

INSTITUTO DE PESQUISAS ENERGÉTICAS E NUCLEARES Autarquia associada à Universidade de São Paulo

Otimização de parâmetros do núcleo do reator IEA-R1 utilizando o algoritmo de enxame de partículas.

THIAGO AUGUSTO DOS SANTOS

Tese apresentada como parte dos requisitos para obtenção do grau de Doutor em Ciências na área de Tecnologia Nuclear – Reatores

Orientador: Prof. Dr. Frederico Antônio Genezini

São Paulo 2023

INSTITUTO DE PESQUISAS ENERGÉTICAS E NUCLEARES Autarquia associada à Universidade de São Paulo

Otimização de parâmetros do núcleo do reator IEA-R1 utilizando o algoritmo de enxame de partículas

Versão Corrigida Versão Original disponível no IPEN

THIAGO AUGUSTO DOS SANTOS

Tese apresentada como parte dos requisitos para obtenção do grau de Doutor em Ciências na área de Tecnologia Nuclear – Reatores

Orientador: Prof. Dr. Frederico Antônio Genezini

São Paulo 2023 Autorizo a reprodução e divulgação total ou parcial deste trabalho, para fins de estudo e pesquisa, desde que citada a fonte.

Como citar:

SANTOS, T.A. Otimização de parâmetros do núcleo do Reator IEA-R1 utilizando o algorítmo de enxame de partículas. 2023. 117 p. Tese (Doutorado em Tecnologia Nuclear), Instituto de Pesquisas Energéticas e Nucleares, IPEN-CNEN/SP, São Paulo. Disponível em: (data de consulta no formato: dd/mm/aaaa).

Ficha catalográfica elaborada pelo Sistema de geração automática da Biblioteca IPEN/USP,com os dados fornecidos pelo(a) autor(a)

dos Santos, Thiago Augusto Otimização de parâmetros do núcleo do reator IEA-R1 utilizando o algoritmo de enxame de partículas/ Thiago Augusto dos Santos; orientador Frederico Antônio Genezini. -- São Paulo, 2023. 117 p. Tese (Doutorado) - Programa de Pós-Graduação em Tecnologia Nuclear (Aplicações) -- Instituto de Pesquisas Energéticas e Nucleares, São Paulo, 2023. 1. Otimização de recarga. 2. Enxame de partículas. 3.Reatores de pesquisa. I. Genezini, Frederico, orient. II. Título.

FOLHA DE APROVAÇÃO

Thiago Augusto dos Santos. Otimização de parâmetros do núcleo do reator IEA-R1 utilizando algoritmo de enxame de partículas. 2023.

Tese apresentada ao Programa de Pós-Graduação em Tecnologia Nuclear – Reatores da Universidade de São Paulo para obtenção do título de Doutor em Ciências.

Data:___/__/____ Banca Examinadora Prof. Dr. _____ Instituição: Julgamento: Assinatura: Prof. Dr. Instituição:_____Julgamento:_____ Assinatura: Prof. Dr. _____ Instituição:_____Julgamento:_____ Assinatura:_____ Prof. Dr. _____ Instituição: Julgamento: Assinatura:

A Alvaro dos Santos

AGRADECIMENTOS

Antes de mais nada, é fundamental dizer que essa tese foi escrita com muito mais do que duas mãos. Sem todas elas eu nunca teria chegado até aqui.

Ao meu orientador, Prof. Dr. Frederico Antônio Genezini por todo suporte, conversas e apoio. Agradeço pela confiança e paciência.

Ao Prof. Dr. Giovanni Laranjo di Stefanni, por toda a ajuda.

À Coordenação de Aperfeiçoamento de Pessoal de Nivel Superior (CAPES) e a ELETRONUCLEAR, pelo suporte financeiro.

Aos professores Roberto Schirru e Alan Miranda Monteiro de Lima da COPPE/UFRJ, por toda presteza e ajuda sempre que necessitei.

À Bruna Roque da Silva, que sempre me ajudou em todos os perrengues burocráticos no IPEN.

A Waldemir Gutierrez e Antônio Carlos Iglesias Rodrigues pelas muitas dúvidas esclarecidas sobre 2DB, CITATION e COBRA.

Aos Luizes, Fernandos, Marcos e todo pessoal de Guarulhos.

Ao pessoal do Ipiranga, que têm a audácia de me chamar de professor.

A todos da GV.

À patota.

Ao povo do IFUSP.

Aos meus sogros, João e Terezinha. Aos meus cunhados Gabi e Iuri, que me deram o prazer de ser tio do Fernando.

A todos do Remanso.

Aos meus companheiros de grupo. Boas 24 horas.

À Silvia, por todas as broncas, conselhos e apoios. Além de me mostrar que minha história não é o meu destino.

À Sonia, minha mãe e Dona Maria, minha vó. Apenas obrigado não é suficiente, mas não vou conseguir muito mais do que isso, como sempre.

À Renata, que como essa tese, não sei como chegou até aqui. Mas chegou. Te amo.

Ao meu pequeno Chico, por quem tento ser alguém melhor todos os dias.

RESUMO

SANTOS, T.A. *Otimização de parâmetros do núcleo do reator IEA-R1 utilizando algorítmo de enxame de partículas*. 2023. 111p. Tese (Doutorado em Tecnologia Nuclear) – Instituto de Pesquisas Energéticas e Nucleares – IPEN-CNEN/USP. São Paulo.

O problema da otimização de recarga (POR) de um reator nuclear consiste na determinação de uma configuração do núcleo tal que, respeitados os parâmetros de projeto e segurança, os valores obtidos sejam os melhores possíveis para determinadas grandezas estudadas. Em decorrência de sua não-linearidade e de um grande número de soluções possíveis (na ordem de n!, onde n é o número de elementos combustíveis presentes no núcleo), trata-se de um problema do tipo NPdifícil. Este trabalho buscou resolver o POR para o reator nuclear de pesquisa IEA-R1, localizado na cidade universitária, em São Paulo. Para solucioná-lo, foi criado um código em ambiente FORTRAN 90, que por meio da meta-heurística de enxame de partículas (*Particle Swarm Optimization – PSO*) integrada aos programas utilizados na metodologia de cálculos neutrônicos e termohidráulicos, com a finalidade de encontrar a melhor solução possivel, dentro de uma combinação de casos e particulas para o código. Para garantir a inexistência de duplicidade de resultados, o método das chaves aleatórias (Random Keys – RK) foi utilizado para codificar cada posição dos elementos combustíveis. Uma função objetivo foi determinada a partir das seguintes grandezas: a constante de multiplicação k (K_{EFF}) e o fluxo de nêutrons (ambas maximizadas), o pico de potência e a variância da distribuição da densidade de potência (ambas minimizadas). Foi feita também a busca pela configuração com a menor variância, com o intuito de garantir uma distrbuição de potência mais estável ("Flat") ao longo do núcleo. Foram estudados casos com um número p de partículas tal que p = 20,30,40,50 e t iterações, sendo t = 500, 1000, 1500. Em um primeiro momento todos os casos são estudados e as duas melhores configurações (p = 30, t = 1500 e p = 50, t = 1500) são escolhidas e comparadas com a configuração 260. O código se mostrou capaz de determinar melhores configurações dentro dos parâmetros estabelecidos e respeitando os limites de temperatura.

Palavras chave: Otimização de recarga; reatores de pesquisa; inteligência de enxame

ABSTRACT

SANTOS, T.A. 111p. Optimization of IEA-R1 reactor core parameters using particle swarm algorithm. 2022. (Doutorado em Tecnologia Nuclear) – Instituto de Pesquisas Energéticas e Nucleares – IPEN-CNEN/USP. São Paulo.

The reload optimization problem (ROP) of a nuclear reactor consists of determining a core configuration such that, respecting the design and safety parameters, the values obtained are the best possible for certain quantities studied. Due to its non-linearity and a large number of possible solutions (on the order of n!, where n is the number of fuel elements present in the core), this is a NP-hard problem. This work sought to solve the POR for the research nuclear reactor IEA-R1, in the University of São Paulo campus, in São Paulo. To solve it, a code was created in a FORTRAN 90 environment, which through the particle swarm meta-heuristic (Particle Swarm Optimization - PSO) integrated into the programs used in the methodology of neutronic and thermohydraulic calculations, with the purpose of finding the best possible solution, within a combination of cases and particles for the code. To ensure the absence of duplication of results, the Random Keys (RK) method was used to create an identification code for, to labeling each position of the fuel elements. An objective function was determined from the following quantities: the multiplication constant k (K_{EFF}) and the neutron flux (both maximized), the peak power and the variance of the power density distribution (both minimized). A search was also made for the configuration with the lowest variance, in order to guarantee a more stable power distribution ("Flat") along the core. Cases were studied with a number p of particles such that p = 20,30,40,50 and t iterations, where t = 500, 1000, 1500. At first, all cases are studied, and the two best configurations (p = 30, t = 1500 and p= 50, t = 1500) are chosen and compared with the 260 configurations. The code proved to be able to determine better configurations within the established parameters, respecting the temperature limits.

Keywords: Loading pattern optimization; research reactors; swarm intelligence.

