



ALGORITMO QUANTUM PBIL AUTOADAPTÁVEL APLICADO NA OTIMIZAÇÃO
DA RECARGA NUCLEAR DE REATORES PWR

Márcio Henrique da Silva

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

Orientador: Roberto Schirru

Rio de Janeiro

Março de 2014

ALGORITMO QUANTUM PBIL AUTOADAPTÁVEL APLICADO NA OTIMIZAÇÃO
DA RECARGA NUCLEAR DE REATORES PWR

Márcio Henrique da Silva

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

Examinada por:

Prof. Roberto Schirru, D.Sc.

Prof. José de Jesús Rivero Oliva, D.Sc.

Prof. Jose Antonio Carlos Canedo Medeiros, D.Sc

Prof. Antonio Carlos de Abreu Mól, D.Sc.

Dr. Antonio César Ferreira Guimarães, D.Sc.

Prof. Francisco Artur Braun Chaves, D.Sc.

RIO DE JANEIRO, RJ – BRASIL

MARÇO DE 2014

Silva, Márcio Henrique da

Algoritmo Quantum PBIL Autoadaptável aplicado na
Otimização da Recarga Nuclear de Reatores PWR / Márcio
Henrique da Silva. - Rio de Janeiro: UFRJ/COPPE, 2014.

XVII, 77 p.:il; 29,7 cm.

Orientador: Roberto Schirru

Tese(doutorado) – UFRJ/COPPE/ Programa de Engenharia
Nuclear, 2014

Referências Bibliográficas: p.72 - 77

1. Algoritmo Quantum PBIL autoadaptável. 2. Otimização
da Recarga Nuclear. I.Schirru, Roberto. II. Universidade
Federal do Rio de Janeiro, COPPE, Programa de Engenharia
Nuclear. III. Título.

Uma mentira repetida mil vezes torna-se uma verdade.

Joseph Goebbels – ministro da propaganda de Hitler na Alemanha nazista.

A verdade não deixa de ser verdade apenas porque não está de acordo com o que
preceitua a sua vã hipocrisia.

O Autor.

Este trabalho é dedicado a todos que defendem a Meritocracia.

Agradecimentos

Depois de todo o tempo dedicado a esta pesquisa, é hora de recolher tudo e encerrar mais uma etapa de vida, fechar mais um ciclo e seguir adiante em busca de novas oportunidades. Se posso hoje estar escrevendo isto é porque tenho com quem contar afinal, sozinho é sempre mais difícil realizar qualquer tarefa. Por esse motivo é que além buscar a realização pessoal é não menos importante saber reconhecer o esforço daqueles que te estendem a mão e te reconhecem, não porque podem tirar vantagem de você, mas porque simplesmente se importam.

Por isso agradeço muito ao apoio de sempre dos meus pais Lourdes e Ronaldo;

Aos professores da Graduação. Neste grupo em especial gostaria de agradecer ao apoio das professoras Maria Darci e Susana Souza Barros;

Agradeço também ao professor Daniel Sazaki, ao David, Juliana e Patrícia “*no creo em brujas*” (infelizmente sou forçado a acreditar).

À Raquel pela amizade e à Tânia pelo empenho;

Ao Cícero e à Maghali.

Enfim, a todos que contribuíram direta ou indiretamente para a realização deste trabalho sem ter obrigação nenhuma em fazê-lo.

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

ALGORITMO QUANTUM PBIL AUTOADAPTÁVEL APLICADO NA OTIMIZAÇÃO DA RECARGA NUCLEAR DE REATORES PWR

Márcio Henrique da Silva

Março/2014

Orientador: Roberto Schirru

Programa: Engenharia Nuclear

O *Quantum* PBIL (QPIL) associa conceitos da computação quântica como o *qubit* e a superposição linear de estados com a rotina de comandos do PBIL original. O QPIL é eficiente na otimização de funções contínuas e discretas, com resultados que rivalizam os das mais conhecidas técnicas de otimização semelhantes descritas na literatura. No entanto, foi possível observar também que o QPIL apresenta uma acentuada tendência a convergir prematuramente. Isso se deve ao fato de que, além de não utilizar uma etapa referente à mutação, o QPIL depende de um número excessivo de parâmetros operacionais.

Esta Tese propõe um novo modelo de otimização que combina a estrutura básica do QPIL a um modelo autoadaptável de determinação dos valores de parâmetros. Além disso, este novo algoritmo evolucionário com inspirações quânticas, denominado *Self-adaptive Quantum PBIL* (SQPIL), é composto por uma fase adicional de mutação, também autoadaptável como uma tentativa de impedir a ocorrência de convergência prematura. Para fins de análise de performance e comparação de resultados, o SQPIL foi utilizado na otimização da recarga nuclear do 7^o ciclo de operação de Angra 1, um problema de grande interesse econômico que consiste em encontrar a melhor configuração de distribuição entre o combustível fresco e o combustível reaproveitado de ciclos anteriores. Os resultados experimentais confirmaram a eficiência do método proposto em comparação a outras técnicas de otimização disponíveis atualmente.

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

SELF-ADAPTIVE QUANTUM PBIL ALGORITHM APPLIED IN THE NUCLEAR
RELOAD OPTIMIZATION OF PWR REACTORS

Márcio Henrique da Silva

March /2014

Advisor: Roberto Schirru

Department: Nuclear Engineering

Quantum PBIL (QPIL) associates quantum concepts as the qubit and the linear superposition of quantum states with PBIL's basic structure. It has proved to be capable to deal with both, continuous and discrete functions, which is confirmed by the results that match to the ones obtained by other similar techniques available in the literature. Nevertheless, it was also observed that QPBIL is susceptible to the premature convergence effects. This is due to the fact that, not only it doesn't have a mutation stage, but also it depends on an excessive set of parameter values.

This Thesis proposes a new optimization model which combines QPBIL's basic structure with a self-adaptive method of parameter values determination. Moreover, this new algorithm, named Self-adaptive Quantum PBIL (SQPBIL), is composed by an additional stage of mutation, which is also self-adaptable, to avoid the effects of premature convergence. Aiming to evaluate its performance and compare its results, SQPBIL was applied in the optimization of nuclear reload in the 7th operation cycle of Angra 1, an issue of great interest regarding economy resources. It consists on finding the optimal configuration of shuffling between fresh and burned fuel assemblies. Experimental results have confirmed the method efficiency in comparison to the ones found by other similar optimization techniques available nowadays.

Sumário

1	Introdução.....	1
2	Fundamentos da computação quântica e o algoritmo PBIL original.....	12
2.1	Fundamentos da computação quântica.....	13
2.1.1	Algoritmos evolucionários com inspirações quânticas.....	14
2.2	O algoritmo PBIL.....	17
2.2.1	Etapas do processo de otimização do algoritmo PBIL original.....	18
3	O Algoritmo <i>Quantum</i> PBIL.....	21
3.1	Descrição do algoritmo.....	22
3.2	Etapas do processo de otimização do <i>Quantum</i> PBIL.....	25
4	Algoritmo <i>Quantum</i> PBIL autoadaptável.....	30
4.1	Algoritmo <i>Quantum</i> PBIL autoadaptável.....	31
4.2	Descrição dos mecanismos autoadaptáveis.....	31
4.2.1	Mecanismo autoadaptável para definição de taxas de aprendizado (θ e ω).....	32
4.2.2	Mecanismo autoadaptável para definição de taxa de mutação (d).....	34
4.2.3	Mecanismo autoadaptável para definição do número de agentes (\mathcal{A}).....	36
5	Gerenciamento do combustível nuclear em uma usina PWR.....	39
5.1	Procedimento da recarga do núcleo.....	40
5.2	Modelagem do problema.....	42
5.3	A Função <i>Fitness</i>	44

6	SQPBIIL aplicado na otimização da recarga nuclear	47
6.1	SQPBIIL aplicado na otimização do ciclo 7 de Angra 1.....	48
6.2	Metodologia	49
6.3	Resultados	50
6.4	Comparação com outros trabalhos	53
6.5	Sobre o número de agentes	54
7	Conclusão	57
7.1	Conclusão.....	58
7.2	Propostas de sequencia de Tese	60
	ANEXO 1 - Teoria dos múltiplos universos	61
	ANEXO 2 - O sentido de colapsar	63
	ANEXO 3 - Conceitos básicos associados à operação de uma central nuclear	65
	ANEXO 4 - Limitações do RECNOD.....	67
	ANEXO 5 - Um exemplo de aplicação do modelo <i>Random Keys</i>	70
	Referências Bibliográficas.....	72

Lista de Figuras

2.1	Superposição de estados quânticos.....	16
2.2	Pseudocódigo do PBIL original.....	20
3.1	Representação de um vetor <i>qubit</i> no plano cartesiano.....	23
3.2	Primeira etapa da atualização vetorial.....	27
3.3	Segunda etapa da atualização vetorial.....	28
3.4	Pseudocódigo do <i>Quantum PBIL</i>	29
4.1	Pseudocódigo do SQPBIL.....	38
5.1	Modelo do núcleo de Angra 1.....	42
5.2	Região de simetria de octeto.....	43
6.1	Modelo de <i>Random Keys</i>	49
6.2	Evolução da concentração de Boro.....	50
6.3	Evolução da potência média.....	51
6.4	Evolução da <i>Fitness</i>	52
6.5	Evolução do número de agentes.....	54
A2.1	Orbital eletrônico do hidrogênio.....	63
A2.2	Orbital eletrônico do hidrogênio reduzido.....	64
A4.1	Esquema da interface entre SQPBIL/RECNOB.....	69

A5.1	Exemplo de vetor binário gerado pelo algoritmo	70
A5.2	<i>String</i> de valores inteiros gerados a partir do vetor binário P.....	70
A5.3	<i>String</i> de valores inteiros reordenados.....	71
A5.4	Padrão analisado	71

Lista de Tabelas

6.1	Lista de experimentos	52
6.2	Técnicas usadas para maximizar o comprimento do 7º ciclo de Angra 1	53
A.1	Exemplo de 17 universos para identificação de fatores primos.....	62

LISTA DE SIGLAS

Pressurized Water Reactor (PWR)

Dióxido de urânio (UO₂)

Elementos combustíveis (ECs)

Problema da otimização da recarga (POR)

Não-polinomial (NP)

Algoritmos evolucionários (AEs)

Fuel Optimization for Reloads: Multiple Objectives by Simulated Annealing (FORMOSA)

Fuel Optimization for Reloads: Multiple Objectives by Genetic Algorithm (FORMOGA)

Simulated Annealing (SA)

Genetic Algorithm (GA)

Algoritmo genético (AG)

Ant Colony System (ACS)

Ant System (AS)

Traveling Salesman Problem (TSP)

Parameter Free PBIL (FPBIL)

Population-Based Incremental Learning (PBIL)

Genetic Quantum Algorithm (GQA)

Quantum Ant Colony Optimization (QACO)

Ant Colony Optimization (ACO)

Quantum PBIL (QPBIL)

Self-adaptive Quantum PBIL (SQPBIL)

Padrão de recarga (PR)

Dias Efetivos à Plena Potência (DEPP)

Concentração crítica de boro (B_c)

Fator de pico de potência radial (F_{XY})

Potência média relativa (P_m)

Fator de multiplicação efetivo (k_{eff})

Veneno queimável (VQ)

LISTA DE SÍMBOLOS

$|\Psi\rangle$ - estado quântico geral

Ψ - função de onda associada aos estados $|0\rangle$ e $|1\rangle$

α – densidade de probabilidade associada ao estado $|0\rangle$

β – densidade de probabilidade associada ao estado $|1\rangle$

$|\alpha|^2$ – probabilidade associada ao estado $|0\rangle$

$|\beta|^2$ – probabilidade associada ao estado $|1\rangle$

α^* - complexo conjugado de α

β^* - complexo conjugado de β

P - vetor de distribuição de probabilidades

α - taxa de aprendizado positivo do PBIL

β - taxa de aprendizado negativo do PBIL

\hat{A} – ângulo inicial do vetor *qubit*

$\mathcal{R}(\theta)$ - portão de rotação quântico etapa 1

$\mathcal{R}(\omega)$ - portão de rotação quântico etapa 2

θ - taxa de aprendizado positivo do QPBIL

ω - taxa de aprendizado negativo do QPBIL

\mathcal{D} - *fitness* proporcional

$F(i)$ *fitness* i -ésimo indivíduo

$\langle F \rangle$ - *fitness* média da população

ζ - taxa autoadaptada de aprendizado positivo

μ - taxa autoadaptada de aprendizado negativo

d - taxa autoadaptada de mutação (fronteira de dilatação)

c - número de componentes “corretas”

c' - número de componentes “incorretas”

I_{Melhor} - *bit* do melhor indivíduo

\mathcal{D}_s - fronteira de contração

\mathcal{A} - número de agentes

n - número de *bits* compondo a *string*

\mathcal{P}_0 (na primeira geração) - população inicial correspondente a $c = 0$

\mathcal{P}_0 (ao longo do processo de busca) – experiência adquirida

Introdução

De todos os setores de infraestrutura, a área energética é apontada como uma das que mais contribuem para o crescimento econômico (ABEN, 2009). Isto se deve ao fato de os projetos relacionados à produção de energia demandarem um volume expressivo de turbinas, geradores e outros equipamentos de grande porte que alimentam a indústria, a construção civil e os serviços de engenharia.

Além disso, a infraestrutura energética é vital para sustentar a expansão da economia já que, como evidenciado pelo então presidente da ABEN, Guilherme Camargo, “sem os investimentos necessários para aumentar a oferta de energia elétrica no país, o crescimento econômico de longo prazo é inviável”.

Camargo também defende a idéia de que para dar estabilidade e segurança ao sistema elétrico nacional, o Brasil não pode prescindir da operação de usinas térmicas. Isto porque, segundo ele, “apesar de sua vocação hidrelétrica, a complementação térmica se faz necessária” pois “a ausência de termelétricas foi uma das causas do apagão de 2001”.

As opções que se apresentam para a expansão térmica são óleo combustível, carvão, gás e energia nuclear. Enquanto as duas primeiras apresentam gravíssimos problemas ambientais como a poluição do ar e a emissão de gases responsáveis pelo efeito estufa, o gás é um insumo que ainda precisa ser importado. Para explorar as reservas nacionais, será preciso um alto investimento na construção de uma rede de gasodutos.

Dessa forma, a energia nuclear é a única que se apresenta como alternativa viável e econômica para a expansão térmica. Além da não emissão de poluentes, a geração nuclear é altamente segura visto que, como afirma o ex-presidente do Clube de Engenharia Helói Moreira, “as usinas nucleares contam com avançados sistemas de supervisão e controle que impedem a ocorrência de acidentes”.

A energia nuclear será essencial para suprir a demanda energética de todo o país nas próximas décadas, sendo especialmente importante para o sudeste, acredita Camargo. Segundo ele, de acordo com estimativas do Ministério de Minas e Energia no ano de 2009, a capacidade instalada terá que crescer num ritmo de 3 mil megawatts médios por ano até 2015 para fazer frente ao aumento da demanda energética.

Segundo o plano nacional de energia 2030 (PNE 2030), a demanda deve crescer 3,5% ao ano até o fim do período. Será necessário adicionar 53,7 gigawatts à matriz elétrica até 2030, o que fará com que a capacidade do sistema interligado chegue a 156,3 gigawatts. Será muito difícil atingir esta meta sem investir em geração nuclear. Diante deste quadro, as centrais nucleares se mostram uma possível alternativa na produção de energia. Apesar dos riscos associados a este tipo de central energética, o bem produzido é considerado limpo e de baixo impacto ambiental, desde que tudo ocorra dentro das condições normais de segurança.

O Brasil passou a utilizar centrais nucleares para complementar sua matriz energética a partir da década de 70, a princípio apenas com a central Angra 1 e, alguns anos mais tarde, Angra 2. Essas centrais nucleares utilizam reatores à água pressurizada (do inglês *Pressurized Water Reactor* – PWR). Nelas o combustível nuclear é composto por um conjunto de pastilhas de dióxido de urânio (UO₂) posicionados no interior de varetas que compõem estruturas maiores conhecidas como Elementos Combustíveis (ECs).

A queima do combustível nuclear libera uma grande quantidade de energia para o meio externo. Quando a energia produzida não é suficiente para manter a central em operação à potência nominal, se torna necessário substituir o combustível queimado por

novos ECs. No entanto, essa queima não ocorre de maneira homogênea e isso permite que vários ECs possam ser reutilizados no momento em que o núcleo é recarregado. O intervalo entre recargas é conhecido como ciclo de operação.

O Problema da Otimização da Recarga (POR) é uma questão de grande interesse econômico para a Engenharia Nuclear. Ele consiste, em linhas gerais, na busca por uma configuração de núcleo onde a distribuição dos ECs novos e reaproveitados que permita a maximização do ciclo de operação. O ciclo maximizado deverá permitir que a central permaneça em operação à plena potência, dentro de seus limites de segurança, pelo maior intervalo possível até que outra recarga se torne necessária.

Trata-se de um problema combinatório de classe não-polinomial (NP), isto é, onde o número de soluções cresce exponencialmente em relação ao número de ECs considerados. Inicialmente, o POR era resolvido por engenheiros da usina que se baseavam apenas na sua experiência adquirida ao longo dos anos pelo método de tentativa e erro associado a heurísticas básicas. Posteriormente, a utilização de modelos de otimização baseados em inteligência artificial como os Algoritmos Evolucionários (AEs), possibilitou a obtenção de melhores resultados em simulações do POR.

Os AEs são na verdade modelos matemáticos desenvolvidos para lidar com problemas de difícil resolução onde métodos clássicos com o *hill climbing* e o método de gradiente não são aplicáveis. Estas ferramentas pertencem ao ramo da Inteligência Computacional conhecido como Computação Evolucionária. Dentre os vários métodos já utilizados na otimização do POR destacam-se:

O método proposto por GALPERIN et al (1989) que consiste num sistema para gerar configurações ótimas de recarga. O sistema era baseado em um método de busca heurística e implementado em linguagem de programação Common Lisp. A base de conhecimento incorporado da física de reatores compreende as operações do reator e uma abordagem geral de estratégia de gestão de combustível. A base de dados inclui

uma descrição do sistema físico envolvido, ou seja, a geometria do núcleo e armazenamento de combustível. O 5º ciclo da Unidade 1 do reator a água pressurizada de *Three Mile Island* foi escolhido como um caso de teste. A aplicação do sistema para o caso de teste revelou um processo de auto-aprendizagem, através do qual um número relativamente grande de “configurações próximas do ideal” foi descoberto. Várias soluções selecionadas foram submetidas a uma análise detalhada demonstrando, de acordo com os autores, um excelente desempenho;

O código FORMOSA (*Fuel Optimization for Reloads: Multiple Objectives by Simulated Annealing* (SA)) desenvolvido por KROPACKZEK e TURINSKY (1991). Nele é utilizada uma técnica estocástica para minimizar funções conhecida como *Simulated Annealing*. Este código foi capaz de obter bons resultados na otimização da recarga de reatores do tipo PWR, chegando a ser comercializado por seus autores;

O código FORMOGA (POON e PARKS, 1992), onde a ferramenta utilizada para minimização da função objetivo são os Algoritmos Genéticos (AGs) em lugar do SA. O uso dos AGs tornou possível obter um desempenho superior àquele atingido com o SA na otimização de um PWR composto por 193 ECs. Os autores ressaltam que o uso dos AGs fornece melhores resultados particularmente para o processamento em paralelo;

O método proposto por CHAPOT (2000), que consistia em uma codificação inovadora, o Modelo de Listas. De acordo com o autor, esse modelo evita o uso de variantes do operador recombinação dos AGs e gera somente configurações de núcleo válidas. Este trabalho também apresenta o código RECNOD que se baseia no Método de Expansão de Fluxo. O sistema genético GENESIS/ALGER/RECNOD mostrou ser quase duas vezes mais rápido que seu antecessor (GENESIS/ALGER/ANC).

