



Universidade Federal
do Rio de Janeiro

Escola Politécnica

O PROBLEMA DA RECARGA DE REATORES NUCLEARES DO TIPO PWR
OTIMIZADO PELO ALGORITMO CUCO

Jéssica Salgado de Mello

Projeto de Graduação apresentado ao Curso de Engenharia Nuclear da Escola Politécnica, Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Engenheiro.

Orientadora: Andressa dos Santos Nicolau

Rio de Janeiro
Março de 2020

O PROBLEMA DA RECARGA DE REATORES NUCLEARES DO TIPO PWR
OTIMIZADO PELO ALGORITMO CUCO

Jéssica Salgado de Mello

PROJETO DE GRADUAÇÃO SUBMETIDO AO CORPO DOCENTE DO
CURSO DE ENGENHARIA NUCLEAR DA ESCOLA POLITÉCNICA
DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE
DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE
ENGENHEIRO NUCLEAR.

Examinado por:

Prof. Andressa dos Santos Nicolau, D.Sc.

Prof. Paulo Fernando Ferreira Frutuoso e Melo, D.Sc.

Dr. Marcelo Dornellas Machado, D.Sc.

RIO DE JANEIRO, RJ – BRASIL
MARÇO DE 2020

Salgado de Mello, Jéssica

O Problema da Recarga de Reatores Nucleares do Tipo PWR Otimizado pelo Algoritmo Cuco/Jéssica Salgado de Mello. – Rio de Janeiro: UFRJ/ Escola Politécnica, 2020. X, 39 p.: il.; 29, 7cm.

Orientadora: Andressa dos Santos Nicolau

Projeto de Graduação – UFRJ/ Escola Politécnica/ Curso de Engenharia Nuclear, 2020.

Referências Bibliográficas: p. 36 – 39.

1. Recarga de Reatores Nucleares. 2. Inteligência Artificial. 3. Otimização com Cuco. 4. Voo de Lévy. I. dos Santos Nicolau, Andressa. II. Universidade Federal do Rio de Janeiro, Escola Politécnica, Curso de Engenharia Nuclear. III. O Problema da Recarga de Reatores Nucleares do Tipo PWR Otimizado pelo Algoritmo Cuco

*Aos meus amados avós paternos,
Nilda e Guiomedes, à minha
querida amiga Lu, e à minha
companheirinha Danda Maria,
com muitas saudades.*

Agradecimentos

A Deus, que não deixou de contar nem um fio de cabelo meu que caiu ao longo dessa grande jornada pela graduação, pela sua divina providência e por sempre colocar pessoas maravilhosas e amáveis em minha vida. *Soli Deo Gloria*.

Aos meus familiares, especialmente meu amado companheiro e namorado Douglas, minha mãe, meus irmãos e meu pai, por aguentarem minhas crises de choro, me ajudarem tanto quanto possível e nunca me deixarem desistir, mesmo em tempos difíceis.

Aos amigos que muito me ajudaram e suportaram durante esses anos, especialmente Domenica, Carol, Fernando, Gilson e Kauan por ajudarem tanto neste trabalho durante o recesso. Às amigas Camila, Gabi e Nay por sempre me incentivarem a persistir desde o começo, mesmo de longe.

Aos professores do LMP/UFRJ, que se dispuseram a ajudar com este trabalho, Alan e Schirru. Aos funcionários do LMP, sempre dispostos a ajudar, especialmente à Si, sempre com uma palavra de incentivo.

À Andressa, especialmente, que com toda paciência e muita dedicação concordou em me orientar e guiar neste final de graduação, sempre solícita e disposta a ajudar em todos os detalhes.

Aos funcionários da Eletronuclear, que me mostraram a prática e vivência da engenharia nuclear no Brasil, especialmente ao Marcelo e Fernando, sempre gentis e solícitos para tirar dúvidas.

Resumo do Projeto de Graduação apresentado à Escola Politécnica/ UFRJ como parte dos requisitos necessários para a obtenção do grau de Engenheiro Nuclear.

O PROBLEMA DA RECARGA DE REATORES NUCLEARES DO TIPO PWR OTIMIZADO PELO ALGORITMO CUCO

Jéssica Salgado de Mello

Março/2020

Orientadora: Andressa dos Santos Nicolau

Curso: Engenharia Nuclear

Em reatores nucleares do tipo PWR (*Pressurized Water Reactor*), existe o problema de otimização da recarga nuclear (POR), um problema de análise combinatória de alta complexidade e de grande interesse na engenharia nuclear, tanto econômico quanto acadêmico. Por este motivo, algoritmos de inteligência artificial (IA) foram sendo desenvolvidos e aplicados ao POR desde meados da década de 1980. O presente trabalho trata sobre o POR ao propor um novo método de otimização de IA para ele, especificamente para o ciclo de operação 7 da usina nuclear Angra 1. Este método é proposto através do desenvolvimento e implantação de um algoritmo de busca, que utiliza o Voo de Lévy, baseado na estratégia de reprodução da família de aves Cuculidae, o algoritmo de otimização cuco (COA). Para que as condições de segurança de cada solução candidata produzida pelo COA para o POR pudessem ser avaliadas, foi feito o uso do código nodal de física de reatores RECNOD, cujos parâmetros utilizados pelo COA foram a concentração de boro solúvel na água do circuito primário e a potência média relativa dos elementos combustíveis. Os resultados obtidos mostraram-se satisfatórios, uma vez que podem ser comparados aos já existentes na literatura, o que ressalta a eficiência e robustez do COA.

Palavras Chave— Recarga de Reatores Nucleares, Inteligência Artificial, Otimização com Cuco, Voo de Lévy

Abstract of Undergraduate Project presented to POLI/UFRJ as a partial fulfillment of the requirements for the degree of Engineer.

PWR NUCLEAR REACTOR RELOAD PROBLEM OPTIMIZED BY CUCKOO
ALGORITHM

Jéssica Salgado de Mello

March/2020

Advisor: Andressa dos Santos Nicolau

Course: Nuclear Engineering

In nuclear reactors of the PWR (Pressurized Water Reactor) type, there is the problem of optimization of the nuclear reload (POR), a problem of combinatorial analysis of high complexity and of great interest in nuclear engineering, both economical and academic. For this reason, artificial intelligence (AI) algorithms have been developed and applied to POR since the mid-1980s. This work deals with POR by proposing a new method of AI optimization for it, specifically for the cycle 7 of operation of the nuclear power plant Angra 1. This method is proposed through the development and implementation of a search algorithm, which uses Lévy Flights, based on the breeding strategy of the Cuculidae bird family, the cuckoo optimization algorithm (COA). So that the safety conditions of each candidate solution produced by the COA for the POR could be evaluated, the nodal reactor physics code of RECNOD was used, whose parameters used by the COA were the concentration of soluble boron in the water of the primary circuit and the relative average power of the fuel assemblies. The results obtained were satisfactory, since they can be compared to those already existing in the literature, which highlights the efficiency and robustness of COA.

Keywords— Nuclear Reactor Reload, Artificial Intelligence, Cuckoo Optimization Algorithm (COA), Lévy Flight

Sumário

Lista de Figuras	ix
Lista de Tabelas	x
1 Introdução	1
2 O Problema da Recarga	4
2.1 Gerenciamento do Combustível no Núcleo do Reator	4
2.2 O Problema da Recarga (POR)	7
2.2.1 Métodos de Otimização Aplicados ao POR	10
3 O Algoritmo de Otimização Cuco	13
3.1 Algoritmos de Otimização Evolucionários e de Enxame	13
3.2 O COA - <i>Cuckoo Optimization Algorithm</i>	15
3.3 Voo de Lévy	18
3.4 Aplicação do COA a Funções de Teste	20
3.4.1 Função Esfera	20
3.4.2 Função Rastrigin	23
3.4.3 Função Griewank	26
4 O COA aplicado ao Problema de Otimização da Recarga	29
4.1 Aplicação do COA à otimização do ciclo 7 de Angra 1	29
4.2 Resultados da aplicação do COA ao POR	31
4.3 Comparação entre os resultados obtidos pelo COA e resultados de outras meta-heurísticas da literatura	33
5 Conclusões e Trabalhos Futuros	34
5.1 Conclusões	34
5.2 Trabalhos Futuros	35
Referências Bibliográficas	36

Lista de Figuras

2.1	Esquema do elemento combustível	5
2.2	Esquema simplificado de uma usina nuclear do tipo PWR	6
2.3	Representação do núcleo do reator de Angra 1	8
2.4	Representação de um quarteto	9
2.5	Representação de um octeto	9
2.6	Representação de um octeto do núcleo do reator de Angra 1	10
3.1	Fluxograma do COA	16
3.2	Processo de busca do COA	17
3.3	Voo de Lévy partindo da origem com 50 passos consecutivos	19
3.4	Pseudocódigo do COA	20
3.5	Função Esfera	21
3.6	Função Rastrigin	23
3.7	Função Griewank	26
4.1	Vetor solução do COA	29
4.2	Procedimento realizado pelo Random Keys	30

Lista de Tabelas

3.1	Resultados dos testes da aplicação do COA à função Esfera	22
3.2	Resultados dos testes da aplicação do COA à função Rastrigin para o intervalo $x \in [-5, 12, 5, 12]$	24
3.3	Resultados dos testes da aplicação do COA à função Rastrigin para o intervalo $x \in [-2, 56, 2, 56]$	25
3.4	Resultados dos testes da aplicação do COA à função Griewank	27
3.5	Melhores resultados para cada função de teste	28
4.1	Resultados do COA aplicado à otimização do problema da recarga para 100 ninhos e diferentes probabilidades de abandono	32
4.2	Resultados do COA aplicado à otimização do problema da recarga	32
4.3	Comparação entre o COA e os algoritmos ACS, PBIL e PSO	33

Capítulo 1

Introdução

O progresso de todas as nações está intimamente relacionado com a capacidade de produzir ou adquirir energia elétrica por parte de cada país. Essa energia é vital para sustentar diversos setores de grande importância econômica para muitos países, como o setor industrial, por exemplo. A matriz elétrica mundial é dividida de acordo com as principais fontes de produção energética, com o carvão sendo responsável pela maior parcela da produção de energia mundial do ano de 2016 com 38,3% da produção total. Em seguida, temos gás natural com 23,1%, e hidráulica (por meio de hidroelétricas) com 16,6% e nuclear com 10,4%, sendo essas as principais fontes [22].

A matriz elétrica no Brasil é composta de forma diferente da matriz mundial. Aqui, no ano de 2017, tivemos a hidráulica como a geradora de mais da metade do total da energia gerada, ou seja com 65,2%, seguida por gás natural com 10,5%, biomassa com 8,2%, solar e eólica com 6,9%, carvão com 4,1%, nuclear com 2,6% e petróleo com 2,5% [22]. Este fato se dá pelo grande potencial hídrico do Brasil, onde há longos períodos de chuva em volumes muito grandes. Isto, porém, pode acarretar impactos ambientais, uma vez que requer o alagamento de grandes áreas que podem conter vegetação e fauna, além de haver a possibilidade de secas e estiagens prolongadas, o que pode provocar apagões no país. Por este motivo, é ideal que a matriz elétrica seja diversificada e o investimento em novas formas de produção de energia procurando reduzir os impactos socioambientais tanto quanto possível é de grande importância.

Para produção em larga escala, a energia nuclear mostra-se uma boa opção, tanto economicamente, quanto para o meio-ambiente, visto que demanda uma menor área para sua construção e possui grande capacidade de produção, além de não estar sujeita às mudanças climáticas locais, e não emite gases poluentes para a atmosfera.

No Brasil, a energia nuclear é responsável por cerca de 3% do total anual da matriz energética. Este percentual corresponde a aproximadamente 1900 MWe, que são produzidos pelas usinas nucleares Angra 1 e Angra 2, localizadas no estado do Rio de

Janeiro, no litoral do município de Angra dos Reis. Ambas as usinas são do tipo PWR (*Pressurized Water Reactor*), isto é, contam com um reator de fissão nuclear por meio de nêutrons lentos contido em um circuito de água pressurizada. Este circuito, chamado primário, troca calor por meio de geradores de vapor com o circuito secundário, onde estão localizadas as turbinas de alta e baixa pressão, que compartilham o mesmo eixo com o gerador elétrico, que, por sua vez, gera a energia elétrica. Este circuito secundário, por sua vez, troca calor com a água do mar, devidamente recolhida e tratada para este fim, atuando como a fonte fria para o circuito secundário. A usina nuclear Angra 1 possui dois *loops*, o que significa que possui dois geradores de vapor, enquanto que, a usina nuclear Angra 2 possui quatro *loops*. Ambas as usinas possuem o mesmo tipo de combustível, que é o óxido de urânio enriquecido, e utilizam o mesmo tipo de moderador de nêutrons, que opera também como refrigerante no circuito primário (água leve) [21].