SUMÁRIO

I INTRODUÇÃO 18
1.1 Justificativa
1.1.1 Contexto da produção de radioisótopos no Brasil
1.1.2 O IEA-R1
1.1.3 Utilização de meta-heurísticas21
1.2 Objetivos22
1.3 Aspectos relevantes e originalidade23
1.4 Estrutura do trabalho24
II ESTADO DA ARTE25
2.1 A resolução do POR por meio de meta-heurísticas
2.2 O POR em reatores de pesquisa28
2.3 A utilização do PSO para a resolução do POR
III FUNDAMENTAÇÃO TEÓRICA
3.1 Otimização: Conceitos preliminares
3.2 Descrição matemática
3.2.1 Definições
3.3 Otimizações Mono-objetivo e Multiobjetivo
3.4 Otimização combinatória 40
3.5 Métodos de resolução 43

3.5.1 Heurísticas43
3.5.2 Meta-Heurísticas44
3.6 Algoritmos bioinspirados45
3.6.1 Algoritmos evolucionários47
3.6.2 Algoritmos de enxame48
3.6.2.1 Conceitos básicos48
3.6.2.2 Principios biológicos da inteligência de enxame
3.6.2.3 Principais algoritmos51
3.7 Conceitos básicos de Física de Reatores e Termohidráulica54
3.7.1 K _{EFF} 54
3.7.2 Fluxo de nêutrons55
3.7.3 Transferência de calor56
№ METODOLOGIA
4.1 Otimização por enxame de partículas - PSO60
4.1.1 Definições básicas60
4.1.2 Bases sociocognitivas61
4.1.3 Descrição do algoritmo62
4.1.4 O modelo de chaves aleatórias (Random Keys) aplicado no PSO67
4.2 Metodologia do grupo de cálculo do reator IEA-R168
4.2.1 2DB
4.2.2 CITATION
4.2.3 COBRA
4.2.4 Descrição da metodologia72

V O CÓDIGO DE OTIMIZAÇÃO73					
5.1 Descrição do código73					
5.2 A função objetivo75					
5.3 Função de teste					
5.4 Resolução do POR utilizando o código de otimização					
5.4.1 Estudo da função objetivo79					
5.4.2 Características das configurações obtidas					
<i>5.4.2.1 Características das configurações obtidas: Casos com 20 partículas</i> 82					
<i>5.4.2.2 Características das configurações obtidas: Casos com 30 partículas</i> 85					
<i>5.4.2.3 Características das configurações obtidas: Casos com 40 partículas</i> 87					
<i>5.4.2.4 Características das configurações obtidas: Casos com 50 partículas</i> 89					
<i>5.4.2.5 Características das configurações obtidas: Conclusões preliminares</i> 92					
5.4.4 Estudo das melhores configurações obtidas pelo código de otimização . 93					
<i>5.4.4.1 Caso 30 partículas e 1500 iterações</i> 94					
<i>5.4.4.2 Caso 50 partículas e 1500 iterações</i>					
VI CONCLUSÕES E COMENTÁRIOS103					
Referências Bibliográficas105					
ANEXO I: Manual de utilização do código de otimização113					

LISTA DE FIGURAS

Figura 1.1 – Representação esquemática do núcleo do reator IEA-R120
Figura 2.1 – Representação do núcleo do reator DNRR 29
Figura 2.2 – Posições dos EC no núcleo do DNRR e no cromossomo 29
Figura 2.3 – Estrutura global da estratégia de otimização realizada por MAZROU e HAMADOUCHE (2006)
Figura 2.4 – Esquema do núcleo do reator PARR-1
Figura 3.1 — Representação gráfica de uma maximização e uma minimização de funções opostas
Figura 3.2 – Conjunto de soluções factíveis, espaço objetivo factível e o grau de dominância em um problema de minimização com dois objetivos
Figura 3.3 – Diagrama das diferentes classificações de meta-heurísticas
Figura 3.4 – Tipos de algoritmos bioinspirados 46
Figura 3.5 – Forrageamento de uma colônia de formigas
Figura 4.1 – Fluxograma representando o funcionamento do PSO original
Figura 4.2 – Representação da movimentação de uma partícula no PSO 59
Figura 4.3 – Comparação da diversidade do enxame de partículas utilizando o termo de inércia ou não
Figura 4.4 – Vetores no RK: (a) Vetores posição e velocidade para um espaço de busca de 5 dimensões. (b) Novas posições como chaves aleatórias. (c) Vetor auxiliar contendo uma solução válida
Figura 4.5 – Diagrama de cálculos neutrônicos e termohidraúlicos no IEA-R1 64
Figura 5.1 – Diagrama de fluxo do código de otimização por PSO

Figura 5.2 – Interface da metodologia de cálculo do IEA-R1
Figura 5.3 – Comportamento da função objetivo em relação ao número de iterações para 20, 30, 40, 50, 60 e 70 partículas
Figura 5.4 – Comportamento da função objetivo em relação ao número de iterações para 20, 30, 40 e 50 partículas
Figura 5.5 – Comportamento da função objetivo em relação ao número de partículas para 500, 1000, 1500 iterações
Figura 5.6 – Comportamento do K _{EFF} decorrente da queima do combustível para 20 partículas
Figura 5.7 – Esquema de posicionamento dos elementos no núcleo do IEA-R1 83
Figura 5.8 – Posições dos EC com maior valor de pico para 20 partículas
Figura 5.9 – Comportamento do K _{EFF} decorrente da queima do combustível para 30 partículas
Figura 5.10 – Posições dos EC com maior valor de pico para 30 partículas
Figura 5.11 – Comportamento do K _{EFF} decorrente da queima do combustível para 40 partículas
Figura 5.12 – Posições dos EC com maior valor de pico para 40 partículas
Figura 5.13 – Comportamento do K _{EFF} decorrente da queima do combustível para 50 partículas
Figura 5.14 – Posições dos EC com maior valor de pico para 50 partículas
Figura 5.15 – Comparação entre as configurações 260: a) Original. b) Calculada com 30 partículas e 1500 iterações
Figura 5.16 – Desempenho da função objetivo no POR comparado a configuração 260 do reator IEA- R1 para o caso p = 30, t = 1500
Figura 5.17 – Distribuição da densidade de potência em 2D e 3D para a configuração 260 (superior) e a calculada (inferior)
Figura 5.18 – Comparação entre as distribuições de temperatura no EC de maior densidade de potência nas configurações 260 e calculada com 30 partículas e 1500 iterações e a temperatura limite
Figura 5.19 – Comparação do Comportamento do K _{EFF} decorrente da queima do combustível entre as configurações 260 e calculada para 30 partículas e 1500 iterações
Figura 5.20 – Comparação entre as configurações 260: a) Original. b) Calculada com 50 partículas e 1500 iterações
Figura 5.21 – Desempenho da função objetivo no POR comparado a configuração 260

LISTA DE TABELAS

Tabela 3.1 – Tipos de Problemas de Otimização
Tabela 5.1 – Resolução para o problema da esfera com diferentes valores de partículas e interações
Tabela 5.2 – Valores da função objetivo para o caso de 30 partículas e 1500 iterações com sementes distintas
Tabela 5.3 – Valores da função objetivo e as grandezas que a compõem para casos com 20 partículas
Tabela 5.4 – Trocas de posição de EC nos casos de 20 partículas
Tabela 5.5 – Temperaturas máximas no EC de maior densidade de potência e a comparação com a temperatura limite do material do revestimento nos casos de 20 partículas
Tabela 5.6 – Valores da função objetivo e as grandezas que a compõem para casos com 30 partículas
Tabela 5.7 – Trocas de posição de EC nos casos de 30 partículas
Tabela 5.8 – Temperaturas máximas no EC de maior densidade de potência e a comparação com a temperatura limite do material do revestimento nos casos de 30 partículas
Tabela 5.9 – Valores da função objetivo e as grandezas que a compõem para casos com 40 partículas
Tabela 5.10 – Trocas de posição de EC nos casos de 40 partículas
Tabela 5.11 – Temperaturas máximas no EC de maior densidade de potência e a comparação com a temperatura limite do material do revestimento nos casos de 40 partículas
Tabela 5.12 – Valores da função objetivo e as grandezas que a compõem para casos com 50 partículas
Tabela 5.13 – Trocas de posição de EC nos casos de 50 partículas
Tabela 5.14 – Temperaturas máximas no EC de maior densidade de potência e a comparação com a temperatura limite do material do revestimento nos casos de 50 partículas

I INTRODUÇÃO

1.1 Justificativa

1.1.1 Contexto da produção de radioisótopos no Brasil

Os Reatores de pesquisa possuem um papel importante no desenvolvimento da energia nuclear voltada a fins pacíficos. Começaram a ser construídos logo após o final da Segunda Guerra Mundial e têm contribuído até hoje com a produção de radioisótopos para uso na medicina, indústria e agricultura, bem como no treinamento e formação de operadores de reatores, além de uma importante ferramenta na pesquisa científica.

No Brasil, o convênio realizado entre o CNPq e a USP em 1956 no qual estabeleceu a criação do Instituto de Energia Atômica (IEA) foi pioneiro, permitindo, dentre outras coisas, a produção de radioisótopos no país (SANTOS-OLIVEIRA e CARNEIRO-LEÃO,2008). Os trabalhos deram inicio em 1959, com a produção de Iodo-131 voltado à medicina e em 1963 a produção rotineira de radioisótopos foi iniciada no IEA-R1, que se trata de um reator nuclear de pesquisa do tipo piscina aberta tendo água como moderador e refrigerante. Atualmente é, dentre os quatro reatores de pesquisa existentes no Brasil, o de maior potência e um dos mais antigos ainda em funcionamento no planeta, localizado no atual Instituto de Pesquisas Energéticas e Nucleares (IPEN - antigo IEA), na cidade universitária, em São Paulo (MAIORINO,1999).

1.1.2 O IEA-R1

Apesar de ter seu núcleo inicialmente projetado para operar ao máximo de 5 MW, até 1961 não havia um regime definido de operação para o reator IEA-R1 e sua potência variou entre 200 kW e 2 MW (RODRIGUES,2016). A partir de 1961, o reator passou a operar a uma potência definida de 2 MW. Entre 1971 e 1991, várias modificações foram sendo introduzidas no reator para adequar as suas instalações às normas de segurança mais recentes. Em 1993, passou a operar em ciclos de 64 horas contínuas semanais a 3,5 MW, permitindo assim irradiar materiais com fluxos de nêutrons térmicos de até 8 x10¹³ n.cm⁻²s⁻¹. O reator passou então por diversas reformas e modernizações que, concluídas em setembro de 1997, permitiram aumentar a potência máxima de operação para 5 MW (TERREMOTO, 2004).

Atualmente é utilizado para as seguintes finalidades: produção de radioisótopos para uso em medicina nuclear e na indústria; produção de fontes radioativas para gamagrafia industrial; irradiação de amostras para a realização de análises multielementares; pesquisas em física nuclear, serviços em neutrongrafia; treinamento de pessoal para licenciamento na operação de reatores. Tamanha versatilidade faz do IEA-R1 um dos grandes pontos de aplicação e pesquisa nuclear no Brasil (RODRIGUES, 2016).

Seu núcleo é montado sobre uma placa motriz com 80 posições formando um paralelepípedo de 8x10, conforme mostrado na figura 1.1, sendo composto por elementos combustíveis (EC), elementos de irradiação e refletores encaixados verticalmente em furos existentes na placa matriz, que se encontra suspensa por uma estrutura de treliças de alumínio preso à uma ponte móvel.

A primeira configuração do núcleo IEA-R1 foi um arranjo 5x6 utilizando Urânio de Baixo Enriquecimento (LEU) de liga U-Al com 1,9 g/cm³ de U (a primeira carga produzida por Babcock & Wilcocx), mas pouco depois estes exibiram corrosão e foram trocados. Em 1968 os elementos combustíveis foram alterados para do tipo placa plana com 93 % de enriquecimento e com 0,6 g/cm³ de U, também fabricados com uma liga U-Al (produzida pela UNC/EUA e NUKEM/Alemanha).