O pacote de programas computacionais de EDORGAN e GEÇKINLI (2003), desenvolvido com o objetivo de auxiliar o procedimento de recarga em usinas PWR. O pacote é composto por uma rede neural artificial associada a uma implementação dos AGs. Enquanto a rede neural tem a função de buscar padrões ótimos de recarga através da previsão de parâmetros como a distribuição de potência, os AGs são encarregados de automatizar a geração dos mesmos. Os autores afirmam que esta forma de tratar o problema torna possível reduzir o tempo de cálculo. Isso permite que um número maior de padrões de recarga seja analisado, aumentando a probabilidade de encontrar um ótimo desejável.

O sistema computacional paralelo baseado no *Ant Colony System* (ACS) denominado *Artificial Ant Colony Networks* para reatores de usinas do tipo PWR apresentado por LIMA (2005). Neste trabalho, os autores propõem o uso do modelo de ilhas (desenvolvido originalmente para ser usado em AGs) adaptado ao ACS. As ilhas se comunicam entre si trocando indivíduos na busca pela obtenção do valor ótimo. O ACS é um algoritmo para problemas de otimização combinatorial inspirado pela observação de formigas reais desenvolvido por DORIGO e GAMBARDELLA (1996). Ele tem seus fundamentos no *Ant System* (AS) e no *Ant-Q* (uma extensão do AS com *Q-learning*, uma técnica de reforço de aprendizagem). Algoritmos como estes pertencem à família das colônias de formigas artificiais, indicadas para a resolução de problemas como o *Traveling Salesman Problem* (TSP), o *Quadratic Assignment Problem* e o *Job-Scheduling Problem*;

O algoritmo *Parameter Free* PBIL (FPBIL) proposto por CALDAS (2006), que parte da premissa de que quantos menos parâmetros um algoritmo utiliza, menor é o risco de que ele não obtenha seu máximo desempenho. Trata-se de uma nova versão do algoritmo PBIL padrão que utiliza mecanismos de autoadaptação para torná-lo mais eficiente. Neste trabalho, o FPBIL é aplicado na otimização da recarga nuclear da central Angra 1 obtendo bons resultados.

De acordo com os exemplos apresentados, a utilização de AEs no POR é uma prática realizada há pelo menos 20 anos com relativo sucesso. Durante este período, estas ferramentas têm evoluído, contribuindo para a consolidação dos sistemas artificiais de aprendizagem em substituição aos modelos clássicos, incapazes de resolver problemas do mundo moderno de maneira eficiente.

Os AEs são inspirados na evolução de populações de indivíduos na natureza. Através das gerações, indivíduos cada vez mais aptos são criados, enquanto a população evolui. Tais indivíduos são aproximações para uma solução idealizada como sendo o valor ótimo da função que está sendo analisada.

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 à evolução de populações de indivíduos que são melhores adaptados ao seu ambiente que os indivíduos que os originaram, da mesma maneira que ocorre na natureza.

Um exemplo deste tipo de ferramenta são os sistemas adaptativos geneticamente motivados conhecidos como Algoritmos Genéticos (AGs) (GOLDBERG, 1989). Eles se baseiam na seleção natural e recombinação genética unindo os princípios darwinianos da sobrevivência do mais apto com uma troca randomizada de informações para guiar a evolução de seus indivíduos.

Desse modo, fenômenos naturais como o *crossing over* e a mutação genética são traduzidos em operadores matemáticos que atuam no algoritmo ao longo da evolução dos agentes artificiais através das gerações.

Baseado no comportamento dos AGs, BALUJA (1994) desenvolveu o algoritmo de incremento de aprendizado baseado na população (do inglês *Population-Based Incremental Learning*, PBIL). Nele, um vetor de distribuição de probabilidades é utilizado para descrever o que para o AG é a população de indivíduos. Embora o PBIL compartilhe vários conceitos empregados pelos AGs, ele não é classificado como um algoritmo com inspirações biológicas mas sim como uma técnica de otimização estocástica simples.

Desde o seu surgimento, os AEs vem se consolidando como ferramentas eficientes e confiáveis na otimização de vários problemas de difícil resolução. Existem exemplos de aplicações de tais métodos em diagnóstico de falhas (SILVA et al, 2009) e gerenciamento de combustível em usinas nucleares (SILVA, 2010), determinação de itinerários de veículos (GAMBARDELLA et al, 1999), catalogação (MERKLE, 2001), formação de união entre robôs (YU et al, 2009), projeto de motores (KOROSEC, 2008), e, até mesmo, reconstrução de concentração clorofilática em amostras de água salgada (SOUTO et al, 2004).

Apesar deste relativo sucesso, estes algoritmos apresentam problemas geralmente associados à dependência de parâmetros como o tamanho da população, valor da taxa de aprendizado (que pode ter diferentes denominações dependendo da ferramenta de otimização em questão), taxa de mutação e valor da função heurística (quando utilizadas) entre outros.

Por essa razão, muitos pesquisadores têm se empenhado ao longo dos anos na busca por tornar os AEs ainda melhores, superando estas limitações. Alguns exemplos deste tipo de prática são encontrados em métodos de otimização multiobjetivo (GAMBARDELLA et al, 1999), algoritmos híbridos (ERDOGAN e GEÇKINLI, 2003), paralelização (RANDALL e LEWIS, 2002) e mecanismos de autoadaptação (CALDAS, 2006) com a intenção de tornar o algoritmo independente de parâmetros.

Uma das mais recentes propostas que visam melhorar a performance destes algoritmos está relacionada à transposição de conceitos oriundos da computação quântica para a linguagem binária convencional, dando origem aos AEs com inspirações quânticas. Estes algoritmos pertencem a uma classe que tem como principal característica o fato de utilizar conceitos da computação quântica associados à rotina de comandos utilizada pelos AEs regulares. Atualmente, existem vários exemplos deste tipo de ferramenta dentre os quais se destacam:

O *Genetic Quantum Algorithm* (GQA) desenvolvido por HAN e KIM (2001), que apresenta uma abordagem com inspirações quânticas dos AGs. Neste trabalho o GQA é aplicado na otimização do problema da mochila (*knapsack*), conhecido por ser um *benchmark* de natureza combinatória bastante utilizado na validação de algoritmos. Ele faz uso de vários vetores *Qubit* trabalhando simultaneamente na geração das soluções binárias enquanto a população evolui;

O *Quantum Ant Colony Optimization* (QACO), um novo algoritmo evolucionário apresentado por Wang et al (2008), que combina a teoria quântica com a aprendizagem coletiva de uma colônia de formigas reais do *Ant Colony Optimization* (ACO) (DORIGO e GAMBARDELLA, 1996). O feromônio é representado por uma *string* de *qubits* e sua atualização depende do ângulo de rotação aplicado a ele. Não se utiliza nenhum tipo de função heurística ou etapa de evaporação, o que compromete o desempenho do algoritmo quando utilizado na otimização de funções complexas e problemas reais (SILVA, 2010).

De acordo com o trabalho desenvolvido anteriormente (SILVA, 2010), foi identificada a possibilidade de utilizar a estrutura básica do algoritmo PBIL para o desenvolvimento de um novo algoritmo de otimização como inspirações quânticas. Este novo método foi denominado *Quantum PBIL* (SILVA e SCHIRRU, 2011a). Por ser uma versão quântica do PBIL original, o QPBIL compartilha de algumas de suas características como o uso de um vetor de distribuição de probabilidades para gerar a população de indivíduos binários. No entanto, o vetor de distribuição de probabilidades do QPBIL é composto por *bits* quânticos (HEY, 1999). Além disso, as componentes desse vetor são atualizadas por meio de um portão de rotação quântico no lugar do processo iterativo utilizado no PBIL.

Anteriormente, o QPBIL foi utilizado na otimização de funções *benchmark* (SILVA e SCHIRRU, 2010a), exemplos de TSP simétrico e assimétrico (SILVA e SCHIRRU, 2011) além do POR (SILVA e SCHIRRU, 2011a) obtendo resultados

satisfatórios em todos estes problemas com relação à qualidade dos resultados obtidos, custo computacional e à complexidade associada à sua implementação.

Porém, considerando que o QPBIL usa uma estrutura muito semelhante à do PBIL, observou-se empiricamente que ambos apresentam problemas parecidos, dentre os quais, a tendência a convergir prematuramente é a mais importante. É comum que o algoritmo fique estagnado em mínimos locais no início do processo de otimização, o que irá impedi-lo de obter melhores soluções comprometendo, assim, sua performance. Esse comportamento foi observado não apenas na otimização de funções contínuas, mas também nos casos de TSP analisados e no POR.

Por outro lado, quando a convergência prematura não é observada, o QPBIL consegue obter boas soluções mais rapidamente que outros métodos analisados. Dessa forma, para melhorar o desempenho do algoritmo e, conseqüentemente, sua capacidade de otimização, o mais importante é encontrar uma maneira de tirar vantagem de sua velocidade para evitar que seus resultados convirjam prematuramente.

Tendo em vista que parâmetros operacionais como tamanho da população, taxas de aprendizado e mutação, entre outros, têm uma grande influência no comportamento do algoritmo, este trabalho propõe um novo método para ajustar alguns deles automaticamente. Para o desenvolvimento deste novo modelo, foi utilizada a seguinte metodologia:

- levantamento bibliográfico enfocando o estado da arte dos algoritmos de otimização baseados em computação quântica;
- desenvolvimento do protótipo do modelo proposto em uma plataforma MatLab 6.5;
- definição de uma plataforma de testes com funções contínuas e discretas, além de problemas de classe NP – completa, de modo a comparar o comportamento do algoritmo em relação a outros disponíveis na literatura;

- utilização no POR para o caso particular do 7^o ciclo de operação da central nuclear Angra 1.

O objetivo desta Tese é propor um novo método para superar a convergência prematura observada em testes com o QPBIL visando melhorar sua performance na otimização do POR e, futuramente, em outros problemas da Engenharia Nuclear. Isto é feito através do uso em conjunto de mecanismos de autoadaptação de parâmetros semelhantes àqueles propostos por CALDAS (2006) para o FPBIL associados à estrutura básica do QPBIL, dando origem a um novo modelo de otimização com inspirações quânticas denominado *Self-adaptive Quantum PBIL* (SQPBIL).

A motivação desta pesquisa é analisar a performance do método através da comparação dos resultados obtidos com o SQPBIL aos de outros métodos semelhantes disponíveis na literatura. Tendo em vista que o POR do 7^o ciclo de operação da central Angra 1 vem sendo objeto de estudos utilizando várias ferramentas de otimização ao longo de pelo menos dez anos, existe uma base de dados muito rica para a realização deste estudo.

Como comprovado pelos resultados publicados em (SILVA e SCHIRRU, 2014), o SQPBIL é um novo método de otimização *quantum*-inspirado perfeitamente viável, eficiente e de fácil implementação. Tendo em vista que se trata da primeira vez que conceitos quânticos são associados a mecanismos de obtenção automatizada de parâmetros, fica evidenciada a originalidade do método. Isso por si só já bastaria para qualificar este estudo como objeto de uma tese de doutorado.

A relevância do método desenvolvido se deve à aplicabilidade do mesmo no POR dado o seu reconhecido apelo financeiro. Considerando que uma das principais razões para o desenvolvimento de um modelo de otimização automatizado, como é o caso do SQPBIL, é a sua utilização em problemas reais, o uso do algoritmo no POR também serve como justificativa para a implementação e desenvolvimento deste projeto.

Este documento é estruturado da seguinte maneira: no capítulo 2 são apresentados os fundamentos teóricos da Computação Quântica de cujos conceitos os algoritmos *quantum* – inspirados se utilizam, além da descrição do algoritmo PBIL padrão;

O capítulo 3 trata do algoritmo *quantum* – inspirado implementado no início do desenvolvimento desta proposta de tese, que associa a representação de *bits* quânticos e o conceito de superposição linear de estados quânticos do qual deriva a teoria do paralelismo quântico, aos mecanismos de aprendizagem do algoritmo *Population-Based Incremental Learning*, o PBIL original, denominado *Quantum PBIL* (QPBIL).

O capítulo 4 descreve o modelo auto-adaptável e apresenta o produto desta tese, o algoritmo *Self-adaptive Quantum PBIL* (SQPBIL) que visa aprimorar a capacidade de otimização do QPBIL por meio de um método autoadaptável para obtenção de valores dos parâmetros operacionais além de uma etapa adicional de mutação.

O capítulo 5 descreve o procedimento da recarga nuclear de um reator do tipo PWR com ênfase no caso particular do reator da Central Angra 1. São discutidos entre outros tópicos, a modelagem utilizada para a análise do problema, que permite sua resolução pelos métodos de otimização analisados neste trabalho, e a função *fitness* adotada;

No capítulo 6 são apresentados e discutidos os resultados obtidos pelo SQPBIL na otimização da recarga nuclear para o caso particular do 7^o ciclo de operação da Central Nuclear de Angra 1.

Finalmente, no capítulo 7, são apresentadas as considerações finais e a proposta para a sequência do trabalho de tese.

Fundamentos da computação Quântica e o Algoritmo PBIL original

Este capítulo apresenta uma breve introdução dos fundamentos da Computação Quântica, desde o seu surgimento no início dos anos 80 do século passado. A forma como tais conceitos foram adaptados aos limites tecnológicos atuais é descrita através da óptica de vários especialistas, tendo como foco contextualizar seu uso associado aos métodos computacionais de otimização conhecidos como Algoritmos Evolucionários.

Além disso, a estrutura e características básicas do algoritmo PBIL original são discutidas, tendo em vista a sua importância conceitual no desenvolvimento do produto final desta Tese.

2.1 Fundamentos da Computação Quântica

A computação quântica é um ramo da ciência proposto no início da década de 80 do século XX por pesquisadores como BENIOFF (1980) e FEYNMAN (1982). Originalmente, buscava-se desenvolver um computador literalmente quântico que pudesse processar suas informações em paralelo. Essa possível forma de processamento foi reconhecida por DEUTSCH (1985) como “fenômeno do paralelismo quântico”.

Segundo NARAYANAN e MOORE (1996), o paralelismo da computação quântica implica que nós podemos calcular de modo eficiente uma solução parcial dentro de cada caminho de um espaço exponencialmente ramificado de uma maneira que não pode ser facilmente simulada em termos clássicos. Isso equivale a dizer que cada caminho de um problema NP-completo ou NP-difícil, onde a dificuldade cresce de forma exponencial, seria explorado num tempo de escala polinomial.

No entanto, ainda citando NARAYANAN e MOORE (1996), existe uma limitação a esta ideia tendo em vista que não é possível observar os resultados de cada caminho computacional separadamente, já que eles estão todos em diferentes universos. Se uma observação é feita, os universos colapsam para um único universo, sendo toda a informação a respeito dos demais perdida. Esta interpretação é conhecida como Teoria dos Múltiplos Universos (EVERETT, 1957) de acordo com a qual devemos imaginar que todos os sistemas quânticos existem em universos paralelos.

Técnicas computacionais que visam explorar o paralelismo quântico devem levar em consideração métodos que garantam que o universo colapsado na observação de alguma forma represente a solução para o problema ou contenha informação suficiente para permitir generalizações para uma solução.

A Teoria dos Múltiplos Universos foi utilizada por SHOR (1994) no desenvolvimento de um método quântico para extrair fatores primos de números inteiros grandes. O algoritmo quântico de Shor foi capaz de solucionar em poucos segundos um problema proposto quase vinte anos antes, conhecido como RSA-129,

feito que até então só havia sido alcançado utilizando 1600 computadores conectados via *internet* ao longo de 8 meses. Este procedimento é descrito no Anexo 1.

Embora mesmo nos dias atuais se mostre inviável o desenvolvimento de um computador quântico operacional “completo”, vários dos conceitos idealizados pelos pesquisadores mencionados, entre outros, fundamentam a metáfora utilizada para definir os algoritmos evolucionários quânticos.

2.1.1 Algoritmos Evolucionários com inspirações quânticas

Um algoritmo evolucionário com inspirações quânticas é o resultado da fusão de conceitos oriundos da computação quântica com a metáfora biológica que estrutura os algoritmos evolucionários regulares. Este tipo de ferramenta armazena sua informação em estruturas conhecidas como *bits* quânticos.

O *bit* quântico ou *qubit* é uma representação probabilística onde existem dois estados possíveis, o estado $|0\rangle$ e o estado $|1\rangle$. Sob a óptica da computação quântica, traça-se um paralelo entre a unidade fundamental de armazenamento de informação dos computadores clássicos com seu ente quântico correspondente. Assim sendo, o sistema quântico descrito por um *qubit* desempenharia no computador quântico a mesma função de armazenar informações que o *bit* clássico tem num computador regular.

Existem pelo menos duas maneiras possíveis de descrever um *qubit*: a polarização de estados de um fóton e a representação dos dois estados do *spin* eletrônico. Nesta tese será utilizada a representação baseada no *spin*, isto é, *down* (\downarrow) e *up* (\uparrow) equivalem aos zeros (‘0s’) e uns (‘1s’) dos dados binários convencionais.

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

A notação *qubit* tem como característica o fato de considerar que o mesmo pode ser encontrado no estado $|0\rangle$, no estado $|1\rangle$ ou em qualquer combinação linear de ambos. De acordo com HEY (1999), a evolução temporal de um sistema quântico é usualmente bem aproximada pela equação de Schroedinger. Em uma representação em coordenadas espaciais, esta equação é diferencial linear parcial, com a propriedade que qualquer superposição linear de autofunções é também uma solução. Esta propriedade de superposição da mecânica quântica significa que o estado geral pode ser descrito como uma superposição de autoestados. No caso do sistema quântico de dois estados utilizado nesta tese, um estado geral $|\Psi\rangle$ será descrito como:

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

Ou seja, uma combinação linear dos estados quânticos $|0\rangle$ e $|1\rangle$. Além disso, Ψ é a função de onda associada aos estados $|0\rangle$ e $|1\rangle$ enquanto α e β são números complexos que especificam as amplitudes de probabilidade dos mesmos. Uma vez que α e β não possuem sentido físico, devem ser consideradas as seguintes relações para determinar a probabilidade de que os estados quânticos $|0\rangle$ e $|1\rangle$ sejam observados:

$$\alpha^* \alpha = |\alpha|^2 - \text{probabilidade associada ao estado } |0\rangle \quad (3)$$

$$\beta^* \beta = |\beta|^2 - \text{probabilidade associada ao estado } |1\rangle \quad (4)$$

Onde α^* e β^* são o complexo conjugado de α e β , respectivamente. O somatório das probabilidades de todos os estados disponíveis deve, de acordo com a condição de normalização, ser igual a 1, isto é:

$$|\alpha|^2 + |\beta|^2 = 1 \quad (5)$$

que também pode ser expressa por (TYPLER, 2001):

$$\int_{-\infty}^{\infty} \Psi^* \Psi dx = 1 \quad (6)$$

Esta propriedade particular dos *qubits* permite representar várias possíveis soluções num espaço de busca binário a partir de um único indivíduo. A figura 2.1 apresenta um exemplo para o uso de um *Qubit* (indivíduo quântico) composto por quatro *qubits*.

