



COPPE/UF RJ

OTIMIZAÇÃO DE PROBLEMAS DA ENGENHARIA NUCLEAR UTILIZANDO
AGENTES QUÂNTICOS

Márcio Henrique da Silva

Dissertação de Mestrado 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 Mestre em Engenharia Nuclear.

Orientador: Roberto Schirru

Rio de Janeiro
Fevereiro de 2010

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

OTIMIZAÇÃO DE PROBLEMAS DA ENGENHARIA NUCLEAR UTILIZANDO
AGENTES QUÂNTICOS

Márcio Henrique da Silva

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

Examinada por:

Prof. Roberto Schirru, D.Sc.

Prof. Jose Antonio Carlos Canedo Medeiros, D.Sc.

Prof. Cláudio Marcio Nascimento Abreu Pereira, D.Sc.

Dr. Jorge Luiz Cachoeira Chapot, D.Sc.

RIO DE JANEIRO, RJ – BRASIL

FEVEREIRO DE 2010

Silva, Márcio Henrique da

Otimização de Problemas da Engenharia Nuclear
Utilizando Agentes Quânticos/ Márcio Henrique da Silva. -
Rio de Janeiro: UFRJ/COPPE, 2010.

XII, 80 p.:il; 29,7 cm.

Orientador: Roberto Schirru

Dissertação(mestrado) – UFRJ/COPPE/ Programa de
Engenharia Nuclear, 2010

Referências Bibliográficas: p.76 - 80

1. Algoritmos Evolucionários com Inspiração Quântica. 2.
Otimização da Recarga Nuclear. I.Schirru, Roberto. II.
Universidade Federal do Rio de Janeiro, COPPE, Programa de
Engenharia Nuclear. III. Título.

*Se ontem eu era um Universitário,
Se hoje sou um Mestre,
Se amanhã talvez eu venha a me tornar um Doutor,
É graças ao esforço e dedicação de duas pessoas às quais sou
sempre grato e a quem eu dedico este trabalho
Dona Lourdes e Seu Ronaldo.*

Agradecimentos

O fim de um trabalho como este que demanda tempo, dedicação e um esforço que não pode ser descrito em palavras, é sempre um momento de satisfação e realização pessoal. Mas também é o momento no qual nos lembramos de todos os amigos que contribuem de forma direta ou indireta, nos ajudando a permanecer focados no objetivo principal que é sempre evoluir e desenvolver um trabalho de qualidade.

Agradeço especialmente ao amigo Kuato (vulgarmente conhecido como Alan Miranda) pelo apoio de sempre como “mestre das formigas” e à Maghali(ce), pela amizade e por ter compartilhado estes momentos por vezes exaustivos que passei como mestrando;

Também agradeço aos amigos que a vida no meio acadêmico me trouxe

Patrícia “no creo en brujas”, David e Juliana ;

Aos amigos que conheci já no curso de Mestrado, Cícero e Rachel entre outros.

À professora Maria Darci, que me incentivou a fazer o curso de álgebra linear que tanto me ajudou no mestrado.

Muito obrigado a todos.

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

OTIMIZAÇÃO DE PROBLEMAS DA ENGENHARIA NUCLEAR UTILIZANDO AGENTES QUÂNTICOS.

Márcio Henrique da Silva

Fevereiro/2010

Orientador: Roberto Schirru

Programa: Engenharia Nuclear

Os algoritmos evolucionários com inspirações quânticas são ferramentas de otimização baseadas em inteligência artificial, desenvolvidas para simular o processamento quântico em computadores clássicos.

Neste trabalho é apresentada uma dessas ferramentas, que une os conceitos quânticos à metáfora biológica da aprendizagem coletiva de uma colônia de formigas reais denominado QACO_Alfa. Nele é introduzida uma etapa de evaporação além de um novo método de atualização para evitar problemas de convergência prematura.

Para comprovar sua eficiência, o QACO_Alfa é aplicado à problemas da engenharia nuclear, em especial à otimização da recarga do ciclo 7 do núcleo de Angra 1. Os resultados experimentais observados são comparados aos de outros métodos de otimização e confirmam que o QACO_Alfa é uma ferramenta adequada para a otimização deste tipo de problema.

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

OPTIMIZATION OF NUCLEAR ENGINEERING PROBLEMS USING QUANTUM AGENTS

Márcio Henrique da Silva

February/2010

Advisor: Roberto Schirru

Department: Nuclear Engineering

Quantum-inspired evolutionary algorithms are optimization tools based on artificial intelligence, developed to simulate the quantum processing in classical computers. In this work it is introduced one of these tools, that adds quantum concepts to the biological metaphor of collective learning of the real ants, named QACO_Alfa. It uses mechanisms developed in order to avoid premature convergence problems like a pheromone evaporation step besides a new updating method.

To show its effectiveness, QACO_Alfa is applied in nuclear engineering problems, especially the optimization of nuclear reload in the cycle 7 of Angra 1. Experimental results are confronted to those obtained with other optimization methods, qualifying QACO_Alfa as a valid optimization tool for this kind of problem.

Sumário

1	Introdução.....	1
2	A Computação Quântica e os Algoritmos Evolucionários.....	7
2.1	A Evolução da Computação Quântica	8
2.1.1	Transposição de Conceitos Quânticos para Computadores Clássicos.....	9
2.2	Algoritmo de Otimização por Colônias de Formigas (ACO)	12
2.2.1	Regra de Atualização Local.....	16
2.2.2	Regra de Atualização Global.....	16
3	O Problema da Otimização da Recarga de Elementos Combustíveis	18
3.1	A Recarga Nuclear	19
3.2	A Modelagem do Núcleo	19
3.3	A Função <i>Fitness</i>	22
4	Otimização com Agentes Quânticos	26
4.1	<i>Quantum Ant Colony Optimization</i> (QACO)	27
4.1.1	Estagnação dos Bits	33
4.1.2	Dependência de Parâmetros.....	36
4.2	Otimização por Colônias de Formigas Quânticas_Alfa (QACO_Alfa)	37
4.2.1	Reforço de Feromônio	38
4.2.2	Evaporação de Feromônio	40
4.3	Problemas Teste	43
4.3.1	Parâmetros dos Testes	44

4.3.2	Função Esfera	45
4.3.3	Função <i>Himmelblau</i>	48
4.3.4	Função <i>Goldstein - Price</i>	50
5	QACO_Alfa Aplicado a Problemas da Engenharia Nuclear	52
5.1	QACO_Alfa Aplicado na Otimização do Ciclo 7 de Angra 1	53
5.1.1	Modelo <i>Random Keys</i>	53
5.1.2	Metodologia.....	55
5.1.2.1	Situação Real.....	55
5.1.2.2	Situação Particular para Efeitos de Comparação	56
5.2	Resultados	57
5.2.1	Resultados Caso 1.....	58
5.2.2	Resultados Caso 2.....	60
5.3	QACO_Alfa Aplicado ao Problema de Identificação de Transientes	63
5.3.1	Metodologia.....	64
6	Conclusões	67
6.1	Considerações Finais	68
6.2	Propostas para Trabalhos Futuros	69
	APÊNDICE - O Código RECNOD.....	71
	Referências Bibliográficas	76

Lista de Figuras

1.1	Visão geral de uma usina nuclear	2
2.1	Representação binária utilizando <i>qubits</i>	11
2.2	Influência do feromônio	13
2.3	Pseudo-código do ACO	17
3.1	Linhas de simetria do núcleo de Angra 1	20
3.2	Região de simetria de octante	21
3.3	Comportamento da função <i>fitness</i>	24
4.1	Caminho binário construído pelas formigas quânticas	29
4.2	Atualização de um <i>qubit</i> no plano cartesiano	32
4.3	Pseudo-código do QACO original	33
4.4	Exemplo de ação do portão quântico sobre um <i>qubit</i>	34
4.5	Casos particulares de estagnação	35
4.6	Atualização do <i>qubit</i> no QACO_Alfa	39
4.7	Determinação de ZERO(j) e UM(j)	40
4.8	Aplicação do portão quântico de evaporação	42
4.9	Exemplo de representação binária	43
4.10	Função esfera $n = 2$	45
4.11	Influência do número de <i>qubits</i> na precisão do algoritmo	46
4.12	Função <i>Himmelblau</i>	48
4.13	Função <i>Goldstein - Price</i>	50
5.1	Exemplo de vetor binário gerado pelo QACO_Alfa	53
5.2	Sequência de valores inteiros gerados a partir do vetor binário P	54
5.3	<i>Random Keys</i>	54
5.4	<i>Random Keys</i> ordenado	55
5.5	Padrão analisado pelo RECNOD	55
5.6	Modelo de <i>Random Keys</i> aplicado ao caso1	56

5.7	Padrão analisado pelo RECNOd caso 1.....	56
5.8	Modelo de <i>Random Keys</i> aplicado ao caso2	57
5.9	Padrão analisado pelo RECNOd caso 2.....	57
5.10	Evolução da potência média	58
5.11	Evolução da concentração de boro	59
5.12	Evolução da <i>fitness</i>	59
5.13	Evolução da potência média - Caso 2.....	62
5.14	Evolução da concentração de boro – Caso 2	62
5.15	Evolução da <i>fitness</i> – Caso 2	63
5.16	Identificação de um evento baseado na diferença entre	65
	as distâncias euclidianas	
5.17	Curva do QACO_Alfa aplicado à identificação de transientes	66
A.1	Equação de difusão de nêutrons independente do tempo	71
A.2	Método de Expansão de Fluxo	72
A.3	Método de Expansão de Fluxo alternativo	73
A.4	Modelagem do reator de acordo com os dados fornecidos ao RECNOd	74

Lista de Tabelas

4.1	Ângulo de rotação do QACO	31
4.2	Resultados de p_e	36
4.3	Ângulo de rotação do QACO_Alfa	38
4.4	Ângulo de evaporação do QACO_Alfa.....	41
4.5	Dados da função esfera.....	45
4.6	Resultados da função esfera dimensão $n = 10$	47
4.7	Resultados da função esfera dimensão $n = 30$	47
4.8	Dados da função <i>Himmelblau</i>	48
4.9	Resultados da função <i>Himmelblau</i>	49
4.10	Dados da função <i>Goldstein - Price</i>	50
4.11	Resultados da função <i>Goldstein - Price</i>	51
5.1	Comparação técnicas aplicadas à maximização	60
	do comprimento do ciclo 7 de Angra 1	
5.2	Resultados Caso 2	61
5.3	Comparação das técnicas utilizadas na identificação de transientes	65
A.1	Coordenadas dos elementos.....	75

Capítulo 1

Introdução

A todo procedimento no qual o objetivo é criar condições favoráveis para a obtenção do máximo rendimento possível, dá-se o nome de otimização. Na área da Engenharia Nuclear, um dos problemas de otimização mais complexos, que vem sendo analisado por vários pesquisadores ao longo dos anos (CHAPOT, 2000), é a otimização da recarga de Elementos Combustíveis (ECs) no núcleo do reator de uma usina do tipo PWR como é o caso das usinas de Angra 1 e Angra 2.

Os ECs são compostos por diversas varetas dentro das quais estão alocadas pastilhas de dióxido de urânio (UO_2). Estas pastilhas sofrem fissão produzindo nêutrons e gerando energia para o meio líquido no qual elas estão imersas, ou seja, a água que preenche o núcleo do reator que age ao mesmo tempo como moderadora de nêutrons e refrigerante para evitar o aumento descontrolado da temperatura. Esta água é mantida a temperaturas muito acima do ponto de ebulição graças ao pressurizador sendo então conduzida para o Gerador de Vapor (GV).

No GV ocorre troca de energia térmica entre a água superaquecida vinda do núcleo e a água do circuito secundário. Com isso é produzido vapor d'água que é então conduzido ao sistema secundário do reator onde cede energia para movimentar as turbinas da usina que podem assim gerar a energia elétrica. O vapor resfriado por ter cedido calor para as turbinas é conduzido ao condensador e retorna ao GV completando o processo, como pode ser observado na figura 1.1.

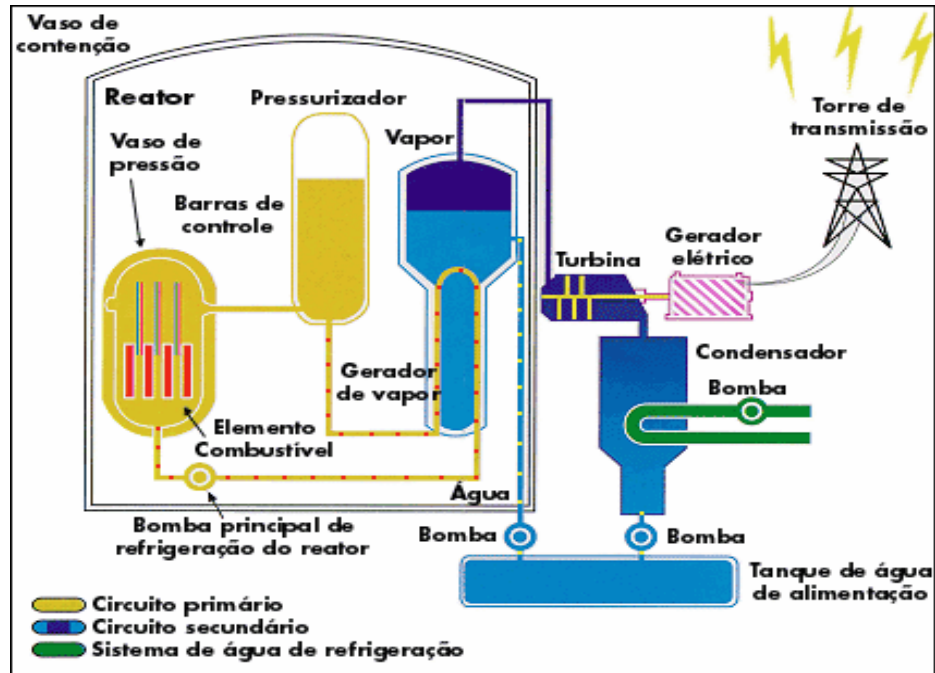


Figura 1.1: Visão geral de uma usina nuclear (Fonte: ELETRONUCLEAR)

A recarga de ECs se torna necessária sempre que o núcleo não consegue manter o reator crítico produzindo energia à potência nominal. O intervalo de tempo entre as recargas de um núcleo recebe o nome de ciclo de operação. Como a queima dos ECs depende da sua posição no núcleo, nem todos precisam ser substituídos ao final de um ciclo apenas aqueles que estiverem mais queimados.

Busca-se encontrar a melhor configuração de embaralhamento entre os elementos combustíveis novos e os remanescentes do ciclo anterior que mantenha o núcleo do reator funcionando à plena potência e satisfaça a critérios de otimização previamente estabelecidos tendo em vista que a recarga só ocorre com a usina desligada, o que acaba gerando lucro cessante. As várias possibilidades de configurações de embaralhamento recebem o nome de configurações de núcleo (CHAPOT, 2000).

Trata-se de um problema não-polinomial difícil, onde o número de Padrões de Recarga (PRs) cresce exponencialmente como o número de ECs analisados. A solução do problema de otimização da recarga também é dificultada pela sua natureza não-linear, pela existência de descontinuidades e múltiplos máximos e mínimos locais no espaço de soluções e pelas restrições de simetria existentes no núcleo (LIMA, 2005).

Inicialmente, o problema de otimização da recarga de ECs no núcleo do reator era resolvido pelo método de tentativa e erro, utilizando o conhecimento de especialistas. Com o desenvolvimento das técnicas de inteligência artificial no início dos anos 80, os pesquisadores chegaram à conclusão que automatizar o processo de otimização através de técnicas computacionais poderia levar a resultados superiores aos obtidos até então (CHAPOT, 2000). Desse modo, várias dessas técnicas, com destaque para os algoritmos evolucionários (HOLLAND, 1975), vêm sendo aplicados ao processo de otimização da recarga de ECs.

Desde o método de busca heurística baseado na experiência de operadores criado por GALPERIN *et al* (1989), passando pelo *Simulated Annealing* (SA) utilizado por KOPRACKZEK e TURINSKY (1991) e os Algoritmos Genéticos (AGs) (HOLLAND, 1975), até chegar à otimização por colônias de formigas (ACO) (DORIGO e GAMBARDELLA, 1996), os métodos de otimização que utilizam técnicas de inteligência artificial se consolidaram como sendo eficientes, rápidos e confiáveis quando aplicados ao problema de otimização da recarga nuclear.

Algoritmos Evolucionários (AEs) são ferramentas de busca e otimização que se baseiam no princípio darwiniano da sobrevivência do mais apto para produzir sucessivamente melhores aproximações para uma solução ao longo das gerações nas quais a população evolui. Eles são caracterizados pela representação do indivíduo, a função objetivo (que determina quão “apto” é o indivíduo na otimização do problema considerado) comumente conhecida como função *fitness* ou aptidão e as dinâmicas populacionais tais como o tamanho da população, operadores variação (como *crossing* e mutação), seleção de pais, reprodução e herança entre outras (GOLDBERG, 1989).

A cada geração, um novo conjunto de aproximações é criado através do processo de escolha de indivíduos pelo seu nível de *fitness* no domínio do problema. Este processo deve levar a evolução de populações de indivíduos melhor adaptados ao seu ambiente que os indivíduos que os originaram, da mesma maneira que ocorre na

natureza. Comparados aos métodos de otimização tradicionais como os baseados em cálculo e estratégias enumerativas, AEs são robustos, globais e têm a vantagem de geralmente não necessitarem de heurísticas específicas do domínio (HAN e KIM, 2002).

A otimização por colônias de formigas (ACO) (DORIGO e GAMBARDELLA, 1996) é um exemplo de algoritmo evolucionário bem conhecido que se destaca na otimização de problemas combinatórios, que foi originalmente desenvolvido para atuar na otimização do Problema do Caixeiro Viajante (PCV) (DORIGO e GAMBARDELLA, 1997). Neste problema, um caixeiro viajante deve visitar todas as cidades consideradas, pelo menos uma vez, sem repetições, e retornar à cidade de partida ao final do percurso. Existem dois tipos de PCVs: os simétricos, onde o tamanho do trajeto é o mesmo na ida e na volta e os assimétricos, onde os caminhos de ida e volta possuem tamanhos diferentes.

Devido à analogia entre o PCV e o problema de otimização da recarga (LIMA, 2005), uma vez que ambos são problemas combinatórios onde os elementos de um conjunto devem ser ordenados de modo que cada elemento ocupe uma única posição na ordenação sem que haja repetições, o ACO passou a ser empregado na otimização da recarga, obtendo para este problema os melhores resultados entre todas as outras técnicas de otimização às quais ele já foi submetido, sejam elas baseadas em inteligência artificial ou não.

O ACO utiliza o conceito de aprendizagem coletiva observado em colônias de formigas reais na natureza. Apesar de serem criaturas frágeis e rudimentares, foi observado (GOSS *et al*, 1989) que as formigas utilizam um método de aprendizagem coletiva para se comunicarem entre si que permite que elas sempre encontrem o menor caminho entre o formigueiro e uma fonte de alimento, independentemente de obstáculos ou alterações no ambiente de busca.

Isso se deve a uma substância conhecida como feromônio. Enquanto caminham, as formigas liberam esta substância por onde passam, atraindo outras formigas. Como o feromônio evapora com o passar do tempo, se o caminho for longo a concentração de feromônio presente nele não será tão intensa quanto num caminho mais curto e cada vez menos formigas seguirão por ele. Numa situação extrema, a quase totalidade das formigas irá optar por seguir pelo caminho mais curto deixando apenas algumas serem

guiadas por um instinto mais investigador, ou “forrageador” como são conhecidas em termos biológicos (GOSS *et al*, 1989).

Paralelamente à evolução dos AEs ao longo dos anos, vários pesquisadores têm buscado alternativas para torná-los mais rápidos e robustos (HAN e KIM, 2001). Sendo assim, foram propostos os algoritmos evolucionários com inspirações quânticas que associam elementos da computação clássica a conceitos da mecânica quântica como as ondas estacionárias e a superposição linear de estados.

Um exemplo é o algoritmo evolucionário quântico QEA – *Quantum Evolutionary Algorithm* (HAN e KIM, 2002), que utiliza uma representação probabilística conhecida como bit quântico ou *qubit* como sua unidade fundamental de informação. Um indivíduo é representado por uma seqüência (*string*) de *qubits* o que permite ao QEA lançar mão do conceito quântico de superposição linear de estados, de acordo com o qual um único indivíduo é capaz de representar várias possíveis soluções num espaço de busca binário, dependendo do número de *qubits* utilizados. A atualização dos indivíduos ao longo das gerações é feita através de um operador variação conhecido como portão de rotação quântico.

WANG *et al* (2008) uniram os conceitos quânticos do QEA com a metáfora biológica da colônia de formigas do ACO para criar um novo algoritmo evolucionário: a otimização por colônias de formigas quânticas QACO – *Quantum Ant Colony Optimization*. Nele o feromônio é descrito por uma *string* de *qubits* a partir da qual são gerados os caminhos percorridos por cada uma das formigas da colônia.