No início dos anos 80 com a conversão do núcleo de alto enriquecimento (HEU) para LEU, cinco elementos (produzidos pela NUKEM com liga U-Al_x - 19,75%) foram colocados na matriz do núcleo do IEA-R1. Ao mesmo tempo, o IPEN começou a desenvolver seus próprios elementos combustíveis. Em 1985 dois protótipos (U₃O₈-Al, 1,9 g/cm³ de U – 19,75%) foram examinados no IEA-R1 e em 1988 os primeiros elementos combustíveis brasileiros foram colocados na matriz de núcleo IEA-R1. Após esta primeira produção de 18 elementos combustíveis em 1997, a densidade de U foi aumentada para 2,3 g/cm³, e a partir de 1998 o IEA-R1 passou a utilizar elementos combustíveis de U₃Si₂ com densidade de 3,0 g/cm³ até a

substituição completa do U₃O₈-Al. O elemento combustível padrão atual é composto por 18 placas com cerca de 10g de ²³⁵U cada, e os elementos combustíveis de controle são compostos por 12 placas no centro e dois canais nas pontas (GENEZINI et al, 2019).

Dada a sua constante utilização, a troca periódica do combustível é um problema relevante a ser considerado. Como um de seus fatores limitantes, recomenda-se que a taxa máxima de queima não ultrapasse 40%, o que exige um monitoramento constante da mesma.

Para realizar tal controle, a metodologia de cálculos do IEA-R1 é demasiadamente complexa, necessitando de vários programas que executam cálculos distintos, desde a obtenção da seção de choque até as temperaturas registradas no núcleo. Tais programas, apesar de agirem separadamente, fornecem grandezas que estão intimamente relacionadas umas com as outras.

Figura 1.1 – Representação esquemática do núcleo do reator IEA-R1



Fonte: DOMINGOS,2010

Adicionalmente, nos últimos anos o IPEN tem sofrido severa restrição de recursos humanos, estando há mais de uma década sem reposição do quadro de funcionários e com a quantidade de aposentadorias aumentando a cada ano. Deste modo, qualquer facilitação de tarefas rotineiras, a escolha de nova configuração

com a troca por um novo elemento combustível quando a queima limite do combustível for atingida é de suma importância para o bom andamento das atividades no reator.

Essa nova configuração é guiada por alguns parâmetros, como o fluxo no reator com a maior uniformidade possível ou a minimização da queima de combustível, e por alguns limites, como não inserir novos combustíveis na região central do reator.

1.1.3 Utilização de meta-heurísticas

Entretanto, para solucionar as questões levantadas anteriormente, ou seja, a minimização da queima do combustível e a maximização da uniformidade do fluxo de nêutrons, faz-se necessário a utilização de um método que garanta a determinação de uma configuração ótima para o núcleo, uma vez que a experiência de especialistas é limitada pela imensa possibilidade de configurações possíveis, já que que o núcleo possui 20 EC e rearranjá-los de maneira a testar todas as configurações possíveis levaria a um cálculo de 20! ($20! \sim 2,4x10^{18}$) possibilidades. Do ponto de vista temporal, se cada configuração fosse calculada em 1 segundo, seriam necessários aproximadamente 77 bilhões de anos para todas as configurações do reator IEA-R1 serem devidamente calculadas.

Um problema como esse é conhecido como Problema de Otimização de Recargas (POR), que justamente busca a melhor configuração possível para o núcleo de um reator nuclear, atendendo a todas as necessidades e limitações previamente impostas, por exemplo, como o máximo fator de pico de potência radial (OLIVEIRA,2013).

Tal problema assemelha-se ao conhecido "problema do caixeiro viajante" (TSP), que consiste na procura de um trajeto que possua a menor distância, começando em certa cidade, dentre várias e retornando a esta após visitar todas as outras cidades apenas uma vez executando o menor caminho possível (NILSSON, 1982).

Seja o conjunto C que denota o número de caminhos possíveis, este aumenta exponencialmente com o avanço de n, o número de cidades, podendo ser

definido conforme a equação 1.1 (NILSSON, 1982):

$$C = \frac{(n-1)!}{2} \approx \frac{1}{2} \sqrt{2\pi(n-1)} \left(\frac{n-1}{e}\right)^{n-1}.$$
 (1.1)

Problemas como esses são do tipo denominado NP (Non deterministic polynomial time) e estão dentro da área da Otimização Combinatória, ou seja, suas variáveis assumem valores discretos e procuram por um inteiro, um conjunto, uma permutação ou até um grafo (FERREIRA, 2018).

Durante décadas, problemas de otimização combinatória como o POR só podiam ser resolvidos de forma manual, contando apenas com a experiência de especialistas. Após a criação de processadores mais rápidos e eficientes, por volta da década de 80 do século passado, as técnicas de otimização começaram a se tornar uma ferramenta na busca por melhores soluções.

Por sua vez, a utilização de técnicas convencionais de otimização esbarravam na alta complexidade desse tipo de problema, o que levou a implantação de modelos de busca de soluções, conhecidas como *heurísticas* e, posteriormente *metaheurísticas*. Ao longo da década de 90, trabalhos mostraram a eficiência de metaheurísticas como a dos Algoritimos Genéticos (GA) e do Recozimento Simulado (SA) (BAPTISTA, 2009).

Com base nas colocações anteriores, verifica-se que a resolução do POR para o IEA-R1 por meio de alguma solução meta-heurística é um caminho possível para a obtenção de novas configurações que permitam atingir os objetivos atuais do reator.

1.2 Objetivos

O objetivo principal deste trabalho é a resolução do POR para o núcleo do reator IEA-R1 a partir da criação de um código computacional que se integre com os códigos que compõem a metodologia do cálculo realizado no Centro do Reator de Pesquisa do Instituto de Pesquisas Energéticas e Nucleares (CERPq – IPEN) utilizando uma meta-heurística adequada.

O código de otimização confeccionado neste trabalho foi feito em FORTRAN

90, uma vez que os demais códigos pertencentes à metodologia de cálculos neutrônicos e termo hidráulicos do IEA-R1 também estão nesta linguagem de programação. Para este trabalho foi escolhido o método de Otimização por Enxame de Partículas (*Particle Swarm Optimization* – PSO). As justificativas para a utilização desta meta-heurística serão feitas no decorrer da tese. Deste modo, pode-se enumerar os objetivos específicos deste trabalho:

 Acoplar uma meta-heurística dentro da rotina de cálculos neutrônicos e termo hidráulicos do reator IEA-R1 visando estabelecer uma configuração ótima para o posicionamento dos EC.

• Verificar os parâmetros neutrônicos e termo hidráulicos para a configuração padrão determinada pelo grupo de cálculo do reator IEA-R1.

 Encontrar uma configuração que otimize os parâmetros encontrados para a configuração padrão.

1.3 Aspectos relevantes e Originalidade.

O presente estudo tem grande relevância pois resulta na simplificação da metodologia de cálculo do núcleo, o que contribui tanto para a rotina atual quanto para o trabalho de futuros pesquisadores que trabalharão no CERPq. Além disso, o trabalho contribui com estudos e informações sobre novas configurações possíveis para o núcleo do reator IEA-R1, além de possibilitar uma ampliação de suas capacidades, uma vez que configurações novas e mais adaptadas às necessidades do reator poderão ser encontradas.

Apesar de existirem outros estudos sobre o núcleo do reator IEA-R1 (JOÃO, 2016), a utilização de meta-heurísticas para a resolução de problemas deste tipo uma novidade dentro do IPEN. Este trabalho também pretende ser uma base para que outros pesquisadores façam estudos relativos ao POR no reator IEA-R1 utilizando outras meta-heurísticas.

Do ponto de vista da originalidade do trabalho, ressalta-se o fato do POR ser resolvido em um reator de pesquisa com um código que, feitas as devidas alterações, poderá ser utilizado em qualquer reator do tipo.

1.4 Estrutura do trabalho.

A tese está estruturada da seguinte maneira: No capítulo 2 é apresentada uma pesquisa do estado da arte relativo ao POR. Um levantamento a cerca dos trabalhos realizados sobre o assunto disponíveis na literatura foi feito, de maneira a ser feita uma breve descrição dos mesmos, evidenciando suas particularidades e permitindo uma melhor compreensão da proposta, originalidade e contribuição deste trabalho.

O capítulo 3 faz uma discussão sobre a estrutura matemática do problema, partindo do conceito de otimização e definindo o POR como um problema de otimização combinatória, tendo na sua função objetivo várias grandezas a serem otimizadas, i.e., trata-se de uma otimização multiobjetivo simplificada a uma monoobjetivo. Além disso é feita uma discussão sobre os métodos de resolução de problemas deste porte, o que é uma introdução para a discussão da metodologia utilizada neste trabalho, tema do quarto capítulo.

Assim, o método de otimização por enxame de partículas – PSO e as metodologias de cálculo utilizadas para dimensionar o núcleo do reator IEA-R1 são discutidas com maiores detalhes no capítulo quatro.

O quinto capítulo é o cerne deste trabalho, uma vez que apresenta o código de otimização, com suas particularidades e a relação com os demais códigos na qual este foi integrado. É feita a validação dele por meio de uma função de teste, uma vez que se trata da primeira resolução do POR para o IEA-R1. Após isso, são apresentados os demais resultados.

O último capítulo, sexto, refere-se às conclusões obtidas e as recomendações e sugestões para trabalhos futuros. São descritos de forma sucinta todos os resultados alcançados ao longo do estudo desenvolvidos nas seções anteriores, avaliando suas contribuições e sugerindo estudos futuros para o POR tanto no IEA-R1 como em reatores de pesquisa em geral.

II ESTADO DA ARTE

Neste capítulo é apresentada uma pesquisa na literatura sobre os trabalhos já desenvolvidos referentes ao tema desta tese. São mostradas as mais variadas pesquisas sobre o POR que utilizam meta-heurísticas em sua resolução, obedecendo uma linha cronológica; na sequência a pesquisa focou-se em resoluções do POR que possuam uma maior semelhança com o problema desta tese, i.e., na resolução do POR em reatores de pesquisa e, por fim, trabalhos que utilizaram a meta-heurística do PSO para a resolução do POR, independente do tipo de reator (potência ou pesquisa).

2.1 A resolução do POR por meio de meta-heurísticas.

Como citado no capítulo anterior, por muito tempo os problemas relacionados ao estabelecimento de novas configurações para núcleos de reatores nucleares eram resolvidos por especialistas de maneira manual, ou seja, buscavam as configurações baseados na experiência adquirida ao longo dos anos pela prática de "tentativa e erro".

Após a evolução dos hardwares em meados dos anos 80, em especial dos processadores, métodos de cálculo mais complexos como as metodologias meta-heurísticas puderam ser utilizados.