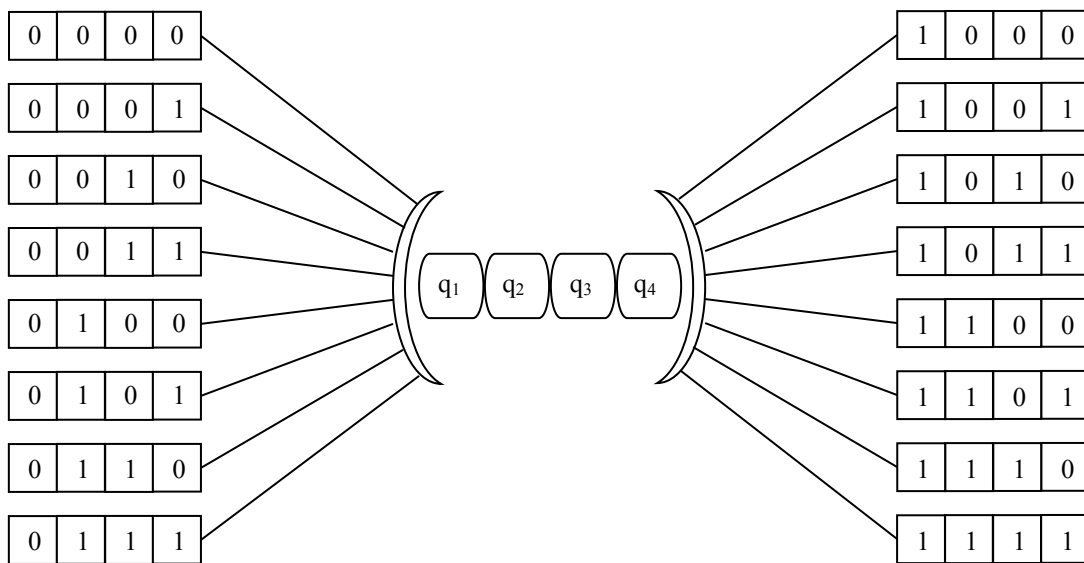


Figura 2.1: Superposição de estados quânticos.

Considerando que cada *qubit* é capaz de representar dois possíveis estados ($|0\rangle$ ou $|1\rangle$), existem 2^4 possíveis soluções binárias geradas por este único indivíduo quântico. Em outras palavras, o *Qubit* da figura 2.1 é capaz de armazenar a informação referente a até 16 indivíduos binários diferentes simultaneamente. Por outro lado, utilizando a notação binária convencional, só seria possível representar estas mesmas soluções se dispuséssemos de 16 *strings*.

Isso significa que a Computação Evolucionária associada à representação *qubit* apresenta uma melhor característica de diversidade de população que outras ferramentas de otimização, uma vez que esta associação permite descrever superposições de probabilidades de estados.

No entanto, a utilização da representação *qubit* torna inviável o uso de portões reversíveis como são os operadores AND, NOT, XOR, OR, entre outros, comumente empregados na atualização dos *bits* convencionais. Por esse motivo foram definidos os portões quânticos. Eles atuam como operadores unitários agindo nos estados básicos dos *qubits* através de pequenas rotações, ora aproximando-o do estado $|0\rangle$ e afastando-o do estado $|1\rangle$, ora fazendo o oposto, de acordo com a evolução da população.

Tendo em vista que o desenvolvimento de um computador literalmente quântico ainda se mostra inviável nos dias atuais, os algoritmos evolucionários inspirados na teoria quântica se mostram como uma importante alternativa no sentido de simular o provável comportamento do mesmo. A performance obtida por tais ferramentas na otimização de problemas de diferentes naturezas comprova sua eficiência a despeito do uso meramente metafórico dos conceitos quânticos.

2.2 O algoritmo PBIL

O PBIL (BALUJA, 1994) foi criado com o objetivo de simular o comportamento do sistema adaptativo motivado biologicamente conhecido como Algoritmo Genético (AG) (GOLDBERG, 1989). Neste algoritmo, os indivíduos são criados geração após geração, a partir de um vetor de distribuição de probabilidades P . Em uma decodificação binária como aquela utilizada nesta tese, cada componente deste vetor representa a probabilidade de que o *bit* correspondente seja igual a 1. O objetivo é criar ativamente um vetor que, com alta probabilidade, descreva uma população de vetores solução com as melhores avaliações.

Diferentemente do que ocorre nos AGs, os operadores variação não são definidos sobre a população. No caso do PBIL as operações ocorrem diretamente sobre

o vetor P através de mecanismos que são derivados daqueles utilizados na aprendizagem competitiva. Por outro lado, a exemplo do que ocorre nos AGs, o algoritmo PBIL também mantém o paralelismo intrínseco (BALUJA, 1994) do processo de busca através da representação de vários pontos distintos do espaço de soluções do problema, representados pelos indivíduos que compõem a sua população.

Além disso, a exploração do espaço de busca converge para um único ponto também no PBIL. Essa característica gera uma dificuldade adicional para o algoritmo, tendo em vista que um único vetor de probabilidades é utilizado, fato que contribui para uma menor diversidade de indivíduos, resultando em problemas de convergência prematura. Visando a obtenção de uma população mais diversificada, no início do processo de busca o vetor P atua como uma distribuição de probabilidades uniforme, isto é, cada uma de suas componentes é igual a 0.5.

$$P_0 = \begin{bmatrix} 0.5 & 0.5 & 0.5 & \dots & 0.5 & 0.5 & 0.5 & 0.5 \end{bmatrix} \quad (7)$$

Isso garante que todas as regiões do espaço de busca tenham a mesma chance de serem escolhidas ao acaso, ou seja, a probabilidade de gerar o valor ‘0’ ou o valor ‘1’ é a mesma em cada posição da cadeia de *bits*. A partir do vetor P serão gerados os indivíduos binários, sequências de zeros e uns que compõem a população do PBIL.

2.2.1 Etapas do processo de otimização do algoritmo PBIL original.

O processo de otimização do PBIL segue a seguinte rotina de comandos: Primeiro, é atribuído o valor inicial, 0.5, a cada uma das componentes do vetor de distribuição de probabilidades, P . É importante lembrar que cada componente deste vetor representa a probabilidade de que o *bit* seja igual a ‘1’, portanto, quando seu valor é 0.5 existe 50% de chance de que o *bit* seja igual a ‘1’ e 50% de chance de que o mesmo *bit* seja igual a ‘0’. Isso irá garantir que, pelo menos na primeira geração, o algoritmo não esteja privilegiando nenhuma região do espaço de soluções realizando, portanto, uma busca aleatória.

Na etapa seguinte, são formados os indivíduos binários. Estes indivíduos são então avaliados pelo seu nível de *fitness*, isto é, uma função de otimização previamente definida, para que o vetor P seja atualizado. A cada geração, são identificados o melhor e o pior indivíduo da população. Se o melhor indivíduo da geração possuir um nível de *fitness* que é melhor (de acordo com o critério do problema em questão) que o da solução ótima conhecida até então, a sua *fitness* passa a ser o novo valor ótimo.

Conhecendo o melhor e o pior indivíduo, é possível realizar a próxima etapa do algoritmo, a atualização das componentes do vetor P . Trata-se de um processo iterativo onde parte do valor original da componente é mantida enquanto um incremento é atribuído ao mesmo. O resultado são alterações nas componentes de P , fazendo com que os indivíduos gerados sejam, ao mesmo tempo, cada vez mais similares ao melhor indivíduo avaliado até então e menos similares ao pior deles. Nesta etapa dois parâmetros desempenham um papel de fundamental importância: as taxas de aprendizado positivo e negativo.

A taxa de aprendizado positivo, α , é utilizada com o objetivo de determinar quanto os novos indivíduos gerados irão se aproximar do melhor indivíduo conhecido até então, ou seja, aumentar a probabilidade de reproduzir as componentes binárias daquele que possui o maior nível de *fitness* da população. A taxa de aprendizado negativo, β , por outro lado, determina quanto os novos indivíduos gerados serão afastados daquele que é o indivíduo com pior nível de *fitness*. Por exemplo, se o pior indivíduo possui um *bit* '1' na posição j , β é utilizada no sentido de diminuir a probabilidade de que a posição j seja também ocupada por um *bit* igual a '1' nos novos indivíduos formados.

Esta é a descrição do algoritmo PBIL padrão considerado para efeito de análise e comparação de resultados nesta tese. Existem diversas variantes do PBIL disponíveis na literatura que utilizam alguns parâmetros e etapas adicionais aos descritos aqui. Na figura 2.2 é apresentado o pseudocódigo do PBIL original.

Inicialização do vetor de probabilidades
Atribuição de P_0

$$P_0 = \begin{array}{|c|c|c|c|c|c|c|c|} \hline 0.5 & 0.5 & 0.5 & \dots & 0.5 & 0.5 & 0.5 & 0.5 \\ \hline \end{array}$$

Cada componente do vetor P , que representa a probabilidade de que o *bit* seja igual a 1, é igual a 0.5 na primeira geração para que nenhuma região do espaço de busca seja privilegiada.

Construção de Indivíduos

```
for i = 1:número de indivíduos
  for j = 1:número de bits
    if P(j) > rand
      I(i,j) = 1
    else
      I(i,j) = 0
    end
  end
end
```

Avaliação da *fitness*

Seleção do Melhor e Pior indivíduo da população.

Atualização do vetor P

Fase 1 : aproxima do Melhor

```
for j = 1:nbits
  P(j) = (1- $\alpha$ )P(j) +  $\alpha$ * I(Melhor,j)
end
```

Fase 2: afasta do Pior

```
for j = 1:nbits
  P(j) = (1- $\beta$ )P(j) +  $\beta$ * I(Pior,j)
end
```

Retorna para a fase de construção de indivíduos

Figura 2.2: Pseudocódigo do PBIL original.

O Algoritmo Quantum PBIL

O PBIL é uma técnica de otimização cuja principal característica é o fato de gerar sua população de indivíduos binários a partir do chamado vetor de distribuição de probabilidades. O uso da estrutura do PBIL associada a conceitos da computação quântica como o *qubit* e a superposição linear de estados quânticos dá origem ao algoritmo *Quantum* PBIL (QPIL).

O objetivo do QPIL é melhorar a performance do PBIL original através da característica de maior diversidade populacional expressa pelo uso da representação *qubit* que permite a obtenção de bons resultados em um menor intervalo de tempo. Neste capítulo serão apresentadas as características básicas desta nova ferramenta de otimização destacando semelhanças e diferenças entre a mesma e sua versão original.

3.1 Descrição do algoritmo.

AEs com inspiração quântica são modelos matemáticos de otimização automatizada que combinam conceitos da computação quântica à metáfora na qual se baseiam os AEs tradicionais. Sua unidade fundamental de armazenamento de informação é o *bit* quântico ou *qubit*. O PBIL é uma técnica de otimização estocástica inspirada pelos AGs que pode ser aplicada a uma grande variedade de problemas. Sua principal área de aplicação compreende problemas nos quais métodos simples ou o método gradiente não são capazes de obter bons resultados.

O uso da estrutura do algoritmo PBIL original associada a conceitos básicos da computação quântica como o *bit* quântico e a superposição linear de estados deram origem a uma nova ferramenta denominada PBIL quântico ou *Quantum PBIL* (SILVA e SCHIRRU, 2011a). De acordo com o que foi apresentado no capítulo 2, o *qubit* é uma representação probabilística do estado binário que pode ser descrita matematicamente como:

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} \quad (8)$$

Onde as componentes α e β são números complexos sendo que $|\alpha|^2$ é a probabilidade de que o *qubit* seja encontrado no estado $|0\rangle$ e $|\beta|^2$ é a probabilidade de que o *qubit* seja encontrado no estado $|1\rangle$, obedecendo a condição de normalização descrita pela relação (5), no capítulo 2. Uma maneira mais simples e didática de representar um *qubit* é mostrá-lo na sua forma vetorial em um plano cartesiano como na figura 3.1:

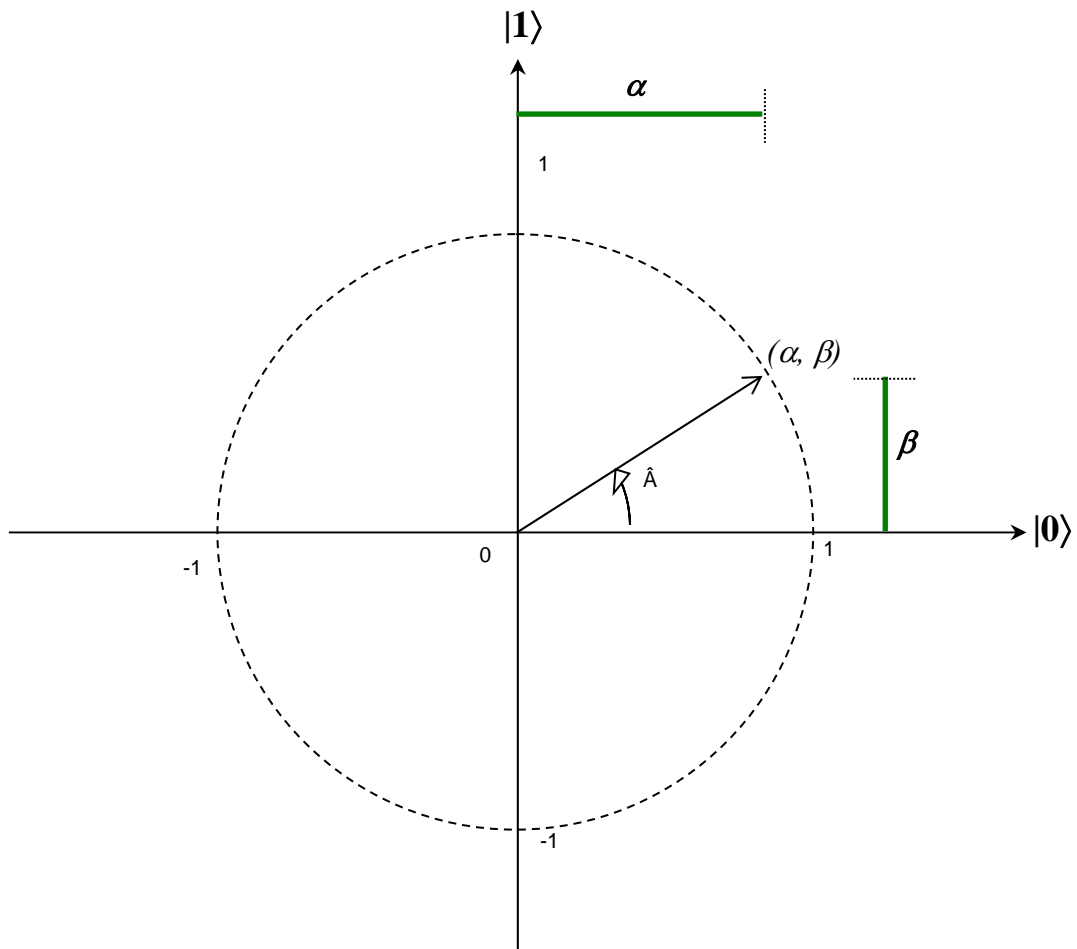


Figura 3.1: Representação de um vetor *qubit* no plano cartesiano.

Nesta figura, o eixo da ordenadas se refere à probabilidade de que o estado $|1\rangle$ seja observado enquanto o eixo das abscissas se refere à probabilidade de que o estado $|0\rangle$ predomine. As projeções do vetor *qubit* em cada um desses eixos fornecem as amplitudes de probabilidade (α, β) . Como $||\textit{qubit}|| = 1$:

$$\alpha = \cos \hat{A} \quad (9)$$

$$\beta = \sin \hat{A} \quad (10)$$

O quadrado destas componentes irá fornecer a probabilidade associada a seus respectivos estados quânticos ($|0\rangle$ ou $|1\rangle$):

$$\alpha^2 + \beta^2 = \cos^2 \hat{A} + \sin^2 \hat{A} = 1 \quad (11)$$

À medida que o ângulo \hat{A} varia, este vetor se aproxima do eixo das ordenadas se afastando do eixo das abscissas e vice-versa. O fato de se aproximar de um desses eixos significa que a probabilidade de que o estado a ele associado seja observado está aumentando. O contrário ocorre em relação ao eixo do qual o vetor está se afastando. A exemplo do que ocorre no PBIL, uma população inteira é criada a cada geração a partir do vetor de distribuição de probabilidades P . No entanto, o vetor P do QPBIL é formado por *qubits*.

$$P = \left[\begin{array}{c|c|c|c} \alpha_1 & \alpha_2 & \cdots & \alpha_m \\ \beta_1 & \beta_2 & \cdots & \beta_m \end{array} \right] \quad (12)$$

Desta maneira, 2^m indivíduos (possíveis soluções no espaço de busca) são criados a partir de P devido ao fenômeno da superposição linear de estados, uma vez que cada um dos *qubits* pode ser observado no estado $|0\rangle$, $|1\rangle$ ou em uma superposição linear dos dois, desde que seja respeitada a condição de normalização (5). Porém, todos os possíveis estados colapsam para uma única solução no momento que ele é observado. O sentido do verbo “colapsar” utilizado aqui é melhor explicado no Anexo 2.

Para que o QPBIL seja capaz de resolver problemas de otimização em espaços de busca contínuos, é necessário converter a notação de *qubits* para a forma binária convencional, ou seja, uma sequência de ‘0s’ e ‘1s’. Isto é feito a partir da observação do vetor de distribuição de probabilidades P que gera soluções válidas mostradas na forma de *strings* binárias.

Para determinar para qual estado quântico o *qubit* irá colapsar, ou seja, em qual estado ele será observado ($|0\rangle$ ou $|1\rangle$), é gerado um número randômico que é então comparado ao quadrado da componente β de cada *qubit*, de acordo com a relação (13).

$$I_{ij} = \begin{cases} 1 & \text{if rand} \leq \beta^2 \\ 0 & \text{if rand} > \beta^2 \end{cases} \quad (13)$$

onde I_{ij} é o j -ésimo bit do i -ésimo indivíduo.

Neste ponto é importante lembrar que a componente β do *qubit* está associada ao estado $|1\rangle$ (β^2 representa a probabilidade de o *bit* ser igual a ‘1’) bem como, no PBIL, cada componente do vetor de probabilidade representa a probabilidade de o *bit* ser igual a ‘1’.