No entanto, os autores do referido trabalho não mencionam o uso de nenhuma heurística ou etapa de evaporação, que desempenham papel de grande importância no ACO original, além de apresentarem resultados apenas para funções teste extremamente simples. De fato, de acordo com testes realizados no nosso trabalho anterior publicado no INAC – *International Nuclear Atlantic Conference* (SILVA *et al*, 2009a), o QACO original apresentou tendência a convergir prematuramente quando aplicado a funções numéricas complexas como a esfera *n*-dimensional ou a problemas combinatórios, evidenciada pela estagnação dos *qubits* antes que fossem obtidos resultados satisfatórios.

Os objetivos desta dissertação são:

- aplicar pela primeira vez um algoritmo evolucionário que utiliza conceitos quânticos a problemas de engenharia nuclear, em especial, a otimização da recarga nuclear;
- propor um novo algoritmo evolucionário baseado no QACO, denominado QACO_Alfa, idêntico a sua versão original exceto em dois aspectos: uma nova maneira de atualizar o feromônio (o que dá origem ao seu nome) e a inclusão de uma etapa de evaporação semelhante à que ocorre no ACO de modo a evitar a convergência prematura.

A motivação deste trabalho é mostrar que o QACO_Alfa é uma ferramenta de otimização comparável às melhores técnicas de busca disponíveis para a resolução de problemas de engenharia nuclear. Sendo assim, o capítulo 2 apresenta os fundamentos teóricos da computação quântica além de descrever o algoritmo ACO.

No capítulo 3 é apresentado o problema da otimização da recarga de ECs, particularizado para o caso de Angra1, descrevendo as simetrias do núcleo e a função *fitness* utilizada para efeitos de análise e comparação de resultados;

O capítulo 4 descreve o algoritmo evolucionário QACO, e o QACO_Alfa, desenvolvido para este trabalho. Ao longo do capítulo são apresentados os pontos em comum entre eles assim como as etapas adicionais, exclusivas do QACO_Alfa, que o tornam mais robusto e eficiente que o QACO original. Também são apresentadas as funções de teste utilizadas para efeitos de comparação entre o QACO e o QACO_Alfa. Os problemas analisados são: as funções numéricas *Himmelblau*, *Goldstein-Price* e esfera *n*-dimensional.

No capítulo 5, o QACO_Alfa é utilizado na otimização de dois problemas da engenharia nuclear de diferentes classes: para maximizar o comprimento do ciclo 7 de Angra 1 e para minimizar o número de falhas no problema de identificação de transientes, um problema numérico de diagnóstico de falhas numa usina nuclear.

O capítulo 6 encerra o trabalho e apresenta as conclusões e sugestões de aperfeiçoamento do algoritmo, possíveis aplicações futuras e considerações finais.

Capítulo 2

A Computação Quântica e os Algoritmos Evolucionários

Neste capítulo são apresentados os fundamentos teóricos da computação quântica. A transposição de conceitos da mecânica quântica para a linguagem binária utilizada nos computadores é descrita de acordo com a sua evolução ao longo dos anos desde o surgimento das primeiras idéias a respeito de um possível computador quântico até sua adequação aos limites tecnológicos existentes nos dias atuais que resultaram no surgimento dos primeiros algoritmos evolucionários com inspiração quântica.

Além disso, são introduzidos os conceitos básicos do algoritmo de otimização por colônias de formigas, ACO, um algoritmo evolucionário que utiliza a metáfora biológica da inteligência coletiva de uma colônia de formigas reais para determinar a tomada de decisão de seus agentes artificiais desenvolvido especialmente para resolver problemas combinatórios como é o caso do problema da otimização da recarga de elementos combustíveis no núcleo do reator.

2.1 A Evolução da Computação Quântica

A necessidade de realizar cálculos cada vez mais complexos foi uma das razões para o surgimento das primeiras idéias relacionadas a transpor conceitos quânticos para a linguagem dos computadores clássicos e até mesmo de criar um computador capaz de processar as informações quanticamente. Diante disso, no início da década de 80 do século passado, pesquisadores como BENIOFF (1980) propuseram a computação quântica.

Pouco tempo depois, o físico Richard Feynman (FEYNMAN, 1982) mostrou como um sistema quântico poderia ser usado para realizar cálculos computacionais e agir como um simulador para processos quânticos probabilisticamente ponderados.

Mais tarde, DEUTSCH (1985) comprovou que qualquer processo físico poderia ser perfeitamente modelado num computador quântico. Tal computador seria capaz de realizar tarefas de maneira mais rápida que as máquinas tradicionais por ter a vantagem de poder explorar o fenômeno do paralelismo quântico, ou seja, ele se comportaria como uma enorme máquina de processamento paralelo que explora cada caminho de um problema NP-completo ou NP-difícil, onde a dificuldade cresce de forma exponencial, num tempo polinomial.

Além desses importantes avanços teóricos no desenvolvimento da computação quântica, deve-se também ressaltar o trabalho de SHOR (1994) que levou à criação de algoritmos polinomiais baseados em computação quântica para resolver problemas de fatoração de números inteiros grandes (cerca de 29 dígitos) em fatores primos que são considerados NP-difíceis em computadores clássicos.

Embora as contribuições feitas por estes pesquisadores (e também por vários outros que não são citados neste trabalho) ainda não tenham sido suficientes para dar suporte à existência de um computador quântico em termos literais (nem poderia ser diferente com o nível de desenvolvimento tecnológico atual), elas fundamentam os conceitos básicos por trás dos algoritmos evolucionários quânticos desenvolvidos para atuar nos computadores disponíveis nos dias de hoje.

2.1.1 Transposição de Conceitos Quânticos para Computadores Clássicos

A unidade fundamental para o armazenamento das informações num computador é o bit. Por isso, se pretendemos transpor conceitos quânticos para a linguagem computacional, devemos correlacioná-los, ou seja, dizer qual grandeza quântica desempenhará o papel do bit.

Dentre as possibilidades comumente abordadas, destaca-se representação dos dois estados do *spin* de um elétron, *up* (\uparrow) e *down* (\downarrow) para representar os 1s e 0s dos dados binários (HEY, 1999):

$$|1\rangle \text{ e } |0\rangle \text{ correspondem a } \uparrow \text{ e } \downarrow, \text{ respectivamente} \quad (1)$$

É como se tivéssemos um sistema quântico com apenas dois estados possíveis, ou 1 ou 0 para armazenar a informação. Um sistema como esse é chamado de *quantum bit* ou *qubit*.

O conceito de *spin* eletrônico nasce da necessidade de diferenciar os elétrons na órbita de um átomo através do princípio de exclusão de Pauli (TIPLER, 2001). De acordo com este princípio, dois elétrons não podem ocupar o mesmo estado quântico simultaneamente, isto é, dois elétrons de um mesmo átomo não podem ter todos os números quânticos iguais.

O princípio de exclusão de Pauli é a razão fundamental para várias características da matéria desde a sua estabilidade até a existência das regularidades expressa pela tabela periódica dos elementos. Até a formulação deste princípio, um elétron era classificado da seguinte maneira (FELTRE, 1996):

- número quântico principal, n , que informa em qual das camadas orbitais (de K até Q, sendo K a mais externa e menos energética) o elétron se encontra e seu nível energético sendo geralmente representado por um inteiro positivo;

- número quântico secundário ou azimutal, l , que descreve as densidades de probabilidade de cada elétron nos subníveis atômicos, ou seja, a forma das órbitas

descritas pelos elétrons. Pode assumir os valores 0, 1, 2 e 3, correspondentes às subcamadas s, p, d e f ;

- número quântico magnético, m_l , fornece informações sobre a orientação de um orbital no espaço. Pode assumir valores inteiros de $+l$ a $-l$. Por exemplo, para uma subcamada d ($l = 2$), m pode ser igual a -2, -1, 0, +1 ou +2, cinco valores que correspondem aos cinco orbitais da subcamada d .

Pauli chegou empiricamente à conclusão de que com apenas estes três números não seria possível diferenciar completamente os elétrons e sugeriu a criação do que hoje é conhecido como número quântico de *spin*. Este número indica a orientação do elétron em torno do seu próprio eixo. Como existem apenas dois sentidos possíveis, este número quântico assume apenas os valores $1/2$ e $-1/2$.

De acordo com esta interpretação, o bit clássico passa a ser analisado sob a ótica da mecânica quântica e como tal está sujeito às suas leis, isto é, ao invés de representar os valores 0 ou 1 simplesmente, a notação *qubit* considera que o bit pode ser encontrado no estado 0, no estado 1 ou numa superposição linear de ambos os estados (HEY, 1999):

$$|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle \quad (2)$$

A expressão acima descreve o estado geral $|\Psi\rangle$ como uma superposição linear dos estados $|0\rangle$ e $|1\rangle$. Essa é uma versão simplificada da equação de Schroedinger que tem a característica de fornecer boas aproximações da evolução temporal de um sistema quântico sendo que Ψ é a função de onda associada aos estados 0 e 1 e α e β são números complexos que especificam as amplitudes de probabilidades dos estados correspondentes, estando sujeitos à seguinte restrição:

$$|\alpha|^2 + |\beta|^2 = 1 \quad (3)$$

sendo $|\alpha|^2$ a probabilidade de que o *qubit* seja encontrado no estado 0 e $|\beta|^2$ a probabilidade de que o *qubit* seja encontrado no estado 1. Essa restrição tem origem na seguinte expressão:

$$\int_{-\infty}^{\infty} \Psi^* \Psi dx = 1 \quad (4)$$

onde Ψ^* é o complexo conjugado de Ψ . Esta expressão garante que o elétron está em algum lugar do espaço (TIPLER, 2001).

A superposição de estados mencionada anteriormente corresponde ao fenômeno de paralelismo quântico previsto por DEUTSCH (1985), ou seja, a notação *qubit* nos permite lançar mão da representação de várias possíveis soluções num espaço de busca binário a partir de um único indivíduo. Por exemplo, se quiséssemos representar um indivíduo com dois *qubits*,

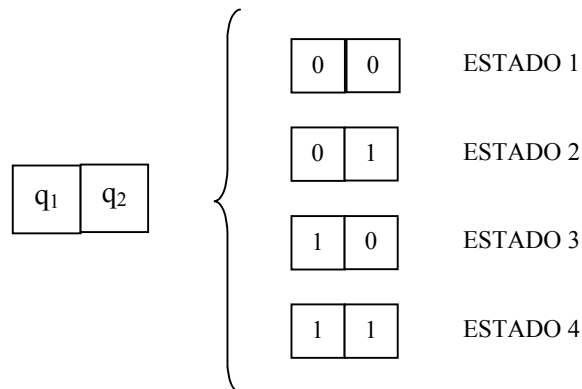


Figura 2.1: Representação binária utilizando *qubits*.

poderíamos obter como resultado qualquer um dos quatro estados quânticos descritos na figura 2.1 acima, ao passo que utilizando a notação binária convencional seriam necessários quatro indivíduos de dois bits cada um.

Os bits clássicos são atualizados através de portões reversíveis, como os operadores AND, NOT, entre outros. De maneira semelhante, foram definidos os portões quânticos que atuam como operadores unitários agindo nos estados básicos dos *qubits*. Dentre os mais conhecidos estão o portão NOT, NOT controlado, portão de rotação e Hadamard (HEY, 1999).

Em linhas gerais, a computação quântica é uma área de pesquisa que inclui conceitos como computadores quânticos e algoritmos quânticos. Embora ainda não exista um computador que seja capaz de processar as informações de maneira quântica, os algoritmos evolucionários inspirados nas teorias quânticas vêm sendo cada vez mais utilizados, obtendo resultados comparáveis aos de ferramentas de otimização tradicionais baseadas em inteligência artificial. Alguns deles serão apresentados em mais detalhes no capítulo 4.

2.2 Algoritmo de Otimização por Colônia de Formigas (ACO)

As formigas são alguns dos animais mais intrigantes da natureza. Apesar de serem criaturas rudimentares e frágeis, algumas espécies não são capazes nem mesmo de enxergar, elas possuem uma sociedade bem estruturada onde cada indivíduo desempenha uma função específica desde a mais nobre, pela qual a rainha é a responsável, até a mais básica, caso das formigas forrageadoras (GOSS *et al*, 1989).

Além disso, são extremamente fortes, resistentes a todo tipo de ambiente e de alguma maneira “sabem” que várias delas juntas conseguem transportar um peso maior que a soma das partes que cada uma transportaria se estivesse sozinha.

Mas a característica mais impressionante das formigas é sua capacidade de sempre encontrar o caminho mais curto para uma fonte de alimento, independentemente da presença de obstáculos ou alterações no ambiente de busca. Segundo estudos de especialistas (GOSS *et al*, 1989), isso é explicado por uma substância liberada pelas formigas enquanto elas caminham, o feromônio. Essa substância age como uma inteligência coletiva através da qual as formigas trocam informações entre si, decidindo qual o caminho mais promissor a ser seguido.

Funciona da seguinte maneira: uma vez que o feromônio evapora com o passar do tempo, caminhos mais longos atraem menos formigas. Em contrapartida, quanto mais curto for o caminho, maior será a chance de que outras formigas sejam atraídas pelo feromônio depositado nele. Uma vez que há novas formigas seguindo por este caminho, elas contribuem depositando mais feromônio fazendo com que ainda mais formigas sejam atraídas ao longo do tempo como pode ser visto na figura 2.2 a seguir.

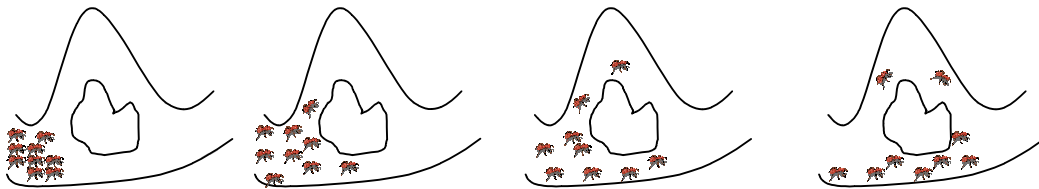


Figura 2.2: Influência do feromônio.

Dessa forma a quase totalidade das formigas acaba sendo guiada para o caminho mais curto em direção à fonte de alimento. Apenas uma pequena quantidade de formigas segue por outros caminhos, adotando uma postura investigadora na busca por fontes alternativas de alimento. O mecanismo de otimização natural observado nas colônias de formigas inspirou o desenvolvimento do algoritmo de otimização por colônias de formigas artificiais – ACO (DORIGO e GAMBARDELLA, 1996).

O ACO foi desenvolvido para resolver problemas combinatoriais complexos como o Problema do Caixeiro Viajante (PCV) (DORIGO e GAMBARDELLA, 1997). O PCV é um problema onde se deve encontrar o caminho mais curto passando por todas as cidades pelo menos uma vez e retornar a cidade de partida no final do trajeto.

Formalmente, sendo G um grafo completo $G = (V,A)$, onde V é um conjunto completo de n vértices e A é um conjunto de arestas que ligam cada par de cidades $i, j \in V$, esta formulação pode ser representada como um problema de minimização (BURIOL, 2000):

$$\text{Min} \sum_{i=1}^n \sum_{j=1}^n d_{ij} X_{ij} \quad (5)$$

Sujeito a

$$X_{ij} \in \{0,1\} \quad i, j = 1, \dots, n \quad (6)$$

$$\sum_{i=1}^n X_{ij} = 1 \quad i = 1, \dots, n \quad (7)$$

$$\sum_{j=1}^n X_{ij} = 1 \quad j = 1, \dots, n \quad (8)$$

$$\sum \sum X_{ij} \leq |S| - 1 \quad \forall S \subset V, S \neq \emptyset \quad (9)$$

A variável inteira $X_{ij} = 1$ indica que a cidade j é visitada imediatamente depois da cidade i . Caso contrário, $X_{ij} = 0$. V é o conjunto $\{1, 2, \dots, n\}$ de cidades do problema e S é um de seus subconjuntos. $|S|$ representa a cardinalidade de S . As restrições (6) e (7) garantem que para cada cidade i existe apenas uma conexão de chegada e uma de saída. A condição (9) garante que não existem sub-rotas, ou seja, rotas que não incluam todas as n cidades.

No ACO, as formigas artificiais cooperam para solucionar o problema através da aprendizagem coletiva da colônia evoluindo ao longo das gerações. Para resolver o PCV, as formigas artificiais precisam escolher as cidades que ainda não foram visitadas através de uma regra de transição de estados (DORIGO e GAMBARELLA, 1996) dada pelas equações (10) e (11) a seguir

$$S = \begin{cases} \max\{[FE(r,s)]^\delta [HE(r,s)]^\beta\} \quad s \in J_k(r) & \text{se } q \leq q_0 \\ P_k & \text{se } q > q_0 \end{cases} \quad (10)$$

onde $FE(r,s)$ é um valor real positivo que representa a quantidade de feromônio associado ao arco (r,s) , $HE(r,s)$ é o valor da função heurística relativa a se mover da cidade r para a cidade s , os parâmetros δ e β pesam a importância relativa do aprendizado $FE(r,s)$ das formigas e do conhecimento heurístico dado pela função heurística $HE(r,s)$, q é um valor aleatório com probabilidade uniforme no intervalo $[0,1]$ e q_0 ($0 \leq q_0 \leq 1$) é um parâmetro do algoritmo.

P_k é uma variável escolhida aleatoriamente em função de q que fornece a probabilidade com a qual uma formiga na cidade r escolha ir para a cidade s de acordo com a equação

$$P_k = \begin{cases} \frac{[FE(r,s)]^\delta [HE(r,s)]^\beta}{\sum_{s \in J_k(r)} [FE(r,s)]^\delta [HE(r,s)]^\beta} \quad \text{se } s \in J_k(r) \\ 0 & \text{caso contrário} \end{cases} \quad (11)$$

onde $FE(r,s)$ e $HE(r,s)$ tem o mesmo significado que na Eq.(10), bem como os parâmetros δ e β e $J_k(r)$ é a lista de cidades não visitadas por uma formiga que está na cidade r . A cooperação entre as formigas é expressa pela modificação dos valores de feromônio $FE(r,s)$ visando favorecer a descoberta de boas soluções. Essa modificação atualiza a quantidade de feromônio de duas maneiras de acordo com as equações a seguir:

2.2.1 Regra de Atualização Local

$$FE(r, s) = (1 - \rho)FE(r, s) + \rho FE_{zero} \quad (12)$$

onde ρ é o parâmetro de evaporação local de feromônio e FE_{ZERO} é a quantidade inicial de feromônio. A regra de atualização local é aplicada depois que todas as formigas tenham utilizado a regra de transição de estado e depois que todas elas tenham escolhido a próxima cidade a ser visitada.

Sendo assim, esta atualização é aplicada enquanto a solução está sendo construída. Seu objetivo é estimular a busca sobre novas regiões do espaço de soluções evitando problemas de convergência prematura. Dessa maneira, a quantidade de feromônio nos arcos é reduzida lentamente, o que possibilita que as formigas artificiais diversifiquem seu espaço de busca num processo conhecido como evaporação de feromônio (DORIGO e GAMBARDELLA, 1996).

2.2.2 Regra de Atualização Global

$$FE(r, s) = (1 - \alpha)FE(r, s) + \alpha(w/bfit) \quad (13)$$

onde α é o parâmetro de evaporação global de feromônio, w é o parâmetro definido pelo usuário que associado ao parâmetro α expressa a média do aprendizado do algoritmo e $bfit$ é a melhor *fitness* da configuração atual. A regra de atualização global é aplicada depois que todas as formigas tenham construído um caminho completo e é por isso considerada o reforço de aprendizado do algoritmo. Um pseudo-código do ACO é mostrado na figura 2.3.