O trabalho de GALPERIN e NISSAN (1988) foi um dos primeiros a inserir o conceito de heurística dentro de problemas voltados à engenharia nuclear. Nele, foi feito um estudo do POR utilizando um código de pesquisa heurística para a geração e a otimização de configurações de recarga em uma simulação de um reator PWR. O desempenho de algumas configurações propostas pelo programa se mostrou consideravelmente satisfatório e impulsionou o uso de códigos computacionais de otimização como uma ferramenta a auxiliar um especialista.

No inicio dos anos 90 foi desenvolvido o código computacional FORMOSA – Fuel Optimization for Reloads: Multiple Objectives by Simulate Annealing (KROPACZEK e TURINSKY, 1991), que utilizava a meta-heurística de Recozimento Simulado (*Simulate Annealing* – SA) (KIRKPATRICK et al, 1982) para a resolução do POR. O código tem como objetivo a maximização do fator efetivo (k_{EFF}) no final do ciclo de queima (*End of cicle* - EOC), a minimização do pico de potência radial ao longo do ciclo ou a maximização da queima de descarga média ao longo do ciclo. Os resultados obtidos indicavam uma redução de 6,1% no pico de potência e o aumento da queima média em 11,4%, maximizando o k_{EFF} no EOC e a diminuição de 11,2% da duração do ciclo. Desta forma, foi comprovada a viabilidade da utilização do método de SA na resolução do POR, a ponto do código ganhar uma versão comercial.

Posteriormente o código sofreu uma alteração de meta-heurística, trocando o SA pelo método dos Algoritmos Genéticos (*Genethic Algorithms* – GA) (HOLLAND,1975). Surgia assim o FORMOGA (POON e PARKS, 1992). O código usava uma função objetivo que minimizava o valor de F_{XY} , o fator de pico, definido como a razão do pico de densidade de potência no plano horizontal onde ocorreu o pico local de potência. Observou-se que com a substituição da meta-heurística houve um ganho no tempo de otimização (MACHADO,2005).

Devido ao êxito desse programa, intensificou-se a pesquisa voltada às vantagens do uso de Algoritmos Genéticos, sobretudo na engenharia nuclear. TANKER, E. e TANKER, A. (1994) fizeram um trabalho comparando a programação linear com o uso de GA e um código de programação, implementando esses dois algoritmos em um código voltado à resolução de um POR em reatores nucleares. O problema visava à maximização do k_{EFF} em EOC. Os autores observaram que as configurações com resultados mais próximos do ideal eram os obtidos utilizando o algoritmo de GA, porém o tempo computacional era consideravelmente maior se comparado ao algoritmo de programação linear.

O código CIGARO (*Code Independent Genetic Algorithm Reactor Optimization*) utiliza GA para um POR de um reator PWR. Nestre trabalho, DeCHEINE e FELTUS (1996) buscaram maximizar o K_{EFF} no inicio do ciclo (*Beginning of Cicle* - BOC) e foi utilizada uma função de penalidade para limitar o pico de potência. O sistema, segundo os autores, teve um bom desempenho, sendo efetivo na maximização do k_{EFF} ao mesmo tempo em que obteve êxito em limitar o pico de potência.

Em vista do aumento de metodologias voltadas à resolução do POR,

YAMAMOTO (1997) fez um estudo comparativo entre alguns desses métodos, incluindo os que fazem uso do conceito de otimização hibrida, isto é, a utilização de mais de um método de otimização no mesmo código. No trabalho em questão, foram comparados GA, SA, o método de busca direta (*Direct Search* – DS) e o método de Troca binária (*Binary exchange* – BE). Desta forma, viu que o método híbrido era capaz de fornecer resultados melhores e ainda concluíu que dos métodos analisados, o que melhor se comporta em um algoritmo hibrido é o método de GA. Em especial, testou-se o algoritmo hibrido formado por GA e BE, uma vez que as fraquezas de cada método foram compensadas entre si, o que gerou uma melhora significativa da otimização.

KASHI et al. (2014) criaram o código BANEC - *Bat Algorithm Nodal Expansion Code*, afim de resolver o POR para um reator PWR da central nuclear de Bushehr, no Irã, utilizando a meta-heurística do Morcego (Bat Algorithm – BA) (SHE YANG E GANDOMI, 2012). O código buscou maximizar o K_{EFF} e minimizar o fator de pico de potência em BOC. Além disso, definiu uma função objetivo dada pela razão das duas grandezas citadas anteriormente. Utilizaram como benchmarck o código CFNEC -*CFA Nodal Expansion Code* (POURZALEHI et al, 2013) que utiliza o CFA (Continuous Firefly Algorithm) e resolve o POR para o mesmo reator. Os resultados do código BANEC formam muito próximos aos do CFNEC, que já tinha se mostrado confiável. Como ganho, o BANEC foi o primeiro código a utilizar o BA para a resolução do POR.

A tese de MACHADO, L. (2001) trata da otimização utilizando o algoritmo Anti-Q (GAMBARADELLA e DORIGO, 1995), que baseia-se na teoria de agentes artificiais com reforço de aprendizado. Neste trabalho, o algoritmo busca a maximização da concentração de boro ou a minimização do fator de pico de potência radial e seus resultados foram comparados com um código semelhante que utilizava GA. Os resultados do Anti-Q se mostraram superiores aos obtidos com GA, indicando ser uma boa opção para a resolução do POR.

2.2 O POR em reatores de pesquisa

Ao contrário de reatores de potência, onde trabalhos focados na resolução do POR são comuns na literatura, a presença de artigos e teses sobre este tipo de problema voltado a reatores de pesquisa é rara. Aqui são apresentados dois trabalhos onde o POR é resolvido por meio da utilização de meta-heurísticas adequadas.

Importante observar que, apesar de possuírem menos EC, os núcleos dos reatores de pesquisa geralmente são assimétricos, o que leva a problemas no POR tão ou até mais complexos do que os de reatores de potência.

DO e NGUYEN (2007), buscaram resolver o POR no Dalat Nuclear Ressearch Reactor (DNRR) do tipo TRIGA MARK III utilizando GA, num código ambientado em FORTRAN 90. A função objetivo buscava a maximização do K_{EFF} e a minimização do fator de pico de potência, utilizando pesos para cada uma das grandezas, como mostrado na equação 2.1:

$$F = A(K_{EFF} - 1) + B(PFF^{0} - PFF)$$
(2.1)

Onde F é a função objetivo, PFF é o fator de pico de potência e PFF⁰ é um fator de entrada constante, escolhido de tal forma que seja sempre superior ao PFF. A e B, no caso, são constantes que funcionam como pesos para cada um dos parâmetros.O reator em questão possui um núcleo maior e de geometria diferente ao núcleo do IEA-R1, como mostrado na figura 2.1.



Figura 2.1 – Representação do núcleo do reator DNRR (DO & NGUYEN, 2007).

Fonte: (DO & NGUYEN, 2007)

O trabalho considerou um número limitado de alteração de posições de EC e realizou uma combinação entre estes e o vetor cromossomo, que abriga 10 EC, conforme mostrado na figura 2.2. Importante observar que as seções de choque foram calculadas pelo código CITATION (FOWLER et al, 1972).

Figura 2.2 – Posições dos EC no núcleo do DNRR e no cromossomo

Padrão de carregamento						Cromossomo	
<u>15</u>	2	<u>30</u>	4	<u>27</u>	6	<u>40</u>	
<u>32</u>	<u>37</u>	10	11	12	<u>39</u>	14	
<u>1</u>	16	17	<u>25</u>	19	20	<u>21</u>	
22	23	24	<u>18</u>	26	<u>5</u>		↔ (15 30 27 40 37 21 18 8 13 38)
28	29	<u>3</u>	31	<u>8</u>	33		
34	35	36	<u>9</u>	<u>41</u>		•	
13	7	38			-		

Fonte: (DO & NGUYEN, 2007)

O código se mostrou eficiente dentro dos objetivos propostos de otimização

e com as restrições das trocas de posições dentro do núcleo. Entretanto, o trabalho não considerou o fluxo de nêutrons como um dos parâmetros a serem maximizados.

O trabalho de MAZROU e HAMADOUCHE (2006) trata da elaboração de um código voltado à resolução do POR, utilizando como *benchmark* os dados de um reator de pesquisa de 10 MW (IAEA TECDOC-643,1992), por meio do método de Redes Neurais Artificiais (*Artificial Neural Network* – ANN) (HOPFIELD,1988).

A função objetivo foi desenvolvida com base em dois parâmetros de desempenho que são o K_{EFF} e fator de pico de potência, também não tratando do fluxo de nêutrons, sendo que o primeiro buscou-se maximizar o K_{EFF}, com a condição de o fato de pico ser menor do que dado valor imposto pelo operador, conforme mostrado na equação 2.2.

$$F = \omega_P P(P_{max}) + \omega_k (K_{EFF} - K^{ref})$$
(2.2)

Onde $\omega_P e \omega_K$ são constantes que definem os pesos para as grandezas otimizadas, K^{ref} é um valor referência para o K_{EFF}, semelhante ao realizado no trabalho descrito anteriormente. P(P_{max}) é uma função penalidade definida por:

$$P(P_{max}) = \begin{cases} P_{max} - P_{lim} & se P_{max} > P_{lim} \\ 0 & se P_{max} \le P_{lim} \end{cases}$$
(2.3)

Com P_{lim} sendo o fator de pico limite (no trabalho dado por 1,9).

No início dos cálculos, uma configuração é escolhida pela ferramenta computacional por meio de um banco de dados disponível ou sugerido pelo próprio operador. Na etapa seguinte, com base em regras de heurísticas estabelecidas previamente, um gerador aleatório sugere um rearranjo na configuração do núcleo. Um algoritmo de rede neural do tipo *back-propagation* (definido pelos autores como BPN) é utilizado para prever os parâmetros principais de segurança operacional.

O método de SA otimiza e determina se a configuração atual é melhor do que a de referência com base nos resultados de previsão, buscando valores dos parâmetros já definidos na função objetivo. Se a configuração ideal não for alcançada, uma nova etapa é executada e o processo é repetido até que o espaço de busca disponível seja totalmente explorado, conforme mostrado na figura 2.3. As regras heurísticas indicadas são expressas em termos de restrições e que não devem ser violados, já que isso representaria um potencial risco aos limites de projeto. As restrições foram definidas como:

- I. EC novos não podem ser movidos para posições centrais do núcleo.
- II. São proibidas trocas entre EC novos e os de maior queima (entre 45% e 50%).
- III. EC de maior queima não podem ser movidos para as regiões periféricas do núcleo.
- IV. As barras de controle têm suas posições fixas.
- V. Apenas um tipo de elemento de controle deve estar presente no núcleo durante um ciclo.
- VI. Não são permitidos quatro ou mais EC novos vizinhos, i.e., que não exista um EC com queima entre quatro ou mais EC novos.