3.2 Etapas do processo de otimização do algoritmo QPBIL

O algoritmo é inicializado com a atribuição dos seguintes parâmetros: número de *bits*, gerações e indivíduos além dos valores de deslocamento angular positivo e negativo, que serão utilizados na etapa de atualização. Além disso, os valores das componentes α e β de cada *qubit* que compõe o vetor P são definidos como sendo iguais a $\frac{\sqrt{2}}{2}$. Como o quadrado destas componentes fornece a probabilidade de que seus respectivos estados ($|0\rangle$ e $|1\rangle$) predominem, isso resulta em:

$$\alpha^2 = \beta^2 = \left(\frac{\sqrt{2}}{2}\right)^2 = 0.5 \quad (14)$$

Isto é, inicialmente todos os *qubits* possuem igual probabilidade de representar os estados $|0\rangle$ e $|1\rangle$. Em seguida, ocorre a construção de indivíduos binários que, posteriormente, serão convertidos para vetores reais para ter seu nível de *fitness* avaliado. Dentre todos os indivíduos que compõem a população, são considerados o melhor e o pior para guiar o processo de atualização das componentes de P . O objetivo

desta etapa é fazer com que os novos indivíduos gerados sejam cada vez mais parecidos com o melhor e cada vez menos parecidos como o pior indivíduo.

Cada *qubit* recebe um deslocamento angular total que é resultado de dois incrementos preliminares. Primeiro, para cada *qubit* j , é determinada uma rotação que o aproxima do melhor indivíduo, isto é, se a componente j do melhor indivíduo for igual a ‘1’, o portão de rotação quântico, R , fará com que o *qubit* desta mesma posição se aproxime do eixo das ordenadas, ou seja, aumentará a probabilidade que o estado $|1\rangle$ predomine. O portão R é uma representação matemática definida como:

$$\mathcal{R}(\theta_j) = \begin{bmatrix} \cos(\theta_j) & -\sin(\theta_j) \\ \sin(\theta_j) & \cos(\theta_j) \end{bmatrix} \quad (15)$$

onde θ_j é o deslocamento angular que representa quanto o *qubit* irá se aproximar do melhor indivíduo.

Em seguida, é aplicado o deslocamento angular ω_j cuja função é determinar o quanto o *qubit* irá afastar os próximos indivíduos gerados do pior indivíduo da população. Por exemplo, se a posição j do pior indivíduo for igual a ‘1’, o portão de rotação quântico fará com que o *qubit* desta mesma posição se afaste do eixo das ordenadas, ou seja, diminuirá a probabilidade de que o estado $|1\rangle$ predomine. É importante destacar que o que muda é o deslocamento angular aplicado sendo o portão quântico utilizado nas sub-etapas de atualização exatamente o mesmo:

$$\mathcal{R}(\omega_j) = \begin{bmatrix} \cos(\omega_j) & -\sin(\omega_j) \\ \sin(\omega_j) & \cos(\omega_j) \end{bmatrix} \quad (16)$$

As figuras 3.2 e 3.3 descrevem o processo de atualização do QPBIL:

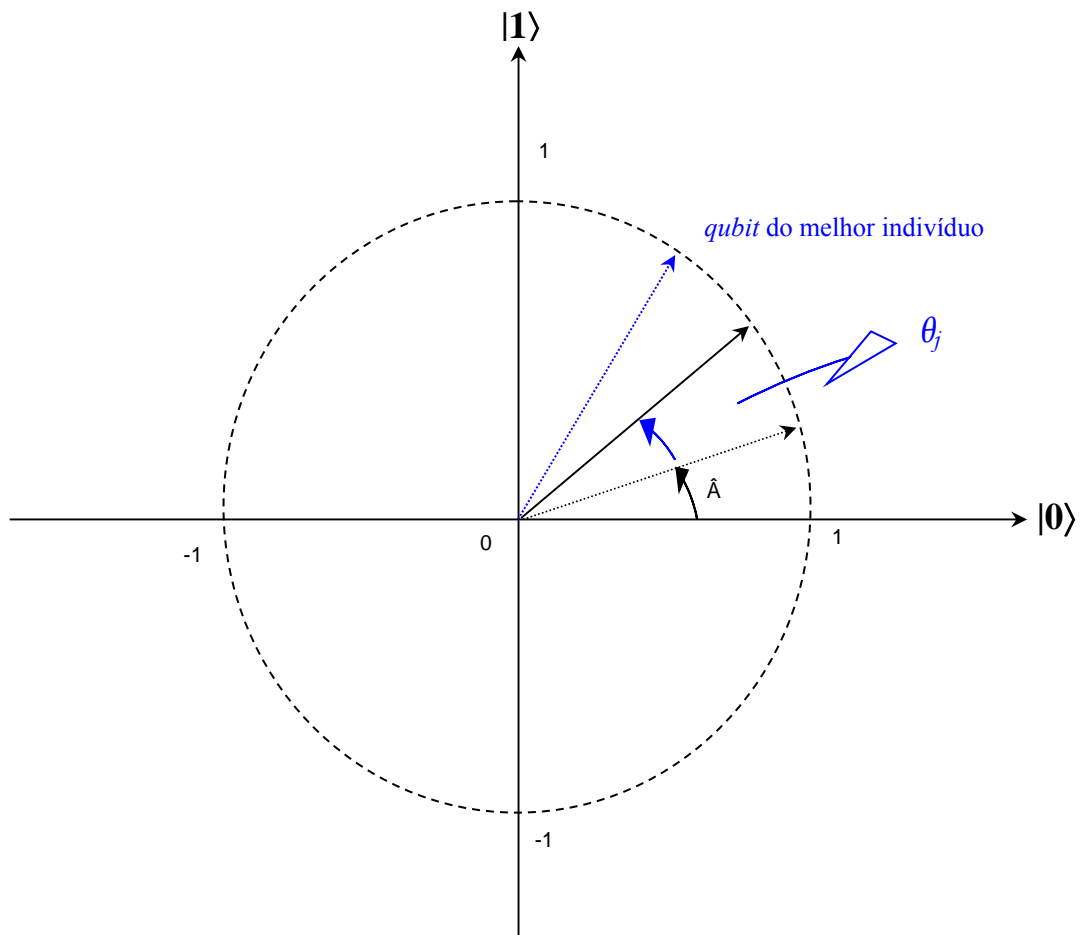


Figura 3.2: Primeira etapa da atualização vetorial.

Na primeira etapa da atualização, o vetor *qubit* é deslocado em direção ao melhor indivíduo por meio de uma rotação de magnitude θ . No exemplo da figura 3.2 é observado que esta rotação aproxima o vetor do eixo das ordenadas, isto é, aumento na probabilidade do estado $|1\rangle$. Em seguida, o vetor *qubit* é deslocado para longe do pior indivíduo por meio de uma rotação de magnitude ω . No exemplo da figura 3.3 é observado que esta rotação aproxima o vetor do eixo das abscissas, isto é, aumento na probabilidade do estado $|0\rangle$.

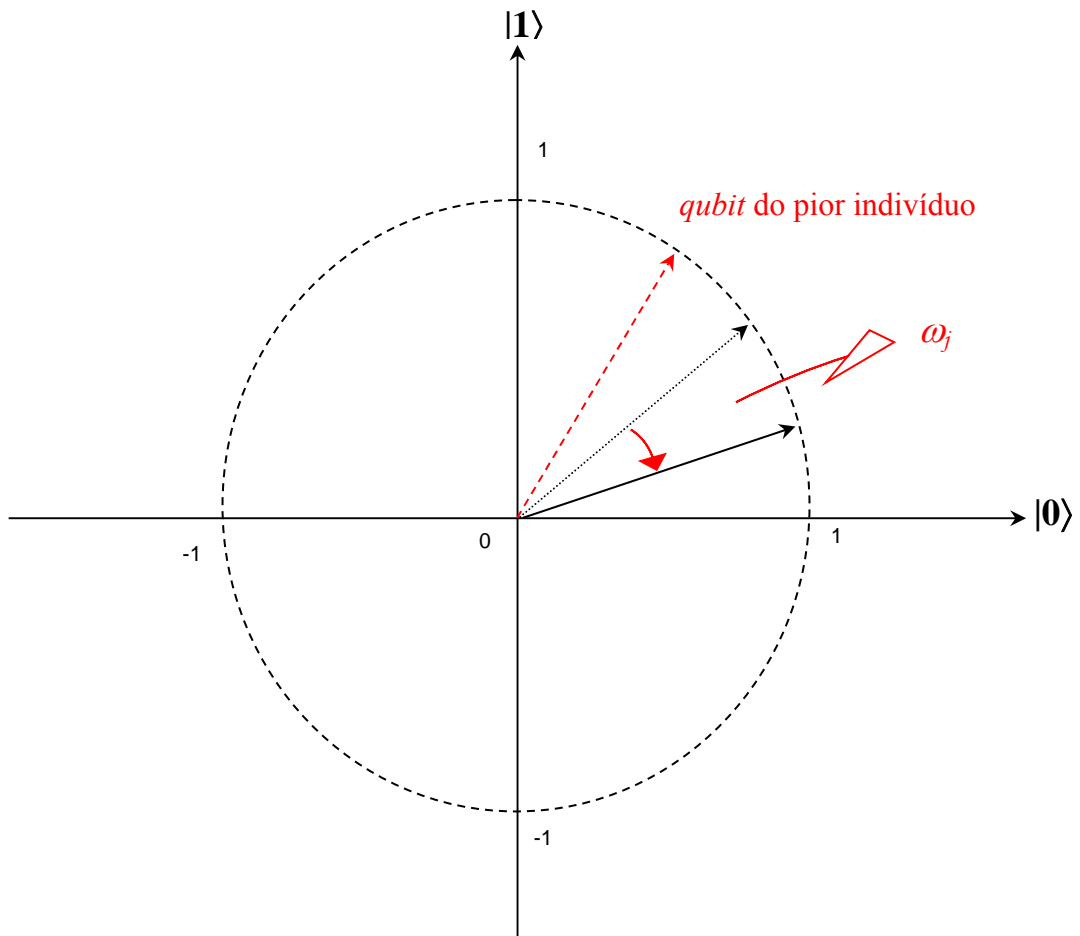


Figura 3.3: Segunda etapa da atualização vetorial.

A figura 3.4 apresenta um pseudocódigo do QPBIL. Devido à já mencionada semelhança que compartilha com o PBIL, o QPBIL mostrou ser suscetível aos mesmos problemas que ele apresenta quando utilizado para otimizar os problemas analisados em (SILVA e SCHIRRU, 2010a), (SILVA e SCHIRRU, 2011) e (SILVA e SCHIRRU, 2011a). A necessidade de encontrar um conjunto ótimo de parâmetros, além da tendência a convergir prematuramente são, sem dúvida, um grande desafio para fazer com que o QPBIL seja capaz de encontrar melhores soluções.

Fase de Inicialização

Atribuição de P_0

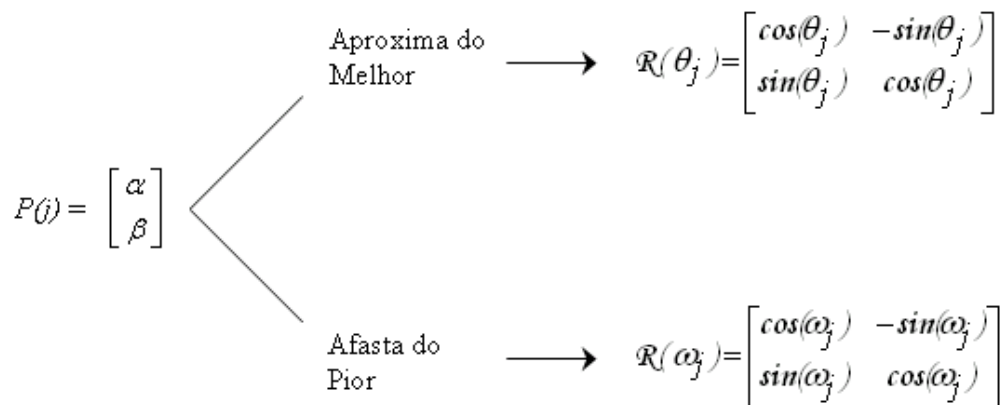
$$P_0 = \begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \cdots & \frac{\sqrt{2}}{2} \\ \frac{2}{\sqrt{2}} & \frac{2}{\sqrt{2}} & \cdots & \frac{2}{\sqrt{2}} \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \cdots & \frac{\sqrt{2}}{2} \\ \frac{2}{2} & \frac{2}{2} & \cdots & \frac{2}{2} \end{bmatrix}$$

Construção de Indivíduos

```
for i = 1:número de indivíduos
  for j = 1:número de bits
    if rand >  $\beta_j^2$ 
       $I_{ij} = 0$ 
    else
       $I_{ij} = 1$ 
    end
  end
end
```

Avaliação da *fitness*

Atualização do vetor P



Retorna para a fase de construção do vetor de distribuição de probabilidades

Figura 3.4: Pseudocódigo do *Quantum* PBIL.

Algoritmo Quantum PBIL autoadaptável

Os resultados obtidos pelo *Quantum PBIL* na otimização de funções *benchmark*, exemplos de TSP simétrico e assimétrico e do POR o qualificam como uma ferramenta capaz de superar vários dos outros métodos baseados em inteligência artificial descritos na literatura. No entanto, nestes mesmos testes, foi possível observar que o algoritmo apresenta uma acentuada a convergir prematuramente, reflexo da estagnação das componentes dos *qubits*.

Visando contornar este problema para evitar a ocorrência da convergência prematura e, conseqüentemente, melhorar capacidade de otimização do QPBIL, foi desenvolvido um novo modelo que busca eliminar a necessidade de se conhecer *a priori* o valor dos principais parâmetros operacionais do QPBIL além de propor uma etapa adicional referente à mutação que é também autoadaptável. Este capítulo descreve este novo algoritmo com inspirações quânticas denominado *Self-adaptive Quantum PBIL* (SQPBIL) que é o produto final desta Tese.

4.1 Algoritmo Quantum PBIL autoadaptável

O *Self-adaptive Quantum PBIL* (SQPBIL) é um novo modelo de otimização que busca superar o problema referente à convergência prematura observada nos testes com o QPBIL. Tal modelo tem como principal característica o fato de ser capaz de definir valores de parâmetros como tamanho da população, ângulos de rotação e taxa de mutação automaticamente, sem a necessidade de nenhum tipo de conhecimento prévio.

Para tanto, o SQPBIL utiliza um método de determinação de componentes corretas bastante similar àquele descrito por CALDAS (2006) para o *Parameter Free PBIL* (FPBIL) segundo o qual, quanto menor o número de parâmetros utilizados, maiores são as chances de o algoritmo obter sucesso no processo de otimização. Ele funciona fazendo com que o QPBIL seja capaz de encontrar o valor de seus principais parâmetros por si mesmo, independentemente do problema que está sendo analisado.

Primeiro, o método proposto é aplicado para definir os valores dos ângulos de rotação θ e ω . O segundo estágio é relacionado à determinação da taxa de mutação. Uma vez que a proposta na qual o SQPBIL se inspira é desenvolvida originalmente para o PBIL e, considerando a ausência de uma etapa de mutação no QPBIL, esta pode ser vista como uma etapa adicional.

É importante mencionar que a mutação existe num algoritmo como o QPBIL para inserir diversidade na população e com isso evitar a convergência prematura. Portanto, o fato de o QPBIL não utilizar uma etapa referente à mutação pode ser considerado uma das causas da estagnação das componentes dos *qubits*. Finalmente, no terceiro estágio, é definido o número de indivíduos da população de agentes artificiais.

4.2 Descrição dos mecanismos autoadaptáveis

Os *bits* quânticos que compõem o vetor de probabilidades do QPBIL são atualizados por meio de dois deslocamentos angulares. No primeiro, é aplicada uma

rotação que deve aproximar os valores de α_j e β_j da j -ésima posição da *string* de cada indivíduo gerado dos valores de α e β da mesma posição na *string* que representa o melhor indivíduo conhecido. No segundo deslocamento, α_j e β_j devem ser atualizados no sentido de serem afastados dos valores de α e β da mesma posição na *string* que representa o pior indivíduo conhecido.

No entanto, como exposto por CALDAS (2006), tal procedimento requer que a população inteira seja avaliada. Por outro lado, apenas a informação referente ao melhor e pior indivíduos é preservada ao longo das gerações. Esta é uma das possíveis causas de convergência prematura em métodos que utilizam uma estrutura semelhante à do PBIL.

4.2.1 Mecanismo autoadaptável para definição de taxas de aprendizado (θ e ω)

A regra de atualização do SQPBIL combina o ângulo de rotação do QPBIL com o modelo proposto por CALDAS (2006). Dessa forma, θ e ω são definidos pelas relações:

$$\theta(k) = \frac{\mathcal{D}(k) * \pi}{\mathcal{A}} \quad (17)$$

$$\omega(k) = \frac{\theta(k)}{20} \quad (18)$$

onde \mathcal{A} é o número de agentes e $\mathcal{D}(k)$ é definido como sendo igual a:

$$\mathcal{D}(k) = \frac{\sum_{i=1}^{\mathcal{A}} \mathcal{F}(i) \cdot I(i, k)}{\sum_{i=1}^{\mathcal{A}} \mathcal{F}(i)} \quad (19)$$

onde $F(i)$ é a *fitness* i -ésimo indivíduo e $I(i,k)$ é seu k -ésimo *bit*. A expressão (19) reflete uma média na qual a informação referente a todos os indivíduos é preservada. Ela pode ser reescrita como:

$$\mathcal{D}(k) = \frac{\sum_{i=1}^{\mathcal{A}} I(i,k)}{\mathcal{A}} + \zeta \cdot \mu \quad (20)$$

onde:

$$\zeta = \frac{1}{\mathcal{A}} \sum_{F(i) > \langle F \rangle} \frac{F(i) - \langle F \rangle}{\langle F \rangle} \quad (21)$$

onde $\langle F \rangle$ é a *fitness* média da população na geração atual e

$$\mu = \frac{\sum_{F(i) > \langle F \rangle} (F(i) - \langle F \rangle) \cdot I(i)}{\sum_{F(i) > \langle F \rangle} (F(i) - \langle F \rangle)} - \frac{\sum_{F(i) < \langle F \rangle} (\langle F \rangle - F(i)) \cdot I(i)}{\sum_{F(i) < \langle F \rangle} (\langle F \rangle - F(i))} \quad (22)$$

A expressão (21) mostra que ζ equivale à taxa de aprendizado positivo do PBIL, embora seja responsável por guiar o processo de busca em direção a melhores indivíduos como um grupo. Por essa razão a informação que é mantida diz respeito à população e não mais a um único indivíduo. De modo semelhante, a expressão (22) mostra que μ tem a mesma função que a taxa de aprendizado negativo do PBIL. Ele guia a busca para longe de soluções indesejáveis também mantendo a informação referente a todos os indivíduos avaliados como uma média.

4.2.2 Mecanismo autoadaptável para definição da taxa de mutação (d)

O segundo estágio propõe a supressão do parâmetro associado à mutação. Independentemente da já mencionada ausência de uma etapa de mutação no QPBIL, ela será considerada para implementar o SQPBIL. Entretanto, será preciso fazer algumas modificações no procedimento descrito por CALDAS (2006).

Originalmente, os valores das componentes de um *qubit* pertencem ao intervalo $[0,1]$. Quando α^2 ou β^2 são iguais a 0 ou 1, existe uma grande chance de ocorrer convergência prematura. Por isso, é utilizada uma estratégia que impede as componentes dos *qubits* de assumirem tais valores. Assim, toda componente α^2 e β^2 têm seus valores limitados ao intervalo $[d, 1-d]$.