Um reator nuclear do tipo PWR precisa ser reabastecido/recarregado a intervalos de tempo estabelecidos de acordo com o seu projeto e com o enriquecimento do combustível utilizado e, diferentemente dos reatores do tipo CANDU [9], sua recarga não pode ser feita durante seu funcionamento (recarga *online*), portanto, é preciso que o reator seja desligado para que a recarga possa ser executada. Durante esse período de desligamento, a usina também passa por processos de manutenção. Esse período é chamado de parada para recarga.

A recarga do reator é um processo que deve ser planejado com antecedência, uma vez que seu componente principal é a encomenda dos EC novos que também irão compor o projeto do próximo ciclo de operação da usina. Este projeto consiste em como os elementos combustíveis (EC) serão organizados dentro do núcleo do reator para o próximo ciclo, de forma tal que o comprimento deste, isto é, o tempo que este próximo ciclo irá durar seja o maior possível, para que a empresa que gerencia a usina tenha mais lucros e menos perdas pois cada dia de operação da usina equivale a centenas de milhares de dólares. Este processo de projetar um núcleo com o maior número de dias efetivos à plena potência (DEPP), quanto possível, é conhecido na literatura como o problema de otimização da recarga (POR) ou problema da recarga nuclear (PRN), e é tido como um dos problemas de maior interesse da engenharia nuclear, assim como um dos mais estudados, devido à sua grande complexidade e importância econômica [38].

Durante três décadas, desde o começo do uso de reatores de potência para produção de energia elétrica em larga escala em meados da década de 1950, o POR foi resolvido de forma manual, com base nos conhecimentos e experiências prévias dos especialistas que projetavam as combinações de núcleos dos ciclos de operação subsequentes. Com a expansão do uso de microcomputadores e do desenvolvimento de ferramentas de informática com maior nível de sofisticação a partir da década de 1980, novas abordagens para o POR começaram a ser propostas por pesquisadores, onde diferentes

algoritmos meta-heurísticos de inteligência artificial foram utilizados para otimizar o POR.

Assim, devido a este fato e à grande importância econômica do POR, técnicas de otimização de inteligência artificial foram sendo implantadas a ele.

O objetivo de se implantar algoritmos evolucionários e de enxame no POR é obter a melhor configuração de núcleo (CN), ou padrão de recarga (PR), tanto do ponto de vista econômico quanto do ponto de vista da segurança nuclear.

Neste projeto, é apresentada uma abordagem para a solução do POR, onde o algoritmo de otimização cuco (COA ou CSA: *Cuckoo Optimization Algorithm* ou *Cuckoo Search Algorithm*) proposto por Yang e Deb, 2009 [52] é usado. O cuco será responsável por produzir soluções otimizadas, padrões de recarga, respeitando as devidas condições e restrições do problema. Neste trabalho, o código nodal de física de reatores RECNOD desenvolvido por Chapot, 2000 [8] será usado para avaliar as soluções fornecidas pelo cuco, ou seja, irá retornar os parâmetros de cada combinação de núcleo que são necessários para determinar se uma combinação de núcleo é válida, por meio de cálculos de física de reatores. Para apresentar a teoria e resultados encontrados o presente trabalho foi estruturado da maneira descrita a seguir:

O Capítulo 2 apresenta uma introdução a respeito do ciclo do combustível nuclear no núcleo do reator, que descreve os principais componentes envolvidos no problema da recarga, bem como a descrição do problema da recarga nuclear.

No Capítulo 3 é apresentada uma introdução dos algoritmos de otimização, bem como a descrição do algoritmo cuco e como ele pode ser aplicado em funções de teste, para fins de validação do próprio algoritmo cuco.

No Capítulo 4 é descrita a implantação do algoritmo cuco na recarga e como esta implantação foi realizada neste trabalho. Há também a descrição do código nodal de física de reatores RECNOD, bem como suas atribuições e propriedades de maior importância para o processo de otimização da recarga realizada pelo algoritmo cuco.

Em seguida, no Capítulo 5 é descrito o método proposto e os principais resultados obtidos para o POR, com a aplicação do algoritmo cuco. Por fim, no Capítulo 6 são apresentadas as conclusões sobre o método proposto e as sugestões para trabalhos futuros.

Capítulo 2

O Problema da Recarga

2.1 Gerenciamento do Combustível no Núcleo do Reator

O núcleo do reator de uma usina nuclear é considerado o coração da usina; ele está para a usina termonuclear assim como a caldeira está para as demais usinas termoelétricas [21]. No núcleo de um reator nuclear do tipo PWR, existem estruturas a que chamamos elementos combustíveis (EC), contendo pastilhas cerâmicas de óxido de urânio (UO_2), que possuem enriquecimento de 5% no máximo. Para o caso específico da usina nuclear Angra 1, há 121 elementos combustíveis no núcleo do reator medindo 4 metros de comprimento. Cada EC possui uma estrutura de 16 x 16 varetas cilíndricas, compostas por liga de zircônio, totalizando 235 varetas contendo urânio enriquecido, 20 tubos-guia (para a inserção de barras de controle e desligamento) e 1 tubo de instrumentação (para a tomada de medições), além das 8 grades espaçadoras e um bocal inferior e outro superior [21] e [25]. A Figura 2.1 apresenta um esquema do elemento combustível.

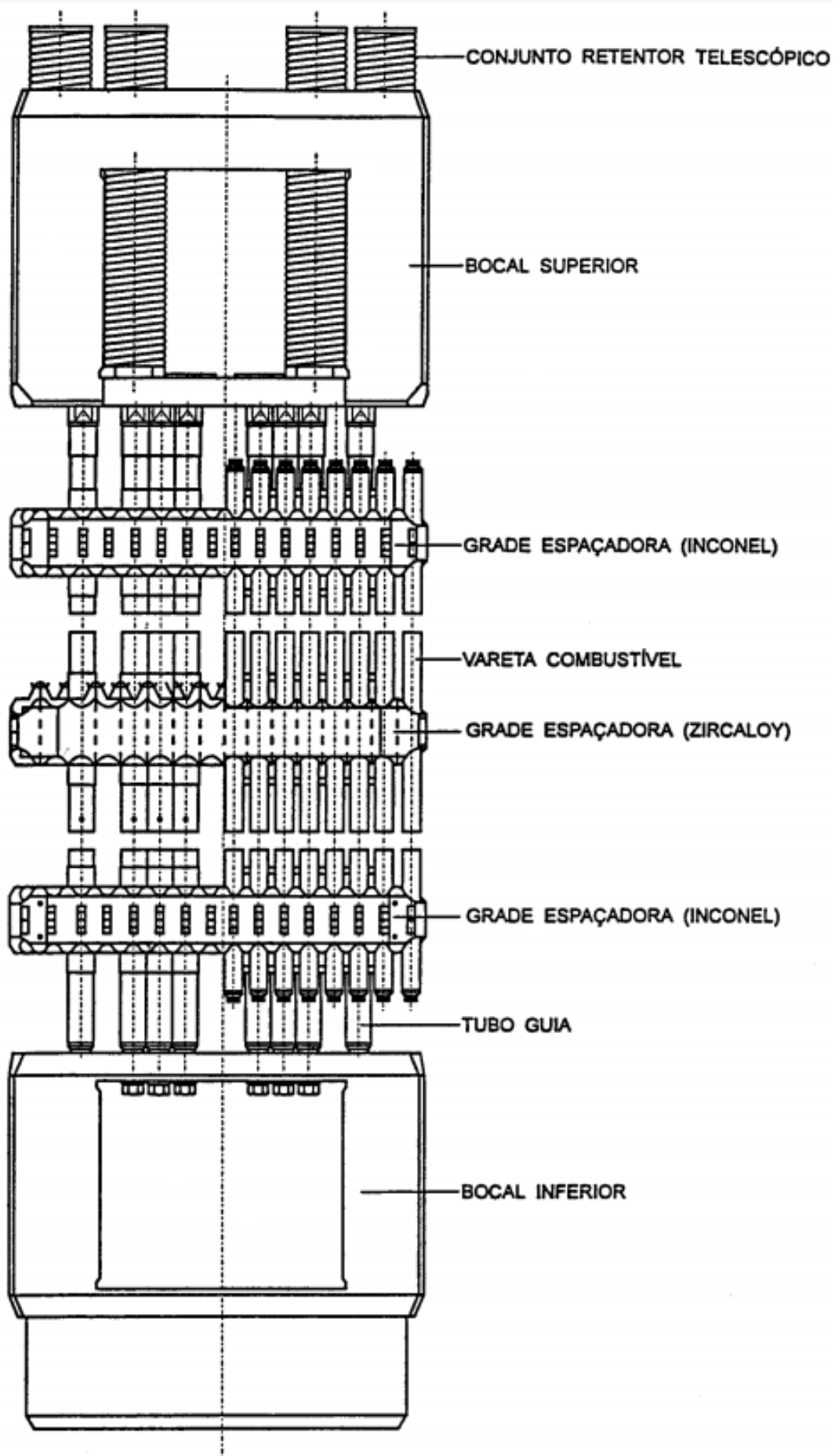


Figura 2.1: Esquema do elemento combustível [20]

Uma usina nuclear do tipo PWR funciona da seguinte forma: o combustível (urânio enriquecido) é carregado no núcleo do reator para que seja iniciada a reação de fissão em cadeia. Esta reação produzirá calor, que irá fazer com que a água em contato com ele esquente até aproximadamente 300°C. Esta água do circuito primário irá trocar calor com a água do circuito secundário para que esta última seja transformada em vapor. O vapor gerado, por sua vez, irá movimentar as turbinas de alta e baixa pressão, que compartilham o mesmo eixo com o gerador elétrico, que irá, por fim, transformar a energia cinética das turbinas em energia elétrica para a rede de distribuição. A Figura 2.2 apresenta um esquema simplificado de um reator do tipo PWR.

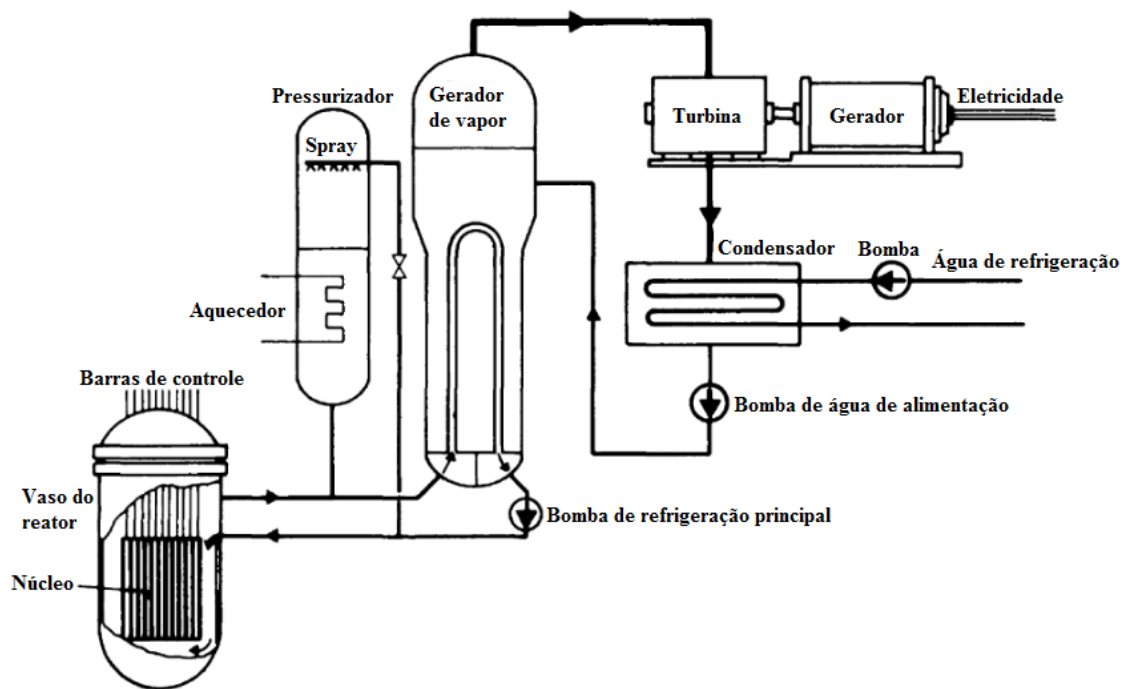


Figura 2.2: Esquema simplificado de uma usina nuclear do tipo PWR [41]

O presente trabalho foi desenvolvido com base nos dados do ciclo 7 de operação da usina nuclear Angra 1. A usina Angra 1 está localizada na Central Nuclear Almirante Álvaro Alberto, estado do Rio de Janeiro, no município de Angra dos Reis. Ela produz cerca de 640 MWe de potência elétrica nominal bruta, foi projetada pela empresa norte-americana Westinghouse, sua construção foi autorizada pela CNEN (Comissão Nacional de Energia Nuclear) no ano de 1970, teve sua primeira criticalidade em 1982, e é operada pela Eletronuclear.