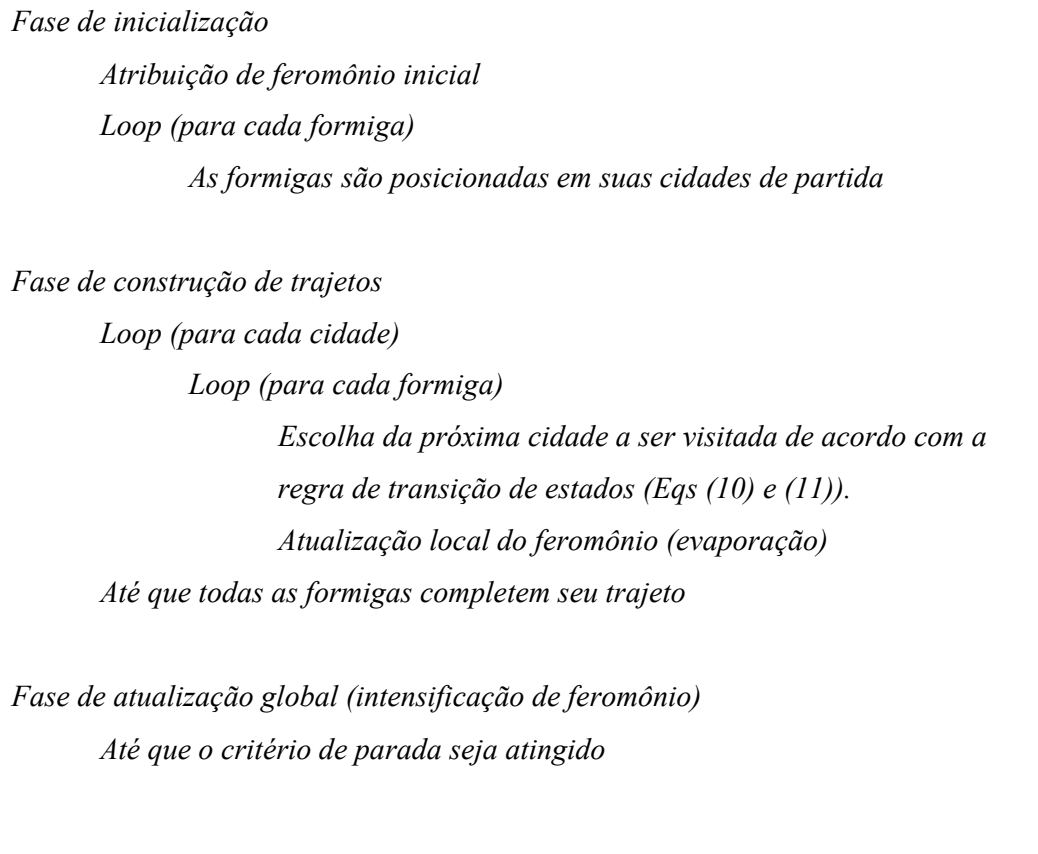


Figura 2.3: Pseudo-código do ACO

Capítulo 3

O Problema da Otimização da Recarga de Elementos Combustíveis

Este capítulo é destinado a um dos problemas mais complexos e de grande interesse da área de Engenharia Nuclear: o problema da otimização da recarga de elementos combustíveis no núcleo do reator. Tendo como exemplo o reator do tipo PWR utilizado em Angra 1, são apresentados:

A recarga nuclear - o que é, quando ocorre e o que precisa ser levado em conta com relação à sua otimização;

Modelagem do núcleo – de que maneira o núcleo é interpretado pela ferramenta de otimização utilizada neste trabalho;

Função *fitness* – de que maneira são avaliadas as soluções obtidas pelos agentes artificiais a partir da análise do espaço de busca.

3.1 A Recarga Nuclear

Diz-se que uma usina está operando à plena potência, ou em sua potência nominal, quando a energia produzida pelos elementos combustíveis que compõem o núcleo de seu reator é a maior possível, dentro dos limites de segurança que garantem a manutenção de sua integridade física.

Com o passar do tempo, o combustível nuclear, ou seja, as pastilhas de dióxido de urânio (UO_2) dentro das varetas que formam os ECs, vai sendo consumido até chegar ao ponto em que não é mais possível para a usina continuar produzindo energia à plena potência (CHAPOT, 2000).

Neste momento, a usina deve ser desligada para que o núcleo seja recarregado. Os ECs queimados são então substituídos por novos e permanecem no núcleo até que seja necessária a realização de uma nova recarga. O intervalo de tempo entre as recargas é denominado ciclo de operação. Entretanto, a queima dos ECs não ocorre de maneira uniforme, ela depende da posição que cada um deles ocupa no núcleo, e nem todos precisam ser substituídos ao mesmo tempo.

Torna-se necessário encontrar uma configuração de embaralhamento (também conhecida como padrão de recarga) entre os novos e antigos ECs que seja capaz de manter a usina funcionando em sua potência nominal, por um período ditado pelo critério de otimização adotado e, conforme o objetivo, aumentar o tamanho do ciclo de operação. Este processo é conhecido como problema da otimização da recarga de ECs no núcleo do reator (CHAPOT, 2000).

3.2 A Modelagem do Núcleo

A otimização da recarga é conhecida por ser um problema de difícil resolução onde a complexidade cresce exponencialmente com o número de ECs considerados. No caso de Angra 1, que contém 121 ECs em seu núcleo, o número de padrões de recarga possíveis é de cerca de 10^{273} (LIMA,2005).

A existência de condições de simetria de 1/4 e 1/8 associada às restrições impostas ao posicionamento dos ECs no núcleo reduz este número para algo em torno de 10^{25} padrões de recarga que, ainda assim, é um valor alto para ser analisado. O núcleo de Angra 1 e suas linhas de simetria são mostrados na figura 3.1 a seguir.

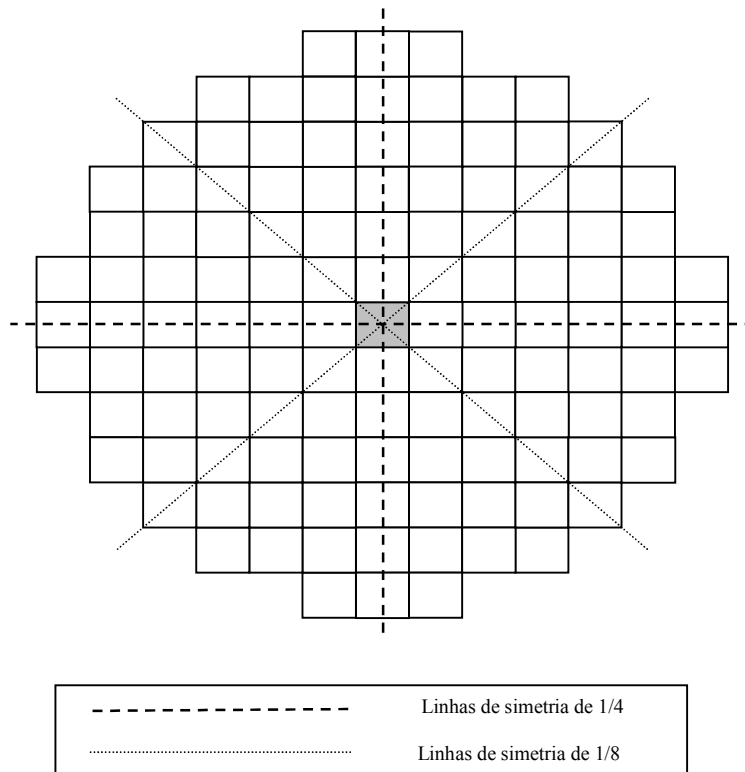


Figura 3.1: Linhas de simetria do núcleo de Angra 1

Os eixos vertical e horizontal dividem o núcleo em quatro regiões idênticas, comumente conhecidas como quadrantes. Na figura 3.1 também é possível observar a presença de dois eixos secundários que dividem cada um dos quadrantes em duas regiões menores, totalizando oito sub-regiões chamadas de octantes (CALDAS, 2006).

Cada elemento posicionado sobre a linha, coluna ou diagonal cortada por um desses eixos recebe o nome de elemento de quarteto. Isso se deve ao fato que existem mais três posições além da atual que este elemento poderá ocupar no caso de não ser descartado ao final de um ciclo de recarga (CHAPOT, 2000).

Os demais elementos são chamados de elementos de octeto (em destaque na figura 3.2). Cada um deles pode ocupar mais sete posições simétricas à sua posição atual em caso de reaproveitamento ao final de um ciclo de recarga. O elemento central é considerado fixo.

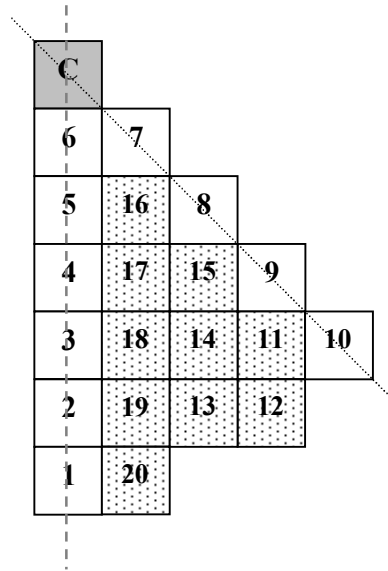


Figura 3.2: Região de simetria de octante

Na prática, os elementos que ocupam posições de octeto podem ser posicionados em posições de elementos de quarteto, o mesmo não podendo ser dito em relação aos elementos de quarteto que podem apenas ocupar posições de elementos de quarteto no núcleo. Para efeitos de análise e comparação de resultados, será utilizada neste trabalho a simetria de 1/8 de núcleo. Isso é feito de duas maneiras distintas:

Primeiro caso: os elementos numerados de 1 a 10 da figura 3.2 só podem ser trocados entre si, o mesmo ocorrendo com os elementos numerados de 11 a 20. Este é o caso real, onde os elementos de quarteto são trocados apenas por um outro elemento de quarteto.

Segundo caso: todos os elementos são considerados como sendo elementos de octeto sendo permitido que qualquer um dos elementos ocupe as posições de 1 a 20 sem a imposição de restrições. Esse caso não tem sentido físico, uma vez que essa situação seria inviável na prática (só seria possível no primeiro ciclo de uma usina, quando todos os elementos apresentam as mesmas características de queima igual a zero) e é considerado aqui apenas para fins de validação e comparação de resultados do algoritmo.

3.3 A Função *Fitness*

O tamanho de um ciclo de recarga é medido em função do número de dias em que a usina permanece em atividade à plena potência até que se torne necessário repetir o procedimento de substituição de ECs. À quantidade de dias que compõem este período dá-se o nome de número de dias efetivos à plena potência - DEPP - (CHAPOT, 2000).

Embora o processo de otimização da recarga seja de natureza multi-objetiva (LIMA, 2005), ele pode ser analisado sob a ótica da maximização do número de DEPPs. Neste trabalho, a função *fitness* será definida em função deste objetivo em particular. Para determiná-la são analisados dois parâmetros principais: a concentração crítica de boro (B_c) no moderador e o fator de pico de potência radial (F_{XY}) de cada uma das varetas dos ECs.

De acordo com MACHADO (1998), F_{XY} é 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. No caso de Angra 1, F_{XY} deve ser menor ou igual a 1,435 (RFAS,1980).

A função do boro é, juntamente com as barras de controle e os venenos queimáveis, controlar o número de nêutrons produzidos no núcleo, ou seja, controlar o valor do fator de multiplicação k impedindo que ele assuma valores maiores que 1. O

valor de k é obtido calculando-se a razão entre o número de nêutrons produzidos em um geração e o número de nêutrons produzidos na geração precedente (DUDERSDADT, 1976).

$$k = \frac{n^{\circ} \text{ nêutrons em } (t)}{n^{\circ} \text{ nêutrons em } (t-1)} \quad (14)$$

Quando $k < 1$, o reator atua em regime subcrítico. Numa situação como essa, o número de nêutrons decresce, geração após geração, e, logo, a reação em cadeia se extingue. O contrário ocorre quando $k > 1$. Nesse caso o número de nêutrons produzidos cresce exponencialmente até chegar ao ponto em que não é mais possível controlar a quantidade de energia produzida no núcleo o que pode levar a fusão do mesmo. Portanto, é preciso encontrar meios de manter o reator crítico, isto é, com $k = 1$, para que o número de nêutrons produzidos esteja sempre sob controle.

Para realizar o procedimento de otimização da recarga a ferramenta utilizada neste trabalho foi associada ao código nodal de física de reatores conhecido como RECNOB que será apresentado em maiores detalhes no Apêndice. Uma vez que o RECNOB não é capaz de calcular o fator de pico de potência radial, utilizamos a potência média para determinar o valor da função *fitness*.

De acordo com o trabalho de CHAPOT (2000), desde que a potência média produzida no núcleo não ultrapasse o valor 1,395 que corresponde ao valor de 1,435 para o fator de pico de potência radial de Angra 1 no sistema genético GENESIS/ALGER/ANC, para efeito do estudo de otimização, podemos assumir que o fator de pico de potência está dentro dos limites de segurança.

O objetivo é encontrar um padrão de recarga que maximize a concentração crítica de boro que satisfaça a restrição de $P_m \leq 1,395$. Existem várias funções *fitness* que podem ser utilizadas para analisar este problema. Neste trabalho adotamos a função descrita a seguir para que seja possível comparar os nossos resultados com os de outros trabalhos onde ela também foi empregada.

$$Fitness = \begin{cases} P_m & \text{se } P_m > 1,395 \\ 1/B_c & \text{se } P_m \leq 1,395 \end{cases} \quad (15)$$

Onde P_m é a potência média no EC e B_c é a concentração crítica de boro. Enquanto a potência média for maior que 1,395, a função *fitness* age minimizando este valor independente do valor de B_c , que permanece variando ao longo do tempo. Uma vez que um padrão de recarga válido é encontrado, a função *fitness* passa a maximizar a concentração de boro (LIMA, 2005) como é ilustrado na figura 3.3 a seguir.

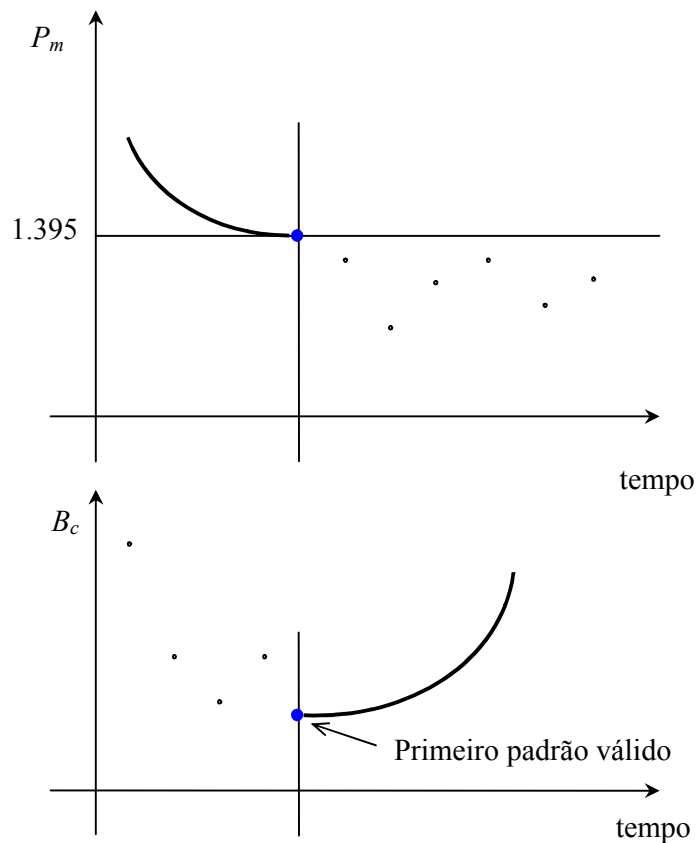


Figura 3.3: Comportamento da função *fitness*

Deve-se ressaltar que, apesar desta função utilizar como parâmetros a concentração de boro e a potência média, o cálculo do número de DEPPs depende única e exclusivamente da concentração de boro no moderador. O fator de pico de potência radial (ou sua aproximação na forma de potência média) é um valor relacionado à segurança da usina.

Cada DEPP de Angra 1 corresponde ao consumo de aproximadamente 4 ppm de boro (CHAPOT, 2000). Baseado nisso, poderia-se imaginar que quanto maior o valor de B_c maior seria o número de DEPPs. Isso é verdade até certo ponto, pois, como já foi discutido na seção anterior, o valor de B_c é uma das principais componentes de controle da reatividade no núcleo, ou seja, um valor muito alto impedirá que a usina seja ligada.

Capítulo 4

Otimização com Agentes Quânticos

Os algoritmos evolucionários com inspirações quânticas foram criados para simular em computadores clássicos a forma como se imagina hoje em dia que um computador quântico processaria as informações. Na verdade, eles são capazes de fazer uso de alguns conceitos oriundos deste ramo da ciência como as ondas estacionárias e a superposição linear de estados apenas de uma maneira metafórica, ainda que eficiente.

Um exemplo deste tipo de ferramenta é o algoritmo QEA que serviu de inspiração para o desenvolvimento da otimização por colônias de formigas quânticas, o QACO original, cujos fundamentos teóricos são descritos neste capítulo. Além disso, é apresentado o novo algoritmo QACO_Alfa que é o produto final desta dissertação de mestrado. O QACO_Alfa é uma versão modificada do QACO onde foi incluída uma nova forma de atualização dos *qubits* além de uma etapa de pseudo-evaporação semelhante àquela realizada no ACO.

O capítulo apresenta ainda as funções teste utilizadas para fins de comparação entre a capacidade de otimização do QACO original e do QACO_Alfa.

4.1 Quantum Ant Colony Optimization (QACO)

O QEA – *Quantum Evolutionary Algorithm* (HAN e KIM, 2002) é um algoritmo evolucionário com inspirações quânticas que utiliza uma nova forma de retratar o estado binário, a representação probabilística conhecida como *qubit*

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} \quad (16)$$

onde α e β são números complexos sendo que $|\alpha|^2$ é a probabilidade de que o *qubit* seja encontrado no estado “0” e $|\beta|^2$ é a probabilidade de que o *qubit* seja encontrado no estado “1”, obedecendo à restrição

$$|\alpha|^2 + |\beta|^2 = 1 \quad (17)$$

O papel de agente quântico é desempenhado no QEA por indivíduo *Qubit* r_i , representado por uma sequência de m *qubits*.

$$r_i = \begin{bmatrix} \alpha_1 & \alpha_2 & \cdots & \alpha_m \\ \beta_1 & \beta_2 & \cdots & \beta_m \end{bmatrix} \quad (18)$$

onde i varia de 1 até o número total de indivíduos da população. Uma vez que cada *qubit* pode representar dois estados quânticos (0 ou 1) cada indivíduo r_i é capaz de representar 2^m estados, ou seja, 2^m possíveis soluções ao mesmo tempo, onde m é o número de *qubits* considerados. Entretanto, no momento em que esse indivíduo é observado, todos os m possíveis estados colapsam para um único estado.

Essa característica é análoga ao fenômeno estudado pela mecânica quântica conhecido como superposição linear de estados quânticos (TIPLER, 2001). De acordo com ele, todos os possíveis estados quânticos de um elétron colapsam para um único estado no momento em que ele é observado.

O mesmo ocorre com o indivíduo r_i , onde todas as possíveis soluções que ele seria capaz de representar acabam colapsadas em uma única solução. Por exemplo, o indivíduo *Qubit* com $m = 3$ a seguir

$$\left[\begin{array}{c|c|c} \alpha_1 & \alpha_1 & \alpha_3 \\ \beta_2 & \beta_2 & \beta_3 \end{array} \right] = \left[\begin{array}{c|c|c} \frac{1}{3} & \frac{\sqrt{2}}{2} & \frac{-\sqrt{3}}{2} \\ \frac{2\sqrt{2}}{3} & -\frac{\sqrt{2}}{2} & \frac{1}{2} \end{array} \right] \quad (19)$$

é capaz de representar os 2^3 estados descritos abaixo

$$-\frac{\sqrt{6}}{12}|000\rangle + \frac{\sqrt{2}}{12}|001\rangle + \frac{\sqrt{6}}{12}|010\rangle - \frac{\sqrt{2}}{12}|011\rangle - \frac{\sqrt{3}}{3}|100\rangle + \frac{1}{3}|101\rangle + \frac{\sqrt{3}}{3}|110\rangle - \frac{1}{3}|111\rangle \quad (20)$$

Isto quer dizer que as probabilidades dos estados $|000\rangle$, $|001\rangle$, $|010\rangle$, $|011\rangle$, $|100\rangle$, $|101\rangle$, $|110\rangle$ e $|111\rangle$ são $1/24$, $1/72$, $1/24$, $1/72$, $1/3$, $1/9$, $1/3$, $1/9$, respectivamente. Cada um desses estados binários representa uma possível solução sendo que apenas uma delas será observada.

A representação *qubit* e o conceito de superposição linear de estados empregados no QEA foram adaptados à metáfora biológica do reforço de feromônio utilizada na otimização por colônias de formigas (ACO) por WANG *et al* (2008) para dar origem ao algoritmo de otimização por colônias de formigas quânticas, o QACO.

Trata-se de um novo algoritmo evolucionário com inspirações quânticas onde o feromônio τ é formado por uma *string* de *qubits*.

$$\tau = \left[\begin{array}{c|c|c|c} \tau_{\alpha,1} & \tau_{\alpha,2} & \cdots & \tau_{\alpha,m} \\ \tau_{\beta,1} & \tau_{\beta,2} & \cdots & \tau_{\beta,m} \end{array} \right] \quad (21)$$

Da mesma forma que o indivíduo r_i do QEA é capaz de representar 2^m possíveis soluções o feromônio τ do QACO pode representar 2^m caminhos binários. Cada um desses caminhos é associado de maneira metafórica a uma formiga que caminha do formigueiro em direção a uma possível fonte de alimento. Isso quer dizer que quando se faz referência a uma formiga quântica o que será avaliado na verdade é o caminho binário associado a ela.

Para que os agentes quânticos sejam capazes de resolver os problemas de otimização é necessário converter a notação de *qubits* para a forma binária convencional ou, em outras palavras, uma seqüência de zeros e uns. Isto é feito no QACO observando o feromônio τ . A partir dele são geradas soluções válidas mostradas na forma de caminhos binários percorridos pelas formigas quânticas.