Para utilizar esta abordagem o SQPBIL trabalha com um modelo de componentes “corretas” e “incorretas”. Em cada geração o melhor indivíduo conhecido até então é avaliado. Em seguida, a *string* binária que lhe originou é usada para determinar o número de componentes “corretas” (c) de acordo com a seguinte expressão:

$$c = c + 1 \quad \left\{ \begin{array}{l} \text{se } I_{\text{Melhor}} = 0 \text{ e } \beta^2 < d \\ \text{ou} \\ \text{se } I_{\text{Melhor}} = 1 \text{ e } \beta^2 > 1 - d \end{array} \right. \quad (23)$$

De modo semelhante, a *string* binária que deu origem ao pior indivíduo conhecido até então é usada para determinar o número de componentes “incorretas” (c') de acordo com a seguinte expressão:

$$c' = c' + 1 \quad \left\{ \begin{array}{l} \text{se } I_{\mathcal{M}ellior} = 0 \text{ e } d' < \beta^2 < \mathcal{D}_s \\ \text{ou} \\ \text{se } I_{\mathcal{M}ellior} = 1 \text{ e } \mathcal{D}_s < \beta^2 < 1 - d' \end{array} \right. \quad (24)$$

Onde \mathcal{D}_s é definido como:

$$\mathcal{D}_s = \frac{1}{c - 1} \quad (25)$$

Os valores de c são usados para definir os valores de d' , uma espécie de fronteira para as componentes de cada *qubit*. A relação entre os valores de c e d' é dada por:

$$d' = \frac{1}{c + 1} \quad (26)$$

Esta expressão descreve como uma fronteira funciona. À medida que os valores de c aumentam, d' tende a 0. Caso contrário, ele tenderá a 1. Considerando que cada componente do *qubit* passa a ser limitada pela fronteira d' , α^2 e β^2 nunca serão iguais a 0 ou 1. Isto evita que o vetor de probabilidade crie um grande número de cópias de indivíduos com nível de *fitness* alto diminuindo assim o risco de ocorrer convergência prematura.

Enquanto os valores de c são utilizados para aumentar os limites das fronteiras, os valores de c' servem para fazer o oposto. No início, $d' = 1/3$, que é o valor mais alto abaixo da condição de busca aleatória, ou seja, $d' = 1/2$, normalmente adotado na primeira geração de um algoritmo como uma tentativa de evitar uma exploração tendenciosa do espaço de busca.

Dependendo dos valores de c em cada geração ele sofrerá um aumento de uma unidade para cada *qubit* que esteja de acordo com qualquer uma das condições estabelecidas em (23). De modo semelhante, valores de c' em cada geração sofrerão um aumento de uma unidade para cada *qubit* que esteja de acordo com qualquer uma das condições estabelecidas em (24).

4.2.3 Mecanismo autoadaptável para definição do número de agentes (\mathcal{A})

O número de agentes (\mathcal{A}) é um dos mais importantes parâmetros de um algoritmo. Ele descreve uma amostra dos 2^n indivíduos binários que podem ser criados pelo vetor de distribuição de probabilidades onde n é o número de *bits* compondo a *string*. Geralmente, \mathcal{A} tem um valor fixo, que é definido antes de o processo de busca ser iniciado. Entretanto, não é trivial encontrar um valor ótimo para este parâmetro.

Além disso, seria interessante fazer o número de agentes variar enquanto a exploração do espaço de busca está sendo realizada, tendo em vista que, se uma região com ótimos locais for encontrada, um número crescente de agentes irá investigá-la de modo mais eficiente e rápido que um conjunto fixo faria. Por outro lado, não há necessidade de manter o mesmo número de agentes utilizados no início da busca quando se observa que as soluções encontradas já estão estagnadas. O terceiro estágio trata desta questão. O número de agentes do SQPBIL é definido em função do valor de componentes corretas (c) descrito na seção anterior:

$$\mathcal{A} = \left(1 + \frac{1}{c}\right)^c \cdot \mathcal{P}_0 \left(\frac{\mathcal{P}_0}{7}\right)^{-\left(\frac{c}{n}\right)} \quad (27)$$

$$\mathcal{P}_0 = 7 \cdot \left(1 + \frac{1}{n}\right)^n \quad (28)$$

Onde \mathcal{P}_0 é a população inicial correspondente a $c = 0$. Quando o valor médio de c sofre uma mudança brusca no seu comportamento, isto é, ele nem cresce nem diminui de maneira regular, \mathcal{P}_0 sofre um incremento de uma unidade. Isto define uma flutuação em c . Caso contrário, o algoritmo é reiniciado mantendo o valor de \mathcal{P}_0 armazenado até então, que é o resultado da aprendizagem adquirida anteriormente. Isto é feito visando aumentar o poder de busca do algoritmo. O primeiro fator do produto que define o número de indivíduos na expressão (27) varia no intervalo $[1, e]$ onde $e = 2,3$. Isto significa que para valores de c próximos de 0, o valor de \mathcal{A} tenderá a \mathcal{P}_0 .

Após um grande número de gerações o valor acumulado de \mathcal{P}_0 pode influenciar negativamente a busca por melhores soluções forçando o algoritmo a criar um número muito grande de cópias de um mesmo indivíduo. Por isso, o algoritmo é capaz de definir quando o valor de \mathcal{P}_0 deverá ser reiniciado.

Embora as equações (27) e (28) sejam idênticas às utilizadas por CALDAS (2006) no FPBIL, foram necessários vários testes para comprovar sua adequação ao SQPBIL. Para tanto, o algoritmo foi utilizado na otimização dos mesmos problemas analisados com o QPBIL em (SILVA e SCHIRRU, 2010a) e (SILVA e SCHIRRU, 2011). Isso foi feito com o objetivo não apenas de avaliar o desempenho do modelo proposto, mas também para comparar os resultados do QPBIL e do SQPBIL também em funções contínuas e exemplos de TSP.

A análise de dados para estes testes se mostrou inconclusiva, pois não foram observados resultados consistentes. Em outras palavras, o SQPBIL ora otimizava o problema de maneira satisfatória, ora encontrava resultados considerados descartáveis, impossibilitando uma análise estatística coerente. Por esse motivo não são apresentados dados relacionados a estes testes. Apesar disso, esta foi uma etapa importante para a definição da forma como o valor de \mathcal{P}_0 é incrementado, bem como para determinar quando ele deve ser reinicializado. A figura 4.1 mostra o pseudocódigo do SQPBIL.

```

-----Definição das Funções  $C_d$  e  $\mathcal{D}_s$ -----
 $C_d = (1/d) - 1$ 
 $\mathcal{D}_s = 1/(c - 1)$ 
-----Fase de inicialização-----
 $\mathcal{P}_0 = 7 * (1+1/n)^n$ 
 $d = 1/3$ 
-----Construção do vetor  $\mathcal{P}$  como no QPBIL-----
if (flutuação em  $c$ )
     $\mathcal{P}_0 = \mathcal{P}_0 + 1$ 
    if  $(\Delta\langle c \rangle < 1\%)$ 
         $d = 1/3$ 
        cada componente de  $\mathcal{P} = \frac{\sqrt{2}}{2}$ 
    end
end
 $c = C_d$ 
 $\mathcal{A} = \text{fix}(\mathcal{P}_0(1 + 1/c)^c * (\mathcal{P}_0/7)^{(-c/n)})$ 
-----Construção de Indivíduos-----
for  $i = 1 : \mathcal{A}$ 
    for  $j = 1 : \text{número de bits}$ 
        if  $\text{rand} > \beta_j^2$ 
             $I_{ij} = 0$ 
        else
             $I_{ij} = 1$ 
        end
    end
end
Avaliação da fitness
-----Atualização do Vetor  $\mathcal{P}$ -----

$$\mathcal{P}(j) = \frac{\sum_{i=1}^{\mathcal{A}} \mathcal{F}(i) I(i, k_j) \cdot \pi}{\sum_{i=1}^{\mathcal{A}} \mathcal{F}(i)}$$

-----Mutação em  $\mathcal{P}$ -----
 $c = \text{contar casos} \quad (\text{If } I_{\text{best}} = 0 \text{ and } \beta_j^2 < d) \text{ or } (\text{If } I_{\text{best}} = 1 \text{ and } \beta_j^2 > 1 - d)$ 
 $c' = \text{contar casos} \quad (\text{If } I_{\text{best}} = 0 \text{ and } d < \beta_j^2 < \mathcal{D}_s) \text{ or } (\text{If } I_{\text{best}} = 1 \text{ and } \mathcal{D}_s < \beta_j^2 < 1 - d)$ 
if  $\beta_j^2 < d$ 
     $\beta_j^2 = d$ 
end
if  $\beta_j^2 > 1 - d$ 
     $\beta_j^2 = 1 - d$ 
end
-----Retorna para a fase de construção o vetor  $\mathcal{P}$ -----

```

Figura 4.1: Pseudocódigo do SQPBIL.

Gerenciamento do combustível nuclear em uma usina PWR

O gerenciamento do combustível é o ramo da Engenharia Nuclear responsável pela resolução do problema da otimização da recarga. Este capítulo apresenta os conceitos básicos associados a ele tendo como referência o reator tipo PWR da central nuclear Angra 1. Além disso são abordados:

- a descrição do procedimento de otimização da Recarga Nuclear, definindo o que é, quando se torna necessário e quais os parâmetros considerados na sua execução;
- a modelagem utilizada para descrever o núcleo, que torna possível a interpretação do mesmo pela ferramenta de otimização desenvolvida neste trabalho de forma correta;
- a definição de uma função *fitness* através da qual os indivíduos gerados pelo método artificial de otimização serão analisados.

5.1 Procedimento da recarga do núcleo

Uma central nuclear é projetada para produzir energia à plena potência, também conhecida como potência nominal, pelo maior tempo possível. Além disso, é preciso que certos limites operacionais de segurança sejam respeitados para que a integridade física da central não seja comprometida. Quando a queima do combustível nuclear não é capaz de atender a demanda energética da usina, se torna necessário desligá-la para que o mesmo possa ser substituído. Nesse momento, o ciclo de operação atual é terminado.

Isso gera o chamado lucro cessante. Considerando o caso da central Angra 1, cada dia de operação corresponde a algumas centenas de milhares de dólares, sendo este o valor que está sendo perdido no mesmo período em que a usina estiver inoperante. Por esse motivo, a otimização do procedimento da recarga nuclear é de grande importância para as divisas do país. A queima do combustível nuclear depende da posição que cada EC ocupa no núcleo do reator. Isto permite que parte do inventário possa ser reutilizada no ciclo de operação que será iniciado quando a usina for reativada.

Portanto, o POR consiste, de um modo geral, em buscar uma configuração de distribuição entre ECs novos e reaproveitados que permita que a usina produza energia à plena potência pelo maior tempo possível até que outra recarga se torne necessária, cumprindo os critérios de segurança. De acordo com a situação ideal simulada neste estudo, onde se considera a otimização da recarga de 1 ciclo independente (a saber o 7^o ciclo de Angra 1), o que se busca é um PR ótimo ao qual está associado o maior ciclo de operação que esteja de acordo com os limites de segurança da central.

Na prática, o procedimento da recarga é projetado considerando vários ciclos subsequentes. Isso quer dizer que, à exceção da situação ideal analisada, não basta conhecer um PR que aumente o tamanho de apenas um dos múltiplos ciclos, já que os mesmos são interdependentes. Dito de outra maneira, existem outros fatores a serem considerados em uma situação real, além do PR ótimo descrito anteriormente.

Na maioria dos casos, fatores que não são “padrões de recarga” são determinados antes que a otimização seja feita. É nesse caso que o problema se reduz a encontrar “o” padrão ótimo (ERDOGAN e GEÇKINLI, 2003). Um PR é construído usando o inventário de ECs disponível. Em seguida, calcula-se a distribuição de potência para uma dada configuração de núcleo do reator. Este procedimento é repetido até que a distribuição de potência esteja de acordo com as restrições. O PR ótimo deve permitir um tamanho de ciclo razoavelmente grande, sem comprometer a integridade dos componentes do núcleo.

O gerenciamento do combustível nuclear inclui limitações relacionadas à segurança e também restrições geométricas enquanto o tamanho do ciclo de operação da central é dado externamente pela demanda de potência planejada. Alguns fatores relacionados à segurança são:

- O fator de canal quente;
- O coeficiente de temperatura do moderador;
- Máxima queima permitida do EC.

A redução dos custos relacionados à produção de energia é planejada em relação ao tamanho do ciclo de utilização do combustível nuclear. Uma solução ótima é aquela que se baseia nos seguintes parâmetros:

- Enriquecimento do combustível fresco;
- Número de ECs frescos inseridos no núcleo;
- Padrões de Recarga;
- Absorvedores presentes nos ECs.

5.2 Modelagem do problema

A dificuldade associada à resolução do POR cresce de modo exponencial em função do número de ECs considerados no procedimento. Por isso, ele pertence a classe de problemas de complexidade não-polinomial. O núcleo da central Angra 1 é composto por 121 ECs. Devido ao fato de apenas 1/8 do núcleo ser considerado para efeitos de cálculos, o espaço de busca do problema, que inicialmente seria composto por mais de 10^{200} padrões de recarga, é reduzido para “apenas” 10^{25} . A simetria de 1/8 é utilizada nesta Tese com dois objetivos básicos: primeiro, para simplificar cálculos e, segundo, para tornar possível a comparação dos resultados do SQPBIL com os de outras ferramentas. A figura 5.1 mostra o modelo que descreve o núcleo de Angra 1.

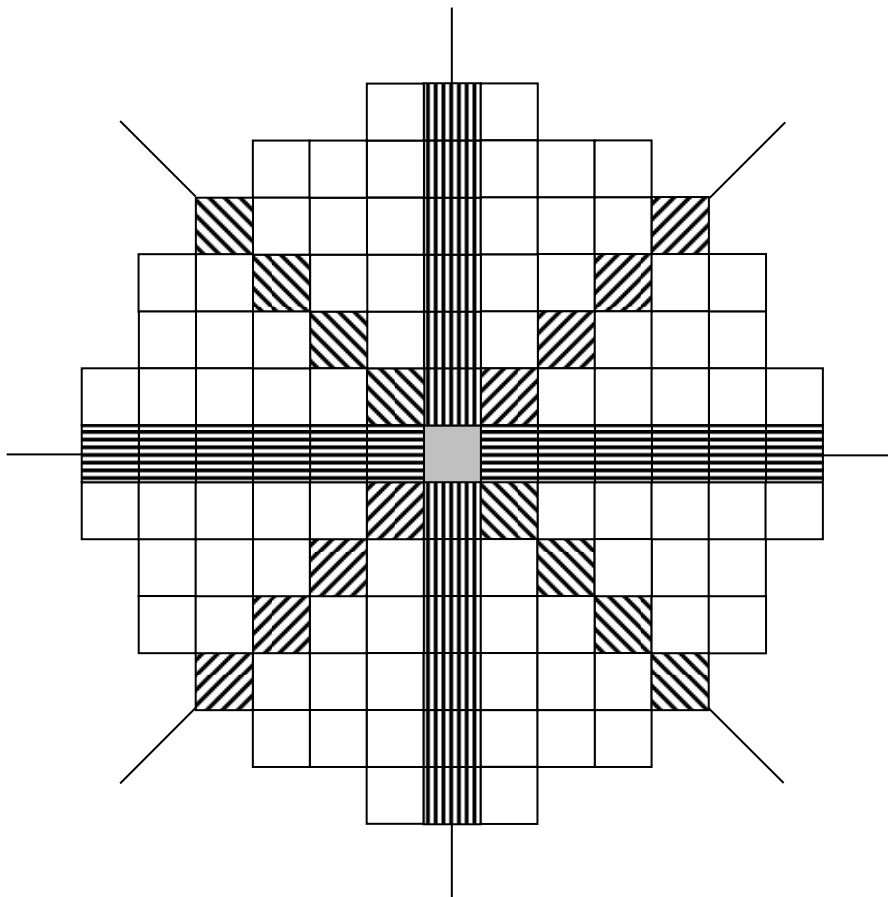


Figura 5.1: Modelo do núcleo de Angra 1.

Os elementos com hachuras estão localizados sobre eixos imaginários que dividem o núcleo em quatro regiões simétricas. Como cada um deles possui exatamente três outras posições equivalentes, totalizando quatro possíveis posicionamentos diferentes, são denominados elementos de quarteto.

Os elementos sem hachura possuem, cada um, outras sete posições equivalentes a sua própria, num total de oito possíveis posicionamentos diferentes. Por essa razão são denominados elementos de octeto. Devido à existência de quatro regiões exatamente simétricas, é possível estimar o comportamento do núcleo por inteiro analisando apenas uma delas. Isso permite a obtenção de resultados que satisfazem aos objetivos do POR com um esforço matemático comparativamente menor.

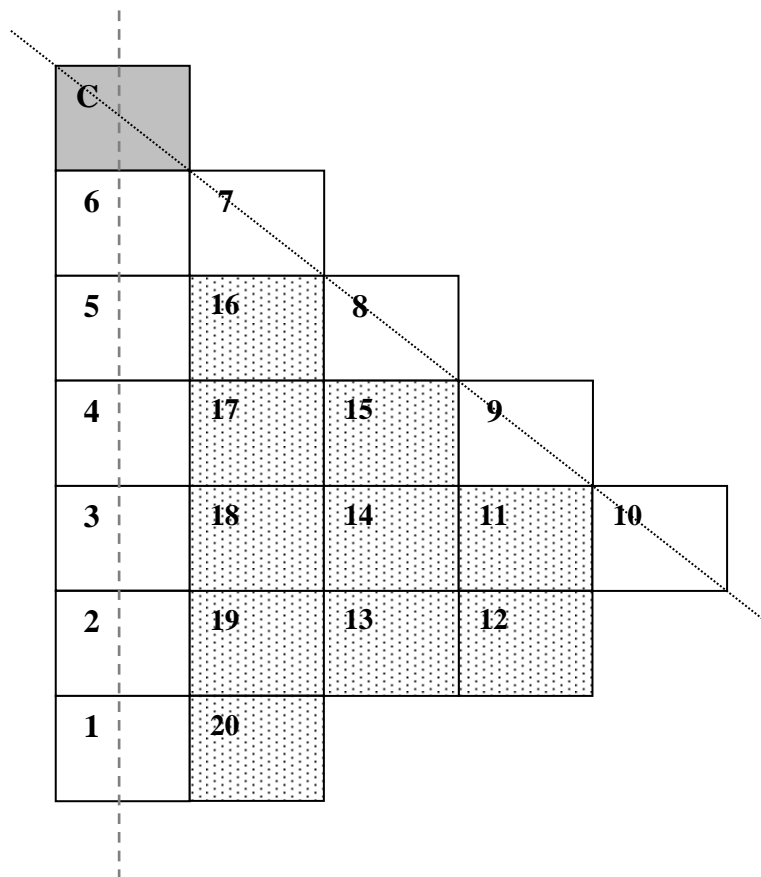


Figura 5.2: Região de simetria de octeto.

Como observado na figura 5.2, em 1/8 de núcleo existem 21 ECs, sendo que o elemento central é considerado como um caso particular visto que não possui pares. Por essa razão ele não participará do processo de otimização analisado nesta Tese, restando 20 elementos a serem considerados: 10 de quarteto e 10 de octeto.