O tempo de duração de um ciclo de operação da usina nuclear Angra 1 é de aproximadamente 12 meses, após esse período é necessário que seja feita a recarga do núcleo do reator, ou seja, é necessário que parte dos elementos combustíveis sejam renovados.

No processo de recarga, uma fração dos EC é removida do núcleo e reposta por EC

novos. No caso de Angra 1, até o ciclo 20, que terminou em maio de 2015, apenas o combustível do tipo 16STD (16×16 *standard*) foi utilizado, o que fazia com que a fração de EC que seria substituída por um novo lote de EC fosse igual a $\frac{1}{3}$ de 121 EC; portanto, 40 EC novos eram carregados a cada novo ciclo de operação. Assim, os 81 restantes seriam removidos do núcleo e encaminhados para a piscina de EC usados, que é uma instalação de armazenamento intermediário dos EC. Estes EC usados podem retornar ou não ao núcleo para os próximos ciclos, a depender dos próximos projetos de recarga, que utilizam EC já utilizados pelo menos uma vez [8].

Cada EC deve ser utilizado em mais de um ciclo, até atingir seu limite de queima (55.000 MWD/MTU [25]). Após atingir esse limite, ou o mais próximo dele quanto possível, ele deve ir para o local de armazenamento definitivo (unidades de armazenamento a seco, por exemplo). Enquanto, porém, ele não atingir tal limite, deve ficar no local de armazenamento intermediário, a este processo chamamos ciclo do combustível nuclear aberto. Já em um ciclo do combustível nuclear fechado, os elementos são enviados para uma planta de reprocessamento do combustível nuclear [46] e [8] (o que, atualmente, não existe no Brasil).

Ao final de cada ciclo de operação, a usina sofre uma parada para recarga, isto é, ela é levada às condições adequadas de desligamento para que o núcleo do reator seja aberto e o processo de recarga possa ser conduzido pelos trabalhadores necessários para realizar este processo. Este período de parada, na usina nuclear Angra 1, tem duração de aproximadamente quatro semanas [8].

Para cada dia que a usina permanece desligada há um impacto significativo na receita da empresa que a gerencia [10]. Assim sendo, é desejável que a duração dos seus ciclos de operação seja tão grande quanto possível, isto é, maximizada, observando sempre as devidas condições e restrições de segurança demarcadas pelo órgão regulador no país em questão [46].

2.2 O Problema da Recarga (POR)

O problema de otimização da recarga (POR) de reatores nucleares consiste em, a priori, como os EC novos e usados serão organizados para compor o projeto do núcleo para o próximo ciclo de operação do reator, de maneira que o tempo de operação do novo ciclo seja maximizado, respeitando sempre as devidas condições e parâmetros de segurança.

Para um reator como o da usina de Angra 1, com 121 EC, o número de possíveis combinações dos EC é de $121! = 8 \times 10^{200}$, este valor cai para 10^{25} quando aplicadas as restrições de posicionamento dos EC. Para serem produzidas todas essas combinações de

núcleo seriam necessários $5,8 \times 10^{19}$ anos fazendo uso das ferramentas computacionais disponíveis atualmente, uma vez que um código de física de reatores leva aproximadamente 30 segundos para realizar todos os cálculos. Este tipo de problema é dito um problema não polinomial completo (NP - completo), e não pode ser resolvido em tempo computacional viável [38].

No caso da Usina Nuclear Angra 1, o núcleo do reator é ordenado de forma tal que pode ser dividido em oito seções simétricas, o que o faz possuir simetria e eixos de simetria imaginários de $\frac{1}{8}$ e $\frac{1}{4}$. Aos EC que se encontram postos sobre os eixos de simetria dá-se o nome de quartetos, uma vez que cada 4 EC formam um conjunto de simétricos com propriedades aproximadamente iguais (enriquecimento, queima, k_{eff} , fator de pico F_{XY}). A cada grupo destes dá-se o nome de EC irmãos. Aos EC restantes, dá-se o nome de octetos por existirem oito posições simétricas no núcleo para cada grupo de oito EC. O elemento central, posicionado na interseção entre os eixos de simetria, é único e não possui posições de simetria ou EC equivalentes a ele. A Figura 2.3 mostra como os EC estão organizados no núcleo do reator de Angra 1; os EC hachurados representam os EC de quarteto, enquanto que os EC em branco representam os EC de octeto [10]. As Figuras 2.4 e 2.5 representam isoladamente um quarteto e um octeto, respectivamente.

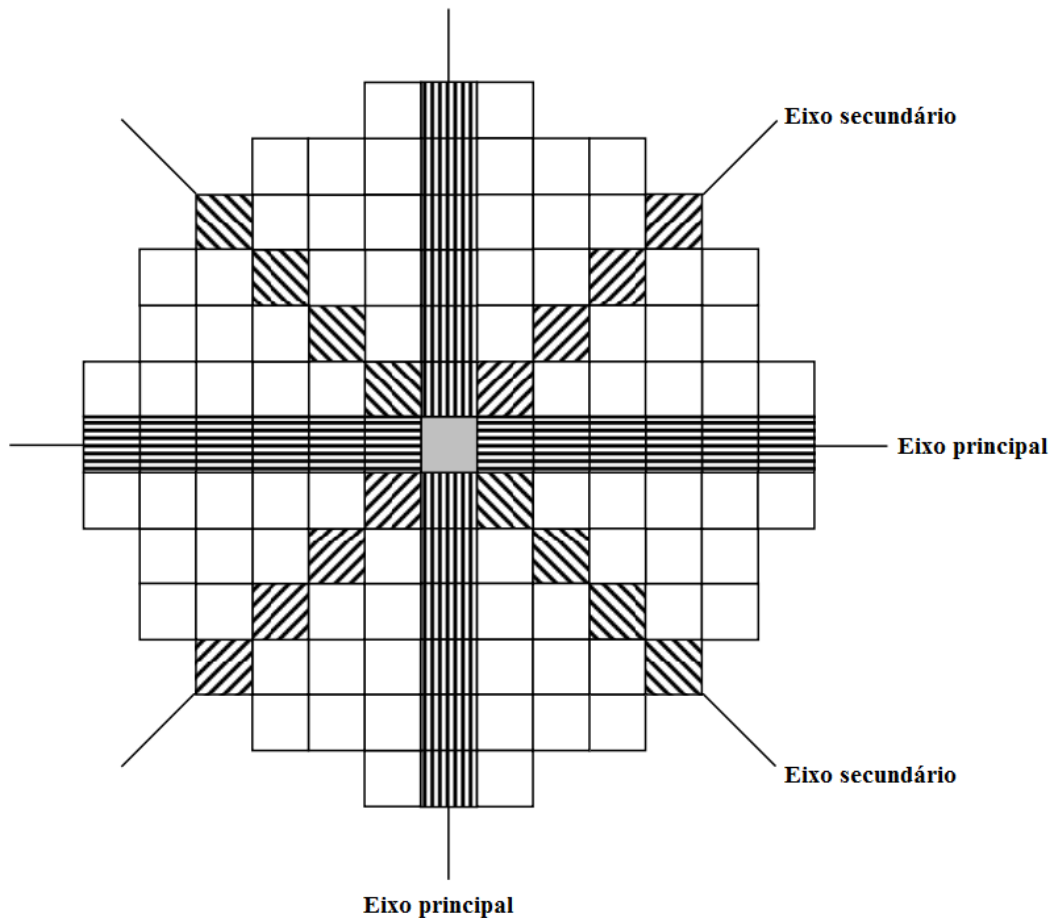


Figura 2.3: Representação do núcleo do reator de Angra 1 [10]

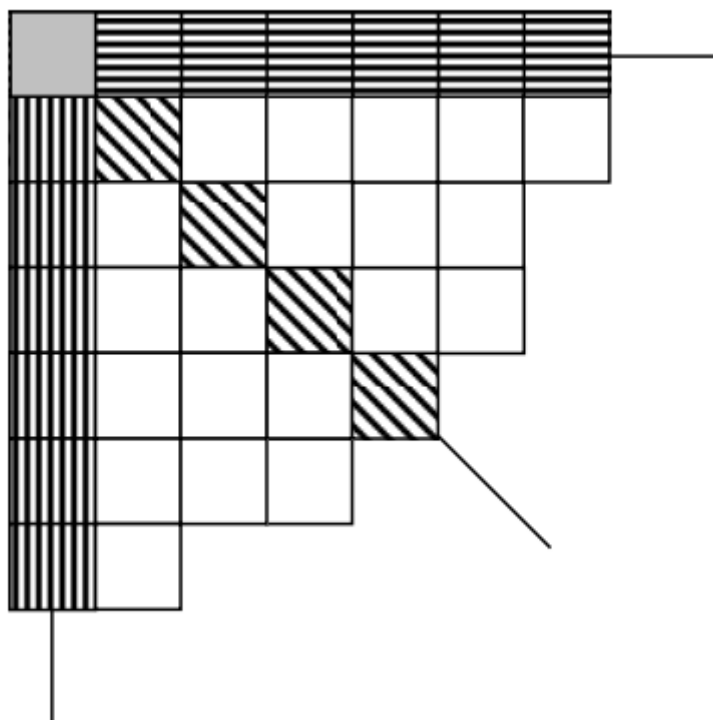


Figura 2.4: Representação de um quarteto [10]

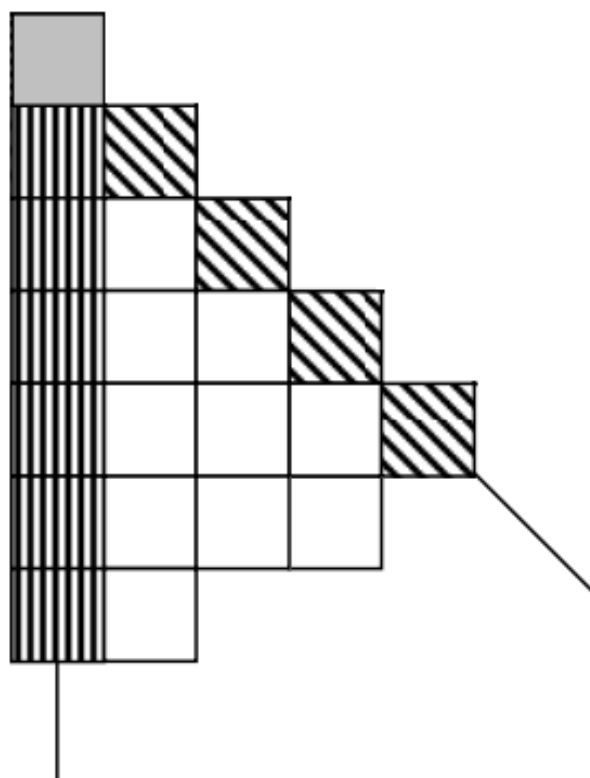


Figura 2.5: Representação de um octeto [10]

Para efeito de simplificação dos cálculos do POR e devido às características de simetria e analogia entre os EC de cada quarteto e suas propriedades, podemos considerar as soluções de apenas um octeto e, então, rebatê-las para as outras sete posições restantes para formar um núcleo inteiro. Assim sendo, apenas 20 EC irão tomar parte na permuta, o que reduz o número de soluções possíveis a $20! = 2,4 \times 10^{18}$, o que não diminui, porém, a complexidade do POR [38]. A Figura 2.6 mostra apenas um octeto, com seus EC numerados de 1 a 20, sendo que de 1 a 10 temos os quartetos e de 11 a 20, os octetos. Cabe ressaltar que o EC central não participa da permuta, permanecendo fixado em sua posição [10].