Figura 2.3 – Estrutura global da estratégia de otimização realizada por MAZROU e HAMADOUCHE (2006).



Fonte: (MAZROU e HAMADOUCHE, 2006)

Os autores concluíram que a metodologia foi eficaz em determinar configurações ótimas para o POR, dentro das restrições impostas, juntamente com um ganho de eficiência computacional.

2.3 A utilização do PSO para a resolução do POR.

Visando um conhecimento maior das capacidades de utilização do algoritmo PSO, há a necessidade de uma pesquisa bibliográfica voltada à resolução do POR utilizando essa metodologia, sua eficácia na resolução do problema, comparações com outros métodos e eventuais modernizações desse algoritmo.

O trabalho de MEDEIROS (2005) utilizou o PSO em diversos problemas relacionados à engenharia nuclear, visando mostrar a eficácia desse método. Em um dos capítulos de sua tese foi tratada a questão do dimensionamento do núcleo de Angra I, problema muito parecido com o POR. Dessa forma, o autor desenvolveu um código que visava minimizar o fator de pico médio e manter tanto o fluxo médio, quanto o K_{EFF} dentro dos limites de projeto, com uma incerteza igual ou menor a 1%. Além disso, comparou-se o código do PSO com um código semelhante que usava GA e a *fitness* encontrada no código do PSO (1,2780) foi menor do que a determinadapor meio do código em GA (1,2949), o que mostrou a eficiência do PSO e sua vantagem em relação ao GA.

WAINTRAUB (2009) fez uma proposta semelhante ao trabalho citado no parágrafo anterior, porém desenvolvendo um programa que utilizava a computação paralela de dois PSO, o chamado PPSO, utilizando diferentes metodologias: Mestreescravo, Celular, N2, N4, Ilhas-100 e Ilhas-1000. Todas estas metodologias foram comparadas entre si e com o PSO original. Todas as metodologias obtiveram ganho de tempo computacional, porém quanto à maximização da concentração de boro, o método mestre-escravo atingiu resultados semelhantes ao PSO e os mais eficazes foram o PPSO-Ilhas-1000 e o PPSO-N4, que atingiram as maiores concentrações de boro.

YADAV e GUPTA (2011) desenvolveram um código com método PSO para a resolução de um POR de um reator com núcleo de 17 x 17 posições nos primeiros dois ciclos de operação, tendo como objetivo a minimização do fator de pico. Os resultados não são muito sensíveis ao número de partículas ou iterações usadas. No entanto, os autores observam que várias experiências devem ser realizadas para chegar ao melhor parâmetro global defunções de aptidão. Além disso, observaram que os fatores de pico de potência obtidos para ambos os ciclos foram razoavelmente baixos.

Já KHOSHAVAL et al (2010) criaram um código para a resolução de um POR em um reator VVER-1000, utilizando o algoritmo PSO contínuo (*Continuous PSO* – CPSO), que consiste em uma versão aprimorada do PSO original. O código, nomeado pelos autores de CRCPSO se mostrou eficiente quando comparado a outras meta-heurísticas como GA, SA e ANN.

Por fim, AHMAD & AHMAD (2018) resolveram o POR para o núcleo do *Pakistan Research Reactor-1* (PARR-1) que se trata de um reator de pesquisa cujo núcleo é composto de 24 EC, de diferentes enriquecimentos e 5 elementos combustíveis de controle (ECC), tendo uma geometria assimétrica, conforme mostrado na figura 2.4.

Para a otimização, foi utilizado o PSO puro e uma versão com uma incorporação do método do peixe – gato (*Catfish PSO* – CFPSO), cuja justificativa foi a de evitar uma convergência prematura à medida que o número de partículas utilizadas no processo aumentam.

mbustivel
controle
grafite
água
de fluxo

Figura 2.4 – Esquema do núcleo do reator PARR-1.

Fonte: (AHMAD e AHMAD, 2018) - adaptado

Os autores dividiram o trabalho em duas etapas: Inicialmente realizaram a integração entre o código de otimização e um código de teoria de difusão afim de resolver uma função objetivo apenas dependente do K_{EFF}, buscando maximizá-lo. Em um segundo momento foi criada uma função objetivo que além do K_{EFF} incluía o fator de pico de potência, ficando então sob a forma

$$F = K_{EFF} + 2(1, 2 - PPF).$$
(2.5)

São comparados os dois modos de otimização e a forma híbrida apresentou uma redução significativa no valor do fator de pico, passando de 1,54 (PSO original) para 1,29 (forma híbrida). Entretanto a forma original do PSO obteve um valor de K_{EFF} maior (1,0324901) se comparado ao obtido pela forma híbrida (1,0170258).

Após a revisão bibliográfica realizada neste capítulo, se faz necessária uma discussão sobre o conceito de otimização, focando-se em especial na otimização combinatória, classe de problema onde o POR se encontra. O terceiro capítulo versará sobre este tema, bem como as formas de solucioná-lo.

III FUNDAMENTAÇÃO TEÓRICA

O presente capítulo faz um breve apanhado sobre os principais conceitos de otimização, tratando da sua definição e formulação matemática. Além disso, são discutidos os conceitos de otimização combinatória e multiobjetivo, para na parte final ser dado um breve panorama sobre algoritmos que resolvem estes problemas.

3.1 Otimização: Conceitos preliminares

Entende-se por otimização o ato de obter o melhor resultado possível sob determinadas circunstâncias, geralmente minimizando o esforço necessário ou maximizando o benefício desejado, para a construção ou o funcionamento de sistemas de interesse para o ser humano.

A aplicação deste conceito atende aos mais variados contextos, como a redução dos custos de fabricação de uma determinada peça de um carro, a determinação de uma rota por um aplicativo que leve o menor tempo possível entre os pontos inicial e final ou mesmo a definição da melhor ação que garanta os lucros máximos de uma empresa.

Uma vez que o esforço exigido ou o benefício desejado podem ser expressos como uma função de certas variáveis de decisão, a otimização é definida então como o processo de encontrar as condições que dão o valor máximo ou o valor mínimo de uma função.

Apesar da otimização ser um conceito aplicável as mais variadas áreas, não existe um método único disponível para resolver todos os problemas de otimização de forma eficiente. A escolha do método dependerá das características do problema a ser otimizado, i.e., se o problema é unimodal ou multimodal, se a função em questão é o não diferenciável, linear ou não linear, convexa ou não convexa. Desta forma, vários métodos foram desenvolvidos para resolver diferentes tipos de problemas de otimização.

Em decorrência da ampla gama de metodologias existentes para a resolução dos mais variados problemas de otimização, exige-se do usuário um conhecimento

considerável sobre estas, de maneira a garantir que a ferramenta mais eficiente para o problema em questão seja utilizada.

Nos próximos tópicos deste capítulo serão descritos os conceitos principais de otimização referentes aos temas abordados neste trabalho, de maneira a garantir um melhor entendimento sobre a tese.

3.2 Descrição Matemática

De acordo com BUTENKO & PARDALOS (2008) seja $\Omega \subseteq \mathbb{R}^n$, um conjunto de vetores n-dimensionais sob a forma $x = [x_1, ..., x_n]^T$ e f(x) uma função tal que $f: \Omega \to \mathbb{R}$, define-se um problema de otimização como:

maximizar ou minimizar
$$f(x)$$
 (3.1)
sujeito a $x \in \Omega$

O conjunto Ω é definido como o conjunto viável do problema e cada um dos elementos x_j , (j = 1, ..., n) são as soluções possíveis. A função f é a função objetivo (*fitness*), tendo esta a capacidade de determinar o quão bem solucionado é o problema a partir de um determinado elemento de Ω . A definição de Ω é dada por um conjunto de restrições a partir de igualdades e desigualdades conforme mostrado em 3.2:

$$h_i(x) = 0, se \ i \in \varepsilon$$

$$g_j(x) \le 0, se \ j \in I$$
(3.2)

Onde $h_i: \mathbb{R}^n \to \mathbb{R}$ e $g_j: \mathbb{R}^n \to \mathbb{R}$ são funções; ε e I são conjuntos de índices para as restrições de igualdade e desigualdade, respectivamente. Ω , por sua vez, é descrita como:

$$\Omega = \{ x \in \mathbb{R}^n | g(x) \le 0, h(x) = 0 \}$$
(3.3)

A solução que, de acordo com o problema, maximiza ou minimiza a função
objetivo é chamada de solução ótima. É importante observar que, do ponto de vista matemático, os problemas de maximização e de minimização não possuem diferenças exceto de um fator -1.

Assim, se a função f(x) é minimizada em um dado problema, basta utilizar a função – f(x) para torna-lo uma maximização (OLIVEIRA,2013). Isso se dá pelo fato das soluções locais e globais nos dois tipos de problemas serem iguais. A figura 3.1 mostra tal conclusão, onde X*, no caso, é o ponto que maximiza f(x) e minimiza seu oposto.

Figura 3.1- Representação gráfica de uma maximização e uma minimização de funções opostas



FONTE: (OLIVEIRA,2013)

As características da região viável determinam o tipo de problema de otimização. Na tabela 3.1 é mostrada uma relação entre os diferentes tipos de regiões viáveis e a classificação de seus respectivos problemas.

Região Viável	Problema
\mathbb{R}^{n}	Otimização sem restrições
$\{x \in \mathbb{R}^n / l < x < u\}$	Otimização em caixas
$\{x \in \mathbb{R}^n / Ax = b, Cx \le d\}$	Otimização com restrições lineares
$\{x \in \mathbb{R}^n/h(x) = 0, \mathbb{R}^n \to \mathbb{R}^m\}$	Otimização com restrições de igualdade
$\{x \in \mathbb{R}^n / h(x) = 0, \mathbb{R}^n \to \mathbb{R}^m \ e \ g(x) \le 0, g: \mathbb{R}^n$	Problema geral de programação não linear
$\rightarrow \mathbb{R}^p$ }	

FONTE: (MARTINEZ & SANTOS, 2020)

Problemas reais em que se busca a otimização de certos parâmetros geralmente podem ser representados matematicamente por um sistema de equações ou inequações, composta por uma função objetivo, e uma série de restrições entre as suas variáveis.