Por exemplo, cada par de nós ‘0’ e ‘1’ na figura 4.1 é representado por um *qubit*, isto é, τ_α e τ_β do *qubit* representam os nós ‘0’ e ‘1’, enquanto $|\tau_\alpha|^2$ e $|\tau_\beta|^2$ são as probabilidades de que as formigas escolham o caminho para o nó ‘0’ e ‘1’ respectivamente.

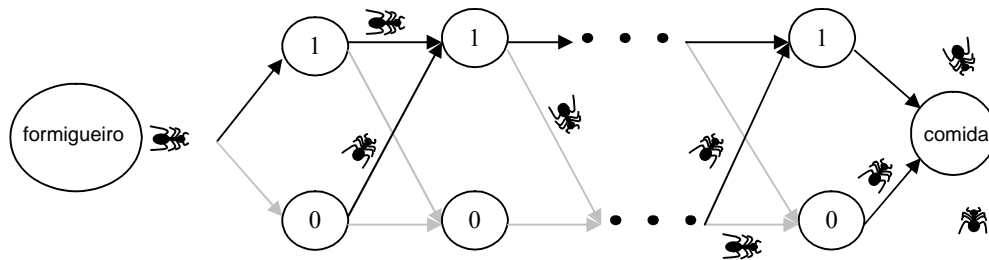


Figura 4.1: Caminho binário construído pelas formigas quânticas

Para determinar para qual estado quântico o *qubit* irá colapsar, ou seja, em qual estado quântico ele será observado (0 ou 1), primeiro um número randômico p é gerado e comparado com o parâmetro de probabilidade de exploração p_e . Se p for maior ou igual a p_e , o j -ésimo bit que compõe o caminho da i -ésima formiga é determinado pela equação a seguir

$$\text{Caminho}_{i,j} = \begin{cases} 0 & \text{if } \tau_{\beta,j} \leq \tau_{\alpha,j} \\ 1 & \text{if } \tau_{\beta,j} > \tau_{\alpha,j} \end{cases} \quad (22)$$

onde $\text{Caminho}_{i,j}$ é o j -ésimo bit da i -ésima formiga. Se p for menor que p_e , o j -ésimo bit que compõe o caminho da i -ésima formiga é determinado pela seguinte função limite onde c é um número randômico e η_0 é uma constante.

$$\eta_c(x) = \begin{cases} 0 & c < \eta_0 \\ 1 & c \geq \eta_0 \end{cases} \quad (23)$$

O parâmetro p_e utilizado no QACO desempenha exatamente a mesma função que o parâmetro q_0 no ACO. Ambos determinam se o processo de tomada de decisão será determinístico ou probabilístico. Isto quer dizer que quanto maior for o valor assumido por q_0 no ACO ou p_e no QACO, maior será a chance de que a próxima decisão tomada seja determinística, baseada no aprendizado que o algoritmo obteve ao longo das gerações em lugar de ser feita de modo probabilístico.

Por outro lado, baixos valores de q_0 ou p_e em seus respectivos algoritmos privilegiam uma tomada de decisão probabilística sem levar em conta as informações obtidas pelo algoritmo durante a evolução das gerações. As equações (22) e (23) correspondem à regra de transição de estado expressa pelas equações (10) e (11) no capítulo 2.

Depois que o feromônio τ colapsa para os caminhos binários convencionais a *fitness* de cada formiga é avaliada e os *qubits* são atualizados através do portão de rotação quântico $R(\theta_{id})$

$$\begin{bmatrix} \tau'_{\alpha j} \\ \tau'_{\beta j} \end{bmatrix} = R(\theta_j) \begin{bmatrix} \tau_{\alpha j} \\ \tau_{\beta j} \end{bmatrix} = \begin{bmatrix} \cos(\theta_j) & -\sin(\theta_j) \\ \sin(\theta_j) & \cos(\theta_j) \end{bmatrix} \cdot \begin{bmatrix} \tau_{\alpha j} \\ \tau_{\beta j} \end{bmatrix} \quad (24)$$

onde θ_j é o ângulo de rotação, responsável por fazer as formigas quânticas convergirem para estados mais aptos. O ângulo de rotação não possui base teórica sendo definido como

$$\theta_j = S(\tau_{\alpha j}, \tau_{\beta j}) \cdot \Delta\theta_j \quad (25)$$

onde $S(\tau_{\alpha j}, \tau_{\beta j})$ é o sinal de θ_j que determina a direção e $\Delta\theta_j$ é a amplitude do ângulo de rotação. Esses valores são determinados, no QACO, pela tabela 4.1 abaixo:

Tabela 4.1. Ângulo de rotação do QACO (FONTE: WANG *et al*, 2008)

x_j	b_j	$f(x) > f(b)$	$\Delta\theta_j$	$S(\tau_{\alpha j}, \tau_{\beta j})$			
				$\tau_{\alpha j} \tau_{\beta j} > 0$	$\tau_{\alpha j} \tau_{\beta j} < 0$	$\tau_{\alpha j} = 0$	$\tau_{\beta j} = 0$
0	0	False	0.01π	-1	+1	± 1	± 1
0	0	True	0.01π	-1	+1	± 1	± 1
0	1	False	0.025π	-1	+1	± 1	± 1
0	1	True	0.025π	+1	-1	± 1	± 1
1	0	False	0.025π	+1	-1	± 1	± 1
1	0	True	0.025π	-1	+1	± 1	± 1
1	1	False	0.01π	+1	-1	± 1	± 1
1	1	True	0.01π	+1	-1	± 1	± 1

onde x_i é o i -ésimo bit do caminho percorrido pela formiga x , b_i é o i -ésimo bit do caminho percorrido pela melhor formiga, $f(x)$ é a *fitness* do caminho percorrido pela formiga x e $f(b)$ é a *fitness* do caminho percorrido pela melhor formiga.

Com os valores obtidos na tabela 4.1, é possível representar de que forma o *qubit* será atualizado. A figura 4.2 mostra como isso é feito.

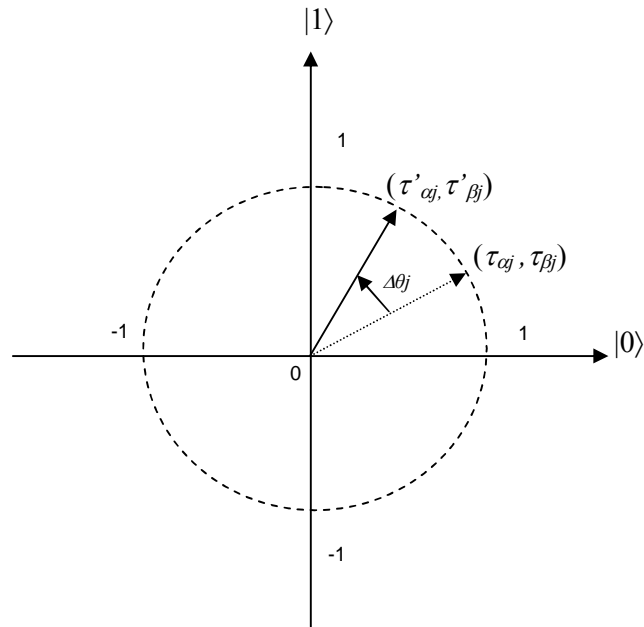


Figura 4.2: Atualização de um *qubit* no plano cartesiano (FONTE: HAN e KIM, 2002)

onde o eixo das ordenadas representa a probabilidade do estado “1” e o eixo das abscissas representa a probabilidade do estado “0”. Um *qubit* $(\tau_{\alpha_j}, \tau_{\beta_j})$ sofre uma rotação de $\Delta\theta_j$, tornando-se $(\tau'_{\alpha_j}, \tau'_{\beta_j})$.

De acordo com a tabela 4.1, ter um bit igual a 0 na posição j da sequência binária fornece um caminho menos promissor dado o aumento na probabilidade do estado “1” e, conseqüentemente, a redução na probabilidade do estado “0”, para o exemplo fictício apresentado na figura 4.2.

Uma vez que as formigas quânticas trabalham em problemas de otimização no espaço binário, isso lhes permite resolver problemas contínuos diferente do ACO que se

destaca na resolução de problemas combinatoriais complexos, mas não é indicado para problemas no espaço real. Na figura 4.3 é apresentado um pseudo-código do QACO original.

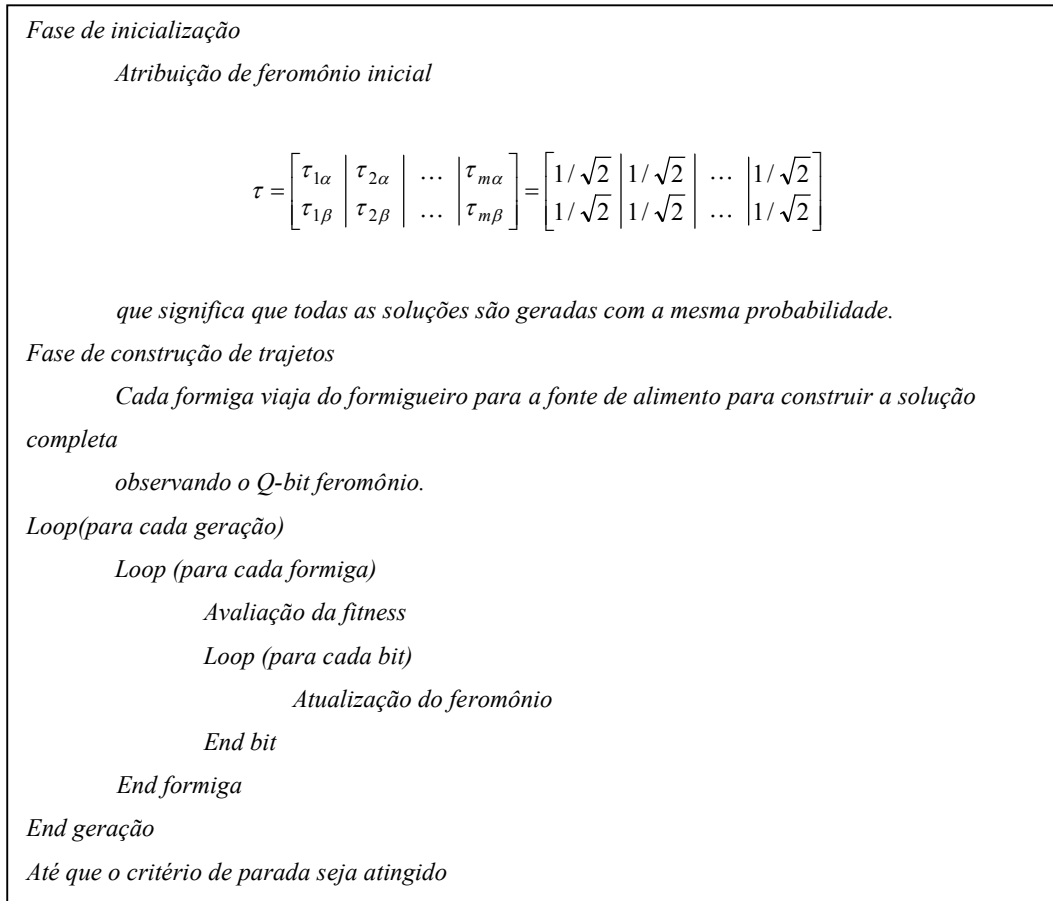


Figura 4.3. Pseudo-código do QACO Original

4.1.1 Estagnação dos Bits

A representação probabilística dos caminhos descritos por *strings* binárias utilizada no QACO original é idêntica à forma de representar os indivíduos no QEA. A atualização dos *qubits*, por outro lado, apresenta algumas diferenças nos dois algoritmos ainda que o portão de rotação quântico por meio do qual essas atualizações são efetuadas seja o mesmo.

Esse portão age, no QACO, aplicando pequenos incrementos no ângulo que o vetor *qubit* faz com o eixo das abscissas. Em outras palavras, o vetor *qubit* gira pelos quadrantes no plano cartesiano, ora se aproximando do eixo das ordenadas, ora se aproximando do eixo das abscissas como é ilustrado no exemplo da figura 4.4, a seguir.

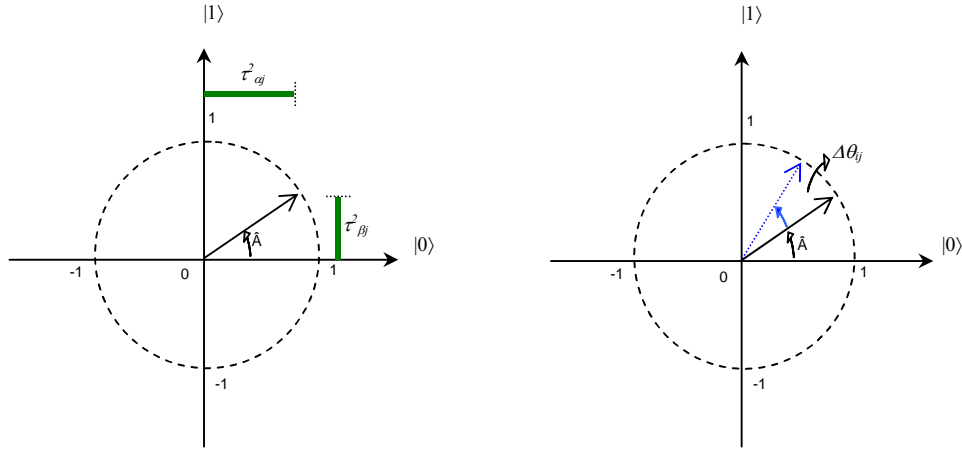


Figura 4.4: Exemplo de ação do portão quântico sobre um *qubit* quando o estado $|1\rangle$ se torna mais provável ($\tau^2_{\beta j} > \tau^2_{\alpha j}$).

Uma vez que, dentro de uma mesma geração, cada um dos i agentes avaliados contribui para atualizar os *qubits* através de um incremento $\Delta\theta_j$, ao final dessa geração o *qubit* da posição j terá sofrido uma rotação total que é igual a

$$\sum_{i=1}^{n^0 \text{ agentes}} \Delta\theta_{ij} \quad (26)$$

O resultado disto pode muitas vezes ser uma rotação total que não altera o valor atual de sua probabilidade de ser encontrado no estado 0 ou no estado 1. Na figura 4.5(a) é mostrado o exemplo do ângulo total sendo igual a múltiplos inteiros de 2π . Neste caso, o *qubit* gira por todos os quadrantes e retorna à sua posição inicial.

Em 4.5(b), é ilustrado que algo semelhante ocorre quando o ângulo de rotação total aplicado ao *qubit* é proporcional a múltiplos inteiros de π . O mesmo ocorre quando o vetor *qubit* sofre um ângulo de rotação total que é igual ao dobro do complemento ou

do suplemento do ângulo original formado entre ele e o eixo das abscissas, situação ilustrada nas figuras 4.5(c) e 4.5(d) respectivamente.

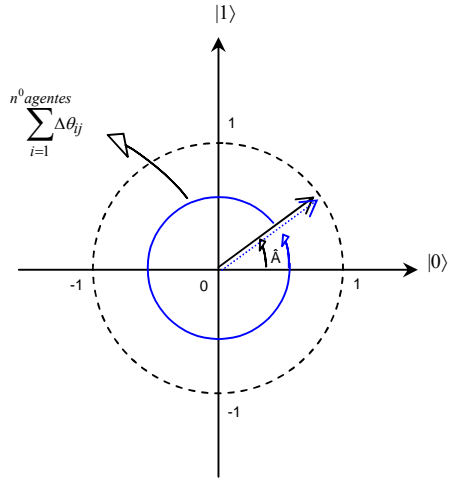


Figura 4.5(a)

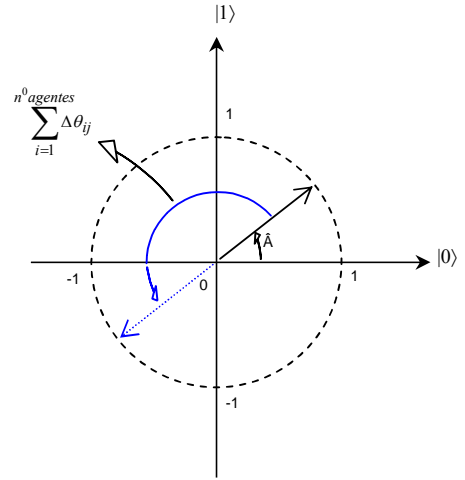


Figura 4.5(b)

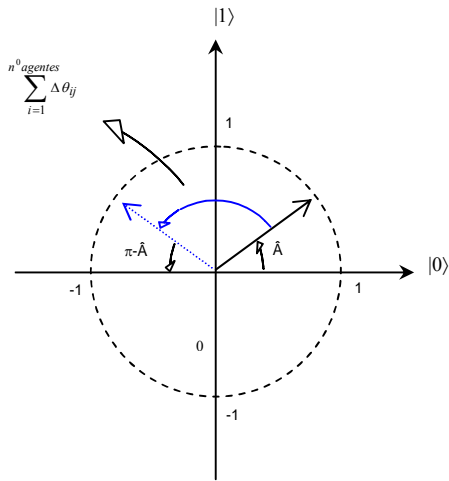


Figura 4.5(c)

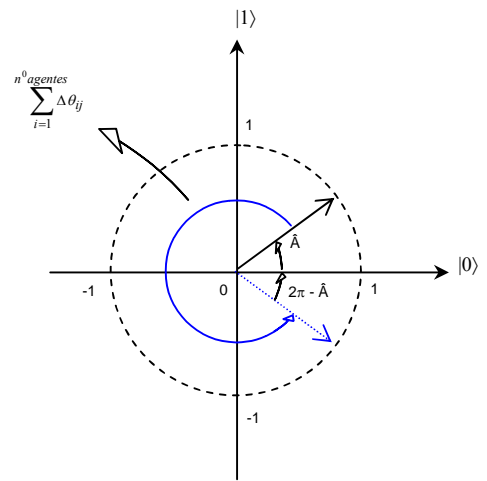


Figura 4.5(d)

Figura 4.5: Casos particulares de estagnação

Nos exemplos mostrados na figura 4.5, a probabilidade do estado dominante atual do *qubit* (0 ou 1) não sofre alteração, ele é o mesmo antes e depois da atualização ser efetuada. Sendo assim, não há aprendizagem efetiva e o indivíduo *Qubit* não consegue evoluir para estados mais aptos. Eles são casos particulares do fenômeno conhecido como estagnação.

À medida que os agentes quânticos evoluem no tempo, existe uma tendência a reforçar a probabilidade de um dos estados do *qubit* para fazer com as novas gerações de agentes se pareçam cada vez mais com o melhor indivíduo encontrado até então. A estagnação consiste no *qubit* de uma determinada posição i ficar “preso” em 0 ou 1, ou seja, a probabilidade de que ele esteja no estado 1(0) é 100% enquanto a probabilidade de que ele esteja no estado 0 (1) é nula.

4.1.2 Dependência de Parâmetros

Além de apresentar problemas com estagnação de *qubits*, o QACO também sofre com a dependência de parâmetros. A escolha do valor apropriado do parâmetro de probabilidade de exploração p_e , por exemplo, pode representar a diferença entre sucesso e fracasso do processo de otimização da função analisada.

Este parâmetro determina se os agentes irão explorar o espaço de busca de modo determinístico ou probabilístico. Quanto maior o valor de p_e , maior será a tendência de um agente ser guiado pelo que o algoritmo aprendeu sobre o problema em lugar de tomar decisões ao acaso.

Os testes realizados no trabalho original deixaram claro que valores de p_e iguais a 1, 0.9 e 0.7 apresentam, dentro da análise de um mesmo problema resultados completamente diferentes como pode ser observado pelos dados da tabela 4.2.

Tabela 4.2: Resultados de p_e (FONTE: WANG *et al*, 2008)

<i>Função Himmelblau</i>		
p_e	Encontra o valor ótimo (%)	Valor ótimo médio
0.7	75	0.0223
0.9	89	0.1462
1	0	39.29

Também foi observado que o número de formigas e o número de bits são parâmetros que influenciam diretamente o comportamento do algoritmo, principalmente no tocante ao desempenho, mesmo em funções bidimensionais. Uma vez que cada agente contribui para determinação do incremento total aplicado ao ângulo de cada *qubit*, quanto maior o número de agentes maior será a tendência a convergir prematuramente caracterizando a estagnação.

O mesmo ocorre em relação ao número de bits utilizados, já que quanto maior for este número, mais refinado é o processo de busca, tornando mais fácil que ocorra a convergência prematura.

4.2 Otimização por Colônias de Formigas Quânticas_Alfa (QACO_Alfa)

O QACO original trabalha bem em funções teste bidimensionais (WANG *et al*, 2008), porém, de acordo com os testes realizados neste trabalho, cujos resultados serão apresentados na seção 4.3, não obtém o mesmo desempenho quando aplicado a funções multidimensionais complexas devido à sua tendência a convergir prematuramente.

Isso ocorre porque existe reforço de feromônio mesmo quando os bits comparados são iguais ($0 \rightarrow 0$ ou $1 \rightarrow 1$), o que contribui para a estagnação. Diante disso, foram feitas alterações no algoritmo original para evitar a estagnação dos *qubits* impedindo que o mesmo convirja antes de obter bons resultados. Tais alterações deram origem ao QACO_Alfa.