5.3 A Função *fitness*

Uma das formas mais comuns de medir o tamanho do ciclo de operação de uma central nuclear é através do parâmetro conhecido como número de Dias Efetivos à Plena Potência (DEPP). Ele representa o número de dias que a usina permanece em atividade em sua potência nominal até que o procedimento da recarga seja realizado.

O objetivo da função *fitness* utilizada nesta Tese é maximizar o número de DEPPs com base na concentração crítica de boro (B_c) no moderador, em um processo orientado pelo valor do fator de pico de potência radial (F_{XY}) do núcleo. Enquanto 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 (MACHADO, 1998), B_c introduz uma reatividade negativa que compensa a reatividade em excesso do combustível no início do ciclo.

Na central nuclear Angra 1, o valor de F_{XY} deve ser menor ou igual a 1.435 para que o limite das Especificações Técnicas não seja violado (CHAPOT, 2000). Nesta Tese, a otimização da recarga nuclear do 7^o ciclo de operação da central Angra 1 é simulada através de um processo automatizado que combina o SQPBIL ao código nodal de física de reatores conhecido como RECNOB.

O RECNOB é um simulador de PRs desenvolvido exclusivamente para fins acadêmicos baseado no método FEM (MONTAGNINI et al, 1994) em substituição ao código ANC, cujo uso é restrito aos operadores da usina. No entanto, devido a questões relacionadas ao tempo de processamento associado a cada configuração de núcleo simulada, não foi desenvolvido o módulo responsável pelo cálculo de F_{XY} (vide anexo 4).

Uma vez que o RECNOd não é capaz de calcular o valor de F_{XY} este parâmetro foi substituído pela máxima potência média relativa (P_m) produzida no núcleo. Assim sendo, o valor de P_m foi determinado empiricamente, através da comparação dos resultados obtidos pelo RECNOd e pelo ANC na 2ª etapa de queima do ciclo 7 de Angra 1. Dois parâmetros foram considerados nesta análise:

- O valor P_m em cada EC;
- O fator de multiplicação efetivo (k_{eff})

O erro percentual entre os valores obtidos pelo ANC e pelo RECNOd referentes ao k_{eff} foi nulo. Com relação a valores de P_m , houve discrepância em todos os ECs analisados. No EC onde foi observado o maior erro percentual (6,3%) foi verificado o maior valor de P_m (1,395). Esse foi então adotado como sendo o maior valor permitido para este parâmetro que estaria de acordo com as normas técnicas de segurança com base nos critérios adotados no experimento.

Por esse motivo o valor limite de P_m passou a ser adotado como sendo equivalente ao valor limite de F_{XY} (1,435), ou seja, dentro da situação particular analisada nesta Tese, enquanto o valor de P_m for menor que 1,395, os limites de segurança não estarão sendo violados.

Após encontrar um padrão de recarga que minimize a potência média relativa no núcleo, deve-se maximizar a concentração crítica de boro no moderador. Dentre as funções objetivo adequadas para a resolução deste problema, foi selecionada a seguinte:

$$Fitness = \begin{cases} P_m & \text{se } P_m > 1,395 \\ 1/B_c & \text{se } P_m \leq 1,395 \end{cases} \quad (29)$$

Onde P_m é a potência média relativa no núcleo e B_c é a concentração crítica de boro no moderador. Um padrão de recarga é considerado válido quando o valor de P_m a ele associado respeita as Especificações Técnicas de segurança, ou seja, quando o valor de P_m está abaixo de 1,395.

A partir da escolha da função *fitness* o engenheiro nuclear optará por utilizar um esquema de carregamento do tipo baixa – fuga ou *out – in*. Quando o objetivo é a maximização do comprimento do ciclo, faz-se uso de um esquema baixa-fuga. Neste caso, ECs novos são inseridos em posições mais internas, enquanto ECs reaproveitados ficam na periferia do núcleo.

Isto implica na ocorrência de elevados picos de potência no núcleo. Venenos queimáveis (VQs) são então utilizados para manter os valores deste parâmetro abaixo dos limites impostos pelas Especificações Técnicas. Por essa razão, maximizar o comprimento de um ciclo sem dispor de VQs é um desafio para o algoritmo.

A maximização do comprimento do ciclo implica na maximização da concentração de boro de final de ciclo. Contudo, a fim de minimizar os custos computacionais, é utilizada a concentração crítica de boro solúvel na 2ª etapa de queima (equilíbrio do xenônio) como parâmetro a ser maximizado. Esquemas baixa-fuga contribuem para o aumento da durabilidade do vaso do reator.

Segundo CHAPOT (2000), “como em um esquema *out - in* os elementos novos ocupam posições fixas na periferia do núcleo e sendo estes ECs idênticos (mesmo enriquecimento, mesmo fabricante), eles não são considerados”. De acordo com o caso particular em análise neste estudo, o objetivo é a maximização do comprimento do ciclo de operação. A exemplo do procedimento adotado em CHAPOT (2000) é utilizado um esquema de carregamento do núcleo do tipo baixa – fuga, mas sem dispor de venenos queimáveis.

SQPIL aplicado na otimização da recarga nuclear

A otimização da recarga nuclear é um dos problemas de grande interesse na área de Engenharia Nuclear. Dada a dificuldade inerente à sua resolução, reforçada pelas restrições de segurança e geometria do núcleo, este problema vem sendo analisado por vários pesquisadores ao longo dos anos.

Esta tarefa, que inicialmente era executada com base na intuição e experiência de especialistas pelo método de tentativa e erro, passou a ser automatizada através de técnicas de inteligência artificial como os algoritmos evolucionários.

Devido ao seu alto grau de complexidade, o problema da otimização da recarga nuclear será utilizado como teste definitivo para determinar a capacidade do SQPIL como ferramenta de otimização.

Neste capítulo é mostrado como o SQPIL pode ser aplicado a este problema. Além disso, seus resultados são comparados aos obtidos a partir de outras técnicas de otimização publicados em trabalhos anteriores.

6.1 SQPBIL aplicado na otimização do ciclo 7 de Angra 1.

Conforme o caso particular de estudo analisado nesta Tese, o problema da otimização da recarga consiste em encontrar o padrão de embaralhamento de ECs que maximize o número de DEPPs e, conseqüentemente, o comprimento do ciclo de operação. Para definir este padrão ótimo, são considerados dois parâmetros: a concentração crítica de boro no moderador (B_c), e a potência média relativa (P_m).

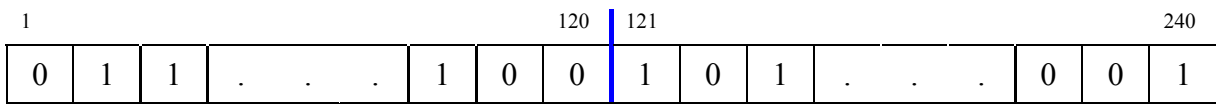
Visto que P_m é utilizado apenas como um parâmetro de controle, a análise do problema é feita em função da maximização de B_c . Assim, o RECNOD simula uma configuração de núcleo que será em seguida analisada pelo SQPBIL. A maximização de B_c está sujeita a duas restrições básicas: a primeira delas é o fator de pico de potência radial (F_{XY}) que, de acordo com as especificações técnicas de segurança, não deve ultrapassar o valor de 1.435. A outra diz respeito à geometria do núcleo. Na simetria de 1/8, um EC de quarteto não pode ser substituído por um EC de octeto em virtude da diferença no número de posições simétricas entre eles.

A interface de trabalho deve levar em consideração que o SQPBIL trabalha com seqüências de *strings* binárias para descrever possíveis soluções no espaço de busca, enquanto no código RECNOD os ECs são identificados por números inteiros que variam de 1-10 para posições de quarteto e de 11-20 para posições de octeto. É necessário, portanto, converter a *string* gerada pelo SQPBIL em uma seqüência de inteiros que irá representar as posições dos ECs no núcleo.

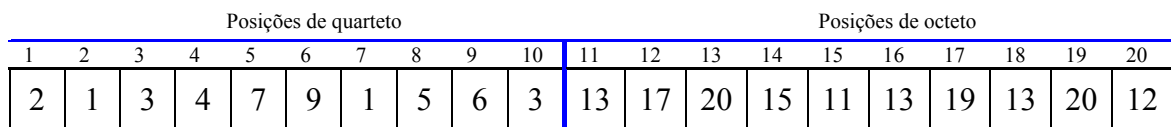
Entretanto, esta prática torna possível a existência de seqüências onde haja repetição de valores. Isto representaria uma impossibilidade física já que um mesmo EC estaria ocupando mais de uma posição na configuração de núcleo em questão, invalidando o resultado. Para contornar este problema, evitando que situações semelhantes ocorram, é utilizado o modelo de *Random Keys* (BEAN,1994).

6.2 Metodologia

Inicialmente, O SQPBIL gera uma *string* binária \mathcal{B} com 240 *bits* composta por 20 fragmentos menores de 12 *bits*. Cada um desses fragmentos é o equivalente binário de um dos ECs sendo distribuídos. Em seguida, ocorre a conversão para a base decimal, onde o vetor I é criado. Ele é composto por 20 valores inteiros variando no intervalo $[0, 4095]$ correspondendo aos 20 ECs compondo o PR.



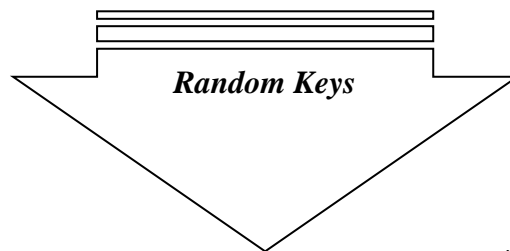
String binária \mathcal{B} gerada pelo SQPBIL



Índices dos ECs de quarteto

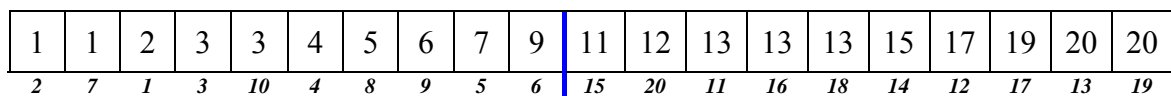
Índices dos ECs de octeto

Padrão Inválido (*String I*)



Índices de quarteto ordenados

Índices de octeto ordenados



Padrão Válido (*String I'* \equiv Índices reordenados da *String I*)

Figura 6.1: Modelo de *Random Keys*.

Para completar o processo, todos os valores de I são ordenados do menor para o maior. Os valores ordenados formam o vetor I' . O índice que acompanha cada posição de I' representa a posição que o EC ocupava na *string* I . Desta forma, o resultado gerado sempre irá corresponder a um PR válido independentemente de haver ou não repetição nos valores de I . Um exemplo de todo o processo é mostrado na figura 6.1.

6.3 Resultados

O código SQPBIL e sua interface de comunicação com o código de física de reatores foram implementados em MatLab 6.5. O RECNOD fornece a informação sobre os dados nucleares para o algoritmo que age minimizando os valores de P_m ao longo das gerações até que um PR válido seja obtido. Para ser considerado como válido, o PR deverá estar associado a um valor de P_m inferior a 1,395. Isto irá garantir que os critérios de segurança estão sendo respeitados.

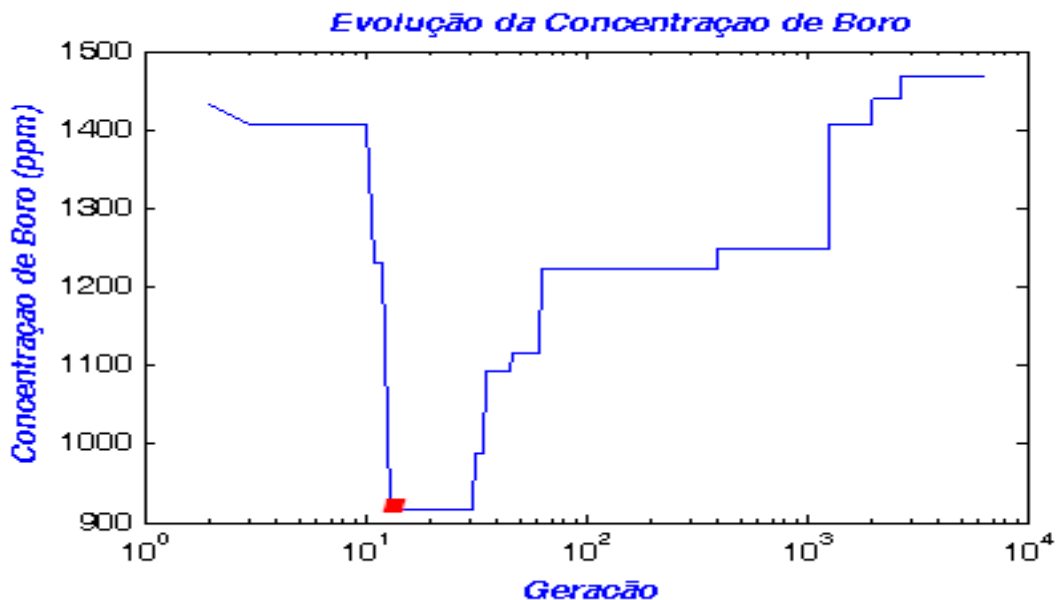


Figura 6.2: Evolução da concentração de Boro.

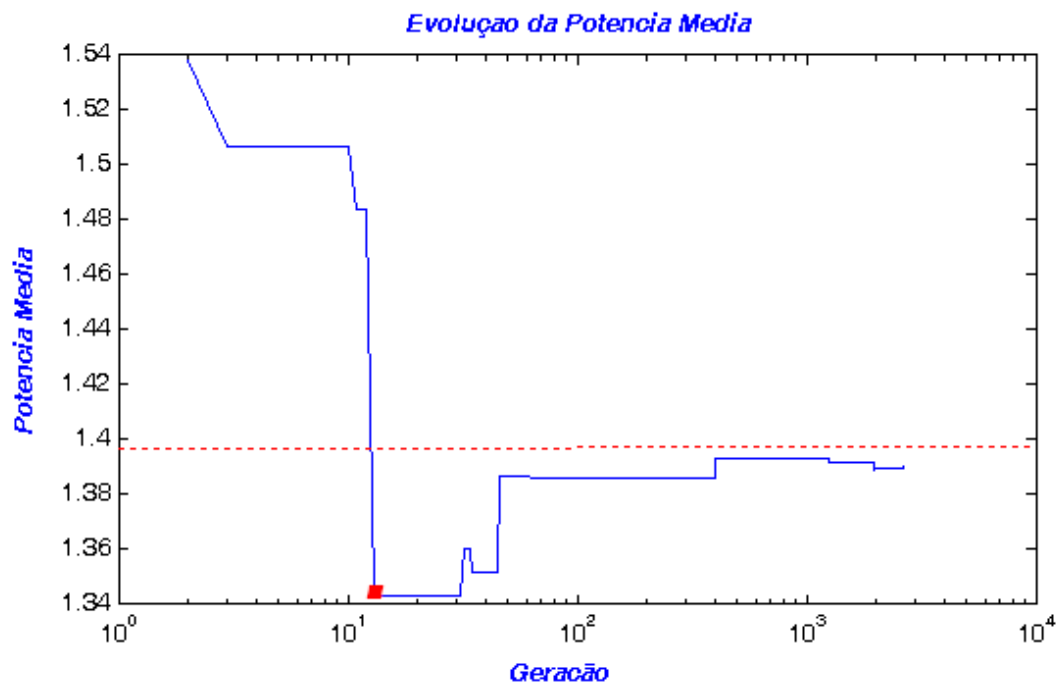


Figura 6.3: Evolução da potência média relativa.

Em seguida, o SQPBIL irá maximizar o valor de Bc . Diante do tempo disponível para a realização desta Tese, foram realizados 10 experimentos para avaliar o desempenho do algoritmo. As figuras 6.2 e 6.3 apresentam, respectivamente, as curvas que descrevem a evolução de Bc e P_m ao longo das gerações. O ponto em destaque marca o momento em que o primeiro PR válido é encontrado. A linha pontilhada na figura 6.3 representa o valor limite de P_m . Na figura 6.4 é mostrada a evolução da *fitness*. Na tabela 6.1 é mostrado o resultado obtido em cada um dos experimentos realizados.

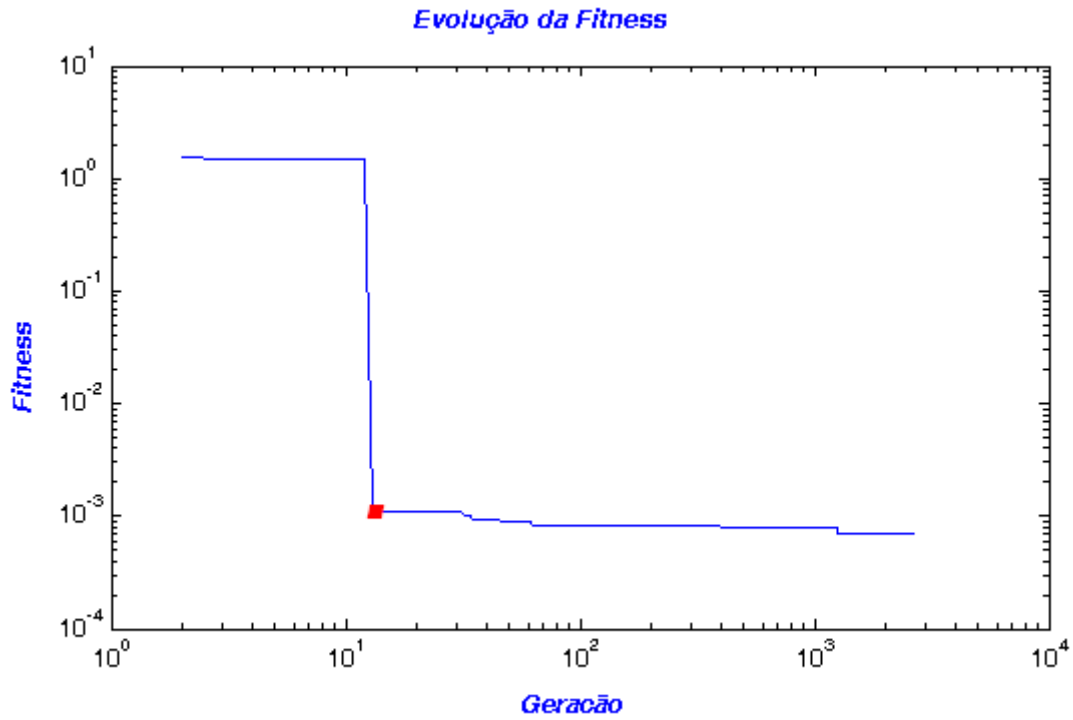


Figura 6.4: Evolução da *Fitness*.

Experimentos	B_c máximo (ppm)			<i>Fitness</i> (* 10^{-6})		
	QACO_Alpha	QPBIL	SQPIL	QACO_Alpha	QPBIL	SQPIL
1	1415	1413	1469	707	708	680
2	1414	1409	1408	707	710	710
3	1410	1408	1405	709	710	711
4	1390	1403	1405	719	713	711
5	1384	1402	1403	723	713	712
6	1382	1402	1403	724	713	712
7	1364	1395	1362	733	717	734
8	1346	1392	1330	743	718	751
9	1325	1331	1327	755	751	753
10	1305	1275	1315	766	784	760
Média	1330	1383	1383	753	723,7	723,4

Tabela 6.1: Lista de experimentos

6.4 Comparação com outros trabalhos

A tabela 6.2 mostra que o maior valor de B_c encontrado pelo SQPBIL é melhor que aquele obtido por todos os demais métodos analisados anteriormente. A diferença entre o valor ótimo do SQPBIL (1469 ppm) e o melhor valor conhecido até então (1428 ppm) é igual a 41 ppm de boro.