3.2.1 Definições

Neste item serão feitas as definições de alguns termos centrais dentro da área de otimização, com o objetivo de proporcionar uma melhor compreensão do tema, conforme mostrado na literatura (HOLTZ,2005).

a) VARIÁVEIS DE OTIMIZAÇÃO

As variáveis de otimização, de decisão ou projeto, são aquelas que se modificam durante o processo de otimização, alterando o valor da função objetivo f(x). De modo geral, essas variáveis podem ser reais, inteiras ou discretas (valores compreendidos dentro de certo conjunto fixo).

b) ESPAÇO DE BUSCA

É o conjunto, espaço, ou região que compreende as soluções possíveis ou viáveis sobre as variáveis do projeto do problema a ser otimizado, sendo delimitado pelas restrições.

c) <u>RESTRIÇÕES</u>

São funções que delimitam o espaço de busca a partir de igualdades e/ou desigualdades. Estas funções atuam sobre as variáveis do projeto que descrevem situações que são consideradas indesejáveis.

d) <u>FUNÇÃO OBJETIVO</u>

É a função de uma ou mais variáveis de projeto que se quer otimizar, de maneira a maximilizá-la ou minimiza-la.

e) PONTO ÓTIMO

É o ponto formado pelas variáveis de projeto que levam ao maior valor da função objetivo e satisfazem as restrições.

f) VALOR ÓTIMO

É o valor da função objetiva no ponto ótimo. Na figura 3.1 é mostrada a representação das soluções ótimas nos casos de máximo e mínimo.

g) <u>ÓTIMO LOCAL E ÓTIMO GLOBAL</u>

Define-se como ótimo local o conjunto de valores para as variáveis do problema que minimiza ou maximiza a função objetivo em um subespaço $S \in x$. O ótimo global, por sua vez, é definido como o maior ou menor valor de todos os ótimos locais em x. Quando este é único, coincide com o ótimo local.

3.3 Otimizações Mono-objetivo e Multiobjetivo

Como definido no início deste capítulo, otimização é um processo que visa encontrar o melhor resultado possível de uma função sob determinadas circunstâncias, de maneira a maximizá-lo ou minimilizá-lo, escolhendo o domínio correntamente.

A otimização mono-objetivo é aquela que visa encontrar o ponto no espaço das variáveis de otimização, no qual a função objetivo alcança o valor máximo ou mínimo. Na vida real, entretanto, os problemas de otimização normalmente nos exigem considerar mais de uma váriável. Além disso, é comum que estas variáveis sejam conflitantes entre si.

Um caso clássico é a compra de uma casa, onde as variáveis custo e comodidade (variável esta que pode considerar, por exemplo, a distância para regiões de comércio, transportes e serviços, a segurança na vizinhança, o relevo da região etc) são conflitantes. Uma vez que fatores que compõem a variável comodidade são favoráveis ao comprador, é esperado que o custo cresça e viceversa. Problemas de otimização deste tipo, onde há a necessidade de otimizar mais de uma variável são conhecidos como problemas de otimização multiobjetivo.

Para resolver problemas deste tipo deve-se construir a função objetivo com significativo cuidado, pois quando são considerados objetivos conflitantes, em geral, não existe uma única solução que seja ótima com respeito a todos os objetivos. De maneira a ilustrar esse conceito, a figura 3.2 mostra o espaço de decisões e o espaço objeto factível de um problema de minimização com dois objetivos (HASHIMOTO,2004).

No caso, observa-se que a imagem de X^* , o espaço objeto factível, é dada por $Z^* = \{f_1(x), f_2(x), ..., f_n(x), x \in X^*\}$. Apesar de os objetivos em questão serem a minimização de $f_1(x)$ e $f_2(x)$, a solução do ponto A oferece um valor menor para f_1 , mas um valor maior para f_2 , em relação ao ponto B, o que leva a conclusão de que o aumento de uma função objetivo acarreta no aumento da outra e vice-versa. Em casos assim, é importante considerar os critérios de dominancia de Pareto, como no trabalho de OLIVEIRA(2013) e MEDEIROS(2005), entre outros.

Figura 3.2- Conjunto de soluções factíveis, espaço objetivo factível e o grau de dominância em um problema de minimização com dois objetivos.





Quando os objetivos não são conflitantes, é possível determinar uma função objetivo única que considera todos os objetivos, onde, para o problema, a otimização se dará sobre essa função objetivo. A discussão sobre a função objetivo específica deste trabalho é feita posteriormente.

3.4 Otimização Combinatória

Problemas combinatórios têm como característica a possibilidade de combinar as variáveis que estejam envolvidas. Problemas de otimização que possuem esta característica são conhecidos como problemas de otimização combinatória.

Os problemas de otimização combinatória, são problemas cujas variáveis

assumem valores discretos procuram por um inteiro, um conjunto, uma permutação ou até um grafo (FERREIRA,2018). O objetivo em um chamado problema de otimização combinatória, conforme definido por PRESTES (2006) é o de atribuir valores a um conjunto de variáveis de decisão, de tal modo que uma determinada função objetivo seja maximizada/minimizada, dependendo do objetivo, atendendo um determinado conjunto de restrições.

O anteriormente citado TSP, bem como o problema da mochila (*Knapsack Problem* – KP) (KELLERER et al, 2004) e a árvore de Steiner (Steiner Tree Problem – STP) (HWANG & RICHARDS, 1992) são exemplos teóricos clássicos. Como aplicações reais, além do POR, podem ser citados problemas referentes a posicionamento de satélites, empacotamento de caixas, estudos de layout, roteamento de veículos entre outros.

Segundo RODRIGUES (1996), os problemas de otimização combinatória podem ser divididos conforme a sua complexidade com base na teoria da NP-Completude. Essa classificação é feita em quatro grandes grupos, listados a seguir:

• *Tipo P (Polinomial Time):* O consumo de tempo é limitado a um algoritmo polinomial do tamanho da instância. São considerados tratáveis e que possuem algoritmos eficientes para sua resolução. Exemplos: Problema da ordenação de elementos, problema da árvore de espalhamento mínimo, problema do fluxo em redes.

• *Tipo NP (Nondeterministic Polinomial Time):* São problemas onde o esforço computacional do melhor algoritmo conhecido cresce não linearmente em função do tamanho da instância, não havendo uma garantia da não-existência de algoritmos melhores e, dada uma solução para o problema em questão, pode-se verificar se a solução satisfaz de maneira positiva ou não o problema de decisão em tempo polinomial. Todos os problemas P ou NP-Completos são exemplos desse tipo.

• *Intratáveis*: Semelhantes aos problemas do tipo NP, entretanto existe a garantia da não existência de algoritmos melhores. Exemplo: O problema da torre de Hanói.

Dentro da classe dos problemas tipo NP, existem outras duas classificações, listadas a seguir:

• **NP-Difícil (NP-Hard):** Um problema é dito NP-Difícil quando todos os problemas da classe NP forem polinomialmente redutíveis a ele. Assim, resolvendo um problema deste tipo em tempo polinomial, todos os problemas da classe NP poderão ser resolvidos em NP.

• *NP-Completo*: São formados pela intersecção dos problemas NP e NP-Difícil, ou seja, é um problema onde um algoritmo polinomial pode ser adaptado para resolver em tempo polinomial qualquer outro problema de complexidade NP. Um exemplo é o chamado problema de parada, que consiste em perguntar se, ao iniciarmos um programa de computador, este rodará infinitamente ou não.

Além da sua divisão pela complexidade, os problemas de otimização combinatória podem ser classificados quanto à forma, segundo MÜLLER-MERBACH (1981):

a) <u>PROBLEMAS DE ATRIBUIÇÃO</u>

Dois ou mais conjuntos discretos de objetos estão disponíveis e o problema consiste em determinar n-uplas a partir da otimização da função objetivo. O Problema da coloração de grafos e o problema do fluxo máximo em redes(HERNANDES et al, 2006) são exemplos.

b) <u>PROBLEMAS DE SEQUÊNCIAS</u>

Trata-se do ordenamento sequencial de um conjunto discreto de dados a partir da otimização de uma função objetivo. O TSP se encontra neste grupo.

c) <u>PROBLEMAS DE SELEÇÃO</u>

Um subconjunto deve ser formado a partir de um conjunto discreto de objetos. Como exemplo o problema da árvore de espalhamento mínimo.

d) <u>PROBLEMAS DE COMPOSIÇÃO</u>

São problemas que possuem características de mais duas ou mais das classes anteriormente citadas. Exemplo: O Problema da Mochila.

3.5 Métodos de resolução

Existem dois tipos principais de métodos para resolução de problemas de

otimização, a saber: Exatos e aproximados (RODRIGUES, 1996).

Os métodos exatos são limitados a um número finito de iterações e após conclui-las encontram a solução ótima. Apesar da precisão do resultado, sua utilização se limita a espaços de busca pequenos, uma vez que o tempo começa a se tornar um fator impeditivo à medida que os espaços de busca aumentam. Algoritmos como o método de força bruta, o método Simplex (GOLDBARG et al, 2016) ou o método *Branch-and-cut* (MÉNDEZ-DIAS & ZABALA, 2006) são exemplos de métodos exatos.

Os métodos aproximados não garantem que a função encontrada corresponda à configuração otimizada, entretanto são capazes de determinar uma solução sub-ótima, isto é, a melhor solução possível dentre as características e restrições do problema proposto. Os métodos aproximados englobam os métodos heurísticos e o metaheurísticos.

3.5.1 Heurísticas

O nome heurística deriva da palavra grega *heurisken*, cujo significado é descobrir. Consiste em métodos e regras que buscam alcançar uma solução viável, o que apenas garante ser um ponto máximo/mínimo local. Entretanto, são úteis problemas onde o espaço de busca é extremamente grande e o tempo de processamento torna-se inviável. Podem ser divididos em sete tipos(SILVA,2018):

- I. Heurística de construção.
- II. Heurística de melhorias.
- III. Heurísticas de decomposição.
- IV. Heurísticas de relaxação.
- V. Heurísticas baseadas em formulações matemáticas.
- VI. Heurísticas de restrição de espaço de soluções.
- VII. Heurísticas de particionamento.

As pesquisas realizadas ao longo de décadas sobre o desempenho de métodos heurísticos e, em particular, sobre as características que conduzem ao êxito de tais métodos levaram à elaboração de estratégias genéricas para a construção de heurísticas. Estas estratégias são conhecidas por meta-heurísticas.

3.5.2 Meta-heurísticas

Partindo da etimologia do termo, agora também há o prefixo "meta", que significa após ou além, dando assim uma ideia de um nível mais avançado de heurística.

Uma meta-heurística é, segundo NETO et al (2006), uma estratégia de busca que tenta explorar da maneira mais eficiente possível o espaço das soluções viáveis de um problema. Uma outra definição de meta-heuristica é dada a seguir:

Trata-se de uma arquitetura geral de regras que, formada a partir de um tema em comum, pode servir de base para o projeto de uma ampla gama de heurísticas computacionais (GOLDBARG et al, 2016, p.75)

Em GOLDBARG et al (2016) é feita uma discussão mais profunda sobre o conceito de meta-heuristica, contando inclusive com definições de outros autores sobre o tema.