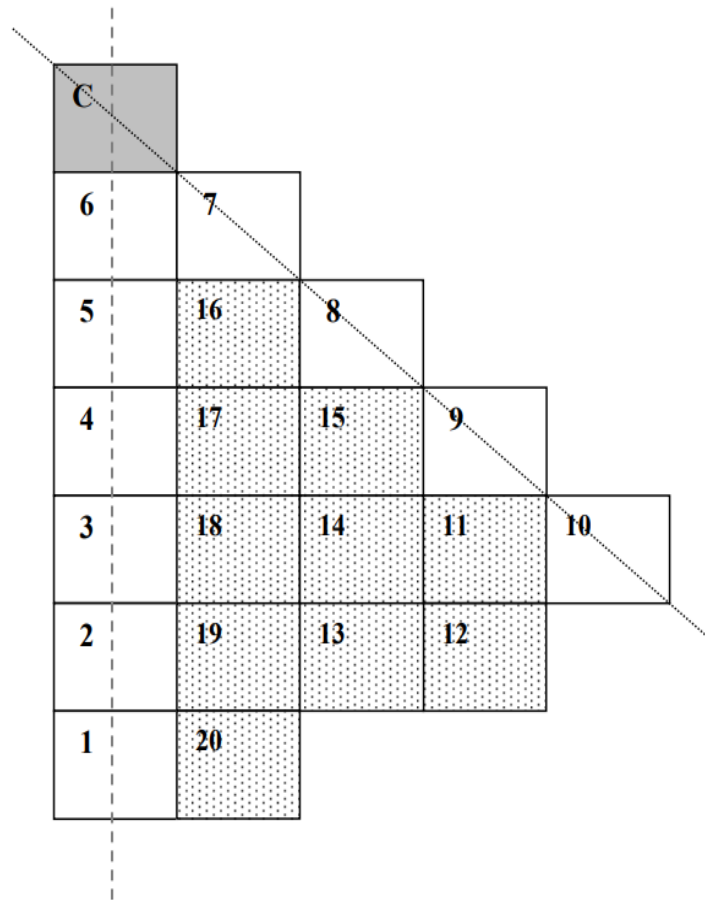


Figura 2.6: Representação de um octeto do núcleo do reator de Angra 1 [10]

2.2.1 Métodos de Otimização Aplicados ao POR

Devido à grande importância econômica do POR, as técnicas de otimização aplicadas para resolvê-lo mostram-se cada vez mais sofisticadas. É de grande importância que o profissional responsável por elaborar os projetos dos ciclos de operação da usina tenha ferramentas e formas de obter um projeto de ciclo ótimo, uma vez que a recarga manual nem sempre fornece resultados que otimizem a queima dos EC.

Dessa forma, podemos citar alguns algoritmos meta-heurísticos já aplicados ao POR: Sistemas Especialistas [23], *Simulated Annealing* [30], Algoritmos Genéticos [42], Redes Neurais e Lógica Fuzzy [28], Pesquisa Tabu (ou Busca Tabu) [7], ACS (*Ant Colony System*) [14], PBIL (*Population-Based Incremental Learning*) [33], PSO (*Particle Swarm Optimization*) [15], algoritmos de inspiração quântica [38] e [10], CE (*Cross-Entropy Algorithm*) e ABC (*Artificial Bee Colony*) [35], e CSA com Voo de Lévy [45].

O objetivo de se implantar algoritmos evolucionários e de enxame no POR é obter a melhor configuração de núcleo (CN), ou padrão de recarga (PR), tanto do ponto de vista econômico quanto do ponto de vista da segurança nuclear. Para que estes algoritmos possam realizar esta otimização adequadamente, é necessário que recebam uma função para maximizar ou minimizar, dada a natureza do problema em questão. Esta função é conhecida como função objetivo, ou função *fitness*. Para o caso do POR, aplicações da literatura apresentam funções objetivos que visam maximizar a concentração de boro crítica no refrigerante do primário respeitando os limites operacionais de segurança, como por exemplo, o fator de segurança conhecido como fator de pico de potência radial. Este limite é determinado através de parâmetros como, por exemplo, o fluxo de nêutrons no interior do núcleo, a reatividade, a queima de combustível, a criticalidade com boro dissolvido no refrigerante, a correção de densidade do moderador, a realimentação Doppler e a reconstrução da densidade de potência pino a pino [8].

Tanto a concentração crítica de boro solúvel quanto o fator de pico de potência são necessários para a função objetivo do POR que será utilizada pelo algoritmo de busca desenvolvido neste trabalho. Entretanto, esses parâmetros são calculados por um código que é programado para este propósito específico, isto é, um código nodal de física de reatores. Nos projetos de recarga reais feitos para a usina Angra 1, é utilizado o código ANC (*Advanced Nodal Code*) [31], licenciado pela Westinghouse. No presente trabalho, será utilizado o código RECNOD, desenvolvido especificamente para o ciclo 7 de operação da usina.

A concentração crítica de boro (diluído no refrigerante do circuito primário como ácido bórico) está relacionada com o comprimento do ciclo, uma vez que a cada dia de operação, a usina consome em média 4 ppm de boro dissolvido. Dessa forma, quanto maior a concentração crítica de boro no refrigerante, maior será seu comprimento de ciclo. Por este motivo, uma das atribuições da função objetivo é maximizar a concentração crítica de boro. Por outro lado, a configuração de núcleo que apresenta a maior concentração de boro, não necessariamente irá ter o seu fator de pico dentro do limite de segurança pré-estabelecido para aquela usina.

Diferentes funções objetivos foram propostas para representar o problema da recarga ao longo dos anos, como, por exemplo, DeChaine e Feltus, 1995 [16], Chapot, 2000 [8] e

Machado e Schirru, 2002 [32].

Neste trabalho, assim como em De Lima, 2005 [14], Nicolau, 2014 [38] e Da Silva, 2014 [10] será usada a seguinte função objetivo, representada pela Eq. (2.1):

$$Fitness = \begin{cases} \frac{1}{C_B} & \text{se } P_{rm} \leq 1,395 \\ P_{rm} & \text{caso contrário} \end{cases} \quad (2.1)$$

onde P_{rm} é a máxima potência média relativa dos EC e C_B é a concentração de boro solúvel.

Capítulo 3

O Algoritmo de Otimização Cuco

3.1 Algoritmos de Otimização Evolucionários e de Enxame

Otimização é o processo pelo qual se pode encontrar os máximos ou mínimos de um dado problema, isto é, o ponto ótimo da função que descreve melhor este problema, e existem diferentes formas de se realizar este processo. Uma destas formas é a otimização por meio de algoritmos de inteligência artificial [11], que, em sua maioria, são inspirados em fenômenos e processos da natureza. Para funções bem comportadas, contínuas que possuem domínio conhecido, a otimização pode ser feita através de testes da derivada segunda, por exemplo.

Dentro desse campo da inteligência artificial, podem-se destacar três linhas principais de algoritmos de otimização: Algoritmos Evolucionários, a exemplo do Algoritmo Genético [24] e do PBIL - *Population Based Incremental Learning* [4], entre outros e os Algoritmos de Enxame, a exemplo do PSO - *Particle Swarm Optimization* [27], ACS (*Ant Colony System*) [18] e do COA - *Cuckoo Optimization Algorithm* [52], entre outros, e os algoritmos de inspiração quântica [37], que são baseados nos algoritmos evolutivos e de enxames, porém usam a teoria de bit quântico da computação quântica, a exemplo do QEA - *Quantum Evolutionary Algorithm* [38], QDPSO - *Quantum Delta Potential Particle Swarm Optimization* e QPBIL - *Quantum Population Based Incremental Learning*.

O Algoritmo Genético se baseia na metáfora com a teoria da evolução das espécies, desenvolvida por Darwin, 1859 [12], e foi proposto por Goldberg, 1989 [24]. Nele, pode-se observar o fundamento de muitas características que foram incorporadas pelos algoritmos que surgiram em seguida. São elas: o estabelecimento de uma população inicial de indivíduos (as possíveis soluções para o problema), a determinação de uma função *fitness*, ou função objetivo, que será a medida da capacidade de cada indivíduo se adaptar ao que se propõe, e o descarte de indivíduos com *fitness* insuficiente. Além dessas, há o cruzamento entre os indivíduos com melhor *fitness* para produzir novos

indivíduos e perpetuar suas características (genes), gerando, assim, uma nova população, o que faz com que o conhecimento adquirido com a população anterior seja passado para as gerações posteriores, assim como há também uma taxa de mutação que se refere à diversidade genética e permite que o algoritmo não fique preso em máximos ou mínimos locais, podendo, assim, variar as características genéticas dos indivíduos das populações subsequentes.

Já o PSO (*Particle Swarm Optimization*), ou enxame de partículas, proposto por Kennedy e Eberhart, 1995 [27], é inspirado no movimento de um cardume de peixes, ou uma revoada de pássaros, em sua procura por alimento, assim como ACS (*Ant Colony System*), proposto por Dorigo e Gambardella, 1997 [18], inspirado no comportamento de colônias de formigas de deposição de feromônios por onde passam.

Além desses, pode-se citar também o COA (*Cuckoo Optimization Algorithm*), proposto por Yang e Deb, 2009 [52], como sendo um algoritmo de enxame. Sua estrutura é baseada no comportamento reprodutivo de uma família de aves, a Cuculidae, cuja característica principal é que, em sua grande maioria, essas aves são parasitas de ninhada.

De acordo com Rajabioun, 2011 [43], dentre as vantagens de se fazer uso desses algoritmos de inteligência artificial, podem-se citar as seguintes:

- têm a capacidade de resolver problemas sem soluções por métodos tradicionais e sem expertise humana;
- podem ser aplicados em qualquer tipo de problema que possa ser descrito através da otimização de uma função objetivo;
- podem ser utilizados com outros métodos tradicionais de otimização bem como combinados entre si; e
- possuem robustez quando submetidos a mudanças dinâmicas, o que os métodos tradicionais de otimização não apresentam, pois precisam ser reinicializados completamente para fornecer uma solução.

Das diferentes áreas do conhecimento em que se podem ser aplicados os algoritmos de inteligência artificial em geral, podem-se citar algumas, tais como [1], [2], [26], [48], [50].

Na engenharia nuclear, os algoritmos de inteligência artificial têm destaque em muitas áreas, tais como: o problema de identificação de acidentes [38], identificação de transientes [39], além do próprio POR [15], [3], [40]. Há também outras aplicações na área de análise de segurança nuclear, tais como [53] e [29].

3.2 O COA - *Cuckoo Optimization Algorithm*

Aves são animais ovíparos que possuem reprodução sexuada, isto é, cada espécie possui exemplares de fêmeas e machos. Sua reprodução tem início com o acasalamento e, após ele, a fecundação. Após este processo, o organismo da fêmea irá produzir um ovo, que será posto e chocado para que, por fim, o filhote possa eclodir e dar continuidade àquela espécie. Para aves que voam, esta estratégia de reprodução é de suma importância, uma vez que o peso do ovo gera dificuldades para a locomoção eficaz da fêmea através do voo. Assim, para a postura dos ovos, muitas aves fazem ninhos fixos, seja em árvores, em nichos de rochas, ou até mesmo na areia, como as aves marinhas. É importante que o ninho fique a salvo de predadores, uma vez que os pais precisam buscar alimento tanto para si quanto para os filhotes.

A expressão parasita de ninhada, ou parasita de ninho, refere-se às espécies que não constroem ninhos, tais como o chupim e a maioria das espécies de cuco, que depositam seus ovos em ninhos de outras espécies para que eles sejam chocados e seus filhotes cuidados pelos pais adotivos dessas espécies hospedeiras. Cabe destacar que este processo, algumas vezes, não tem sucesso, visto que a espécie hospedeira pode abandonar o ninho se identificar que os ovos do parasita não são os seus próprios, ou pode expulsar os ovos alienígenas, jogando-os para fora do ninho. Para minimizar esta possibilidade, algumas espécies de cuco são especialistas em mimetizar o tamanho e aparência dos ovos de algumas espécies hospedeiras quando na postura dos seus próprios ovos. Há também a possibilidade de a fêmea do parasita ser flagrada durante a sua postura no ninho e entrar em conflito com a espécie hospedeira. Para evitar este encontro, ela precisa de não mais do que dez segundos para se dirigir ao ninho da espécie hospedeira, retirar alguns dos ovos já postos ali e pôr os seus próprios. Ela pode escolher diferentes ninhos e pôr um ou mais ovos por ninho [51] e [13].

O COA (*Cuckoo Optimization Algorithm*), proposto por Yang e Deb, 2009 [52] e posteriormente aperfeiçoado por Rajabioun, 2011 [43], tem como metáfora este comportamento de parasitismo dos cucos. A sua estrutura se inicia, assim como outros algoritmos de inteligência artificial, através de uma população inicial de indivíduos. No COA, sua população pode ser de cucos ou de ovos de cuco, uma vez que os cucos põem ovos em ninhos de aves hospedeiras, e esses ovos, se não forem descobertos, transformar-se-ão em cucos.

Este trabalho irá abordar a forma mais simples do COA, onde há apenas um ovo por ninho. Cada cuco irá pôr um ovo por vez em um ninho escolhido de forma aleatória. O melhor ninho, com o ovo de melhor qualidade, será carregado para as próximas gerações. O número de ninhos hospedeiros é fixo e o ovo posto por um cuco é descoberto pela ave hospedeira com uma probabilidade $p_a \in [0, 1]$. Nesse caso, o pássaro hospedeiro pode tanto abandonar o ninho em prol de construir um novo ninho em outro ponto quanto pode

descartar o ovo. Em outras palavras, uma fração p_a dos n ninhos hospedeiros é repostada por novos ninhos com novas soluções aleatórias [51].