Autor	B_c (ppm)	P_{rm}	Técnica	Heurística	Avaliações
-	955	1,345	Manual	-	-
CHAPOT et al (1999)	1026	1,390	GA	Não	4000
MACHADO (1999)	1242	1,361	PBIL – N	Não	6000
MACHADO e SCHIRRU (2002)	1297	1,384	Ant – Q	Sim	200
MACHADO (2005)	1305	1,349	PBIL – MO	Não	10000
LIMA et al (2008)	1424	1,386	RCCA	Sim	420000
CALDAS (2008)	1428	1,392	FPBIL	Não	430364
SILVA e SCHIRRU (2011b)	1415	1.389	QACO_Alpha	Não	99240
SILVA e SCHIRRU (2011a)	1413	1.384	QPBIL	Não	49680
Presente Trabalho	1469	1.391	SQPBIL	Não	622219

Tabela 6.2: Técnicas usadas para maximizar o comprimento do 7º ciclo de Angra 1

Isto corresponde a um ganho de 10 Dias Efetivos à Plena Potência (DEPPs) já que, de acordo com CHAPOT (2000), a cada DEPP são consumidas 4 ppm de boro aproximadamente apesar de o SQPBIL ter requerido um número maior de avaliações para obter tal resultado. Isto é, o tamanho do ciclo de operação foi maximizado de uma maneira como nunca antes havia sido. Como cada DEPP equivale a centenas de milhares de dólares em energia produzida, um ciclo otimizado desta maneira representaria uma economia de mais de seis milhões de dólares em energia produzida.

6.5 Sobre a evolução do número de agentes

A curva da figura 6.5 descreve a evolução do número de agentes associada ao melhor resultado obtido pelo SQPBIL ao longo do processo de otimização da recarga analisada nesta Tese. De acordo com o modelo proposto, a fronteira que limita os valores das componentes do vetor de probabilidades pode ser dilatada (ou comprimida) em função do aumento (ou redução) no número de componentes corretas encontrado.

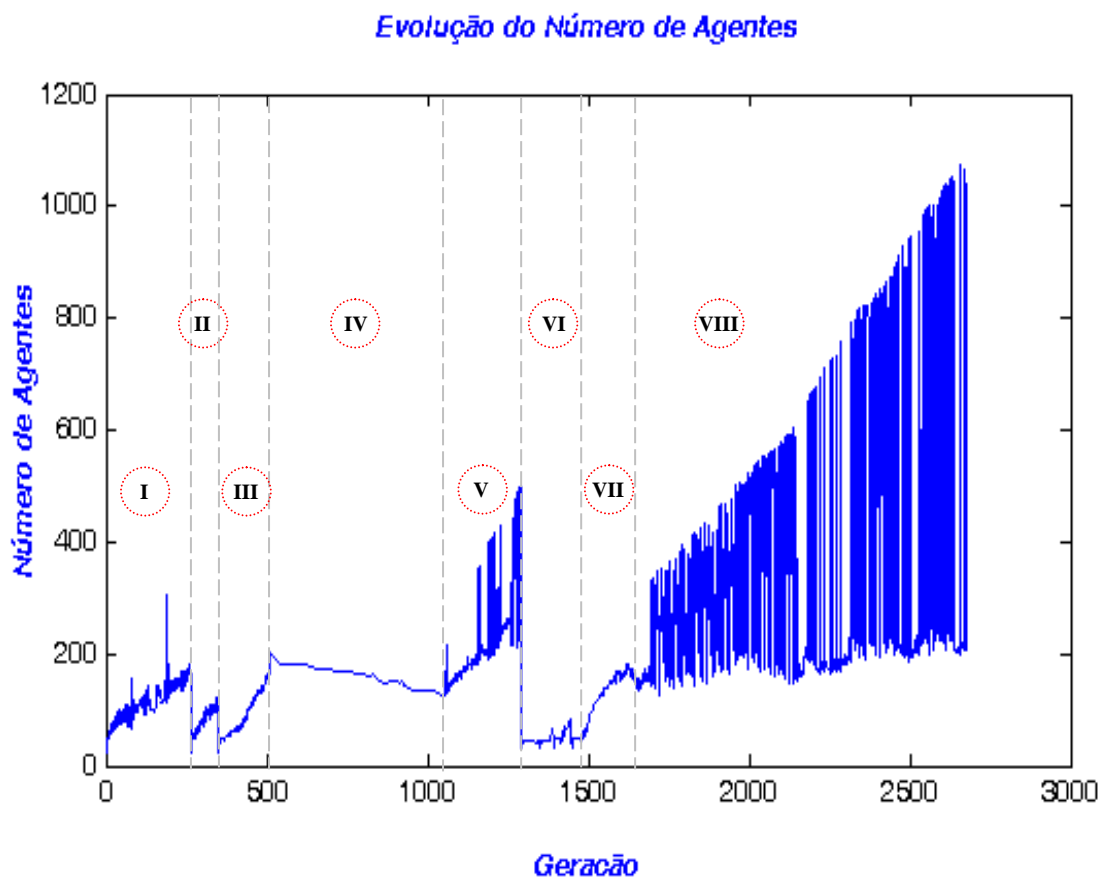


Figura 6.5: Evolução do número de agentes.

Como no início da exploração os valores de c ainda são baixos, o número de indivíduos descrito pela expressão (27), no capítulo 4, tende a \mathcal{P}_0 . À medida que os valores de c aumentam, o segundo fator passa a exercer maior influência no número de indivíduos gerados. Este comportamento fica evidenciado nas regiões destacadas na figura 6.5.

O modelo teórico prevê um padrão de crescimento gradativo no número de agentes à medida que o conhecimento acumulado pelo SQPBIL na exploração do espaço de busca aumenta. No intervalo $[0, 500]$, este padrão se repete três vezes, nas regiões **I**, **II** e **III**. Como o número de agentes ainda não é muito grande, existe uma maior facilidade para reinicializar o valor de c (e, por consequência, o valor da fronteira d) quando uma região não muito promissora é explorada. Esta prática é responsável por impedir que o algoritmo convirja prematuramente.

Após o término da região **II**, o número de agentes aumenta até atingir um ponto onde ocorre uma leve estagnação, ou seja, a região **IV** no intervalo $(500,1000)$. Isso pode ser interpretado como um momento de indefinição, no qual o algoritmo ainda não foi capaz de “decidir” se já esgotou ou não a análise de dados daquela região do espaço de busca, resultando num padrão quase-estático em média.

Na região **V** o padrão da curva é retomado, porém partindo de valores mais altos de \mathcal{P}_0 . A região **VI** reflete um comportamento influenciado pela manutenção do valor resultante do término da região **V**. Trata-se de um comportamento anômalo, onde é possível verificar dois pontos de máximo. Esta discrepância tem influência na região **VII**, onde o padrão esperado da evolução de agentes é retomado, partindo de um \mathcal{P}_0 mais baixo após ser reinicializado.

A região **VIII** preserva o padrão previsto pelo modelo teórico já observado em **I**, **II** e **III**. Porém, há uma diferença em relação ao módulo do número de agentes utilizado. Neste ponto é importante mencionar que o valor ótimo obtido pelo SQPBIL (1469 ppm) foi encontrado próximo da geração 2300 que, portanto, pertence à região **VIII**.

Com um número de gerações maior que 2000, o algoritmo, baseado no conhecimento acumulado nas gerações anteriores, investe um número cada vez maior de agentes para explorar esta região que é resultado do aumento gradativo do valor de \mathcal{P}_0 acumulado em conjunto com o número crescente de componentes corretas sendo encontradas.

Ou seja, o padrão da curva é mantido, mas com um número de agentes que é maior em módulo do que aquele utilizado em **I**, **II** e **III**. Isso quer dizer que o algoritmo, com base nos seus critérios internos, decidiu que a região **VIII** é promissora em soluções de alta qualidade e com isso parte para uma investigação mais detalhada da mesma. Essa é a razão de o número de avaliações utilizadas pelo SQPBIL ter excedido aquele descrito por CALDAS (2006) para FPBIL, por exemplo.

Portanto, o maior número de avaliações utilizadas pelo SQPBIL para a obtenção de seu valor ótimo não deve ser visto como um ponto negativo. Ao contrário, isto confirma que a demanda intrínseca referente à forma como o algoritmo se propõe a analisar o espaço de busca pode contribuir para a obtenção de uma solução que provavelmente não seria encontrada sem que uma investigação tão minuciosa fosse realizada. Isso é confirmado pelo valor ótimo de B_c encontrado pelo SQPBIL (1469 ppm), que supera o de todas as outras ferramentas descritas na tabela 6.2.

Conclusão

Este capítulo trata da conclusão desta Tese. São feitas as considerações finais em relação a toda informação obtida acerca da evolução do algoritmo QPBIL desde a sua implementação até o seu uso em conjunto com as técnicas autoajustáveis de obtenção de parâmetros que deram origem ao SQPBIL. Além disso, são apresentados os possíveis caminhos que o produto da pesquisa poderá seguir a partir deste ponto.

7.1 Conclusão

Esta Tese propõe uma nova abordagem usando uma técnica de obtenção de parâmetros autoadaptável associada ao algoritmo QPBIL. Isto dá origem a um novo modelo automatizado de busca estocástica denominado *Self-adaptive Quantum PBIL*. Nesta nova ferramenta os seguintes parâmetros são determinados automaticamente: taxa de aprendizado positivo, taxa de aprendizado negativo, taxa de mutação e tamanho da população.

As taxas de aprendizado negativo e positivo são definidas em função de um valor que representa a informação do nível de *fitness* de todos os indivíduos compondo a população a cada geração. Em um primeiro momento, a média dos melhores indivíduos guia a busca em direção a melhores soluções, orientada pelo valor da melhor *fitness* conhecida até então.

Em seguida, a média dos piores indivíduos guia a busca para longe de regiões pouco promissoras tendo como referência a pior *fitness* conhecida. Com isso, ainda que a totalidade da população tenha que ser analisada, nenhuma parte desta informação é ignorada, contribuindo para uma exploração mais consistente do espaço de busca.

A taxa de mutação é definida em função do número de componentes consideradas corretas ou não de acordo com critérios pré-estabelecidos. Estas componentes são utilizadas para controlar a fronteira que limita os valores de α^2 e β^2 dos *qubits*, minimizando efeitos de convergência prematura.

O tamanho da população também é função do número de componentes corretas. Ele aumenta (ou diminui) dependendo de uma maior necessidade (ou não) de exploração do espaço de busca. Isso contribui para um maior equilíbrio no processo de otimização.

Para avaliar a performance do método proposto e comparar seus resultados àqueles obtidos com o QPBIL e outros métodos semelhantes disponíveis na literatura, o SQPBIL foi aplicado na otimização do caso particular do 7^o ciclo de operação de Angra 1.

Diante dos resultados obtidos, a conclusão a que se chega é que o SQPBIL cumpriu o fim a que se destina no sentido de ser capaz de evitar a estagnação das componentes dos *qubits* e, conseqüentemente, de impedir que a convergência prematura seja observada no QPBIL. Isso é evidenciado por um tamanho de ciclo que equivale a um ganho de 56 ppm de boro, o que representa uma melhora no algoritmo.

Como a cada DEPP são consumidas aproximadamente 4 ppm de boro (CHAPOT, 2000), o ciclo otimizado pelo SQPBIL representa um ganho de 14 DEPPs. Considerando que a central Angra 1 produz em energia o equivalente a 600.000 dólares por DEPP, o uso do SQPBIL permitiu a obtenção de um resultado que representa o equivalente a 8.400.000 dólares em relação ao resultado obtido com o QPBIL.

Considerando o FPBIL, de quem o SQPBIL pode ser visto como o equivalente quântico, houve um ganho de 41 ppm de boro. Como já mencionado na seção 6.4, isto corresponde a mais de 10 DEPPs, resultando em um ganho de mais de 6.000.000 de dólares em energia produzida.

O maior número de avaliações requerido pelo SQPBIL para obter seu melhor resultado não deve ser interpretado como um ponto negativo para o algoritmo já que, como observado pela análise da figura 6.5, isso é reflexo da forma como ele se propõe a explorar o espaço de busca. Em outras palavras, essa se mostra uma característica inerente ao seu *modus operandi*, sem a qual provavelmente o SQPBIL não encontraria um resultado tão expressivo.

Isso deixa claro que o uso do modelo de autoadaptação de parâmetros em conjunto com conceitos quânticos pode ser mais eficiente que o uso destes métodos separados.

O SQPBIL também conseguiu ser mais eficiente que os Algoritmos Genéticos (CHAPOT, 1999), Ant-Q (MACHADO, 2002), PBIL – N (MACHADO, 1999) e PBIL – MO (MACHADO, 2005), na otimização dos valores de concentração B_c além do RCCA (LIMA et al., 2008) enquanto o QPBIL não obteve êxito nesta questão em particular.

Por tudo o que foi apresentado, pode ser afirmado que o SQPBIL teve uma performance satisfatória na otimização da recarga nuclear. Considerando que ele conseguiu superar cada um dos demais métodos submetidos à análise, o SQPBIL se qualificou como uma eficiente ferramenta de otimização na resolução deste problema.

7.2 Proposta de sequência de tese

Embora o SQPBIL tenha obtido uma performance relativamente aceitável na otimização do POR, é importante salientar que não se trata de um modelo de otimização definitivo, tendo em vista que o mesmo ainda apresenta problemas a serem solucionados. Um exemplo disto está no seu uso em funções *benchmark* como, por exemplo, as funções Esfera, Griewank e Rosembrock. Nestes exemplos algoritmo geralmente não apresenta um bom desempenho, dificilmente conseguindo encontrar o valor ótimo. Esse problema também foi observado nos exemplos de TSP analisados, onde os dados obtidos não permitiram a realização de uma análise estatística coerente.

Seria interessante um estudo investigativo no sentido de avaliar se o critério adotado para a definição de componentes corretas e incorretas pode ser aprimorado ou até mesmo propor uma forma alternativa de determinar como é feito o incremento nestes valores. Nada garante que soma uma unidade sempre que uma dessas componentes é identificada seja a melhor forma de proceder. Existe ainda a necessidade de encontrar um meio mais claro de determinar como deve ser feito o incremento do valor de \mathcal{P}_0 , que pode ser melhor elaborado, o que, possivelmente, resultaria em uma melhora significativa do método.

Para uma melhor avaliação do método seria interessante aplicá-lo em outros problemas de otimização e fazer um estudo mais aprofundado sobre qual o melhor procedimento para definir o que pode ser considerado como uma componente correta ou não. Outra sugestão seria a paralelização do método proposto através do uso de GPUs.

ANEXO 1 – Teoria dos múltiplos universos

Para fatorar um número n que é produto de dois fatores primos, define-se um número arbitrário de universos paralelos p (0,1,2,...) e seleciona-se randomicamente um inteiro x entre 0 e n . Em seguida, para cada universo, eleva-se x à potência do índice do universo. O resultado é então dividido por n sendo o resto desta operação armazenado em uma lista.

Para o próximo número da sequência, x é elevado à potência do último valor armazenado. Este novo resultado é dividido por n e o resto é armazenado. O processo continua em cada um dos universos como observado na tabela A.1, onde é apresentado um exemplo para $n = 33$, $x = 7$ ($0 < x < n$) e $p = 17$ (NARAYANAN, 1996). Ela fornece uma descrição detalhada do que acontece em cada um dos universos (u_0 a u_{16}).

Por exemplo, o universo u_2 é descrito pela sequência cujo valor inicial é 16. Eleva-se x ao índice do universo na base n , ou seja, 7^2 na base 33 cujo resto é 16.

$$\begin{array}{r|l} 49 & 33 \\ 16 & 1 \end{array} \quad (A1.1)$$

Para determinar o segundo valor da sequência que descreve o universo u_2 , devemos elevar x ao resto da operação anterior (7^{16}) na base 33, cujo valor é igual a 4. O terceiro valor é obtido ao elevar 7^4 e assim sucessivamente. A frequência vertical f_v é definida como sendo o intervalo no qual um determinado valor se repete entre os universos. No exemplo citado f_v é igual a 10 já que $u_0 = u_{10}$, $u_1 = u_{11}$, $u_2 = u_{12}$ e assim por diante.

Tabela A.1: Exemplo de 17 universos para identificação de fatores primos.

Adaptado de NARAYANAN (1996)

u ₀	1	7	28	31	7	28	31	7
u ₁	7	28	31	7	28	31	7	28
u ₂	16	4	25	10	1	7	28	31
u ₃	13	13	13	13	13	13	13	13
u ₄	25	10	1	7	28	31	7	28
u ₅	10	1	7	28	31	7	28	31
u ₆	4	25	10	1	7	28	31	7
u ₇	28	31	7	28	31	7	28	31
u ₈	31	7	28	31	7	28	31	7
u ₉	19	19	19	19	19	19	19	19
u ₁₀	1	7	28	31	7	28	31	7
u ₁₁	7	28	31	7	28	31	7	28
u ₁₂	16	4	25	10	1	7	28	31
u ₁₃	13	13	13	13	13	13	13	13
u ₁₄	25	10	1	7	28	31	7	28
u ₁₅	10	1	7	28	31	7	28	31
u ₁₆	4	25	10	1	7	28	31	7

ANEXO 2 – O sentido de colapsar

No capítulo 3, durante a explicação a respeito da superposição linear de estados quânticos nos quais um elétron (e por analogia um *qubit*) pode ser observado, é utilizado o termo “colapsar”. Este verbo pode gerar dúvidas quanto ao real significado físico do fenômeno, tendo em vista que, no dicionário, ele é definido como sendo sinônimo de “desmoronar”, “desintegrar”, entre outros.

De acordo com a teoria dos orbitais atômicos, orbital é a região do espaço onde é máxima a probabilidade de se encontrar um determinado elétron. Segundo este modelo, o elétron é uma partícula-onda que se desloca (ou vibra) no espaço, mas com maior probabilidade dentro de uma esfera (orbital) concêntrica ao núcleo. Devido à sua velocidade, o elétron fica como que “esparramado” dentro do orbital, assemelhando-se, então, a uma “nuvem eletrônica” (FELTRE, 1996) como observado na figura A2.1:

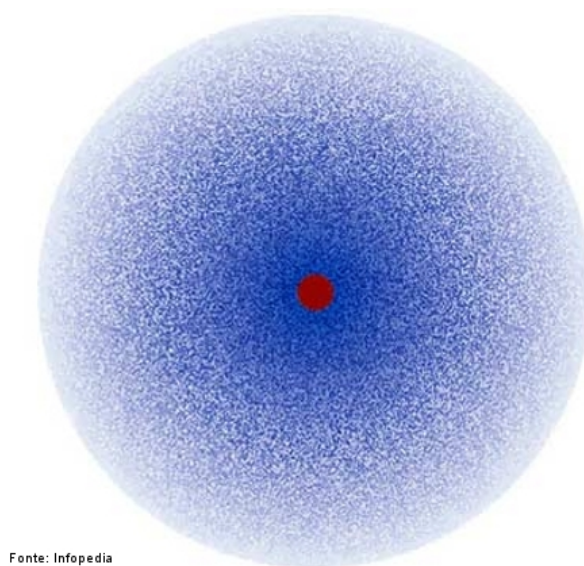


Figura A2.1 Orbital eletrônico do hidrogênio.