O QACO_Alfa é uma versão alternativa desenvolvida numa tentativa de superar as limitações relacionadas a convergência prematura observadas no QACO original. A forma como atuam estas alterações será apresentada em maiores detalhes ao longo desta seção.

4.2.1 Reforço de Feromônio

No QACO_Alfa a atualização dos *qubits* é feita em duas etapas. Primeiro aplica-se o portão de rotação quântico responsável por girar o vetor *qubit* ao longo dos quadrantes exatamente como ocorre no QACO original. No entanto, o efeito da atualização é diferente nos dois algoritmos.

Originalmente, o bit da j -ésima posição da solução i é comparado com o bit da mesma posição da melhor solução global (ver tabela 4.1); dependendo do valor da *fitness*, é determinado o ângulo de atualização que será aplicado ao *qubit*.

Isso também ocorre no QACO_Alfa, no entanto, a atualização do ângulo passa a ser feita somente quando os bits comparados forem diferentes e passa a depender do fator de inércia α_0 , uma constante no intervalo $[0,1]$, utilizada para efeito de ajuste no valor da atualização de acordo com a tabela 4.3:

Tabela 4.3. Ângulo de Rotação do QACO_Alfa

x_i	b_i	$f(x) > f(b)$	$\Delta\theta_j$	$S(\tau_{\alpha,j}, \tau_{\beta,j})$			
				$\tau_{\alpha,j} \tau_{\beta,j} > 0$	$\tau_{\alpha,j} \tau_{\beta,j} < 0$	$\tau_{\alpha,j} = 0$	$\tau_{\beta,j} = 0$
0	1	False	$0.025\pi^* \alpha_0$	-1	+1	± 1	± 1
0	1	True	$0.025\pi^* \alpha_0$	+1	-1	± 1	± 1
1	0	False	$0.025\pi^* \alpha_0$	+1	-1	± 1	± 1
1	0	True	$0.025\pi^* \alpha_0$	-1	+1	± 1	± 1

onde x_i , b_i , $f(x)$ e $f(b)$ têm o mesmo significado que o descrito para a tabela 4.1.

O fator de inércia α_0 tem a função de reduzir o efeito do ângulo de rotação aplicado ao *qubit* no momento da atualização. Ele atua como um “freio”, na tentativa de impedir a convergência prematura.

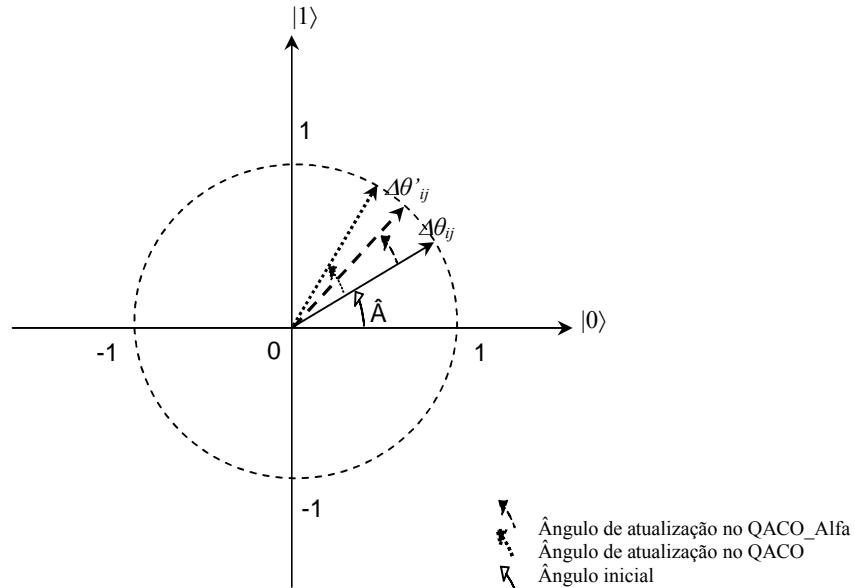


Figura 4.6: Atualização do qubit no QACO_Alfa

Na figura 4.6 é possível observar que o ângulo de rotação aplicado a um *qubit* no QACO ($\Delta\theta'_{ij}$ para o vetor pontilhado) é maior que o aplicado ao *qubit* no QACO_Alfa ($\Delta\theta_{ij}$ para o vetor tracejado). Esse efeito se deve à constante de inércia α_0 , que reduz o incremento aplicado ao ângulo original entre o *qubit* e o eixo das abscissas permitindo controlar de maneira mais eficiente a evolução dos agentes quânticos e impedir a convergência prematura.

Embora a presença da constante de inércia α_0 seja importante na evolução dos indivíduos ao longo das gerações, ela não é suficiente para torná-lo imune aos efeitos da convergência prematura, sendo capaz apenas de contribuir para minimizar seus efeitos. Diante disso, foi desenvolvida uma etapa adicional de evaporação para tornar o QACO_Alfa mais robusto e eficiente como ferramenta de otimização.

4.2.2 Evaporação de Feromônio

Nessa segunda etapa, o *qubit* da posição j , que acabou de ser atualizado, é submetido a um outro portão quântico para que a ele seja aplicado o ângulo de evaporação $\omega(j)$.

$$\begin{bmatrix} \tau'_{\alpha j} \\ \tau'_{\beta j} \end{bmatrix} = R(\omega(j)) \begin{bmatrix} \tau_{\alpha j} \\ \tau_{\beta j} \end{bmatrix} \quad (27)$$

onde $R(\omega(j))$ é o portão de rotação utilizado na evaporação

$$R(\omega_j) = \begin{bmatrix} \cos(\omega(j)) & -\sin(\omega(j)) \\ \sin(\omega(j)) & \cos(\omega(j)) \end{bmatrix} \quad (28)$$

Este ângulo age diminuindo a probabilidade do estado mais observado ('0s' ou '1s') na mesma posição j das demais *strings* que compõem a população. Um exemplo para uma população de cinco formigas, cada uma com seu respectivo caminho de cinco bits, é mostrado na figura 4.7:

Caminho da formiga 1	1	0	0	0	1
Caminho da formiga 2	1	1	1	0	0
Caminho da formiga 3	0	0	1	1	1
Caminho da formiga 4	1	0	0	1	0
Caminho da formiga 5	1	0	1	0	1

Figura 4.7. Determinação de ZERO(j) e UM(j)

Para cada coluna, ou posição j de todos os indivíduos da população dentro de uma mesma geração, é contado o número total de ‘0s’ e ‘1s’ para determinar o valor de $ZERO(j)$ e $UM(j)$, respectivamente. A coluna em destaque ($j = 1$) possui quatro ‘1s’ e um ‘0’. Logo, $UM(1) = 4$ e $ZERO(1) = 1$. Com isso é possível utilizar o ângulo de evaporação $\omega(j)$ descrito na tabela 4.4, cuja função é dar um incremento no ângulo para forçar o *qubit* no sentido contrário ao seu estado atual dominante, ‘0’ ou ‘1’, com o intuito de diminuir a tendência de estagnar no mesmo.

Tabela 4.4. Ângulo de Evaporação do QACO_alfa

	UM(j)>ZERO(j)	UM(j)<ZERO(j)
	$\omega(j)$	$\omega(j)$
$\tau_{\alpha,j} \tau_{\beta,j} > 0$	$-\rho^*(0.025 * \pi)^*(sub(j)/nant)*\lambda$	$\rho^*(0.025 * \pi)^*(sub(j)/nant)*\lambda$
$\tau_{\alpha,j} \tau_{\beta,j} < 0$	$\rho^*(0.025 * \pi)^*(sub(j)/nant)*\lambda$	$-\rho^*(0.025 * \pi)^*(sub(j)/nant)*\lambda$
$\tau_{\alpha,j} = 0$	$\rho^*(0.025 * \pi)^*(sub(j)/nant)*\lambda$	
$\tau_{\beta,j} = 0$		$\rho^*(0.025 * \pi)^*(sub(j)/nant)*\lambda$

onde ρ é uma constante, $sub(j)$ é o resultado da diferença entre o número de zeros ($ZERO(j)$) e uns ($UM(j)$), $nant$ é o número de formigas consideradas no problema e λ é o coeficiente de evaporação, uma constante no intervalo $[0,1]$.

O resultado de $sub(j)/nant$ é uma proporção que fornece a porcentagem de bits iguais a 0 ou 1 presentes numa determinada posição j dentro de uma geração. Quanto maior for esse resultado, maior é o número de 0’s ou 1’s sendo reproduzidos na posição j analisada. Com isso, existe uma maior chance de que o *qubit* fique estagnado, como ocorre no exemplo da figura 4.7, onde houve 80% de 1’s presentes na posição $j = 1$. Sendo assim, mais forte será o efeito da evaporação tendo em vista que o valor de $\omega(j)$ é diretamente proporcional a $sub(j)/nant$.

A forma como este o ângulo $\omega(j)$ age sobre um *qubit* é ilustrado na figura 4.8, a seguir:

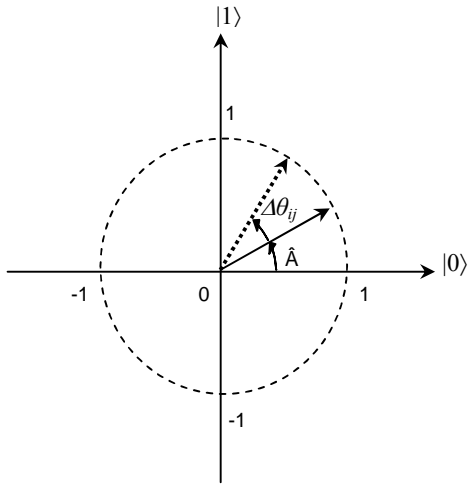


Figura 4.8(a) Etapa de atualização

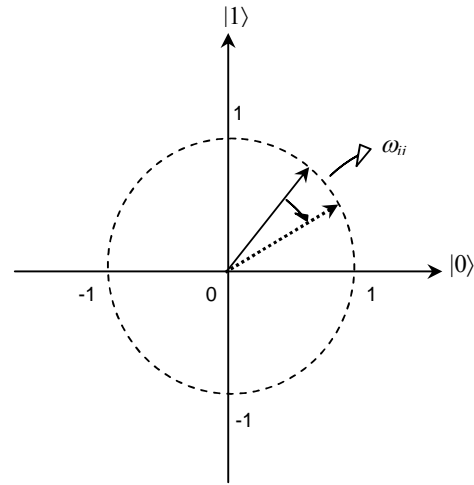


Figura 4.8(b): Etapa de evaporação

Figura 4.8: Aplicação do portão quântico de evaporação

Uma vez que este procedimento é realizado *qubit a qubit*, enquanto o caminho está sendo construído por cada agente, ele é uma analogia perfeita para a etapa de atualização local do feromônio utilizada no ACO (vide capítulo 2) que também é utilizada com a finalidade de impedir a convergência prematura para ótimos locais.

4.3 Problemas Teste

Tendo em vista o QACO havia sido testado apenas na otimização de funções numéricas bidimensionais, que apresentam baixa complexidade, não era possível determinar que tipos de dificuldades ele poderia encontrar quando aplicado a funções de maior complexidade. Por isso, o algoritmo foi submetido a alguns problemas teste para fins de análise de seu comportamento.

Além disso, foram feitas análises na otimização das funções *Himmelblau* e *Goldstein-Price* apenas para ilustrar alguns dos exemplos nos quais o QACO havia sido originalmente testado.

A primeira das funções analisadas foi a esfera de dimensão $n = 10$. Já neste exemplo o QACO apresentou tendência a convergir prematuramente sendo este problema também verificado quando o algoritmo foi aplicado à esfera de dimensão $n = 30$. Em cada uma das análises foram utilizadas sementes randômicas e variáveis com 12 bits de precisão. Por exemplo, numa função bidimensional, considera-se uma *string* de 24 bits.

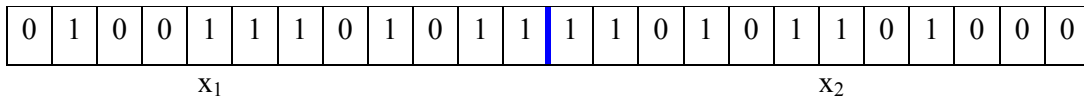


Figura 4.9: Exemplo de representação binária

Em seguida, cada um dos segmentos desta *string* (as variáveis x_1 e x_2 mostradas na figura 4.9, acima) é convertido da base binária para a base decimal para que o valor da *fitness* da função analisada possa ser determinado. O número de agentes considerados varia entre 20 e 200.

A constante α foi incluída na determinação do ângulo de atualização do QACO original depois de constatada a tendência de convergência prematura numa tentativa de evitá-la. Ela tem exatamente a mesma função da constante de inércia α_0 do QACO_Alfa e foi inserida no QACO em testes preliminares para determinar se apenas esta alteração seria suficiente para que ele supera-se sua tendência a convergir prematuramente. Quando $\alpha = 1$, significa que o algoritmo não sofre alterações.

4.3.1 Parâmetros dos Testes

As mesmas funções utilizadas nos testes do QACO original foram analisadas também pelo QACO_Alfa. As tabelas 4.6, 4.7, 4.9 e 4.11 apresentam os resultados obtidos por cada algoritmo em função dos seguintes parâmetros:

- α

Parâmetro auxiliar utilizado em testes preliminares no QACO original que desempenha a mesma função da constante de inércia α_0 do QACO_Alfa,

- λ

Coefficiente de evaporação do QACO_Alfa

- N° agentes

Número de agentes considerados na análise das funções

- Melhores Resultados

Apresenta os melhores resultados obtidos pelos algoritmos ao longo de 10000 avaliações.

- Média

Apresenta o valor médio obtido em 20 execuções do algoritmo com sementes aleatórias para cada função.

4.3.2 Função Esfera

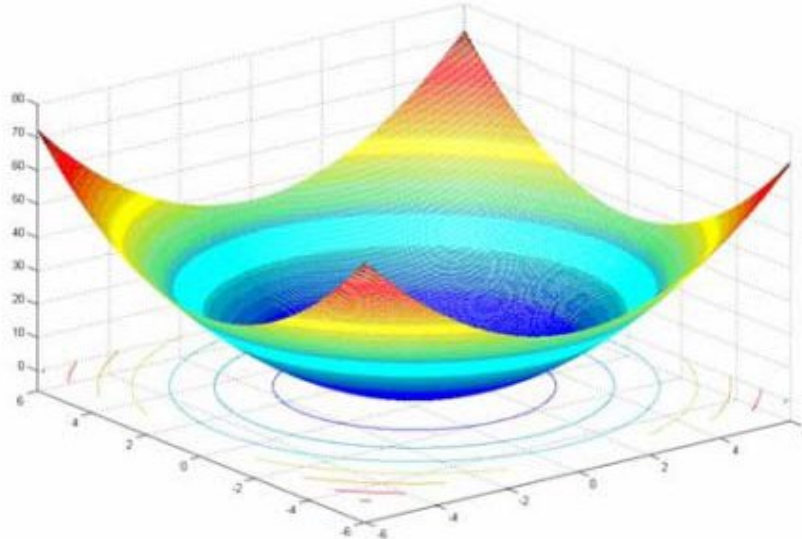


Figura 4.10: Função esfera $n = 2$

Tabela 4.5: Dados da função esfera

Função Esfera dimensão - n	Melhor Valor Conhecido
$\sum x_i^2 \quad i = 1, \dots, n$ Domínio $-5,12 \leq x_i \leq 5,12$	$x_i = 0,000$ 0,000

No problema conhecido como esfera, cujo gráfico é apresentado na figura 4.10 e os dados são apresentados na tabela 4.5 acima, o objetivo é minimizar a função

$$x_1^2 + x_2^2 + \dots + x_{i-1}^2 + x_i^2 + x_{i+1}^2 + \dots + x_n^2 \quad n \in \mathbb{N} \quad (29)$$

Trata-se de uma função numérica onde não existem restrições quanto ao número de variáveis consideradas. É fácil concluir que o mínimo da função é obtido quando o valor de cada uma das variáveis é igual a zero, entretanto, obter este valor vai se tornando mais difícil à medida que o número de variáveis consideradas aumenta.

Isso se deve ao fato de que em algoritmos como o QACO e, por analogia, o QACO_Alfa, o número de bits utilizados para representar uma variável é um parâmetro que está diretamente relacionado à precisão dos resultados obtidos. Quanto maior o número de bits empregados, mais refinado é o processo de exploração do espaço de busca e maiores são as chances de que os agentes converjam para um ótimo local.

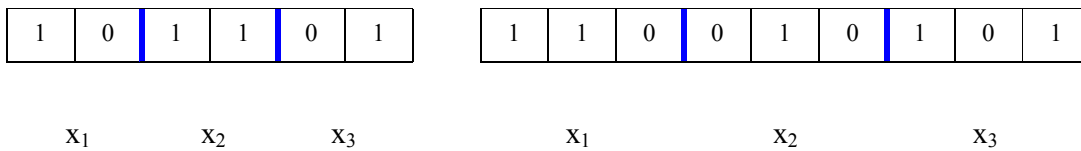


Figura 4.11(a)

Figura 4.11(b)

Figura 4.11: Influência do número de bits na precisão do algoritmo

A figura 4.11 mostra dois exemplos de *strings* binárias. No exemplo 4.11(a) existem 3 variáveis, cada uma com 2 bits de precisão, isto é, cada uma delas pode representar 4 algarismos distintos quando decodificadas para a base decimal.

No exemplo 4.11(b) também são mostradas três variáveis, mas agora cada uma delas utiliza 3 bits de precisão. Com isso, cada uma delas pode representar 8 algarismos distintos após a decodificação, resultando numa exploração mais detalhada considerando que o espaço de busca é o mesmo nos dois casos. Os resultados confirmaram a influência que este parâmetro exerce no comportamento dos algoritmos.

No primeiro teste foram consideradas 10 variáveis, num total de 120 bits. Em todos os casos analisados o QACO_Alfa encontrou o valor mínimo esperado, enquanto

o QACO original, mesmo com o uso da constante auxiliar α , obteve como melhor resultado um valor que está pelo menos três ordens de grandeza acima como pode ser observado pelos dados da tabela 4.6.

Tabela 4.6: Resultados função esfera dimensão 10

α	λ	Nº agentes	Melhores resultados		Média	
QACO	QACO_Alfa		QACO	QACO_Alfa	QACO	QACO_Alfa
1	0,001	20	0,2373	$1,563 \cdot 10^{-5}$	3.9110	$1,563 \cdot 10^{-5}$
0,2	0,01	100	0,0141	$1,563 \cdot 10^{-5}$	0.0520	$1,563 \cdot 10^{-5}$
0,1	0,01	200	0,0923	$1,563 \cdot 10^{-5}$	0.5723	$1,563 \cdot 10^{-5}$

No teste com 30 variáveis, totalizando 360 bits, o valor ótimo não foi obtido em nenhum dos algoritmos de acordo com os resultados da tabela 4.7, abaixo.

Tabela 4.7: Resultados função esfera dimensão 30

α	λ	Nº agentes	Melhores resultados		Média	
QACO	QACO_Alfa		QACO	QACO_Alfa	QACO	QACO_Alfa
1	0,01	20	14,2449	0,3990	24,6711	0,5394
0,2	0,02	100	3,2277	1,4730	6,8070	2,4815
0,1	0,01	200	8,3024	0,5149	20,7499	0,9907

4.3.3 Função *Himmelblau*

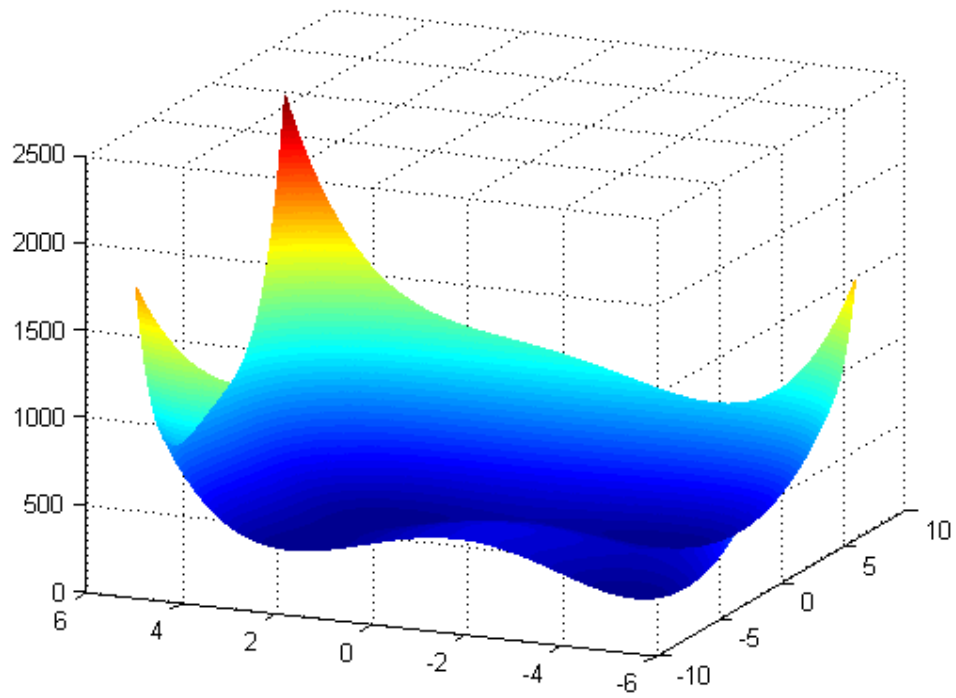


Figura 4.12: Função *Himmelblau*

Tabela 4.8: Dados função *Himmelblau*

Função <i>Himmelblau</i>	x_1	x_2	Melhor valor conhecido
$((x_1^2) + x_2 - 11)^2 + (x_1 + (x_2^2) - 7)^2$	-3,779	-3,283	0,000
	-2,805	3,131	
	3,584	-1,848	
Domínio: $-10 \leq x_i \leq 10$	3,000	2,000	

A função *Himmelblau* cujo gráfico é apresentado na figura 4.12, é uma função numérica de fácil resolução geralmente utilizada nos primeiros testes de uma ferramenta de otimização. O objetivo é encontrar o mínimo da função no domínio de busca. Existem quatro pontos de mínimo diferentes representados pelos pares ordenados na tabela 4.8 na página anterior.