Assim, cada ovo irá representar uma solução possível para o problema e cada cuco pode pôr um ovo. Portanto, na abordagem utilizada neste trabalho, não há distinção entre ovo, ninho ou cuco, na prática, uma vez que cada ninho corresponde a um ovo que também representa um cuco [51]. Tendo em vista estas propriedades do algoritmo, pode-se descrever o funcionamento do COA com os passos a seguir: um cuco é iniciado e põe seus ovos nos ninhos de aves hospedeiras, os que tiverem melhor desempenho para não serem descobertos permanecerão nas próximas gerações, já os que forem descobertos pelas aves hospedeiras serão abandonados juntamente com o ninho ou serão descartados e terão, portanto, um pior desempenho em comparação com os demais. A Figura 3.2 ilustra a metáfora entre este processo de busca descrito do algoritmo e a forma de reprodução dos cucos. A Figura 3.1 representa o fluxograma do algoritmo de busca cuco.

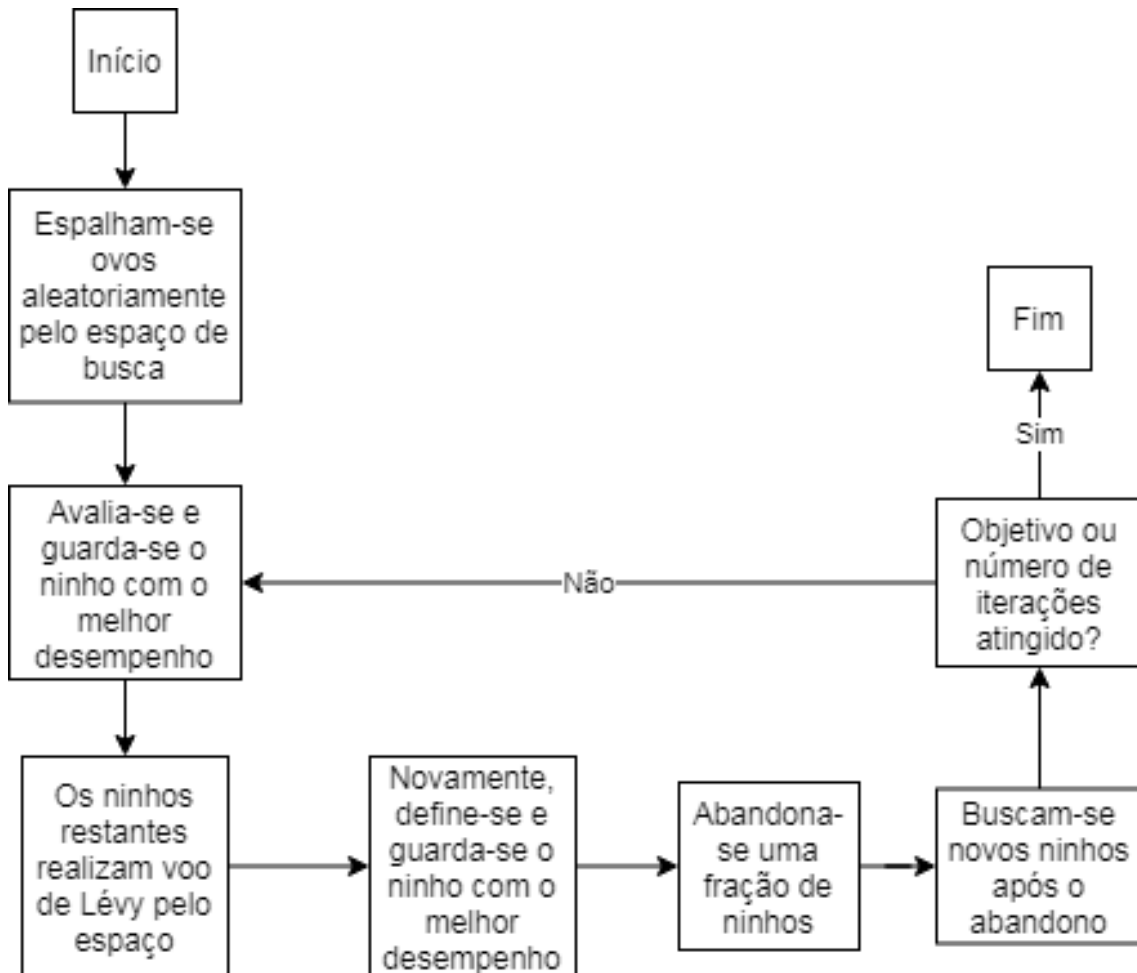


Figura 3.1: Fluxograma do COA [19]



Figura 3.2: Processo de busca do COA [5]

As soluções do problema em questão precisam ter alguma representação em nível computacional. No algoritmo genético, por exemplo, elas são representadas por vetores com valores binários em cada bit, esses vetores são chamados cromossomos, e cada bit representa um gene. No COA, esta representação é feita pelo ninho, que deve possuir a mesma dimensão do espaço de busca. Assim, para um problema de dimensão igual a N , deve haver um vetor $Ninho$ para representar a posição de uma possível solução do problema dentro deste espaço de tamanho N , com cada componente deste vetor sendo um número real. É representado como na Eq. (3.1):

$$Ninho = [x_1, x_2, x_3, \dots, x_i, \dots, x_N] \quad (3.1)$$

onde x_i é a coordenada i de um ninho específico no espaço de busca.

O desempenho de cada ninho é medido através da função objetivo f_O , conforme a Eq. (3.2):

$$f_O(Ninho) = f_O(x_1, x_2, x_3, \dots, x_N) \quad (3.2)$$

Para dar início ao COA, é preciso gerar uma matriz de ninhos $N_{ninhos} \times N$. Neste processo, um cuco i em um tempo t é selecionado e as novas soluções são obtidas quando o voo de Lévy é executado, conforme a Eq. (3.3):

$$Ninho_i^{(t+1)} = Ninho_i^{(t)} + \alpha \times Lévy \quad (3.3)$$

onde Lévy é o passo da distribuição de Lévy e α é um parâmetro de escala cujo valor é dependente do problema em questão, porém na maioria dos casos, pode-se usar α igual a 1 [52].

3.3 Voo de Lévy

O voo de Lévy consiste em um salto aleatório cujo comprimento do passo é dado pela distribuição de Lévy. Pode ser observado em padrões de forrageamento (busca por alimento) de albatrozes, moscas-de-fruta e macacos-aranha e até mesmo humanos, como o conjunto de tribos *Ju/'hoansi* de caçadores-coletores [51].

Na implantação do COA proposta por Yang e Deb, 2009 [52], o voo de Lévy é utilizado para simular o voo do cuco no processo de otimização. A distribuição de Lévy pode ser expressa de forma simplificada pela lei de potência $L(s) \sim u^{-\lambda}$, com $1 < \lambda < 3$, ou definida pela Eq. (3.4):

$$L(s, \gamma, \mu) = \begin{cases} \sqrt{\frac{\gamma}{2\pi}} \exp\left[-\frac{\gamma}{2(s-\mu)}\right] \frac{1}{(s-\mu)^{3/2}}, & 0 < \mu < s < \infty \\ 0 & \text{C.C.}, \end{cases} \quad (3.4)$$

onde γ é um fator de escala que depende do problema a ser tratado e $\mu > 0$ é o mínimo valor possível para um passo. Se $s \rightarrow \infty$, obtém-se um caso de generalização especial da distribuição de Lévy, dada pela Eq. (3.5):

$$L(s, \gamma, \mu) \approx \sqrt{\frac{\gamma}{2\pi}} \frac{1}{s^{3/2}} \quad (3.5)$$

A distribuição de Lévy pode também ser definida como uma transformada de Fourier como representada na Eq. (3.6):

$$F(k) = \exp[-\alpha|k|^\beta], \quad 0 < \beta \leq 2, \quad (3.6)$$

onde α é um parâmetro de escala. Esta equação não possui forma inversa analítica, portanto é preciso que se avalie casos especiais, como quando $\beta = 2$, onde se obtém uma equação cuja inversa é uma distribuição gaussiana, dada pela Eq. (3.7):

$$F(k) = \exp[-\alpha k^2] \quad (3.7)$$

Se $\beta = 1$, obtém-se uma distribuição de Cauchy, conforme as Eqs. (3.8) e (3.9):

$$F(k) = \exp[-\alpha|k|] \quad (3.8)$$

$$p(x, \gamma, \mu) = \frac{1}{\pi} \frac{\gamma}{\gamma^2 + (x - \mu)^2}, \quad (3.9)$$

onde μ é um parâmetro de localização e γ é um fator de escala para controle da distribuição.

Para o caso geral da transformada de Fourier, a inversa da equação é dada pela Eq. (3.10):

$$L(s) = \frac{1}{\pi} \int_0^{\infty} \cos ks \exp[-\alpha|k|] dk \quad (3.10)$$

Quando $s \rightarrow \infty$, obtém-se a Eq. (3.11):

$$L(s) \rightarrow \frac{\alpha\beta\Gamma(\beta) \sin(\pi\beta/2)}{\pi|s|^{1+\beta}}, \quad s \rightarrow \infty, \quad (3.11)$$

onde $\Gamma(z)$ é a função gama definida pela Eq. (3.12):

$$\Gamma(z) = \int_0^{\infty} t^{z-1} \exp(-t) dt \quad (3.12)$$

Para os casos em que $z = n$, com n sendo um número inteiro, faz-se $\Gamma(n) = (n-1)!$.

Há diferentes formas de se realizar o voo de Lévy, e uma dessas formas, tida como uma das mais eficientes, se baseia em uma distribuição simétrica, onde os passos podem ser tanto positivos quanto negativos em torno de um ponto no espaço [34]. A Figura 3.3 representa um caminho característico escolhido em um espaço de duas dimensões, com cinquenta passos.

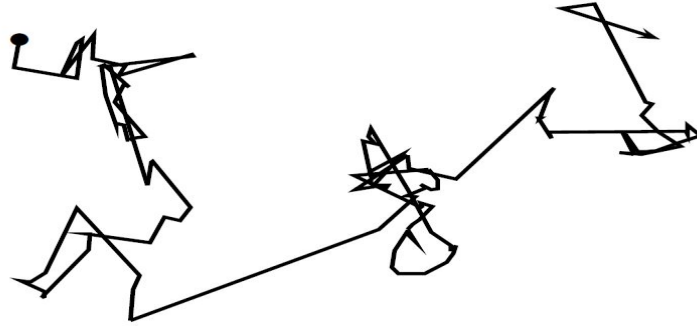


Figura 3.3: Voo de Lévy partindo da origem com 50 passos consecutivos [51]

O comprimento do passo s pode ser calculado pela Eq. (3.13), onde u e v são dados por uma distribuição normal, com $u \sim N(0, \sigma_u^2)$ e $v \sim N(0, \sigma_v^2)$, cujas variâncias σ_u e σ_v são descritas na Eq. (3.14):

$$s = \frac{u}{|v|^{1/\beta}} \quad (3.13)$$

com

$$\sigma_u = \left\{ \frac{\Gamma(1+\beta) \sin \pi\beta/2}{\Gamma[(1+\beta)/2] \beta 2^{(\beta-1)/2}} \right\}^{1/\beta}, \quad \sigma_v = 1 \quad (3.14)$$

O voo de Lévy, em suma, consiste em gerar números aleatórios e escolher uma direção aleatória e gerar o passo a ser dado, também de forma aleatória. Estudos mostram que voos de Lévy podem maximizar a eficiência de recursos de pesquisa em

ambientes incertos, uma vez que são observados, de fato, em comportamentos vistos na natureza [51]. A Figura 3.4 mostra o pseudocódigo que ilustra o funcionamento do algoritmo.

```
Início:  
  Função objetivo:  $f(\mathbf{x})$ ,  $\mathbf{x} = (x_1, \dots, x_n)^T$ ;  
  Gerar a população inicial:  
    Para  $N$  ninhos, gerar  $x_j$ , com coordenadas aleatórias;  
    Guardar  $x_j$  em  $X$ ;  
  Avaliar a função objetivo para cada  $x_j$  em  $X$ ;  
  Definir  $x_{best}$ ;  
While ( $t < \text{Gerações máximas}$ ) ou ( $\text{Critério de parada atendido}$ ):  
  Voo de Lévy é realizado pelos ninhos diferentes do  $x_{best}$ ;  
  Para cada  $x_j$  em  $X$ :  
    Avaliar a função objetivo para cada  $x_j$  em  $X$ ;  
    If ( $f(x_j) < f(x_{best})$ ):  
       $x_{best} = x_j$ ;  
    end  
  Descartar um fração de ninhos ( $p_a$ ) em  $X$ ;  
  Gerar novos ninhos com coordenadas aleatórias em  $X$ ,  
  para os ninhos que foram descartados;  
  Para cada  $x_j$  em  $X$ :  
    Avaliar a função objetivo para cada  $x_j$  em  $X$ ;  
    If ( $f(x_j) < f(x_{best})$ ):  
       $x_{best} = x_j$ ;  
    end If  
end While  
  Pós processamento e visualização dos resultados;  
Fim
```

Figura 3.4: Pseudocódigo do COA [51]

3.4 Aplicação do COA a Funções de Teste

Com o intuito de avaliar a capacidade de otimização do COA e de validação do código, ele foi implantado a três funções de teste, cuja escolha foi baseada em seu amplo uso para fins de validação de algoritmos na literatura. São elas: função Esfera [17], função Rastrigin [44] e função Griewank [49]. Um algoritmo COA, programado na linguagem Python [47], foi desenvolvido para ser aplicado às funções de teste utilizadas neste trabalho.

