE/UFRJ, Rio de Janeiro, Brasil.

MENESES, A. A. M., MACHADO, M. D., SCHIRRU, R., 2009, “*Particle Swarm Optimization Applied to the Nuclear Reload Problem of a Pressurized Water Reactor*”. *Progress in Nuclear Energy* 51, pp. 319-326.

MENESES, A. A. M., 2010, “*Otimização com Enxame de Partículas e Busca Baseada em Classes Aplicadas ao Problema da Recarga de um Reator Nuclear*”. Tese de Doutorado, Programa de Engenharia Nuclear, COPPE/UFRJ, Rio de Janeiro, Brasil.

OLIVEIRA, I. M. S., SCHIRRU, R., 2010, “*A Discrete Binary Version of the Artificial Bee Colony (ABC) Optimization Algorithm*”. XIII Encontro de Modelagem Computacional, Instituto Politécnico (IPRJ), Campus Regional da UERJ, Nova Friburgo/RJ, Brasil. 03-05 Nov. 2010. Associação Brasileira de Engenharia e Ciências Mecânicas – ABCM.

OLIVEIRA, I. M. S., SCHIRRU, R., 2011, “*Swarm Intelligence of Artificial Bees Applied to In-Core Fuel Management Optimization*”. *Annals of Nuclear Energy* 38 (2011) 1039-1045.

POON, P. W., PARKS, G. T., 1992, “*Optimizing PWR Reload Core Design, Parallel Problem Solving from Nature*”. V.2, pp. 371-380.

PRICE, K. V., STORN, R. M., LAMPINEN, J. A., 2005, “*Differential Evolution: A Practical Approach to Global Optimization*”. Springer-Verlag Berlin Heidelberg.

RASHIDI, M., GALANIS, N., NAZARI, F., PARSA, B., A., SHAMEKHI, L., 2011, “*Parametric Analysis and Optimization of Regenerative Clausius and Organic*

Rankine Cycles with Two Feedwater Heaters using Artificial Bees Colony and Artificial Neural Network". Energy 36 (2011) 5728 - 5740.

REKLAITIS, G. V., RAVINDRAN, A., RAGSDELL, K. M., 1983, "*Engineering Optimization: Methods and Applications*". John Wiley & Sons, USA.

SAMANTA, S., CHAKRABORTY, S., 2011, "*Algoritmos Genéticos: Uma Importante Ferramenta da Inteligência Computacional*". Ed. Brasport, 2ª Edição, Rio de Janeiro.

SCHAFFER, J. D., 1984, "*Multiple Objective Optimization with Vector Evaluated Genetic Algorithms*". PhD. Dissertation, Vanderbilt University.

STORN, R., PRICE, K., 1995, "*Differential Evolution: A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces*". Technical Report TR-95-012, International Computer Science Institute, Berkeley.

STORN, R., PRICE, K., 1995, "*Differential Evolution: A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces*". Journal of Global Optimization, v.11, pp. 341-359.

TOSCANI, L. V., VELOSO, P. A. S., 2001, "*Complexidade de Algoritmos: Análise, Projetos e Métodos*". Ed. Sagra Luzzato, Porto Alegre.

VAN DER SCHAAR, M., VERHAGEN, F. C. M., KRUIJF, W. J. M., VAN DE WETERING, T. F. H., 1996, "*ROSA: A Code for Fuel Management*". Proc. Jahrestagung Kerntechnik 96, Mannheim, Germany, May 21-23, Vol. 1, pp. 15-18, Deutsches Atomforum.

VAN DER SCHAAR, M., VERHAGEN, F. C. M., KRUIJF, W. J. M., VAN DE WETERING, T. F. H., JONES, R. D., 1997, "*ROSA: A Utility Tool for Loading Pattern Optimization*". Proc. Of the ANS Topical Meeting – Advances in Nuclear Fuel Management II, Myrtle Beach, South Carolina, March 23-26, 1997, Vol. 1, pp. 8.31-8.38, American Nuclear Society.

VANDERPLAATS, G. N., 1999, “*Numerical Optimization Techniques for Engineering Design*”. 3.ed. Vanderplaats Research & Development, Inc., Colorado Springs, CO.

WAINTRAUB, M., 2009, “*Algoritmos Paralelos de Otimização por Enxame de Partículas em Problemas Nucleares*”. Tese de Doutorado, COPPE/UFRJ, Rio de Janeiro, Brasil, Março.

XU, C., DUAN, H., 2010, “*Artificial Bee Colony (ABC) Optimized Edge Potential Function (EPF) Approach to Target Recognition for Low-Altitude Aircraft*”. Pattern Recognition Letters 31 (2010) 1759 - 1772.

ZANG, H., ZHU, Y., ZOU, W., YAN, X., 2011, “*A Hybrid Multi-Objective Artificial Bee Colony Algorithm for Burdening Optimization of Copper Strip Production*”. Applied Mathematical Modelling (2011).