Tabela 4.9: Resultados função *Himmelblau*

N° agentes	Melhores resultados		Média	
	QACO	QACO_Alfa	QACO	QACO_Alfa
20	$7,076 \cdot 10^{-4}$	$2,215 \cdot 10^{-4}$	0,0158	0,0107

Para cada semente, o algoritmo obtém a *string* do indivíduo de melhor *fitness* e este indivíduo irá representar um desses mínimos. Com o passar do tempo, devido ao processo conhecido como “*genetic drift*” (GOLDBERG, 1989) os indivíduos de melhor *fitness* irão convergir para um desses mínimos.

O *genetic drift* pode ser definido como a mudança na frequência com que um indivíduo aparece de uma geração para outra dentro de uma população. Os resultados para os testes com esta função são apresentados na tabela 4.9.

4.3.4 Função *Goldstein-Price*

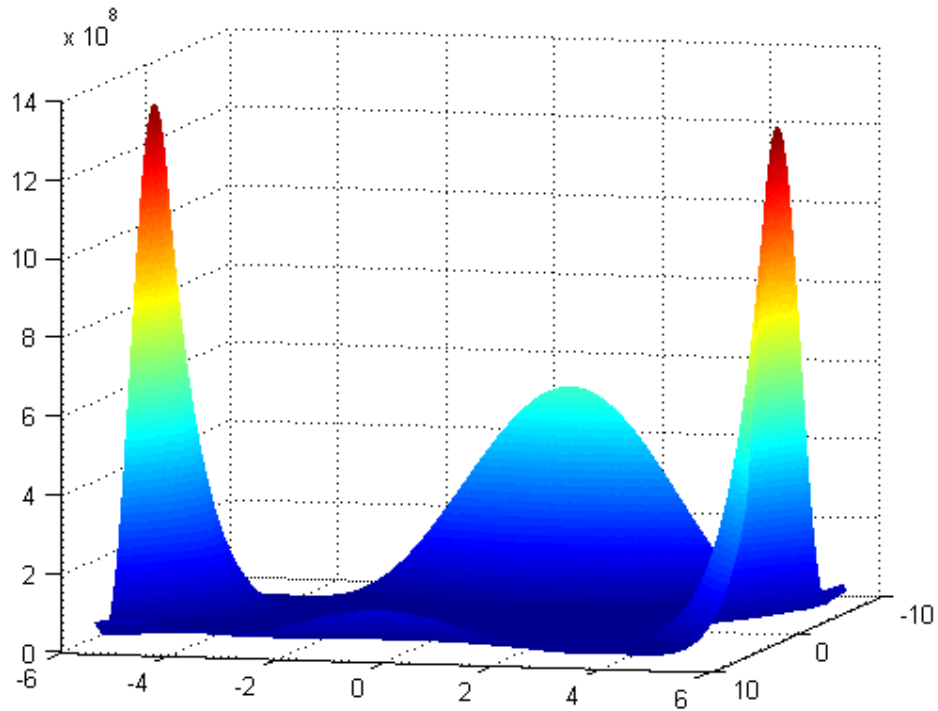


Figura 4.13: Função *Goldstein – Price*

Tabela 4.10: Dados função *Goldstein – Price*

Função <i>Goldstein-Price</i>	x_1	x_2	Melhor valor conhecido
$(1+(x_1+x_2+1)^2*(19-14*x_1+3*x_1^2-14*x_2+6*x_1*x_2+3*x_2^2))*(30+(2*x_1-3*x_2)^2*(18-32*x_1+12*x_1^2+48*x_2-36*x_1*x_2+27*x_2^2))$	0,000	-1,000	3,000
Domínio: $-2 \leq x_i \leq 2$			

Assim com a função *Himmelblau* a *Goldstein-Price* é uma função numérica de fácil resolução cujo gráfico é apresentado na figura 4.13. Na tabela 4.10 são mostrados os dados característicos dessa função cujo objetivo é obter o valor mínimo no domínio de busca. Tanto o QACO como o QACO_Alfa não apresentaram dificuldades para resolvê-la como pode ser observado pelos resultados na tabela 4.11 a seguir.

Tabela 4.11: Resultados função *Goldstein – Price*

N° agentes	Melhores resultados		Média	
	QACO	QACO_Alfa	QACO	QACO_Alfa
20	3,0001	3,0001	3,0060	3,0001

Capítulo 5

QACO_Alfa Aplicado a Problemas da Engenharia Nuclear

Após a etapa de análise de funções teste, cujos resultados foram apresentados no capítulo anterior, o QACO_Alfa foi aplicado na otimização de dois problemas reais relacionados à usinas nucleares do tipo PWR que serão utilizados como teste definitivo para determinar a capacidade do algoritmo como ferramenta de otimização.

- A otimização da recarga nuclear, um problema combinatório que consiste na substituição do combustível queimado no núcleo do reator;

- Identificação de transientes, um problema numérico onde o objetivo é minimizar o número de falhas de diagnóstico de acidentes.

Neste capítulo são apresentados os resultados obtidos a partir dos testes realizados com o QACO_Alfa na otimização destes dois problemas da Engenharia Nuclear para compará-los com os resultados publicados em trabalhos anteriores, especialmente o problema da recarga nuclear.

5.1 QACO_Alfa Aplicado na Otimização do Ciclo 7 de Angra 1.

O problema da otimização da recarga nuclear consiste em encontrar o padrão de embaralhamento de ECs que maximize o número de DEPPs e, conseqüentemente, o comprimento do ciclo (CHAPOT, 2000). Isso implica na maximização da concentração crítica de boro (B_c), tarefa desempenhada neste trabalho pelo QACO_Alfa. A maximização de B_c está sujeita a duas restrições básicas (CALDAS, 2006).

A primeira delas diz respeito ao fator de pico de potência radial do núcleo (F_{XY}) que, de acordo com as especificações técnicas de Angra 1, não deve ultrapassar o valor 1,435. A função *fitness* é então penalizada para solucionar esta restrição.

A outra restrição está relacionada à geometria do núcleo. Na simetria de 1/8 de núcleo utilizada neste trabalho um elemento de quarteto não pode trocar de lugar com um elemento de octeto, tendo em vista que os números de posições simétricas de elementos de quarteto e octeto são diferentes. Da mesma forma, o elemento central não pode mudar de posição. Esta restrição é resolvida através da representação *Random Keys* (BEAN, 1994) de cada arranjo.

5.1.1 Modelo *Random Keys*

O QACO_Alfa trabalha diretamente com *strings* binárias que são convertidas em seqüências de números inteiros para representar as 20 posições que os ECs irão ocupar em 1/8 de núcleo. Tal conversão permite que haja repetições entre estes valores. Em uma situação fictícia, onde existe uma região de simetria no núcleo que corresponde a apenas oito posições disponíveis para os ECs, representadas por p_k (com k variando de 1 a 8)

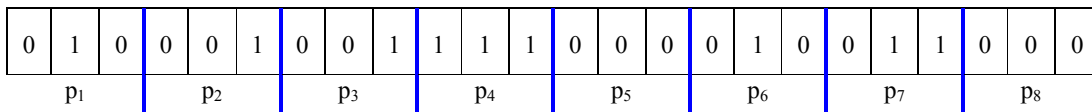


Figura 5.1: Exemplo de vetor binário gerado pelo QACO_Alfa

a conversão do vetor binário P da figura 5.1 para a base decimal dá origem ao seguinte padrão de recarga

2	1	1	7	0	2	3	0
---	---	---	---	---	---	---	---

Figura 5.2: Seqüência de valores inteiros gerados a partir do vetor binário P

que, como pode ser observado pela figura 5.2, apresenta valores repetidos compondo a *string*.

O problema da otimização da recarga de elementos combustíveis no núcleo do reator, por outro lado, é um problema discreto no qual as posições do núcleo são ocupadas por EC's representados por números inteiros sem repetição. Para resolver este problema foi utilizada a modelagem conhecida como *Random Keys* (BEAN, 1994), onde os números reais são transformados em seqüências de números inteiros sem repetição.

Esta modelagem é eficiente, rápida e fácil de ser implementada já que uma *string* de números inteiros, repetidos ou não, pode ser facilmente convertida em uma cadeia de números inteiros sem repetição. A figura 5.3 apresenta um exemplo meramente ilustrativo de *string* de números reais como aquelas que podem ser geradas pelo algoritmo caso os valores sejam normalizados.

1	2	3	4	5	6	7	8	9	10
9,5	0,3	2,0	9,7	5,7	9,1	2,1	1,0	8,8	1,5

Figura 5.3: *Random Keys*

No modelo *Random Keys* esses números são ordenados de forma crescente e os números inteiros gerados são as posições que os números reais ocupavam na cadeia antes da ordenação. A nova *string* é então ordenada de maneira ascendente como pode ser observado na figura 5.4, a seguir.

2	8	10	3	7	5	9	6	1	4
0,3	1,0	1,5	2,0	2,1	5,7	8,8	9,1	9,5	9,7

Figura 5.4: *Random Keys* ordenado.

Com isso obtém-se uma solução discreta através de números inteiros sem repetição. A solução que será avaliada pelo código de física de reatores é mostrada na figura 5.5.

2	8	10	3	7	5	9	6	1	4
---	---	----	---	---	---	---	---	---	---

Figura 5.5: Padrão analisado pelo RECNOB

5.1.2 Metodologia

O QACO_Alfa gera *strings* binárias a partir do vetor *Qubit* feromônio. Cada uma das *strings* é então decodificada em uma configuração de núcleo válida que, em seguida, é processada pelo RECNOB, um código de física de reatores (vide Apêndice) que simula o ciclo 7 de operação de Angra 1.

Duas situações são consideradas na análise dos resultados obtidos. Em cada uma delas é utilizada a simetria de 1/8 de núcleo, no entanto, a maneira como o problema é abordado nos dois casos apresenta importantes diferenças como será mostrado a seguir.

5.1.2.1 Situação real

O primeiro dos casos analisados está de acordo com as restrições geométricas inerentes ao uso da simetria de 1/8 de núcleo. Isso quer dizer que os elementos de quarteto só podem ser substituídos por um outro elemento de quarteto, o mesmo ocorrendo com os elementos de octeto.

As *strings* binárias geradas pelo QACO_Alfa são compostas por 240 bits, subdivididas em 20 segmentos de 12 bits cada. O próprio algoritmo faz a conversão de cada um destes segmentos para base decimal dando origem a seqüências de 20 números

inteiros variando entre 0 e 2047 (nas figuras 5.3 e 5.4 são utilizados valores reais ilustrativos). Estas por sua vez são submetidas ao método *Random Keys* para gerar um padrão de recarga válido.

As 10 primeiras posições desta seqüência correspondem aos elementos de quarteto enquanto as posições de 11 a 20 são destinadas aos elementos de octeto. Este padrão é então analisado pelo RECNOd para que seu nível de *fitness* possa ser avaliado, como visto em maiores detalhes no capítulo 3. Estas etapas são apresentadas nas figuras 5.6 e 5.7, abaixo.

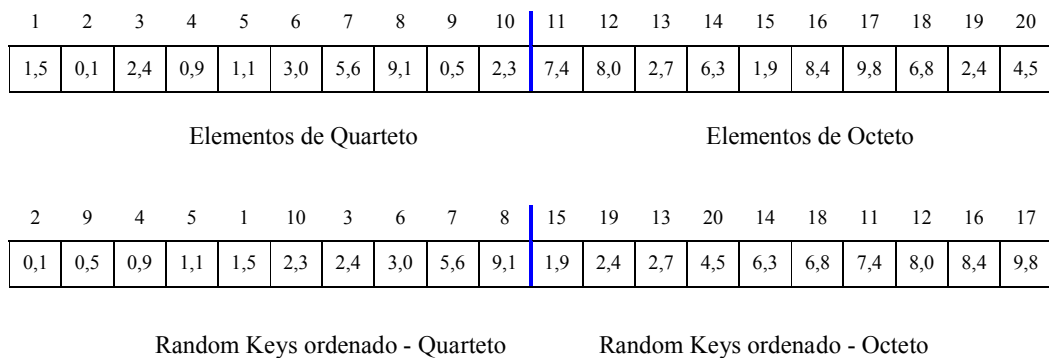


Figura 5.6: Modelo de Random Keys aplicado ao caso 1

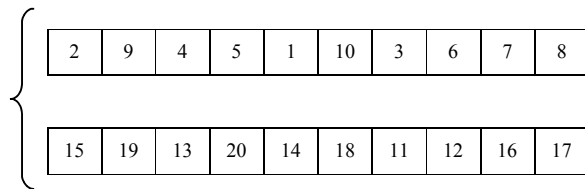


Figura 5.7: Padrão analisado pelo RECNOd no caso 1.

5.1.2.2 Situação particular para efeitos de comparação

No segundo exemplo analisado também é levada em consideração a simetria de 1/8 de núcleo. Porém, é permitido que os elementos de quarteto possam ser substituídos por um elemento de octeto sem a imposição de qualquer restrição.

Essa é uma situação anormal de realizar a recarga que, embora não seja impossível, é praticamente inviável, só podendo ocorrer no primeiro ciclo de uma usina, quando todos os ECs estão em nível de igualdade em relação à queima tendo em vista que todos estão sendo utilizados pela primeira vez.

Como pode ser observado na figura 5.8, o procedimento é idêntico ao do caso 1, no entanto, o padrão de recarga que resulta da aplicação do *Random Keys* é analisado pelo RECNOd como uma única seqüência de 20 números que correspondem às posições dos ECs no núcleo de acordo com a figura 5.9.

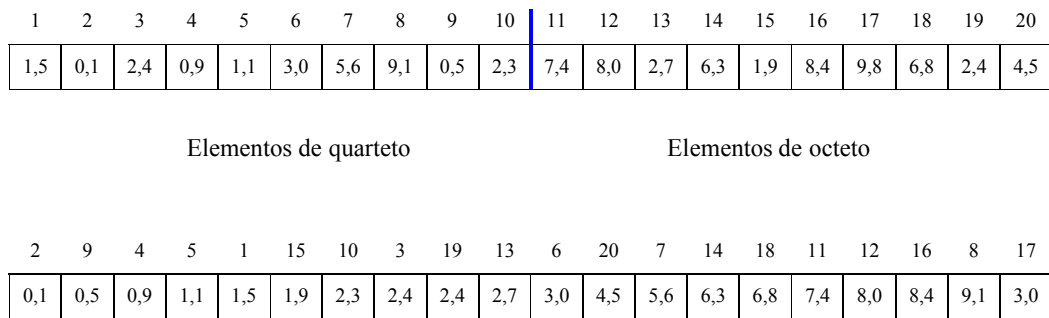


Figura 5.8: Modelo de Random Keys aplicado ao caso 2

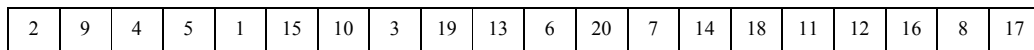


Figura 5.9: Padrão analisado pelo RECNOd caso 2

5.2 Resultados

Nesta seção são apresentados os resultados obtidos pelo QACO_Alfá na otimização do ciclo 7 da recarga da usina nuclear de Angra 1 para os dois casos descritos anteriormente. O QACO_Alfá foi desenvolvido em linguagem MatLab 6.5, bem como a interface de comunicação com o código RECNOd, elaborado por CHAPOT (2000), que é utilizado aqui para comparação de resultados.

5.2.1 Resultados Caso 1

O QACO_Alfa foi utilizado como ferramenta para determinar a concentração ótima de boro (B_c) em de 15 experimentos com sementes aleatórias, utilizando 20 agentes ao longo de 20.000 gerações num total de 400.000 avaliações para cada execução do algoritmo. A figura 5.10 mostra a evolução dos valores da potência média (P_{rm}) ao longo das gerações. A linha pontilhada corresponde ao valor limite $P_{rm} = 1,395$. Enquanto não se obtém um padrão de recarga válido, o algoritmo age minimizando este valor (mais detalhes no capítulo 3). A partir deste ponto, o algoritmo irá atuar de modo a maximizar a concentração de boro (B_c), como pode ser visto na figura 5.11. Depois que o primeiro padrão de recarga válido é encontrado (ponto sobre a curva), o valor de B_c passa a aumentar gradativamente.

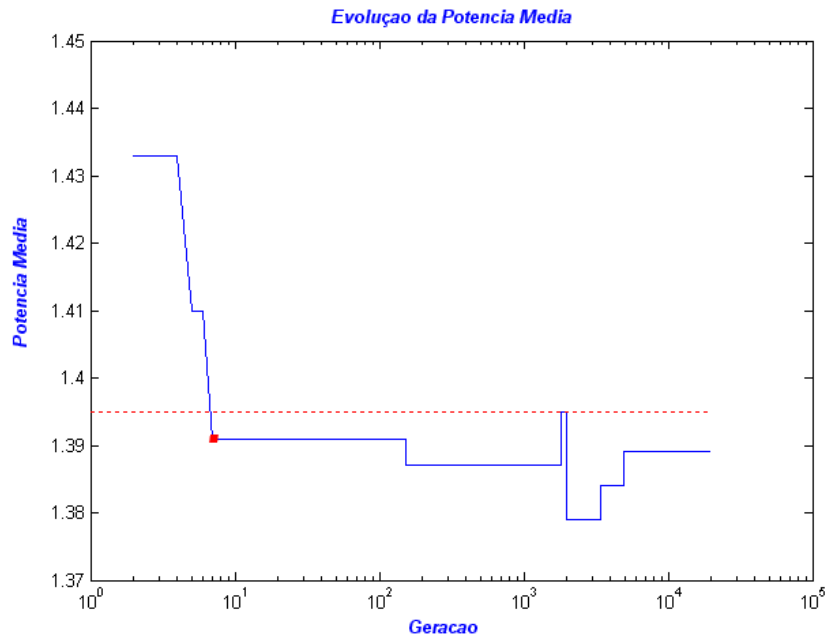


Figura 5.10: Evolução da potência média

A partir da análise da figura 5.11, é possível observar que o QACO_Alfa obtém seu melhor resultado ($B_c = 1415$) na geração 4962, ou seja, com um número de avaliações igual a 99240.

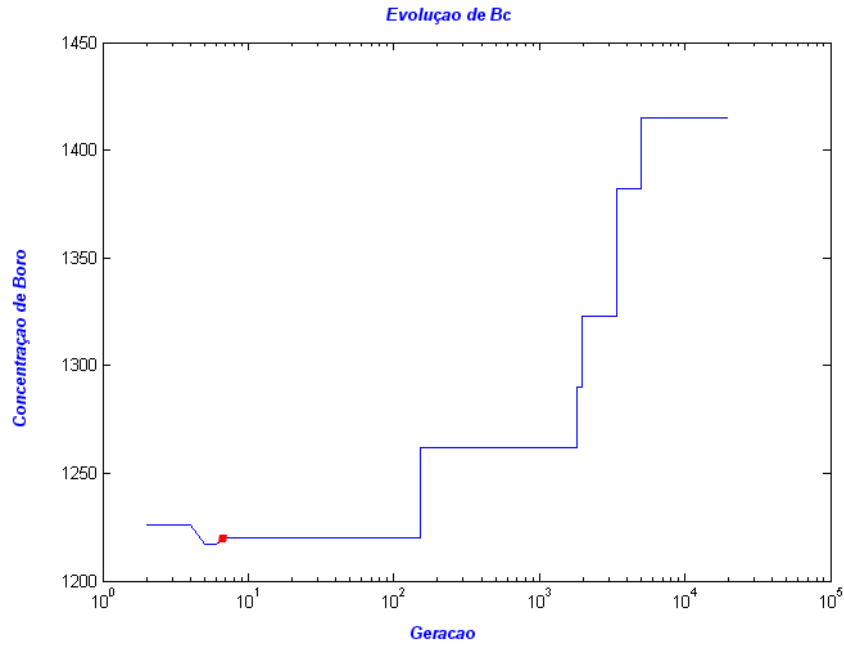


Figura 5.11 : Evolução da concentração de boro

Em geral, o QACO_Alfa encontra o primeiro padrão de recarga válido ($P_{rm} \leq 1,395$), em média, com um número menor que 1500 avaliações como é ilustrado pelo ponto escuro sobre as curvas das figuras 5.11 e 5.12.