No caso particular do átomo de hidrogênio, a nuvem tem forma de uma casca esférica, pois há apenas 1 elétron. A casca é o análogo físico da notação matemática $|\Psi\rangle$ que descreve o somatório dos possíveis estados do *qubit*. Se o elétron do átomo de hidrogênio pudesse ser observado em uma posição de coordenadas (x,y,z) , por exemplo, esta passaria a descrever sua localização naquele momento. A casca esférica que até então descrevia a probabilidade de todas as possíveis localizações do elétron deixaria de fazer sentido.

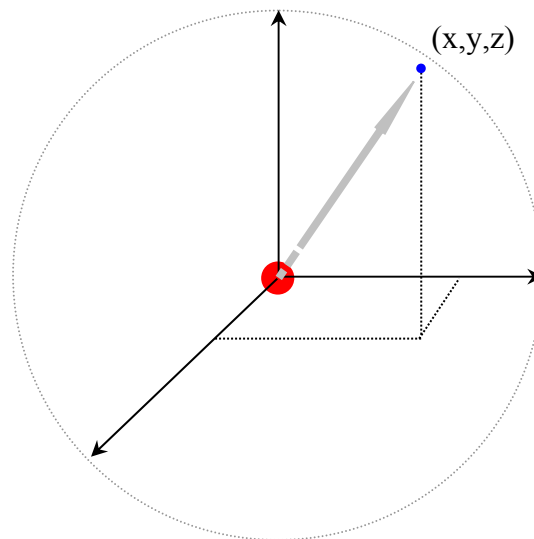


Figura A2.2 Orbital eletrônico do hidrogênio reduzido.

Ou seja, a figura A2.2 mostra que ela se reduz ao ponto (x,y,z) , sendo este talvez o melhor sentido para definir o verbo colapsar em se tratando deste fenômeno. Portanto, “colapsar” deve ser entendido aqui como “reduzir-se a” ou algum termo que denote esta ideia.

ANEXO 3 - Conceitos básicos associados à operação de uma central nuclear.

Centrais nucleares do tipo PWR, como Angra 1 e 2, produzem energia elétrica através da fissão do UO₂ levemente enriquecido (entre 2 e 5 %). O UO₂ é um material cerâmico com alto ponto de fusão ($2,8 * 10^3$ °C) geralmente utilizado na forma de pastilhas cilíndricas com 1 cm de diâmetro e 2 cm de aresta, ligeiramente côncavas nas extremidades.

As pastilhas são posicionadas em tubos de *zircaloy* com aproximadamente 4 metros de altura. Na temperatura de operação, estas pastilhas expandem axialmente, preenchendo o espaço entre elas. O revestimento de *zircaloy*, além de sustentar as pastilhas, evita que produtos de fissão contaminem o líquido refrigerante. Os tubos de *zircaloy* compõem estruturas maiores conhecidas como Elementos Combustíveis (ECs). Considerando o núcleo de Angra 1 em particular, cada EC é composto por arranjos de 16x16 varetas.

O núcleo do reator é o lugar onde ocorrem reações em cadeia controladas. Isso é possível graças à existência dos chamados nêutrons atrasados, que permitem que haja tempo suficiente para controlar o reator. Portanto, fazer com que o reator só seja crítico com a contribuição dos nêutrons atrasados é de fundamental importância.

Isso permite um tempo de vida média dos nêutrons da ordem de 0.08s, totalmente controlável por inserção de barras de controle, por exemplo. Quando um nêutron fissiona o combustível nuclear são gerados produtos de fissão (núcleos menores, nêutrons etc.) e uma grande quantidade de energia para o meio externo, ou seja, a água do moderador. Em um reator do tipo PWR a água age ao mesmo tempo como moderadora de nêutrons e refrigerante.

O calor gerado pela fissão aquece a água até temperaturas muito acima do ponto de ebulição. No entanto, não ocorre evaporação, já que o pressurizador age garantindo a manutenção do estado líquido. A água é então levada até o gerador de vapor completando o ciclo do chamado circuito primário.

No gerador de vapor, a água do circuito primário troca calor com a água fornecida pelo circuito secundário. Com isso, é gerada uma grande quantidade de vapor d'água a altas temperaturas. A pressão gerada pelo vapor irá movimentar as turbinas que alimentam geradores elétricos, produzindo assim a energia elétrica. O vapor resfriado é então levado ao condensador onde transfere o calor excedente para a água do ambiente, completando o ciclo do circuito secundário.

A geração de potência em um reator depende da manutenção de um balanço de nêutrons a uma dada taxa de fissões. Além disso, o procedimento de remoção da energia gerada no sistema nuclear é mais complexo que em sistemas convencionais como máquinas a vapor. Isso se deve entre outras causas, a características inerentes ao processo como, por exemplo, às altas densidades de potência, visto que uma dada geometria de núcleo deve ser mantida por muitos anos de operação e à necessidade de refrigeração contínua do reator mesmo após o seu desligamento devido ao decaimento dos produtos de fissão, o que gera uma fonte de calor que permanece ativa por muito tempo.

ANEXO 4 – Limitações do RECNOd

Um código de física de reatores de uso comercial para estudos de otimização de recargas de reatores deve conter os seguintes módulos:

- fluxo-potência-reatividade;
- queima de combustível;
- pesquisa de criticalidade com boro solúvel;
- x - modelos de realimentação
 - correção de densidade do moderador
 - realimentação *Doppler*
 - outros
- x - reconstrução da distribuição de densidade de potência pino - a - pino.

Por ser um código de uso estritamente acadêmico, no RECNOd estão ausentes os módulos marcados por um “x”. Por essa razão, se tornou necessário utilizar o valor da potência média relativa em lugar do fator de pico de potência radial, como explicado anteriormente.

O módulo de queima do RECNOd resolve equações que descrevem as mudanças isotópicas na composição do núcleo durante a operação do reator. Estas equações estão relacionadas à queima do combustível e ao crescimento de produtos de fissão em função da queima. Como se trata de um código simplificado, apenas produtos de fissão com alto valor de seção de choque macroscópica serão considerados de maneira explícita: ^{135}Xe e ^{149}Sm .

O RECNOd utiliza polinômios baseados em conjuntos de parâmetros do ANC para o 7^o ciclo de Angra 1, ou seja, todos os dados nucleares gerados pela biblioteca do ANC para este ciclo servem de referência para o RECNOd. Os cálculos neutrônicos foram executados em duas etapas de queima: tempo $t = 0$ e na situação de equilíbrio de ^{135}Xe .

Os cálculos da evolução do ^{135}Xe e do ^{149}Sm no tempo baseiam-se em um modelo simplificado segundo o qual estes nuclídeos são produzidos diretamente da fissão pelo decaimento β do ^{135}I e ^{149}Pm , respectivamente. O RECNOOD possui uma sub-rotina denominada QUEIMIN que:

- faz a distribuição dos ECs;
- armazena informações relativas aos parâmetros nucleares de cada um dos elementos combustíveis que ocupavam o núcleo no ciclo anterior àquele que será otimizado;
- transfere esses dados para as novas posições que esses elementos irão ocupar no ciclo em análise.

A figura A4.1 mostra um esquema representando a interface entre SQPBIL e RECNOOD.

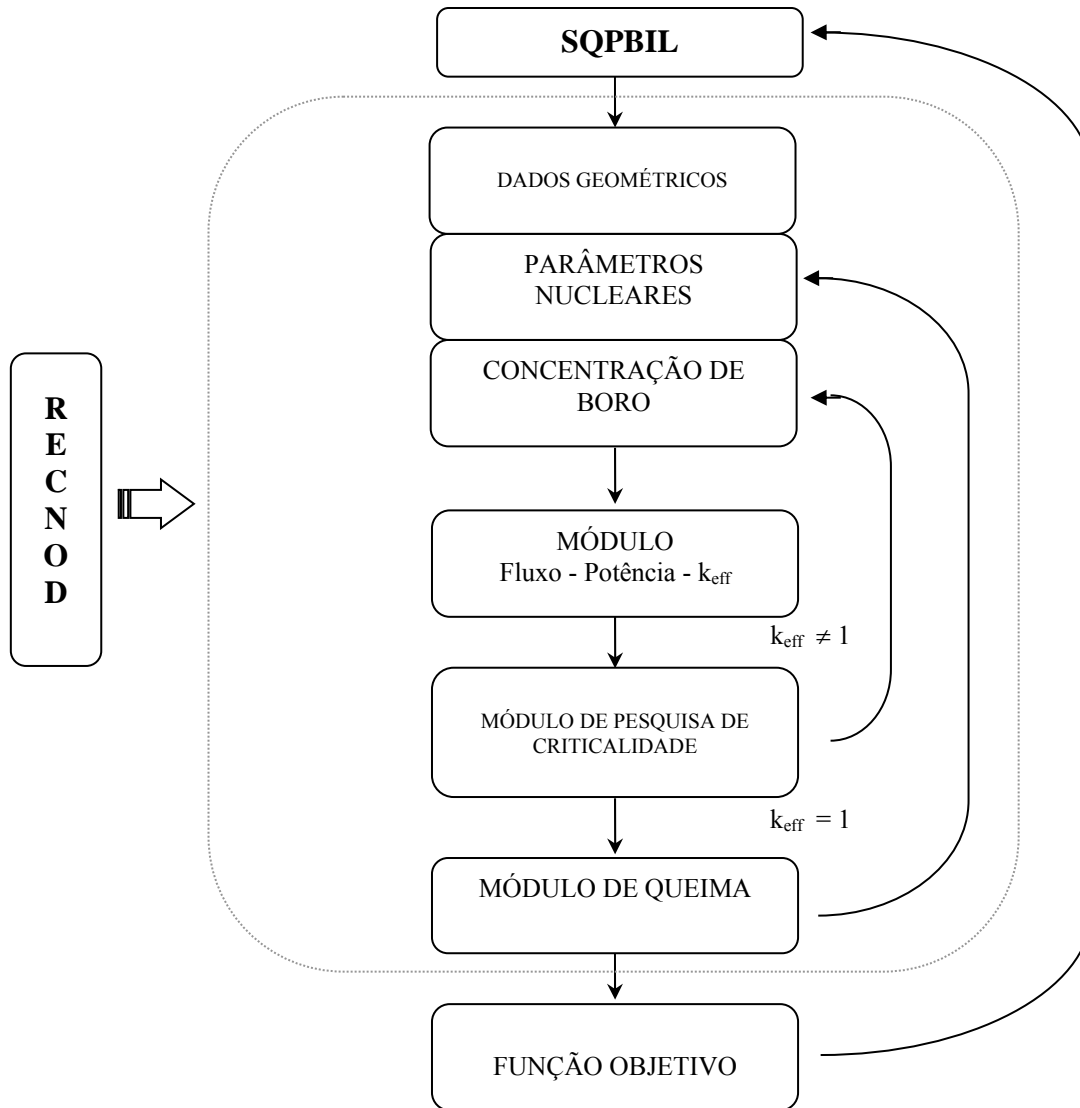


Figura A4.1: Esquema da interface entre SQPBIL/RECNO D

ANEXO 5 – Um exemplo de aplicação do modelo *Random Keys*

Algoritmos que utilizam a representação binária têm como característica a possibilidade de, no momento da conversão para a base decimal, gerar valores repetidos entre si. Dependendo do problema sendo analisado, isso pode se tornar um inconveniente. Alguns problemas de complexidade não-polinomial, como é o caso do POR, não permitem que a situação mencionada descreva uma solução válida. Por essa razão utiliza-se o modelo *Random Keys* (BEAN, 1994).

A figura A5.1 apresenta um exemplo meramente ilustrativo de um vetor binário, como são aqueles gerados pelo SQPBIL. Ele é composto por 24 *bits* e subdividido em 8 regiões menores, compostas por 3 *bits*. Cada uma delas representa uma variável no problema que estaria sendo analisado.

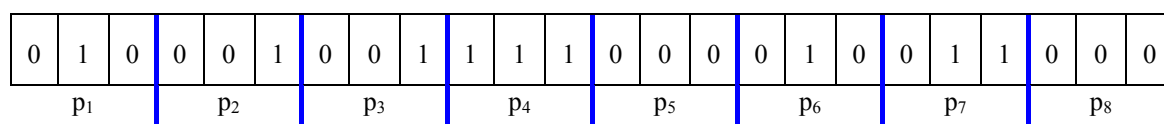


Figura A5.1: Exemplo de vetor binário gerado pelo algoritmo.

A conversão do vetor binário P desta figura para a base decimal dá origem a uma seqüência de oito valores inteiros mostrados na figura A5.2:

2	1	1	7	0	2	3	0
1	2	3	4	5	6	7	8

Figura A5.2 : *String* de valores inteiros gerados a partir do vetor binário P.

É possível observar que entre as componentes deste vetor existem valores repetidos. Para corrigir este problema o modelo *Random Keys*, age da seguinte maneira: primeiro, os valores que compõem o vetor de inteiros da figura A5.2 são reposicionados em ordem crescente, mantendo a informação da posição que ocupavam anteriormente como mostra a figura A5.3.

0	0	1	1	2	2	3	7
5	8	2	3	1	6	7	4

Figura A5.3 : *String* de valores inteiros reordenados

Como os índices que descrevem suas antigas posições serão sempre diferentes, não existirá risco de haver alguma solução inválida sendo gerada. Esta modelagem é eficiente, rápida e fácil de ser implementada já que uma *string* de números inteiros (ou reais se os valores estiverem normalizados), repetidos ou não, pode ser facilmente convertida em uma cadeia de números inteiros sem repetição. Desta forma, o padrão que será analisado pelo algoritmo será aquele mostrado pela figura A5.4 a seguir.

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

Figura A5.4: Padrão analisado.

REFERÊNCIAS BIBLIOGRÁFICAS

ABEN, 2009, *Revista Brasil Nuclear*, Ano 15, n.35 (Set), pp.14 – 17.

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

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.

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

CALDAS, G.H.F., 2008, “Parameterless evolutionary algorithm applied to the nuclear reload problem”. *Annals of Nuclear Energy*, v.35, pp.583-590.

CHAPOT, J.L.C., DA SILVA, F.C., SCHIRRU, R., 1999, “A new approach to the use of genetic algorithms to solve the pressurized water reactor’s fuel management optimization problem”, *Annals of Nuclear Energy* ,v.26, pp.641–655.

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

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

ERDOGAN, A., GEÇKINLI, M., 2003, “A PWR reload optimization code (XCore) using artificial neural networks and genetic algorithms”, *Annals of Nuclear Energy*, v.30., pp.35-53.

EVERETT, H., 1957, “Relative State”. Formulation of Quantum Mechanics. *Preview of Modern Physics*, 29, pp. 454-462.

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.

GAMBARDELLA, L., M., TAILLARD, E., AGAZZI, G., 1999, “A multiple ant colony system for vehicle routing problems with time windows”.

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

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.

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.

KOROSEC, P., SILK, J., 2008, “ The distributed multilevel ant-stigmergy algorithm used at the electric – motor design”, *Engineering Applications of Artificial Intelligence*, v.21, pp. 941-951.

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.

LIMA, A.M.M., SCHIRRU, R., SILVA, F.C., MEDEIROS, J.A.C.C., 2008, “A nuclear reactor core fuel reload optimization using artificial ant colony connective networks”, *Annals of Nuclear Energy*, v.35, pp.1606-1612.

MACHADO, M. D., 1998, *Aplicação de uma nova Metodologia Baseada na Propagação Estatística das Incertezas para Avaliar os Limites do DNBR*. Dissertação de MSc., COPPE/UFRJ, Rio de Janeiro, RJ.

MACHADO, M. D., 1999, *Um Novo Algoritmo Evolucionário com aprendizado LVQ para a Otimização de Problemas Combinatórios como a Recarga de Reatores Nucleares*. Dissertação de M.Sc., COPPE/UFRJ, Rio de Janeiro, 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., 2005. Algoritmo Evolucionário PBIL Multi-Objetivo Aplicado ao Problema da Recarga de Reatores Nucleares. Tese de D.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.

MERKLE, D., MIDDENDORF, M., 2001, “A new approach to solve permutation scheduling problems with ant colony optimization”, E.J.W. Boers et al. (Eds.) *Applications of Evolutionary Computing: Proceedings of EvoWorkshops 2001*, Springer, LNCS 2037, pp.484-492.

MONTAGNINI, B., SORAPERRA, P., TRENTAVIZI, C., SUMINI, M., ZARDINI, D. M., 1994, “A Well-Balanced Coarse-Mesh Flux Expansion Method”, *Annals of Nuclear Energy*, v.21, n.11, pp. 45 – 53.

NARAYANAN, A., MOORE, M., 1996, “Quantum – inspired Genetic Algorithms”, *Proceedings of IEEE. International Conference on Evolutionary Computation*, pp. 61-66.

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

RANDALL, M., LEWIS, A., 2002, “A Parallel Implementation of Ant Colony Optimization. *Journal of Parallel and distributed Computing*, 62(9): 1421 – 1432.

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., SCHIRRU, R., LIMA, A. M. M.,MEDEIROS J.A.C.C., 2009, “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.

SILVA, M., H., 2010, “*Otimização de Problemas da Engenharia Nuclear Utilizando Agentes Quânticos*”. Dissertação de MSc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.

SILVA, M. H., SCHIRRU, R., 2010a “*A Comparative Analysis between QPBIL and QACO*”, XIII Encontro de Modelagem Computacional. Instituto Politécnico (IPRJ), Novo Friburgo, Brasil.03-05 Nov.

SILVA, M. H., SCHIRRU, R., 2011 “An Approach Using Quantum PBIL to Solve the Traveling Salesman Problem”, International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering (M&C 2011), Rio de Janeiro, Brazil. 08-12 (May)

SILVA, M. H., SCHIRRU, R., 2011a, “Optimization of Nuclear Core Fuel Reload Using the New Quantum PBIL”, *Annals of Nuclear Energy*, v.38, pp. 610 – 614.

SILVA, M. H., SCHIRRU, R., LIMA, A. M. M., 2011b, “QACO_Alpha Applied to the Nuclear Reactor Core Fuel Reload Optimization”, *Progress in Nuclear Energy*, v.53, pp. 80 – 85.

SILVA, M. H., SCHIRRU, R., 2014, “A self-adaptive quantum PBIL method for the Nuclear Reload Optimization”, *Progress in Nuclear Energy*, v.74, pp. 103 – 109.

SOUTO, R. P., DE CAMPOS VELHO, H.F., STEPHANY, S., 2004, Reconstruction of Chlorophyll Concentration Profile in Offshore Ocean Water using a Parallel Ant Colony Code. Proc. 16th European Conference on Artificial Intelligence (ECAI – 2004), Hybrid Metaheuristics (HM - 2004).

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

WANG, L., NIU, Q., FEI, M., 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.

YU, Z., SHUHUA, L., SHUAI, F., DI, W., 2009, “A Quantum – Inspired Ant Colony Optimization for Robot Coalition Formation”. Chinese Control and decision Conference (CCDC2009).