Para auxiliar no processo de busca, conhecimentos específicos do problema podem ser utilizados na forma de heurísticas, como o caso das heurísticas de mutação e crossover no método de algorítmos genéticos, por exemplo. Mecanismos como estes tentam evitar confinamentos de soluções em mínimos ou máximos locais. Sua estratégia de busca dependerá da filosofia por ela empregada, como será visto posteriormente no PSO, que é a meta-heurística utilizada neste trabalho.

Atualmente existem várias meta-heurísticas e para classificá-las toma-se como base as origens do algoritmo. A figura 3.3 dá um panorama de como os tipos de meta-heurísticas se relacionam.

Figura 3.3- Diagrama das diferentes classificações de meta-heurísticas



FONTE: (LIBÓRIO,2019)

Uma classe de meta-heurísticas que tem recebido bastante atenção é constituída por ferramentas de computação inspirada em fenômenos biológicos. A chamada computação bio-inspirada é a linha de pesquisa que emprega metáforas e modelos de sistemas biológicos no projeto de ferramentas computacionais de solução de problemas complexos.

3.6 Algoritmos Bioinspirados

Pesquisadores observaram que diversos fenômenos biológicos, como um determinado bando de animais migrando ou procurando comida, por exemplo, executam estratégias relativamente simples para atingirem seus objetivos e que existe um processo de aprendizagem inserido nesta prática, o que garante que uma solução futura não será pior do que a já obtida.

Devido a essas características, muitos pesquisadores criaram novos métodos mimetizando a natureza das mais variadas formas. Foram utilizados inicialmente métodos específicos para determinados problemas, porém com o avanço das pesquisas e a melhor assimilação das heurísticas envolvidas, foi possível desenvolver métodos capazes de serem aplicados de maneira mais ampla em problemas de otimização.

Os algoritmos bioinsnpirados são divididos aqui em algoritmos evolutivos e algoritmos de enxame, como mostrado na figura 3.4.



Figura 3.4 – Tipos de algorítmos bioinspirados

FONTE: (MEDINA, 2014) adaptado

3.6.1 Algoritmos Evolucionários

Os Algoritmos Evolucionários (AE) são algoritmos dentro da área da computação inspirada na natureza que se baseiam nas leis da evolução para

encontrar soluções.

Sua origem remonta a década de 30 do século passado, quando sistemas evolutivos naturais passaram a ser investigados como algoritmos de exploração de múltiplos picos de uma função objetivo. Entretanto, tal estudo estagnou-se por décadas em decorrência da limitação computacional da época. Cerca de três décadas depois, com a criação de computadores que permitiam uma maior capacidade de cálculos, as pesquisas em AE ganharam novo fôlego.

Segundo LINDEN(2012), os algoritmos evolucionários funcionam mantendo uma população de estruturas, denominadas indivíduos ou cromossomos, operando sobre estas de forma semelhante à evolução das espécies. A estas estruturas são aplicados os chamados operadores genéticos, como a recombinação e a mutação, que são aproximações computacionais de fenômenos naturais, tais como a reprodução sexuada, a mutação genética etc.

Cada indivíduo recebe então uma avaliação, que quantifica a sua qualidade como solução do problema em questão. A partir desta avaliação são então aplicados os operadores genéticos de forma a simular a sobrevivência do mais apto.

GOLDBARG et al (2016) cita as seguintes características que definem os AE's, além do fato de se basear nas leis naturais da evolução:

- É baseada em uma população.
- Sua realização se dá por meio de um processo iterativo.
- Possui uma arquitetura de processamento paralelizável.
- Trata-se de um processo de busca guiada.

A computação evolutiva é uma área com amplo potencial de crescimento. GABRIEL & DELBEN (2008) consideram como motivos para tal conclusão o desenvolvimento de algoritmos pioneiros em encontrar soluções para determinados problemas complexos, simplicidade dos métodos e uma grande capacidade de adaptação para problemas de diferentes áreas.

Dentro da área, diversos algoritmos foram desenvolvidos ao longo dos anos, sendo o método de Algoritmos Genéticos (*Genethic Algorithm* – GA) HOLLAND(1975) o mais explorado. Além de GA, destacam-se como AE's, por exemplo, métodos como a Evolução Diferencial (*Differential Evolution* – DE) (FEOKTISKOV,2006) e as Estratégias Evolutivas (*Evolution Strategies* - ES) (BÄCK et al, 1991).

3.6.2 Algoritmos de Enxame

Os algoritmos de enxame (*Swarm Algoritms* - SA) são uma dividão dos chamados algoritmos bioinspirados, bem como os AE's. Neste tópico será feita uma breve descrição deste tipo de algoritmo, tratando inclusive de sua concepção a partir do ponto de vista biológico. Uma vez que o PSO é um algoritmo de enxame, pretende-se fazer aqui uma discussão um pouco mais aprofundada se comparada ao tópico anterior.

3.6.2.1 Conceitos básicos

A inteligência de enxame é, segundo PARSOPOULOS & VRAHATIS (2010), um ramo da inteligência artificial que estuda o comportamento coletivo e propriedades de sistemas complexos, auto-organizados e descentralizados com estrutura social. Tais sistemas consistem de simples agentes interativos organizados em pequenas sociedades, chamadas de enxames.

Seu conceito está intimamente ligado à ideia de inteligência coletiva, que se refere a indivíduos que apresentam um nível de inteligência superior dentro de um determinado comportamento social. A inspiração partiu da observação de grupos de variadas espécies animais (vários tipos de insetos, peixes e pássaros, por exemplo) em suas sociedades buscando um determinado objetivo comum: A busca por alimentos, ou a migração para um ambiente mais favorável, divisão de tarefas, melhor aproveitamento de suas capacidades, evitar predadores, entre outros (BONABEAU et al, 1999).

Embora os membros dessas colônias, se observados separadamente, são vistos como indivíduos não sofisticados, quando analisados em conjunto com demais indivíduos de sua espécie, são capazes de realizar tarefas complexas em cooperação. O comportamento coordenado de toda a colônia emerge de ações relativamente simples ou interações entre os membros individuais. Muitos aspectos das atividades coletivas dos insetos sociais são auto-organizados e funcionam sem um controle central, por exemplo.

BENI & WANG (1989) foram os primeiros a utilizar a expressão "inteligência de enxame" (*Swarm Inelligence* – SI). No caso, tratava-se do contexto de sistemas robóticos celulares, onde muitos agentes simples ocupam ambientes unidimensionais ou bidimensionais para gerar padrões e autoorganizar-se por meio de interações com o vizinho mais próximo.

Embora existam diferenças filosóficas e operacionais entre algoritmos de inteligência evolutiva e de enxame, todos eles foram categorizados como AE's num primeiro momento em meados dos anos 90. Essa ligação foi feita devido às suas semelhanças inerentes, como o comportamento estocástico, o uso de populações, tipos de campos de aplicação, bem como o público científico que estava interessado nessas abordagens (BONABEAU et al, 1999).

3.6.2.2 Princípios biológicos da inteligência de enxame

As decisões coletivas tomadas em um exame contam com o componente de auto-organização (*Self Organization* – SO) que é um grupo de mecanismos dinâmicos onde estruturas aparecem no nível global de um sistema a partir das interações entre os componentes dos níveis mais baixos, sem serem codificadas explicitamente no nível individual. Segundo GOLDBARG et al (2016), a SO independe da inteligência.

Vale notar que a SO não deve ser confundida com capacidades definidas como emergência da inteligência (GOLDBARG et al, 2016), que tratam do raciocinar, planejar, resolver problemas, abstrair ideias, linguagens e aprender. A emergência da inteligência é marcada pela intencionalidade, que se trata de um comportamento de poucos sistemas biológicos.

As teorias de SO, como em NICOLIS & PRIGOGINE (1977), foram originalmente desenvolvidas no contexto da física e da química para descrever o surgimento de padrões macroscópicos de processos e interações definidas no nível microscópico, podendo ser estendida a sociedades com o objetivo de mostrar que o comportamento coletivo complexo pode emergir de interações entre indivíduos que apresentam comportamento simples, o que implica na não necessidade de invocar a complexidade individual para explicar o comportamento coletivo complexo.

Conforme BONABEAU et al (1999), a auto organização de um sistema qualquer depende de quatro regras básicas:

- i. Experiência Positiva (*Positive Feedback*): Execução de regras gerais comportamentais simples que promovem a criação de estruturas. No caso de um recrutamento para uma fonte de alimento, por exemplo, há a necessidade da criação de trilhas para algumas espécies de formigas, ou a execução de danças em abelhas.
- ii. Experiência Negativa (*Negative Feedback*): Tem como objetivo contrabalancear a experiência positiva e leva à estabilização do padrão de comportamento coletivo, podendo ser resultado de algum fator externo, como mudanças no ambiente ou aparecimento de componentes não pertencentes ao grupo. Pode assumir a forma de saturação, exaustão ou competição. No exemplo de forrageamento, a experiência negativa decorre do número limitado de indivíduos que estejam buscando comida, a saciedade, as condições da fonte de alimento etc.
- iii. Amplificação de flutuações aleatórias. A SO depende da amplificação de flutuações (passeios aleatórios, erros, trocas de tarefas de maneira aleatória), que nada mais são do que um comportamento estocástico dos componentes do grupo, que podem ser uma arma poderosa na busca de novas soluções e explorar caminhos alternativos. Por exemplo, formigas forrageadoras podem se perder, porque seguem trilhas com algum nível de erro. Embora tal fenômeno possa parecer ineficiente, elas podem encontrar novas fontes de alimento inexploradas e recrutar companheiros de ninho para esses achados.
- iv. Múltiplas interações diretas entre indivíduos. Ao contrário de um único indivíduo, que pode gerar pode uma estrutura autoorganizada sem maiores problemas, um exame requer uma densidade mínima de organização entre os indivíduos para atingir a auto-organização. Além disso, os indivíduos devem ser capazes de

fazer uso dos resultados de suas próprias atividades, bem como das atividades dos outros.

Na natureza existem vários exemplos de propriedades surpreendentes e desassociadas das características típicas de inteligência como pensamento abstrato, capacidade de planejamento, capacidade de aprender e memorizar coisas úteis, habilidade de comunicação, intuição, criatividade, consciência, emoção, empatia, dentre outras. As bactérias, por exemplo, são capazes de explorar oportunidades e evitar situações desfavoráveis, a fim de garantir sua reprodução e principalmente a sua sobrevivência. Outro exemplo são os comportamentos de cardumes de peixes ou enxames de gafanhotos, que podem assumir o comportamento de um único organismo, lhe garantindo vantagens.