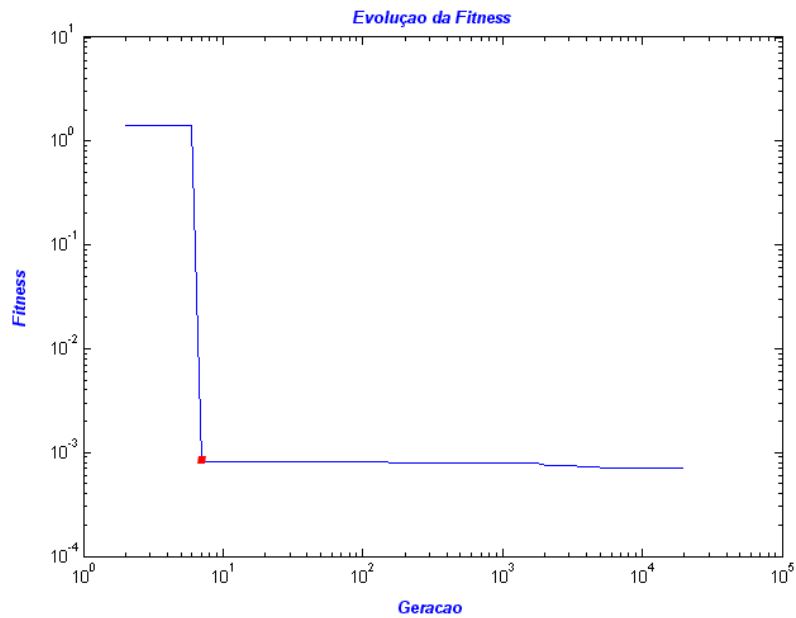


Figura 5.12 : Evolução da *fitness*.

Os dados obtidos foram então comparados aos de outros trabalhos que também utilizavam o RECNOD e uma modelagem semelhante àquela adotada neste trabalho. Estes resultados são apresentados na tabela 5.1 abaixo.

Tabela 5.1: Comparação de técnicas aplicadas à maximização do comprimento do ciclo 7 de Angra 1

Autor	B_c (máximo)	P_{rm}	Técnica	Heurística	Avaliações
-	955	1,345	Manual	-	-
CHAPOT, (2000)	1026	1,390	GA	Não	4000
MACHADO, L. (2002)	1297	1,384	Ant – Q	Sim	200
MACHADO, M., (2005)	1242	1,361	PBIL – N	Não	6000
MACHADO, M., (2005)	1305	1,349	PBIL – MO	Não	10000
LIMA, (2005)	1424	1,386	RCCA	Sim	420000
CALDAS, (2006)	1428	1,392	FBIL	Não	430364
Presente Trabalho	1414	1,393	QACO	Não	400000
Presente Trabalho	1415	1,389	QACO_Alfa	Não	400000(*)

(*) Melhor resultado obtido com 99240 avaliações

A tabela 5.1 mostra que o QACO_Alfa é uma ferramenta de otimização comparável aos métodos utilizados nos demais trabalhos. Seu melhor resultado, 1415 de B_c máximo, é superior à quase todos os métodos considerados com um valor de potência média muito próximo do limite de 1,395. Embora tenha sido superado pelo RCCA e pelo FBIL, o QACO_Alfa obteve resultados próximos mesmo com um número de avaliações menor.

5.2.2 Resultados Caso 2

Tendo em vista que o caso 2 não é utilizado na prática, não existem dados disponíveis na literatura para comparação dos resultados, à exceção do trabalho de WAINTRAUB (2008), que em sua tese de doutorado analisou o comportamento do PSO paralelo quando aplicado a esta situação particular. O mesmo procedimento adotado no Caso 1, com 15 experimentos ao longo de 20.000 gerações foi utilizado para o caso 2, para comparar o QACO_Alfa com o PSO paralelo cujos resultados são apresentados na tabela 5.2, a seguir:

Tabela 5.2: Resultados Caso 2

Técnica	Boro Máximo	Boro Médio	DEPP	Avaliações
PSO Paralelo	1557	1429	389	960000
QACO_Alfa	2090	1686	522	400000 (*)
QACO	1694	1513	423	400000

(*)Melhor resultado obtido na avaliação 351020

É importante deixar claro que no PSO paralelo os valores de concentração de B_c foram limitados a, no máximo, 1600 ppm. Dessa forma, a única comparação válida entre os resultados obtidos pelo QACO_Alfa e o PSO paralelo diz respeito ao número de avaliações utilizados por cada algoritmo.

Os resultados apresentados na tabela 5.2 deixam claro que o QACO_Alfa obtém um rendimento superior ao do PSO paralelo tendo em vista que obtém seu melhor resultado, 2090 ppm de boro, com um número de avaliações que é menor que a metade daquele utilizado no trabalho de WAINTRAUB (2008).

Quando comparado ao QACO original, o QACO_Alfa também, se destaca, com um valor máximo de DEPP's superior e maior robustez, confirmando que as alterações propostas neste trabalho surtiram um efeito satisfatório no desempenho do algoritmo. Nas figuras 5.13, 5.14 e 5.15 são apresentados os gráficos, para o caso 2, da evolução da potência média, concentração de boro e *fitness*, respectivamente.

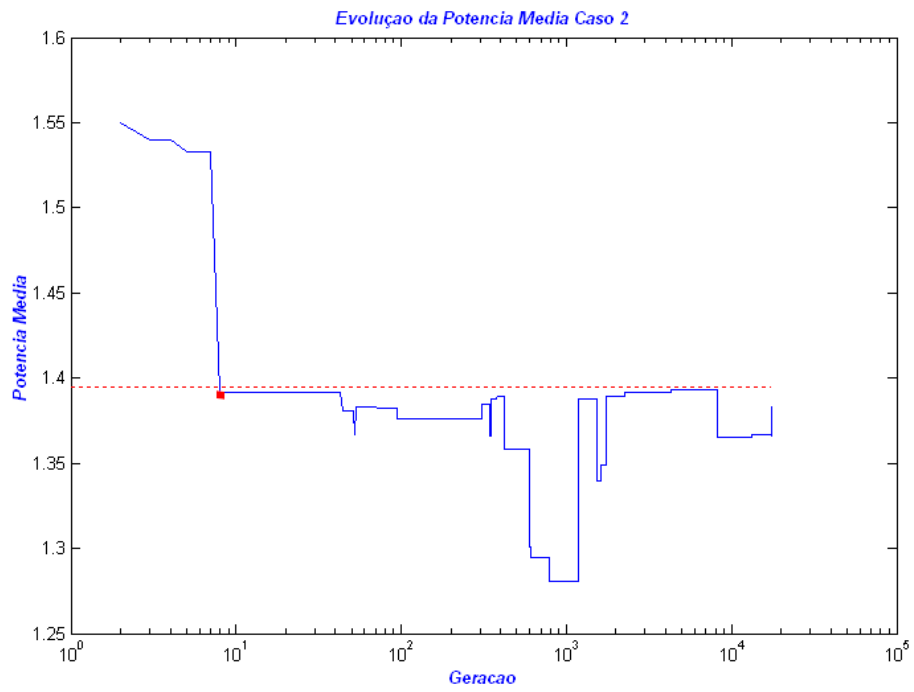


Figura 5.13 : Evolução da potência média - Caso 2

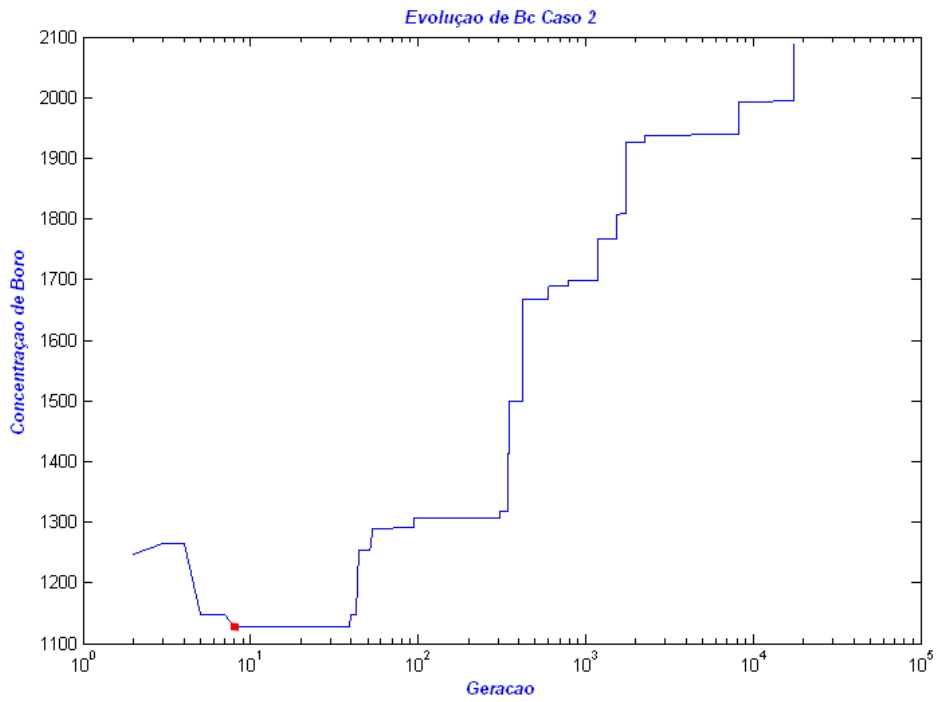


Figura 5.14 : Evolução da concentração de boro – Caso 2

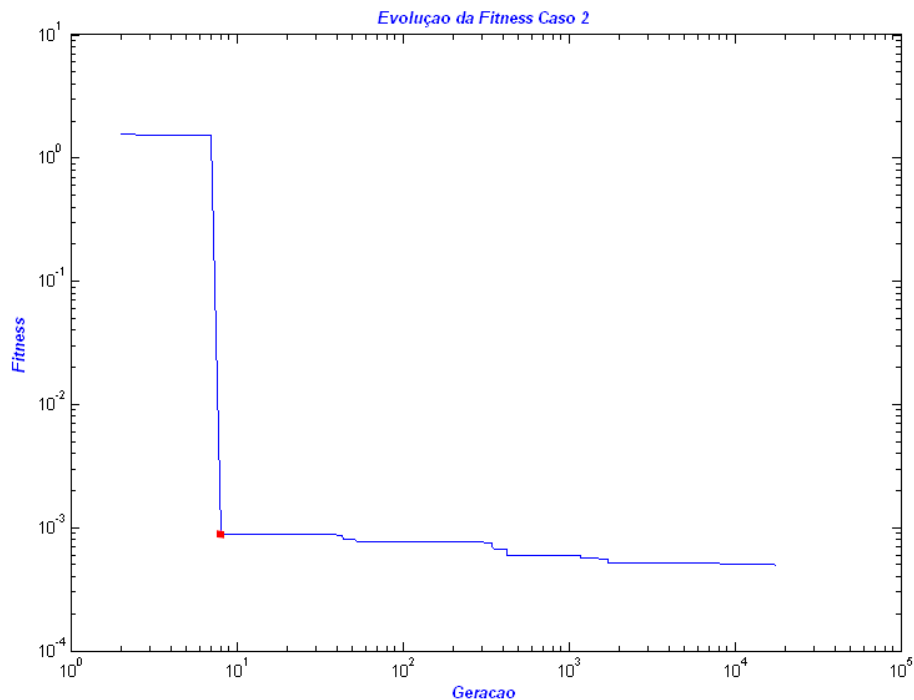


Figura 5.15 : Evolução da *fitness* – Caso 2

5.3 QACO_Alfa Aplicado ao Problema de Identificação de Transientes

Para que uma usina nuclear opere normalmente, é necessário o monitoramento de um grande número de variáveis (pressão, temperatura etc.), ou seja, é preciso conhecer de que forma essas variáveis evoluem no tempo. Isso torna possível, diante da alteração dos padrões normais de funcionamento da usina, identificar o evento anômalo e agir para solucioná-lo evitando que ocorra um acidente (MEDEIROS, 2007).

Com o objetivo de auxiliar o operador da usina a reduzir seu esforço cognitivo, aumentando assim seu tempo disponível para mantê-la funcionando nos seus padrões normais de segurança, possibilitando que ele tenha condições de tomar a melhor decisão no devido tempo, foram criados os sistemas de identificação de transientes (MEDEIROS, 2007, MOL, 2002, PEREIRA e SCHIRRU, 2002).

É classificado como transiente um evento anômalo que ocorra na usina, ou seja, um evento no qual a evolução temporal do comportamento das variáveis analisadas para determinar se a usina opera dentro dos limites de segurança não esteja de acordo com seu padrão normal. O problema da identificação de acidentes num reator do tipo PWR é resolvido pela classificação de transientes.

5.3.1 Metodologia

Para avaliar seu desempenho em um problema numérico não-combinatorial e poder compará-lo com outros trabalhos, o QACO_Alfa foi utilizado na otimização do problema de identificação de transientes. Tendo em vista que a análise deste problema não é o foco deste trabalho, uma abordagem mais detalhada da aplicação do QACO_Alfa na otimização do mesmo encontra-se em (SILVA, et al, 2009b), onde são consideradas as evoluções das variáveis em função do tempo referentes a três acidentes:

BLACKOUT – perda de energia elétrica externa;

LOCA – acidente de perda de refrigerante por grande ruptura;

SGTR – ruptura de tubo do gerador de vapor.

O método adotado para fazer a classificação dos transientes é baseado na minimização das distâncias euclidianas entre o centróide da curva do evento desconhecido e o centróide da curva dos acidentes postulados (MEDEIROS, 2007). O conceito de centróide é semelhante ao conceito físico de centro de massa. Cada acidente possui uma curva própria, como uma impressão digital, que representa a evolução temporal das variáveis envolvidas no processo de análise.

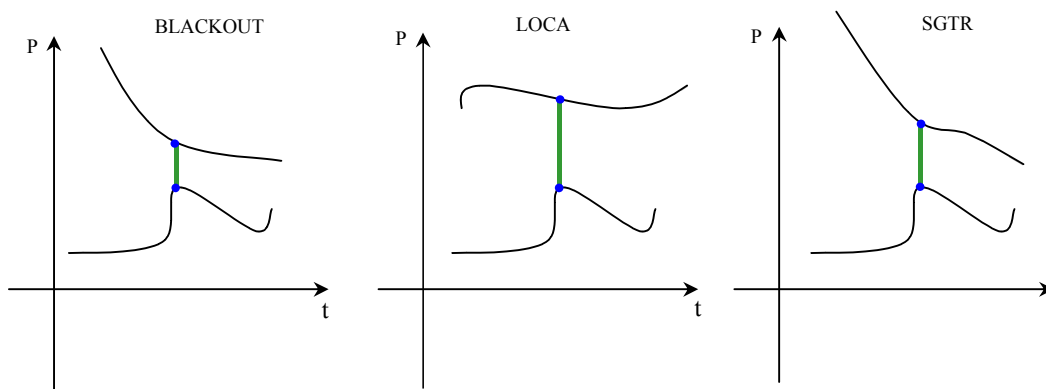


Figura 5.16: Identificação de um evento baseado na diferença entre as distâncias euclidianas

Quando ocorre um evento (curva inferior de cada acidente na figura 5.16), são comparadas as distâncias entre o centróide do mesmo ao centróide de cada um dos acidentes postulados. O evento passa a ser classificado como o acidente ao qual seu centróide mais se aproximar como é mostrado no exemplo da figura 5.16 onde é monitorada a evolução temporal da pressão para cada acidente.

Os três acidentes são postulados em função de 18 variáveis mais o tempo num total de 57 variáveis, cada uma utilizando precisão de 12 bits de precisão. Ou seja, para que o QACO_alfa fosse aplicado como ferramenta para realizar a otimização deste problema foram utilizadas *strings* binárias compostas por 684 bits. A tabela 5.3 apresenta os resultados obtidos pelo QACO_Alfa na otimização deste problema.

Tabela 5.3: Comparação das técnicas utilizadas na identificação de transientes

Classificações corretas	QACO	QACO_Alfa	GA	PSO
Máximo	174	177	183	177
Mínimo	9	6	3	15
Média	141,8	176,2	159,9	160,2

Onde Máximo e Mínimo representam respectivamente o número de acertos máximos e mínimos obtidos através da análise feita por cada uma das ferramentas consideradas. Os resultados mostram que o QACO_Alfa possui um valor médio de obtenção de máximo de classificações corretas maior que o do GA e PSO, o que demonstra maior robustez. Na figura 5.17, a seguir, é apresentada a evolução das classificações de falhas observadas pelo algoritmo.

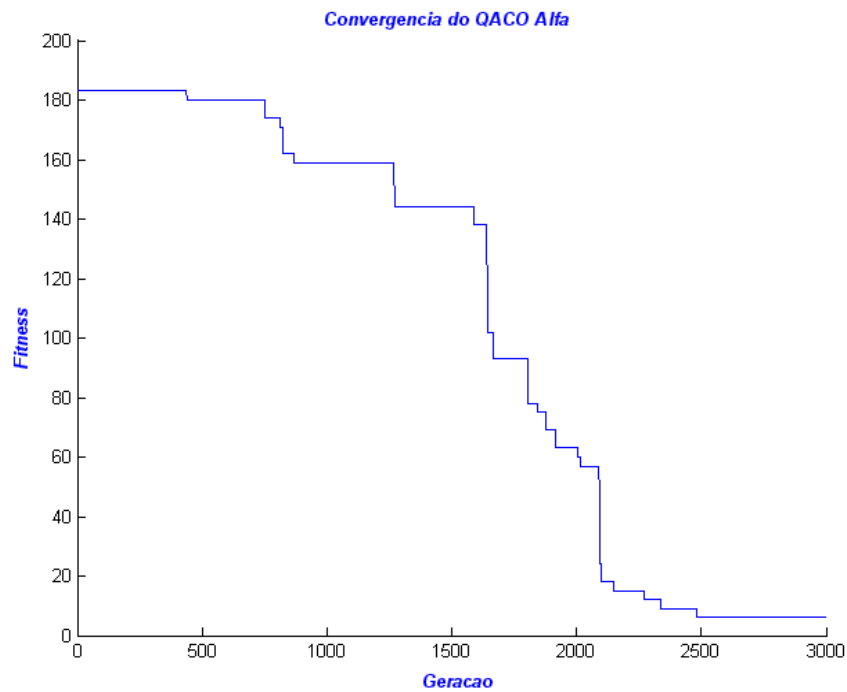


Figura 5.17: Curva do QACO_Alfa aplicado à identificação de transientes

Capítulo 6

Conclusões

Neste capítulo são apresentadas as considerações finais acerca do trabalho realizado nesta dissertação de mestrado que consistiu em desenvolver um novo algoritmo com inspirações quânticas. O mesmo foi então submetido a uma série de testes e, posteriormente, foi aplicado na otimização problemas relacionados à Engenharia Nuclear com foco na recarga nuclear.

Além disso, são discutidas propostas para trabalhos futuros no sentido de melhorar o desempenho deste algoritmo.

6.1 Considerações Finais

Baseado no *Quantum Ant Colony Optimization* – QACO foi desenvolvido um algoritmo evolucionário com inspirações quânticas para ser aplicado a problemas da Engenharia Nuclear, em especial à otimização da recarga do ciclo 7 de Angra 1. Este algoritmo recebeu o nome de QACO_Alfa, cujo método de aprendizagem dos indivíduos foi modificado com o objetivo de superar os problemas de convergência prematura e estagnação de bits observada na otimização de funções teste como a esfera n – dimensional.

A primeira modificação em relação ao algoritmo original consistiu de uma atenuação no reforço de feromônio feita através do coeficiente de inércia α_0 . Trata-se de uma constante no intervalo $[0,1]$ que age como um “freio”, reduzindo o impacto do incremento dado ao ângulo do *qubit*. A segunda modificação é a inclusão de uma etapa adicional de evaporação que age no sentido de diminuir a probabilidade do estado dominante atual (0 ou 1) de um determinado *qubit*, um equivalente da regra de atualização local do ACO.

As alterações propostas neste trabalho atingiram as expectativas e só os resultados obtidos na resolução das funções teste apresentadas no capítulo 4 já bastariam para classificar o QACO_Alfa como uma ferramenta de otimização mais eficiente e robusta que o QACO original, além de inovadora.

Esta, no entanto, constitui apenas uma etapa preliminar para o principal objetivo deste trabalho que consistia em aplicar o QACO_Alfa na otimização problemas da Engenharia Nuclear, em especial ao problema da recarga de uma usina do tipo PWR.