3.4.1 Função Esfera

A função esfera é simétrica, convexa, unimodal e não apresenta descontinuidades em seu domínio. Por não possuir restrições quanto ao número de variáveis consideradas, sua

complexidade aumenta à medida que o número de parâmetros considerados aumenta. Sua representação na forma analítica é dada pela Eq. (3.15):

$$f(x) = \sum_{i=1}^n x_i^2 \quad (3.15)$$

onde n significa o número de parâmetros da função e representa a dimensão do problema.

Neste trabalho, foram realizados testes para $n = 3, 10$ e 20 dimensões e todos os testes foram realizados com o parâmetro do cuco, probabilidade de abandono $p_a = 0,25, 0,75$ e $0,12$, que são valores já usados na literatura [5]. Para cada dimensão, foram testados 10, 20 e 50 ninhos, e número máximo de gerações igual a 1000. No caso de 20 dimensões, para cada número de ninhos, foram testados os números máximos de gerações de 1000 e 2000. O intervalo de busca utilizado para realizar estes testes foi de $x \in [-50, 50]$. A representação gráfica da função Esfera segue na Figura 3.5:

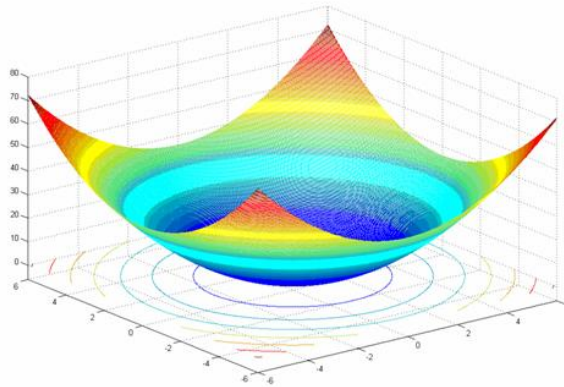


Figura 3.5: Função Esfera [17]

Os resultados para os testes do COA com a função Esfera são apresentados na Tabela 3.1, com o número de ninhos, dimensões, gerações executadas e o menor valor da função *fitness* encontrado para cada caso executado.

Tabela 3.1: Resultados dos testes da aplicação do COA à função Esfera

Dimensão	Probabilidade de Abandono	Ninhos	Gerações Executadas	Fitness
3	0,12	10	343	4,30973E-08
		20	394	4,60712E-08
		50	393	5,03176E-08
10		10	889	8,84988E-08
		20	953	8,69874E-08
		50	1000	1,48121E-07
20		10	1000	2,82681E-05
		20	1000	1,93377E-04
		50	1000	1,91966E-04
3	0,25	10	260	8,96040E-08
		20	211	5,78426E-08
		50	226	2,12870E-09
10		10	571	8,56194E-08
		20	706	8,38705E-08
		50	706	9,72390E-08
20		10	1000	1,39647E-07
		20	1000	8,76507E-07
		50	1000	1,10046E-05
3	0,75	10	134	7,86472E-08
		20	155	7,28451E-08
		50	159	4,78727E-08
10		10	419	8,92087E-08
		20	501	9,71916E-08
		50	573	8,79282E-08
20		10	846	9,33546E-08
		20	970	8,78240E-08
		50	1000	2,48571E-05

3.4.2 Função Rastrigin

A função Rastrigin é baseada na função Esfera acrescida de um módulo cossenoidal, tendo como efeito a produção de vários mínimos locais, o que torna a função Rastrigin não linear e multimodal. Tem a sua representação na forma analítica dada pela Eq. (3.16) e sua representação gráfica dada pela Figura 3.6 Os intervalos de busca utilizados para os testes desta função foram de $x \in [-5, 12, 5, 12]$ e de $x \in [-2, 56, 2, 56]$. A função Rastrigin possui mínimo global zero (0).

Os valores dos resultados para os testes com a função Rastrigin para cada intervalo considerado constam nas Tabelas 3.2 e 3.3.

$$f(x) = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10) \quad (3.16)$$

onde n = número de parâmetros e representa a dimensão do problema.

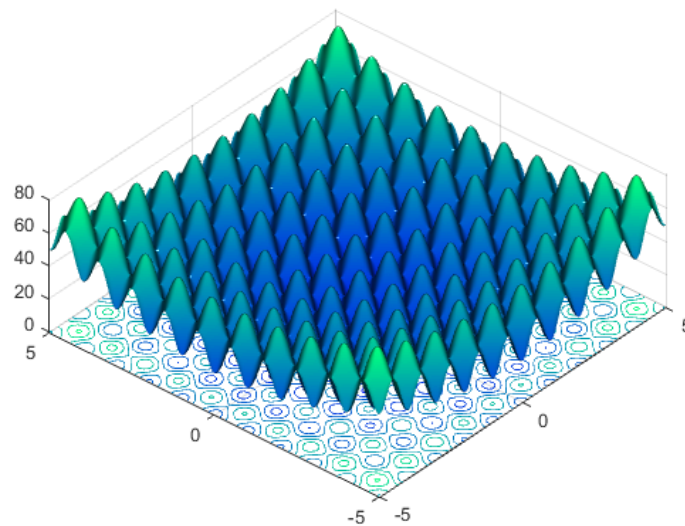


Figura 3.6: Função Rastrigin [44]

Tabela 3.2: Resultados dos testes da aplicação do COA à função Rastrigin para o intervalo $x \in [-5, 12, 5, 12]$

Dimensão	Probabilidade de Abandono	Ninhos	Gerações Executadas	Fitness	
3	0,12	10	639	6,83261E-08	
		20	871	9,55469E-08	
		50	717	2,42686E-08	
10		10	1000	1,00097E+00	
		20	1000	3,31258E-02	
		50	1000	1,48246E-02	
20		10	1000	1,16927E+01	
		20	1000	8,70890E+00	
		50	1000	4,48678E+00	
3		0,25	10	383	5,83874E-08
			20	423	4,21057E-08
			50	451	5,59460E-08
10	10		1000	9,97549E-01	
	20		1000	7,13519E-02	
	50		1000	4,50320E-02	
20	10		1000	2,47880E+01	
	20		1000	2,44268E+01	
	50		1000	1,79812E+01	
3	0,75		10	506	9,54642E-08
			20	460	7,24095E-08
			50	347	4,88773E-08
10		10	1000	1,00162E+01	
		20	1000	1,33163E+01	
		50	1000	4,04455E+00	
20		10	1000	4,16839E+01	
		20	1000	4,12364E+01	
		50	1000	4,22332E+01	

Tabela 3.3: Resultados dos testes da aplicação do COA à função Rastrigin para o intervalo $x \in [-2, 56, 2, 56]$

Dimensão	Probabilidade de Abandono	Ninhos	Gerações Executadas	Fitness	
3	0,12	10	1000	1,97114E-07	
		20	894	9,83818E-08	
		50	730	3,73703E-08	
10		10	1000	2,17111E+00	
		20	1000	1,65897E-02	
		50	1000	2,78821E-03	
20		10	1000	1,01714E+01	
		20	1000	4,31246E+00	
		50	1000	4,60126E+00	
3		0,25	10	526	6,93999E-08
			20	456	8,58608E-09
			50	448	6,90744E-08
10	10		1000	3,99362E-03	
	20		1000	2,64852E-02	
	50		1000	3,32674E-02	
20	10		1000	1,84060E+01	
	20		1000	1,16038E+01	
	50		1000	1,32620E+01	
3	0,75		10	482	7,93678E-09
			20	384	6,60526E-08
			50	586	7,72360E-08
10		10	1000	6,85412E+00	
		20	1000	7,05605E+00	
		50	1000	7,04153E+00	
20		10	1000	2,22157E+01	
		20	1000	2,22168E+01	
		50	1000	2,23992E+01	

3.4.3 Função Griewank

Assim como a função Rastrigin, a função Griewank é convexa, não linear e multimodal, porém apresenta mais pontos de máximo e mínimo locais, o que aumenta a complexidade para se encontrar o seu mínimo global, localizado no ponto zero (0). As suas representações na forma analítica e forma gráfica estão na Eq. (3.17) e Figura 3.7. O espaço de busca utilizado para a realização do teste para esta função foi $x \in [-50, 50]$.

$$f(x) = \sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \quad (3.17)$$

onde n é o número de parâmetros da função.

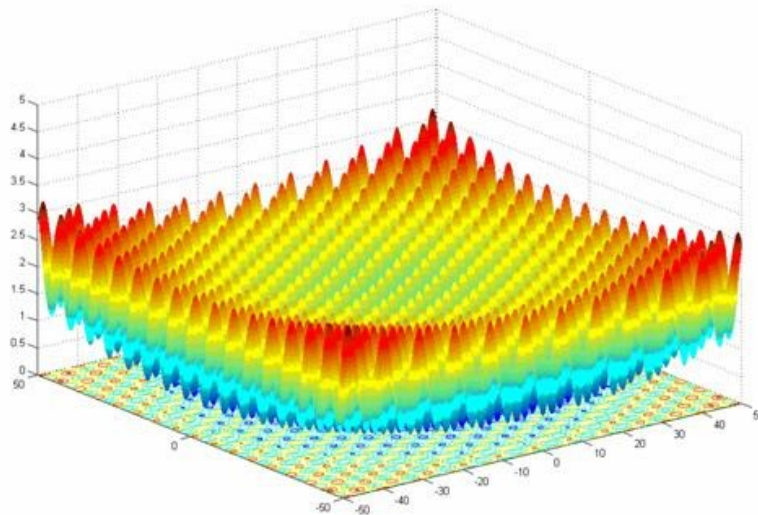


Figura 3.7: Função Griewank [49]

Os valores dos resultados dos testes para a função Griewank estão listados na Tabela 3.4:

Tabela 3.4: Resultados dos testes da aplicação do COA à função Griewank

Dimensão	Probabilidade de Abandono	Ninhos	Gerações Executadas	Fitness	
2	0,12	10	1000	2,92893218818E-01	
		20	1000	2,92893218817E-01	
		36	1000	2,92893218813E-01	
		50	1000	2,92893218813E-01	
3		10	1000	5,9175170977E-01	
		20	1000	5,9175170954E-01	
		36	1000	5,9175170956E-01	
		50	1000	5,9175170969E-01	
5		10	1000	9,10283120E-01	
		20	1000	9,08712908E-01	
		50	1000	9,08712909E-01	
2		0,12	50	1000	9,08712910E-01
	10		1000	2,928932188135E-01	
	20		1000	2,928932188134E-01	
	50		1000	2,928932188134E-01	
	50		1000	2,928932188134E-01	
3	10		1000	5,91751710065E-01	
	20		1000	5,91751709562E-01	
	50		1000	5,91751709537E-01	
	50		1000	5,91751709538E-01	
5	10		1000	9,08712907E-01	
	20		1000	9,08712909E-01	
	50		1000	9,08712907E-01	
2	0,12		50	1000	9,08712909E-01
			10	1000	2,92893218E-01
			20	1000	2,92893218E-01
			50	1000	2,92893218E-01
		50	1000	2,92893218E-01	
3		10	1000	5,9332297474E-01	
		20	1000	5,9175170953E-01	
		50	1000	5,9175170954E-01	
		50	1000	5,9175170953E-01	
5		10	1000	9,08712907E-01	
		20	1000	9,08712915E-01	
		50	1000	9,08712909E-01	
		50	1000	9,08712910E-01	

Na Tabela 3.5 estão listados os menores valores de *fitness* encontrados para cada função testada.

Tabela 3.5: Melhores resultados para cada função de teste

Função de teste	Probabilidade de Abandono	Dimensão	Ninhos	Gerações Executadas	Fitness
Esfera	0,25	3	50	226	2,12870E-09
Rastringin	0,12	3	50	717	2,42686E-08
Rastringin $x \in [-2, 56, 2, 56]$	0,75	3	10	482	7,93678E-09
Griewank	0,12	2	10	1000	2,92893E-01

Os testes realizados mostram que o COA desenvolvido para as funções de teste obteve resultados satisfatórios para as funções estudadas. Pode-se notar que, quanto maior a dimensão da função considerada, maior é a dificuldade do COA para encontrar o ponto de mínimo global, sendo necessária uma pesquisa exaustiva para ajustar os parâmetros do COA, para que ele tenha melhor desempenho na otimização. Ao observar os dados da Tabela 3.5, é possível notar que o melhor resultado obtido com o algoritmo desenvolvido foi para a função Rastringin de 3 dimensões, para o intervalo $x \in [-2, 56, 2, 56]$, usando os parâmetros: probabilidade de abandono igual a 0,75, 10 ninhos e 482 gerações.