Na resolução deste problema real de natureza combinatória, o QACO_Alfa obteve um bom desempenho, comparável ao de ferramentas de otimização tradicionais baseadas em inteligência artificial como colônias de formigas, GA, PBIL e PSO paralelo, chegando até mesmo a superá-las em alguns resultados como pode ser observado em maiores detalhes nos dados apresentados no capítulo 5.

O QACO_Alfa obteve valores de concentração de boro crítico (B_c) maiores que os obtidos com métodos como GA, Ant-Q, PBIL-N, PBIL-MO e manual. Quando comparado ao FPBIL e ao RCCA (que utiliza heurística) o QACO_Alfa obtém valores de B_c muito próximos utilizando menos avaliações.

Diante desses resultados podemos afirmar que o desempenho do QACO_Alfa na otimização da recarga nuclear foi satisfatório e o qualifica como uma ferramenta de otimização eficiente e robusta, capaz de superar muitas das mais tradicionais ferramentas de otimização baseadas em inteligência artificial disponíveis atualmente, a saber, GA, Ant-Q, PBIL-N, PBIL-MO e PSO paralelo.

Além disso, é importante ressaltar o pioneirismo deste trabalho em aplicar um algoritmo evolucionário com inspirações quânticas a problemas reais de uma usina do tipo PWR, como a otimização da recarga nuclear. Na verdade este trabalho é a primeira aplicação deste tipo de algoritmo a um problema real multidimensional tendo em vista que o problema de diagnóstico de falhas abordado por WANG *et al* (2008) na verdade reproduz o caso de funções teste com apenas duas variáveis (x,y) como aquelas analisadas no capítulo 5 desta dissertação.

Os resultados obtidos a partir da análise de problemas de diferentes categorias, a saber, combinatorial e numérica, deixam clara a versatilidade do QACO_Alfa. Além disso, ele é um algoritmo de implementação simples que não requer que os indivíduos de uma população sejam armazenados em estruturas de dados complexas.

6.2 Propostas para Trabalhos Futuros

Embora o QACO_Alfa tenha obtido bons resultados nos problemas analisados, ele ainda apresenta algumas dificuldades relacionadas principalmente a convergência prematura e dependência dos seguintes parâmetros:

n^0 de agentes – número de formigas utilizadas na análise do problema;

n^0 de bits – determina a precisão dos valores inteiros (ou reais se houver normalização) obtidos a partir da conversão do vetor binário gerado pelo QACO_Alfa para a base decimal;

valor do parâmetros de exploração p_e – determina se a tomada de decisão para escolha do próximo bit será feita de maneira determinística, ou seja, baseada no

conhecimento adquirido ao longo das gerações nas quais o algoritmo evolui, ou probabilística;

constante de inércia α_0 – atenuação aplicada ao ângulo de rotação do *qubit*;

coeficiente de evaporação λ - diminuição do ângulo de rotação aplicado a um *qubit*

Na otimização da esfera dimensão $n = 30$, por exemplo, o QACO_Alfa não encontrou o valor ótimo em nenhum dos testes realizados. Tendo em vista que o mesmo não ocorreu para a esfera dimensão $n = 10$, onde os parâmetros utilizados eram os mesmos, a única causa para explicar a diferença de desempenho é o número de bits utilizados em cada situação.

Isto significa que, mesmo com as alterações aplicadas à atualização dos bits, o algoritmo continua sensível aos parâmetros utilizados. Uma maneira de minimizar esta sensibilidade pode ser obtida através de algum método heurístico como os *building blocks* dos Algoritmos Genéticos.

Além disso, pode-se realizar um estudo mais detalhado com a realização testes no sentido de encontrar um padrão ótimo de parâmetros que possibilite obter os melhores resultados para o problema analisado ou até mesmo, tornar o algoritmo livre de parâmetros, através de um método semelhante ao aplicado em (CALDAS, 2006).

Apêndice – O Código RECNOD

Inicialmente a equação de difusão de nêutrons bidimensional (vide figura A.1) era resolvida, para poucos grupos de energia, pelo Método de Diferenças Finitas (MDF).

$$\begin{aligned} & -\frac{\partial}{\partial x}\left(D_g(x,y)\frac{\partial}{\partial x}\phi_g(x,y)\right) - \frac{\partial}{\partial y}\left(D_g(x,y)\frac{\partial}{\partial y}\phi_g(x,y)\right) + \Sigma_{Rg}(x,y)\phi_g(x,y) = \\ & = \sum_{\substack{g'=1 \\ g' \neq g}}^2 \Sigma_{gg'}(x,y)\phi_{g'}(x,y) + \frac{1}{k_{eff}} \chi_g \sum_{g'=1}^2 \nu \Sigma_{fg'}(x,y)\phi_{g'}(x,y) \end{aligned}$$

Onde

g = grupo de energia;

$\phi_g(x,y)$ = fluxo de nêutrons do grupo g no ponto (x,y) ;

$D_g(x,y)$ = coeficiente de difusão do grupo g no ponto (x,y) ;

$\Sigma_{Rg}(x,y)$ = seção de choque macroscópica de remoção de nêutrons do grupo g no ponto (x,y) ;

$\Sigma_{gg'}(x,y)$ = seção de choque macroscópica de transferência de nêutrons do grupo g' para o grupo g no ponto (x,y) ;

k_{eff} = fator de multiplicação efetivo;

χ_g = espectro de fissão do grupo g ;

$\nu \Sigma_{fg'}(x,y)$ = produto do número médio de nêutrons emitidos por fissão (ν) pela seção de choque fissão do grupo g' no ponto (x,y) .

Figura A.1: Equação de difusão de nêutrons independente do tempo, bidimensional, em geometria cartesiana, para dois grupos de energia (FONTE: CHAPOT, 2000).

Entretanto, o uso deste método requer uma grande quantidade de memória, em outras palavras, alto custo computacional, tornando-se necessário buscar métodos alternativos para realizar esta tarefa. Com isso foi criada uma técnica de cálculo menos rigorosa e mais eficiente, os métodos nodais.

Esses métodos partem do pressuposto de que é possível decompor o núcleo de um reator em sub-regiões, chamadas nodos. Para geometrias cartesianas bidimensionais

geralmente são utilizados quadrados ou retângulos para representar os nodos, enquanto a forma de paralelepípedos retângulos é a mais comum em análises tridimensionais.

A técnica dos métodos nodais parte dos pressupostos que os parâmetros nucleares são uniformes dentro de cada nodo e, além disso, que o fluxo pode ser representado por uma expansão em funções polinomiais.

Posteriormente, foi desenvolvido o Método de expansão de Fluxo (MEF) (LANGENBUCH, *et al*, 1977), no qual são calculados os fluxos Φ em cinco pontos da cada nodo: no ponto central e no ponto médio de cada uma das suas faces. Com este método os autores encontraram erros de até 7% na distribuição de potência do núcleo. Este método é apresentado de maneira esquemática na figura A.2.

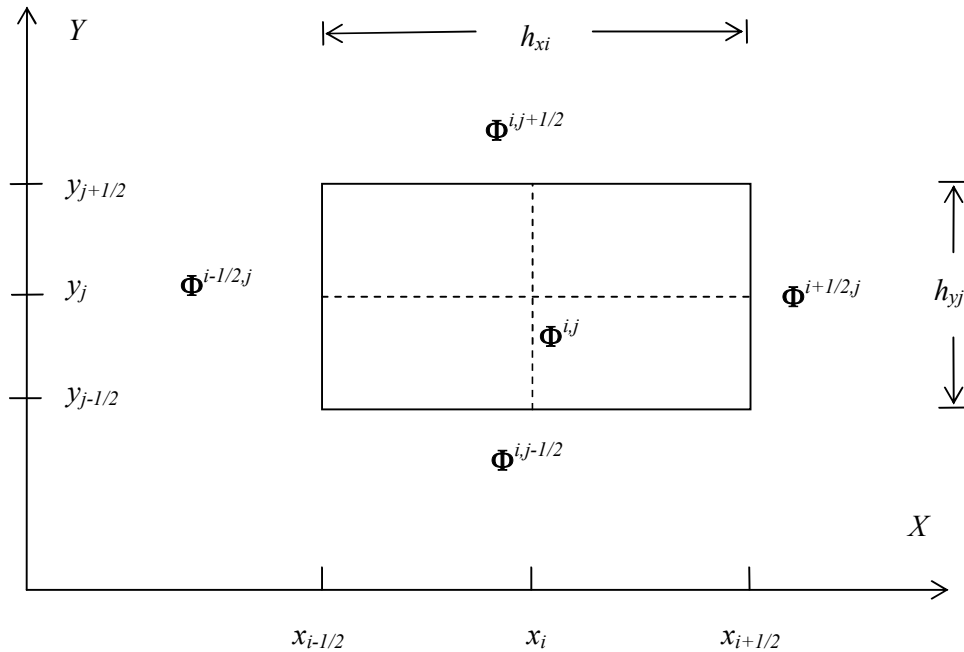


Figura A.2: Método de Expansão de Fluxo

Baseados no MEF, MONTAGNINI *et al* (1994) criaram um novo método nodal. Nele passa-se a calcular o fluxo médio ϕ em cada uma das seguintes regiões dos nodos: na área do nodo e em cada um dos seus lados de acordo com a figura A.3. Os autores

afirmam que com este método os erros encontrados para a distribuição de potência são inferiores a 1.5% sendo por esse motivo adotado no trabalho que originou o código RECNOB (CHAPOT, 2000).

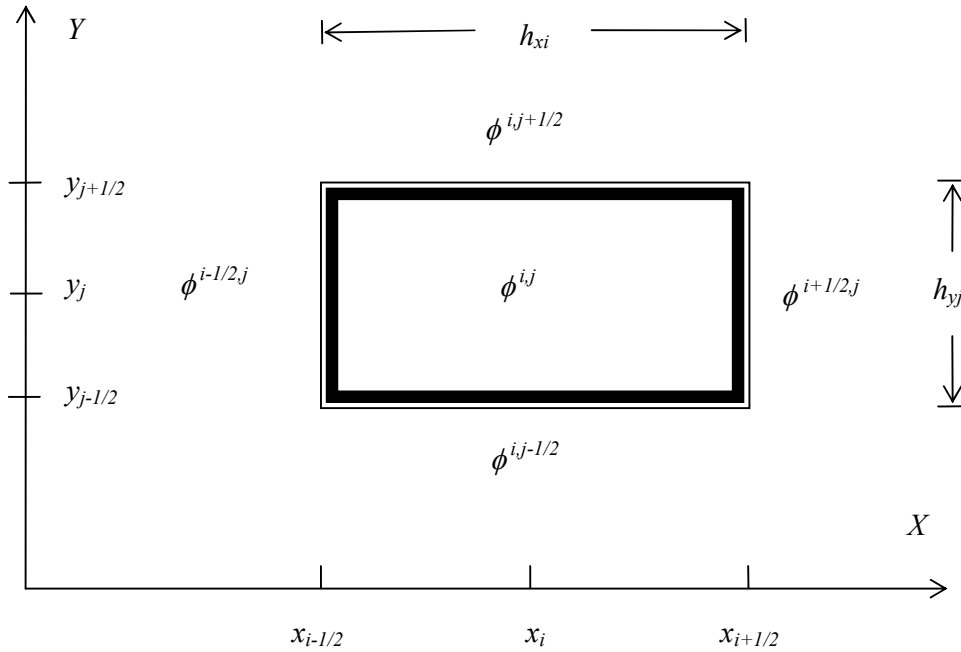


Figura A.3: Método de Expansão de Fluxo alternativo

O RECNOB é um código de física de reatores bidimensional a dois grupos de energia para cálculos neutrônicos desenvolvido para ser mais rápido que o ANC (CHAPOT, 2000) e com isso reduzir os custos computacionais do sistema de otimização de recargas. Por não ser destinado ao uso comercial, o RECNOB não possui todos os módulos que um código nodal de física de reatores como o ANC possui.

Dentre os módulos ausentes no RECNOB destacam-se a realimentação termo-hidráulica e a reconstrução da distribuição de densidade de potência pino a pino necessária para o cálculo do fator de pico de potência radial F_{XY} . Por isso nas comparações de resultados apresentadas neste trabalho, esta grandeza será substituída pela máxima potência média relativa dos elementos combustíveis - P_m .

Devido às discrepâncias entre as distribuições de potência calculadas pelo RECNOd e pelo ANC, o valor 1,395 é adotado como valor limite de P_m e deve, por isso, corresponder a 1,435 como valor real limite de F_{XY} . O RECNOd recebe como entrada a configuração de 1/8 de núcleo de acordo com a figura A.4 a seguir:

1,1 C					
2,1 6	2,2 7				
3,1 5	3,2 16	3,3 8			
4,1 4	4,2 17	4,3 15	4,4 9		
5,1 3	5,2 18	5,3 14	5,4 11	5,5 10	
6,1 2	6,2 19	6,3 13	6,4 12		
7,1 1	7,2 20				

Figura A.4: Modelagem do reator de acordo com os dados fornecidos ao RECNOd

A partir da figura A.4 é possível observar que cada elemento é diferenciado pelo seu número (variando de 1 a 20 mais o elemento central, C) e por um par de coordenadas. Essas coordenadas ajudam a determinar a nova posição que um determinado elemento irá ocupar no ciclo seguinte.

	P_i	P_{i-1}	Tipo
Central	(1,1)	(1,1)	4
Quartetos	(2,1)	(3,1)	4
	(3,1)	(2,2)	5
	(4,1)	(7,1)	2
	(5,1)	(3,3)	4
	(6,1)	(6,1)	2
	(7,1)	(4,4)	1
	(2,2)	(5,1)	4
	(3,3)	(5,5)	2
	(4,4)	(2,1)	6
	(5,5)	(4,1)	4
Octetos	(3,2)	(5,3)	6
	(4,2)	(5,2)	4
	(5,2)	(5,4)	5
	(6,2)	(7,2)	5
	(7,2)	(4,3)	6
	(4,3)	(3,2)	4
	(5,3)	(6,4)	2
	(6,3)	(6,2)	3
	(5,4)	(6,3)	3
	(6,4)	(4,2)	4

Tabela A.1: Coordenadas dos elementos

A tabela A.1, acima, fornece para o RECNOd a posição do elemento combustível P_i , e sua respectiva posição no ciclo anterior, P_{i-1} além do tipo de elemento.

REFERÊNCIAS BIBLIOGRÁFICAS

BEAN, J. C., 1994, “Genetics Algorithms and Random Keys for Sequencing and Optimization”, *ORSA Journal of Computing*, v.6, n.2.

BENIOFF, P., 1980, “The Computer as a Physical System: A Microscopic Quantum Mechanical Hamiltonian Model of Computers as Represented by Turing Machines”, *Journal of Statistical Physics*, vol. 22, n. 5 (May), pp. 563 – 591.

BURIOL, L. S., 2000, *Algoritmo Memético para o Problema do Caixeiro Viajante Assimétrico como Parte de Framework para Algoritmos Evolutivos*. Dissertação de M. Sc., DENNIS/FEE/UNICAMP, Campinas, SP, Brasil.

CALDAS, G. H. F., 2006, *Algoritmo Evolucionário não Parametrizado Aplicado ao Problema da Otimização de Recargas Nucleares*. Tese de D.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.

CHAPOT, J. L. C., 2000, *Otimização Automática de Recargas de Reatores a Água Pressurizada Utilizando Algoritmos Genéticos*. Tese de D.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.

DEUTSCH, D., 1985, “Quantum Theory, the Church – Turing Principle and the Universal Quantum Computer”, *Proceedings of the Royal Society of London*, vol. A400, n. 1818 (July), pp.97 – 117.

DORIGO, M., GAMBARDILLA, L. M., 1996, “Solving Symmetric and Asymmetric TSPs by Ant Colonies”. *IEEE Conference on Evolutionary Computation, ICEC’ 96*, Nagoya Japan, 20-22 May.

DORIGO, M., GAMBARDELLA L. M., 1997, “Ant Colony System: A Cooperative Learning Approach to Traveling Salesman Problem”, *IEEE Transactions and Evolutionary Computation*, v.1, n.1, pp.53 – 66.

DUDERSDADT, J. J., HAMILTON, L. J., 1976, *Nuclear Reactor Analysis*. John Wiley and Sons, Inc.

FELTRE, R., 1996, *Fundamentos da Química*, 2ed., São Paulo, SP, Editora Moderna.

FEYNMAN, R., 1982, “Simulating Physics with Computers”, *International Journal of Theoretical Physics*, vol. 21, n. 6-7 (June), pp. 467 – 488.

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

GOLDBERG, D. E., 1989, *Genetic Algorithms in search, Optimization and Machine Learning*. Massachusetts, Addison – Wesley Publishing Company Inc.

GOSS, S. *et al*, 1989, “Self-organized shortcuts in the Argentine ant”, *Naturwissenschaften*, v. 76, n.12, pp. 579-581.

HAN, K. H., KIM, J. H., 2001, “Genetic Quantum Algorithm and its Application to Combinatorial Optimization Problem”. *Congress on Evolutionary Computation*, Piscataway, NJ: IEEE Press, v.2 (May), pp.1354 – 1360.

HAN, K. H., KIM, J. H., 2002, “Quantum – Inspired Evolutionary Algorithm for a Class of Combinatorial Optimization”, *IEEE Transactions on Evolutionary Computation*, v. 6, n.6 (Dec), pp. 580 – 593.

HEY, T., 1999, “Quantum Computing: An Introduction”, *Computing and Control Engineering Journal*, vol. 10, n. 3 (June), pp.105 – 112.

HOLLAND, J. H., 1975, *Adaptation In Natural and Artificial Systems*, Ann Arbor, University of Michigan Press.

KROPACKZEK, D. J., TURINSKY, P. J., 1991, “In – Core Nuclear Fuel Management Optimization for Pressurized Water Reactors Using Simulated Annealing”, *Nuclear Technology*, v.95, n. 9 (July), pp. 9 – 31.

LANGENBUCH, S., MAURER, W., WERNER, W., 1977, “Coarse – Mesh Flux Expansion Method for Analysis of Space – Time Effects in Large Water Reactors Cores”, *Nuclear Science Engineering*, vol. 63, n. 4, pp. 437 – 456.

LIMA, A. M. M., 2005, *Recarga de Reatores Nucleares Utilizando Redes Conectivas de Colônias Artificiais*. Tese de D.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.

MACHADO, L., SCHIRRU, R., 2002, “The Ant-Q Algorithm Applied to the Nuclear Reload Problem”, *Annals of Nuclear Energy*, v.29, n.12, pp. 1455 – 1470.

MACHADO, M. D., *Aplicação de uma Nova Metodologia Baseada na Propagação Estatística das Incertezas para Avaliar os Limites do DNBR*. Dissertação de M.Sc., COPPE/UFRJ, Rio de Janeiro, Brasil.

MACHADO, M. D., *Um Novo Algoritmo Evolucionário com aprendizado LVQ para a Otimização de Problemas Combinatórios como a Recarga de Reatores Nucleares*. Dissertação de M.Sc., COPPE/UFRJ, Rio de Janeiro, Brasil.

MEDEIROS, J.A.C.C., SCHIRRU,R., 2007, “Identification of nuclear power plant transients using the particle swarm optimization algorithm”, *Annals of Nuclear Energy*, 1806, 1-7.

MOL, A. C. A., 2002, *Um Sistema de Identificação de Transientes com Inclusão de Ruídos e Indicação de Eventos Desconhecidos*, Tese de D.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.

MONTAGNINI *et al*, 1994, “A Well-Balanced Coarse-Mesh Flux Expansion Method”, *Annals of Nuclear Energy*, v.21, n.11, pp.45-53

PEREIRA, C. M. N. A., SCHIRRU, R., 2002, “Genetic Based Identification System Design With Automatic Selection of Meaningful Variables”. Word Scientific.

RFAS – *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.

SHOR, P. W., 1994, “Algorithms for Quantum Computation: Discrete Logarithms and Factoring”, *35th Annual Symposium on Foundations of Computer Science*, IEEE Press, pp. 124- 134 , 20 – 22 Nov.

SILVA, M. H., et al., 2009a, “An Approach Using Quantum Ant Colony Optimization Applied to the Problem of Nuclear Reactors Reload”, *International Nuclear Atlantic Conference – INAC 2009*, Rio de Janeiro, RJ, Brasil, 27 Set – 02 Out.

SILVA, M. H., et al., 2009b, “An Approach Using Quantum Ant Colony Optimization Applied to the Problem of Identification of Nuclear Power Plant Transients”, *International Nuclear Atlantic Conference – INAC 2009*, Rio de Janeiro, RJ, Brasil, 27 Set – 02 Out.

TIPLER, P. A., LLEWELLYN, R. A., 2001, *Física Moderna*, 3 ed. Rio de Janeiro, RJ, LTC.

WAINTRAUB, M., 2009, Algoritmos Paralelos de Otimização por Enxame de Partículas em Problemas Nucleares, Tese de D.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.

WANG, L., et al., 2008, “A Novel Quantum Ant Colony Optimization Algorithm and its Application to Fault Diagnosis”, *Transactions of the Institute of Measurement and Control*, v. 30, n. 3-4 (August), pp. 313-329.

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)