Capítulo 4

O COA aplicado ao Problema de Otimização da Recarga

4.1 Aplicação do COA à otimização do ciclo 7 de Angra 1

No modelo de COA desenvolvido para a otimização do ciclo 7 de operação da Usina Angra 1, cada solução candidata do COA é um vetor de números reais, com 20 elementos, onde cada elemento do vetor representa a posição de um elemento combustível (EC) no núcleo do reator de Angra 1, sendo 10 EC de quarteto e 10 EC de octeto. Assim, uma solução do COA representa um padrão de recarga, e um possível candidato a solução do POR. A Figura 4.1 mostra o vetor-solução do COA, onde nas posições de 1 a 10 estão representados os elementos de quarteto e de 11 a 20, os elementos de octeto.

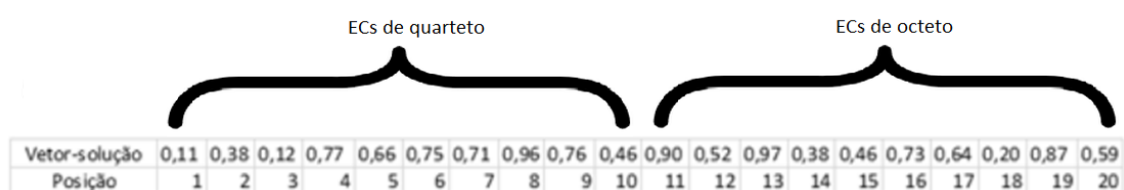


Figura 4.1: Vetor solução do COA

No vetor solução do COA, podem existir elementos com valores iguais, o que seria uma impossibilidade física do ponto de vista do POR, visto que um mesmo EC ocuparia mais de uma posição na configuração de núcleo em questão, o que consiste em uma impossibilidade física. Com o intuito, portanto, de que esta repetição de EC não ocorra durante o processo de otimização, faz-se o uso do Modelo de Chaves Aleatórias, conhecido na literatura como Random Keys, e proposto por Bean, 1994 [6].

O Random Keys tem como função mapear um vetor de números reais em um

vetor-solução de números inteiros sem repetição, o que permite a produção de um padrão de carregamento válido. Este método é realizado da seguinte forma: em uma solução candidata gerada pelo COA de 20 elementos, primeiro localiza-se, independentemente, o elemento de menor valor. A seguir, este elemento é representado em um vetor-solução, de 20 elementos também, utilizando-se o número de sua posição no vetor da solução candidata. Este processo é realizado para o próximo EC de menor valor, e assim por diante. Este processo é realizado de forma separada para os EC de quarteto e para os EC de octeto, com respeito às suas posições no vetor-solução. Dessa forma, uma solução candidata gerada pelo COA não irá apresentar repetição de EC ao se fazer uso do Random Keys. Se, por acaso, houver duas ou mais posições com o mesmo valor, o critério adotado neste trabalho foi o de numerar a posição que aparecer primeiro com o menor valor para vetor produzido após o uso do Random Keys. A Figura 4.2 apresenta o procedimento realizado pelo Random Keys.

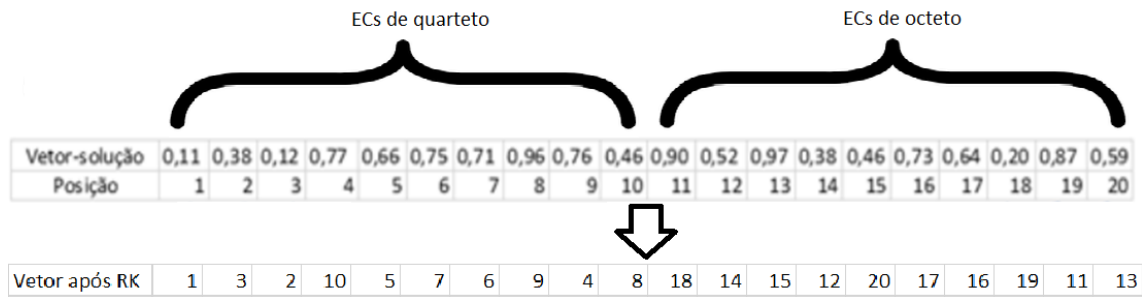


Figura 4.2: Procedimento realizado pelo Random Keys

Após este processo ser realizado pelo Random Keys, o COA irá fornecer uma configuração de núcleo válida, ou seja sem repetição de EC. Porém, é necessário que cada solução candidata do COA seja avaliada por um código nodal de física de reatores.

Neste trabalho, como em outros da literatura [33], [14], [38] e [10], foi usado o código RECNOd, cuja função é realizar cálculos neutrônicos, projetado para uma simetria de $\frac{1}{8}$ de núcleo, com duas dimensões, com os cálculos realizados em duas etapas de queima, sendo estas o início de vida e equilíbrio de xenônio no combustível. Pelo fato do RECNOd não calcular o fator de pico de potência radial F_{XY} , este parâmetro é substituído pela máxima potência média relativa dos EC [8], que substitui o fator de pico de potência radial com erro de $\pm 2\%$. Esse fato, em termos computacionais, é irrelevante, pois o foco é a otimização, e a variação deste parâmetro equivale ao comportamento não linear do fator de pico de potência radial. Em termos físicos, porém, a limitação do valor da potência média relativa a 1,395 para o ciclo 7 de Angra 1, especificamente, garante que o fator de pico de potência radial seja mantido abaixo de 1,435, que representa o valor para o limite estabelecido para as especificações técnicas de Angra 1 [38] e [10].

Os parâmetros avaliados pelo RECNOd são: a concentração de boro (C_B) e a máxima

potência média relativa dos EC (P_{rm}). Neste caso, a P_{rm} substitui o fator de pico de energia e é usado como um parâmetro de segurança, uma vez que, como citado, o RECNOd não calcula o fator de pico de potência radial. No entanto, o uso da P_{rm} implica na não violação das especificações técnicas da central nuclear. Neste trabalho, como em outros da literatura, usou-se o valor limite de $P_{rm} = 1,395$, para o ciclo 7 da usina nuclear Angra 1. Por outro lado, a C_B fornecida pelo RECNOd é dada no equilíbrio do xenônio, o qual é outro aspecto que reduz o custo computacional do processo, sem afetar a validade do propósito da otimização. [8] demonstra que é possível extrapolar e prever o tamanho do ciclo do reator baseado no valor da C_B e no equilíbrio do xenônio.

Neste trabalho, a função objetivo desenvolvida por De Lima, 2005 [14] foi utilizada para avaliar cada solução candidata do COA. Observa-se que essa função leva em consideração dois parâmetros fundamentais: a concentração de boro (C_B) e a máxima potência média (P_{rm}) no EC, parâmetros esses fornecidos pelo RECNOd, como apresentada na Eq. (4.1).

$$Fitness = \begin{cases} \frac{1}{C_B} & \text{se } P_{rm} \leq 1,395 \\ P_{rm} & \text{caso contrário} \end{cases} \quad (4.1)$$

Desta forma, o COA deverá atender a uma função objetivo com duas condições: a priori, deverá minimizar a potência média relativa para que fique dentro do valor delimitado de 1,395, para o ciclo 7 de Angra 1. Assim sendo, se este valor for respeitado pela configuração de núcleo em questão, então o COA deverá maximizar a concentração crítica de boro solúvel, com a finalidade de maximizar o número de dias efetivo a plena potência (DEPP), isto é, aumentar tanto quanto possível a duração do ciclo.

4.2 Resultados da aplicação do COA ao POR

O algoritmo COA desenvolvido para o ciclo 7 de Angra 1, assim como para as funções de teste, foi implantado na linguagem Python. Em resumo, o COA irá gerar uma possível configuração de núcleo (ovo), que será decodificada em uma configuração de núcleo válida por meio do modelo Random Keys. Em seguida, esta configuração será avaliada pelo código RECNOd, e, após isto, seu valor de *fitness* será avaliado pelo COA a partir dos valores retornados pelo RECNOd.

A Tabela 4.1 mostra os resultados obtidos para testes preliminares, com o intuito de determinar a probabilidade de abandono que seria utilizada para que fossem produzidos os demais resultados. Foram escolhidos os parâmetros de número de ninhos igual a 100 e as diferentes probabilidades de abandono mais utilizadas na literatura: 0,12, 0,15, 0,25 e 0,90 [5]. Com base nos dados obtidos, a probabilidade de abandono p_a escolhida para que

o restante dos resultados fosse executado foi de 0,25, uma vez que foi a que apresentou melhor resultado nos testes preliminares.

Tabela 4.1: Resultados do COA aplicado à otimização do problema da recarga para 100 ninhos e diferentes probabilidades de abandono

Ninhos	Probabilidade de Abandono	Gerações	Potência média relativa P_{rm}	Concentração de boro C_B
100	0,12	500	1,385	1311,0
100	0,15	1500	1,389	1383,0
100	0,25	1000	1,370	1397,0
100	0,90	1000	1,395	1281,0

O método desenvolvido foi usado como ferramenta de otimização para determinar os valores de C_B e P_{rm} para o ciclo 7 de Angra 1. A Tabela 4.2 apresenta os resultados encontrados para o POR, onde se variaram os valores dos parâmetros número de ninhos e número de gerações e fixou-se o valor do parâmetro probabilidade de abandono em 0,25, pois esse apresentou melhores resultados na convergência do algoritmo de acordo com os parâmetros do COA utilizados.

Tabela 4.2: Resultados do COA aplicado à otimização do problema da recarga

Ninhos	Probabilidade de Abandono	Gerações	Potência média relativa P_{rm}	Concentração de boro C_B
30	0,25	500	1,394	1336,0
40	0,25	500	1,395	1339,0
50	0,25	500	1,394	1282,0
60	0,25	600	1,382	1335,0
70	0,25	100	1,392	1250,0
70	0,25	200	1,392	1292,0
80	0,25	800	1,387	1339,0
80	0,25	800	1,394	1450,0
80	0,25	500	1,388	1364,0
100	0,25	1000	1,370	1397,0

Pode-se observar, na Tabela 4.2, que o melhor resultado encontrado com o algoritmo COA desenvolvido para o POR foi o valor para a concentração de boro de 1450 ppm. Considerando o fato de que a usina nuclear Angra 1 produz em média 627 MW por dia e, que aproximadamente 4 ppm de boro são consumidos em um dia efetivo a plena potência, pode-se dizer que 1450 ppm de boro correspondem a aproximadamente 362 dias de operação da usina.

4.3 Comparação entre os resultados obtidos pelo COA e resultados de outras meta-heurísticas da literatura

Sendo o POR um problema de grande interesse, tanto econômico quanto acadêmico, tem sido ao longo dos anos estudado amplamente por pesquisadores no Brasil e no mundo. Por isso, existem diversos algoritmos aplicados ao problema, o que o torna um *benchmark* no Brasil [38].

A Tabela 4.3 mostra os resultados das implantações de outras abordagens da literatura, tais como: ACS [32], PBIL_MO [33] e PSO [36]. Os resultados apresentados na Tabela 4.3 mostram os melhores valores encontrados para a C_B , sendo que todos estes algoritmos também foram desenvolvidos para o ciclo 7 de operação de Angra 1, fazendo uso do código RECNOd.

Tabela 4.3: Comparação entre o COA e os algoritmos ACS, PBIL e PSO

Meta-heurística	Concentração de boro C_B	C_B média	Desvio-padrão	Avaliações
Ant-Q	1297	1098	94	200
PBIL_MO	1305	1004	71	10000
ACCN	1424	1350	45	329000
PSORK	1394	1168	95	4000
COA	1450	1340	62	64000

Ao se analisar os dados apresentados na Tabela 4.3, pode-se observar que o COA encontrou um valor de C_B superior a todas as outras abordagens. Este fato demonstra que o algoritmo COA desenvolvido não só é comparável com os outros métodos da literatura, mas que é capaz de encontrar soluções melhores para o problema em questão.

Capítulo 5

Conclusões e Trabalhos Futuros

5.1 Conclusões

O presente trabalho teve como objetivo desenvolver um estudo acerca do desempenho do algoritmo baseado no comportamento dos pássaros da família Cuculidae (COA) com aplicação à otimização do POR, para usinas do tipo PWR. A linguagem de programação Python foi a ferramenta utilizada para o desenvolvimento da metodologia utilizada. Este algoritmo foi implantado e testado em funções de teste já conhecidas na literatura, sendo estas a função Esfera, função Rastrigin e função Griewank, com o intuito de avaliar a convergência do algoritmo desenvolvido bem como de validar a sua aplicação ao POR.

Após isto, o COA foi usado no POR, fazendo uso do código nodal de física de reatores RECNOd. Esta implantação aconteceu através de uma interface que realiza a comunicação entre o RECNOd e o COA. Uma vez que um código de física de reatores demanda um maior tempo computacional para que sejam realizados todos os cálculos necessários, 10 resultados foram produzidos para este trabalho ao se testar o COA no POR. Pode-se observar que para 80 ninhos, probabilidade de abandono de 0,25 e 800 gerações, o algoritmo demonstrou seu melhor desempenho, ou seja, foi capaz de encontrar um valor de concentração de boro igual a 1450 ppm.

O valor de concentração de boro igual a 1450 ppm é capaz de manter o ciclo 7 de operação da usina Angra 1 por, aproximadamente, 362 dias a plena potência, considerando que a cada dia de operação são consumidos 4 ppm de boro. Além disso, se compararmos com o padrão de carregamento manual (955 ppm), ou seja, realizado por especialistas, a metodologia desenvolvida foi capaz de fornecer aproximadamente 123 dias a mais para a duração do ciclo 7 de operação da Usina Angra 1.

Ao ser comparado com outros algoritmos da literatura, o COA demonstrou desempenho superior às abordagens consideradas: Ant-Q, *PBIL_MO*, ACCN e PSORK, sendo capaz de fornecer aproximadamente 6 dias a mais para a duração do ciclo 7 de operação da

Usina Angra 1, quando comparado com o segundo melhor resultado 1424 ppm encontrado pelo algoritmo ACCN. Além disso, podemos destacar o menor esforço computacional do COA, quando comparado ao ACCN, ou seja, o COA encontrou o melhor resultado com 64000 avaliações, enquanto que o ACCN encontrou o melhor resultado com 329000 avaliações. Esse é um fato muito relevante, quando se trata da otimização de problemas complexos.

Cabe ainda ressaltar que a metodologia desenvolvida neste trabalho se mostrou robusta, pois mesmo variando os parâmetros: ninho, probabilidade de abandono e gerações, o COA foi capaz de apresentar valores eficazes para a solução do POR, o que destaca a eficiência do método desenvolvido e, que o COA é uma potencial ferramenta para ser utilizada na otimização de problemas complexos da engenharia nuclear.

5.2 Trabalhos Futuros

A fim de estender os resultados obtidos neste trabalho e aperfeiçoar o método desenvolvido, ficam como sugestões para trabalhos futuros: aplicar o método desenvolvido em plataforma paralela CUDA/GPU, com o intuito de acelerar o processo de otimização, visto que um código de física de reatores demanda maior tempo computacional para sua execução, bem como testar novas *fitness* neste método para o POR, e implantar o método para o COA proposto por Rajabioun, 2011 [43], em que há a possibilidade de haver mais de um ovo para cada ninho.

Referências Bibliográficas

- [1] ABBASS, H. A., 1994, “An Evolutionary Artificial Neural Networks Approach for Breast Cancer Diagnosis”, .
- [2] ABO-ZAHHAD, M., AHMED, S. M., SABOR, N., et al., 1994, “A New Method for Fastening the Convergence of Immune Algorithms Using an Adaptive Mutation Approach”, *Journal of Signal and Information Processing*.
- [3] AUGUSTO, J. P. C., NICOLAU, A. D. S., SCHIRRU, R., 2015, “PSO with Dynamic Topology and Random Keys method applied to Nuclear Reactor Reload”, *Progress in Nuclear Energy (New Series)* 83, pp. 191–196.
- [4] 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*.
- [5] BARBOSA, C. E. M., 2017, *Algoritmos Bio-inspirados para Solução de Problemas de Otimização*. Tese de Mestrado, Centro de Informática/UFPE.
- [6] BEAN, J. C., 1994, “Genetics Algorithms and Random Keys for Sequencing and Optimization”, *ORSA Journal of Computing*.
- [7] BEN HMAIDA, I., CARTER, J., DE OLIVEIRA, C., et al., 1999, “Nuclear In-Core Fuel Management Optimization Using the Tabu Search Method”. In: *Proceedings of the Conference on Mathematics & Computation, Reactor Physics and Environmental Analysis in Nuclear Applications*, p. 1658–1666.
- [8] CHAPOT, J. L. C., 2000, *Otimização Automática de Recargas de Reatores a Água Pressurizada Utilizando Algoritmos Genéticos*. Tese de Doutorado, COPPE/UFRJ.
- [9] CHOI, H., ROH, G., DO, Q., 2006, “Optimal Refueling Pattern Search for a CANDU Reactor Using a Genetic Algorithm”. In: *Proceedings of ICAPP '06*, pp. 2422–2431.
- [10] DA SILVA, M. H., 2014, *Algoritmo Quantum PBIL Autoadaptável aplicado na Otimização da Recarga Nuclear de Reatores PWR*. Tese de Doutorado, COPPE/UFRJ.
- [11] DANTZIG, G. B. “The Nature of Mathematical Programming”. Disponível em: <https://glossary.informs.org/second.php?page=nature.html>.

- [12] DARWIN, C. R., 1985, *On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life*. John Murray.
- [13] DAVIES, G. H. “The Life Style of Birds, Parenthood”. Disponível em: <<http://www.pbs.org/lifeofbirds/home/index.html>>.
- [14] DE LIMA, A. M. M., 2005, *Recarga de Reatores Nucleares Utilizando Redes Conectivas de Colônias Artificiais*. Tese de Doutorado, COPPE/UFRJ.
- [15] DE LIMA, A. M. M., DALVI, D. P., ARAÚJO, E. L., et al., 2019, “PSO with Lévy Flight in the Problem of the Nuclear Reactor Reload”. In: *International Nuclear Atlantic Conference - INAC*.
- [16] DECHAINE, M. D., FELTUS, M. A., 1995, “Nuclear Fuel Management Optimization Using Genetic Algorithms”, *Nuclear Technology*, pp. 109–114.
- [17] DIXON, L. C. W., SZEGO, G. P., 1978, “The global optimization problem: an introduction”, *Towards global optimization 2*, pp. 1–15.
- [18] DORIGO, M., GAMBARDELLA, L. M., 1997, “Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem”, *IEEE Transactions on Evolutionary Computation*, pp. 53–66.
- [19] DOS SANTOS, D., 2019, “O Problema de Identificação de Acidentes em Usinas Nucleares Otimizado pelo Algoritmo Cuco”, *Projeto de Graduação, Escola Politécnica/UFRJ*.
- [20] ELETRONUCLEAR, 1987, *Curso de Formação de Operadores Licenciáveis*.
- [21] ELETRONUCLEAR. “Informações de Angra 1”. Disponível em: <<https://www.eletronuclear.gov.br/Nossas-Atividades/Paginas/Informacoes-de-Angra-1.aspx>>.
- [22] EPE. “Matriz Energética”. Disponível em: <<http://www.epe.gov.br/pt/abcdenergia/matriz-energetica-e-eletrica>>.
- [23] GALPERIN, A., KIMHI, S., SEGEV, M., 1989, “A Knowledge-Based System for Optimization of Fuel Reload Configurations”, *Nuclear Science and Engineering*, pp. 43–53.
- [24] GOLDBERG, D. E., 1989, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Professional.
- [25] INB. “Montagem do Elemento Combustível”. Disponível em: <<http://www.inb.gov.br/Nossas-Atividades/Ciclo-do-combustivel-nuclear/Montagem-do-Elemento-Combustivel>>.
- [26] JOHNSTON, R. L., 2004, *Applications of Evolutionary Computation in Chemistry*. Springer-Verlag.

- [27] KENNEDY, J., EBERHART, R., 1995, “Particle Swarm Optimization”. In: *Proceedings of ICNN'95 - International Conference on Neural Networks*, pp. 1942–1948.
- [28] KIM, H., CHANG, S., LEE, B., 1993, “Optimal Fuel Loading Pattern Design Using an Artificial Neural Network and a Fuzzy Rule-Based System”, *Nuclear Science and Engineering*, pp. 152–163.
- [29] KOBACZY, K. A. H., 2012, “Application of Artificial Intelligence in Maintenance Modelling and Management”. In: *IFAC Proceedings Volumes*, pp. 54–59.
- [30] KROPACZEK, D., TURINSKY, P., 1990, “In-Core Nuclear Fuel Management Optimization for Pressurized Water Reactors Utilizing Simulated Annealing”, *Transactions of the American Nuclear Society*, pp. 9–31.
- [31] LIU, Y. S., 1985, *ANC: A Westinghouse Advanced Nodal Computer Code*. WCAP-10965.
- [32] MACHADO, L., SCHIRRU, R., 2002, “A Otimização Automática da Recarga Nuclear Utilizando o Algoritmo ANT-Q”. In: *International Nuclear Atlantic Conference - INAC*.
- [33] MACHADO, M. D., 2005, *Algoritmo Evolucionário PBIL Multi_Objetivo Aplicado ao Problema a Recarga de Reatores Nucleares*. Tese de Doutorado, COPPE/UFRJ.
- [34] MANTEGNA, R. N., 1994, “Fast, accurate algorithm for numerical simulation of Levy stable stochastic processes”, *Physical Review E*, pp. 4677–4683.
- [35] MENESES, A. A. M., ARAUJO, L. M., NAST, F. N., et al., 2018, “Application of metaheuristics to Loading Pattern Optimization problems based on the IAEA-3D and BIBLIS-2D data”, *Annals of Nuclear Energy*, pp. 329–339.
- [36] MENESES, A. A. M., MACHADO, M. D., SCHIRRU, R., 2009, “Particle Swarm Optimization applied to the nuclear reload problem of a Pressurized Water Reactor”, *Progress in Nuclear Energy*, pp. 319–326.
- [37] MOORE, M., NARAYANAN, A., 1995, “Quantum-inspired computing”, .
- [38] NICOLAU, A. D. S., 2014, *Algoritmo Evolucionário de Inspiração Quântica Aplicado na Otimização de Problemas da Engenharia Nuclear*. Tese de Doutorado, COPPE/UFRJ.
- [39] NICOLAU, A. D. S., SCHIRRU, R., MENESES, A. A. M., 2011, “Quantum evolutionary algorithm applied to transient identification of a nuclear power plant”, *Progress in Nuclear Energy* 53, pp. 86–91.
- [40] NICOLAU, A. D. S., SCHIRRU, R., DE LIMA, A. M. M., 2012, “Nuclear reactor reload using Quantum Inspired Algorithm”, *Progress in Nuclear Energy (New Series)* 55, pp. 40–48.

- [41] PERSHAGEN, B., BOWEN, M., 1989, *Light Water Reactor Safety*. Pergamon.
- [42] POON, P., PARKS, G., 1992, “Optimizing PWR Reload Core Design”, *Parallel Problem Solving from Nature*, pp. 371–380.
- [43] RAJABIOUN, R., 2011, “Cuckoo Optimization Algorithm”, *Applied Soft Computing*, pp. 5508–5518.
- [44] RASTRIGIN, L., 1974, *Systems of extremal control*. Mir, Moscow.
- [45] SILVA, P. V., NAST, F. N., SCHIRRU, R., et al., 2017, “Busca Cuco com Voo de Lévy Aplicado ao Problema de Recarga de Combustíveis em Reatores Nucleares”. In: *International Nuclear Atlantic Conference - INAC*.
- [46] TSOULFANIDIS, N., 2013, *The Nuclear Fuel Cycle*. American Nuclear Society.
- [47] VAN ROSSUM, G., 1991. “Python Software Foundation”. Disponível em: <<https://www.python.org/>>.
- [48] VAROL, H., BINGUL, Z., 2004, “A new PID tuning technique using ant algorithm”. pp. 2154–2159.
- [49] WEISSTEIN, E. W., 2020. “Griewank Function”. Disponível em: <<http://mathworld.wolfram.com/GriewankFunction.html>>.
- [50] WU, C. C., LAI, K. C., SUNA, R. Y., 2008, “GA-based job scheduling strategies for fault tolerant grid systems”. In: *IEEE Asia-Pacific Services Computing Conference*, p. 27–32.
- [51] YANG, X. S., 2010, *Nature-Inspired Metaheuristic Algorithms*. Luniver Press.
- [52] YANG, X., DEB, S., 2009, “Cuckoo Search via Lévy Flights”. In: *Proceeding of World Congress on Nature & Biologically Inspired Computing (NaBIC 2009)*, pp. 210–214.
- [53] ZHOU YANGPING, Z. B., DONGXIN, W., 2000, “Application of genetic algorithms to fault diagnosis in nuclear power plants”, *Reliability Engineering & System Safety*, pp. 153–160.