3.6.2.3 Principais algoritmos

Neste tópico será feita uma abordagem superficial de alguns métodos de algoritmos de enxame, dando prioridade para os mais conhecidos na literatura. Entretanto, o PSO, método a ser utilizado neste trabalho, será descrito separadamente e em mais detalhes no capítulo 5.

a) <u>OTIMIZAÇÃO POR COLÔNIA DE FORMIGAS (*ANT COLONY OPTIMIZATION* - ACO)</u>

O algoritmo de colônia de formigas, criado por GAMBARDELLA & DORIGO(1995), inspira-se no comportamento de forrageamento de formigas reais, que são capazes de encontrar um caminho mais curto entre o formigueiro e uma fonte de alimento, sem auxilio de fontes visuais, mesmo que ocorram mudanças no ambiente, como a introdução de um obstáculo. A figura 3.5 ilustra essa ideia.

Isso acontece em função da utilização do feromônio¹ pelas formigas como uma espécie de indicador. No caso, os trechos mais longos recebem menos

¹ Substâncias quimicas emitidas por algumas espécies com o intuito de atrair um parceiro, seja por intenções sexuais ou a busca de alimentos.

feromônios do que os trechos mais curtos, e as formigas são atraídas de maneira diretamente proporcional à quantidade de feromônios. Assim as formigas são atraídas para caminhos mais curtos (LIMA,2005).



Figura 3.5 – Forrageamento de uma colônia de formigas.

Uma das ideias centrais do algoritmo é justamente mimetizar esta forma de cooperação entre as formigas baseadas em trilhas de feromônios a fim de encontrar soluções melhores para um determinado problema. Para isso, o algoritmo utiliza a ideia das formigas "artificiais". Segundo LIMA(2005), estas se tratam de modelos modificados das formigas "reais", i.e., aquelas encontradas na natureza.

No algoritmo, as formigas "artificiais" modificam informações numéricas armazenadas localmente em cada estado do problema por elas visitados. Essa informação que é compartilhada considera o histórico de busca e cada formiga individualmente. As trilhas, que compõem uma forma de comunicação indireta entre as formigas, tornam-se um tipo de informação que é distribuída e modificada por elas mesmas para então refletir suas experiências adquiridas.

Apesar das muitas semelhanças entre as características das formigas "reais" e as "artificiais" do algoritmo, GAMBARDELLA & DORIGO (1995) citam algumas diferenças:

- I) A natureza do movimento, que é de maneira discreta entre as formigas "artificiais" e contínua entre as "reais".
- II) O depósito de feromônio no algorítmo ocorre baseado na qualidade

FONTE: LIMA(2005)

da solução encontrada.

III) As formigas encontradas na natureza não possuem uma estrutura de memória como no caso das formigas do algoritmo.

b) MÉTODO DA COLÔNIA ARTIFICIAL DE ABELHAS (*ARTIFICIAL BEE COLONY* – <u>ABC</u>)

O ABC teve como inspiração o comportamento forrageiro de abelhas e foi proposto por KARABOGA(2005) como uma metaheurística baseada em populações para a otimização de funções multimodais e de múltiplas variáveis em espaços de busca contínuos.

As abelhas "artificiais" realizam um "voo" sobre o espaço de busca multidemensional buscando as melhores fontes de alimento, que representam as melhores soluções para o problema em questão. De acordo com OLIVEIRA(2005), existem três tipos de abelhas no método:

- Exploradoras: Abelhas que procuram por boas fontes de alimento de maneira aleatória.
- II) Operárias: São abelhas que anteriormente eram exploradoras e após terem encontrado uma boa fonte de alimento, passam a explorá-la. Carregam informações sobre essa fonte e as compartilham com as abelhas observadoras.
- III) Observadoras: São abelhas que ficam na colmeia à espera de informações trazidas pelas abelhas operárias e, com base nestas informações, tentam descobrir novas fontes.

O que determina a qualidade de uma fonte de alimento para as abelhas, no caso, é a quantidade de néctar presente nesta fonte; se, ao encontrar uma nova fonte a abelha identificar que esta possui mais néctar do que a fonte tida como boa, esta nova fonte é armazenada em sua memória e a fonte anterior é descartada.

Deste modo o ABC realiza uma combinação de buscas locais com buscas globais, sendo as locais realizadas pelas abelhas operarias e as globais pelas exploradoras. Além disso, realiza o compartilhamento de informações (experiências adquiridas) e os aproveita com as observadoras.

Além dos dois métodos brevemente discutidos, merecem destaque dentro dos algorítmos de enxame os métodos de Cardume artificial de Peixes (*Artificial Fish Swarm Algorithm* – AFSA) (WANG et al, 2005), Sistema imune artificial (*Artificial Immune System* - AIS) (DASGUPTA, 1993) e o algoritmo de vagalumes (*Firefly Algorithm* - FA)(YANG,2009).

O terceiro capítulo discutiu o conceito de otimização e suas formas de resolução, concentrando-se especificamente no caso de otimizações combinatórias. No capítulo seguinte será discutido em especial o PSO, a meta-heurística utilizada para a resolução do POR para este trabalho.

3.8 Conceitos básicos de Física de Reatores e Termohidráulica

3.7.1 K_{EFF}

A energia nuclear é liberada por meio de uma reação em cadeia de fissão. Neste processo, nêutrons emitidos por núcleos fissionais induzem fissões em outros núcleos físseis ou fissionáveis; os nêutrons dessas fissões induzem fissões em outros núcleos físseis ou fissionáveis; e assim por diante. Tal reação em cadeia pode ser descrita quantitativamente em termos do fator de multiplicação, denotado por K_{EFF}. Isso é definido como a razão do número de fissões (ou fissões de nêutrons) na geração anterio, ou seja:

$$K_{EFF} = \frac{n_t}{n_{t-1}} \tag{3.4}$$

Onde n_t é o número de nêutrons em uma dada geração e n_{t-1} é o número de nêutrons na geração anterior.

Ao analisar a Eq. 3.4, tem-se três casos possíveis para K_{EFF}:

- Se K_{EFF} for maior que 1, então o número de fissões aumenta de geração em geração. Nesse caso, a energia liberada pela reação em cadeia aumenta com o tempo, e a reação em cadeia, ou o sistema em que ela ocorre, é considerada supercrítica.
- No entanto, se K_{EFF} for menor que 1, o número de fissões diminui com o

tempo e a reação em cadeia é chamada de subcrítica.

Finalmente, na situação especial em que K_{EFF} é igual a 1, a reação em cadeia ocorre a uma taxa constante, a energia é liberada no nível estável e o sistema é considerado crítico.

3.7.2 Fluxo de nêutrons

Para compreender melhor o conceito de fluxo de nêutrons, dado por ϕ , LAMARSH & BARATTA (2014) propõem imaginar um experimento onde um feixe de nêutrons de intensidade I atinge um determinado alvo. Assim, a sua taxa de colisões por cm³/s (R) é definida como:

$$R = \Sigma_t I \tag{3.5}$$

Onde Σ_t é a seção de choque macroscópica total, i.e., a soma de todas as seções de choque macroscópicas.

Assim, pode-se então generalizar o experimento citado para não um, mas vários feixes incidindo sobre o mesmo alvo. As intensidades dos feixes são diferentes, mas supõe-se que os nêutrons em todos os feixes tenham a mesma energia.

Tendo em vista que a interação de nêutrons com núcleos é independente do ângulo em que os nêutrons colidem com os núcleos, a taxa total de interação torna-se então o produto da seção de choque macroscópica total. Assumindo os nêutrons como monoenergéticos, chega-se a:

$$R = \Sigma_t (n_A + n_B + n_C + \dots) v \tag{3.6}$$

Onde n_i são as densidades dos nêutrons de cada feixe e v a velocidade deles. Tomando n como a densidade total, R pode ser definido como

$$R = \Sigma_t n v \tag{3.7}$$

Em um reator nuclear é possível imaginar uma situação semelhante a tal experimento, entretanto com os nêutrons se movendo em todas as direções, o que valida a eq. 3.7 para este caso.

Por fim, chega-se ao conceito de é fluxo de nêutrons, que nada mais é do que a quantidade nv presente na eq. 3.7. TERREMOTO (2004) generaliza essa expressão de maneira a tornar as grandezas dependentes da energia.

$$R = \int_0^\infty \phi(\vec{r}, E) \Sigma_T(E) dE$$
(3.7)

3.7.3 Transferência de calor

Para TERREMOTO (2004) a análise térmica de um reator nuclear permite fixar na prática a máxima potência liberada tendo em vista as propriedades dos materiais do reator. Além disso, possibilita determinar as temperaturas dos componentes do reator nuclear (combustível nuclear, revestimento do combustível nuclear e refrigerante) em qualquer ponto do mesmo.

A energia é liberada pela fissão em meio ao combustível, sendo transferida por condução de calor para a superfície do combustível e através do revestimento. O calor é então transferido por convecção da superfície do revestimento para o refrigerante, que circula ao longo da superfície do revestimento, realizando a transferência de calor por convecção para um fluído que escoa em torno do elemento combustível. Finalmente, essa energia é transportada para fora do reator quando o refrigerante sai do núcleo e ingressa em trocadores de calor externos, nos quais pode ser gerado vapor por intermédio de um sistema de potência termodinâmico. Com isso, é possível notar a importância do processo de análise térmica em um reator nuclear.

O revestimento que envolve o combustível nuclear evita a liberação de produtos de fissão no refrigerante quando o mesmo passa pelo núcleo do reator, proporciona suporte e reforço estrutural ao combustível nuclear e amplia a área da superfície externa do elemento combustível em certos tipos de reatores (principalmente naqueles refrigerados a gás). Para sua confecção, busca-se sempre materiais com baixa seção de choque para captura radiativa de nêutrons, elevada condutividade

térmica, boa resistência mecânica em altas temperaturas e inércia química em relação ao combustível nuclear e ao refrigerante, como alumínio, ligas de magnésio (denominadas Magnox, atualmente em desuso), aço inoxidável e ligas de zircônio (denominadas Zircaloy) (TERREMOTO,2004).

Para definir a equação do calor utilizada no reator, utiliza-se a lei definida pelo matemático francês Joseph Fourier, supondo um meio homogênio e isotrópico² (OZISIK, 1993):

$$\nabla^2 T = -\frac{q^{\prime\prime\prime}}{k} \tag{3.8}$$

Onde k, neste caso, é a condutividade térmica do material em questão, q^{'''} é a taxa de liberação de energia no meio (W/m³) e ∇^2 é o operador laplaciano.

Considerando-se que toda a energia liberada por fissão nuclear é depositada no combustível, a taxa de liberação de energia por unidade de volume do combustível, segundo TERREMOTO (2014), é dada por:

$$q^{\prime\prime\prime} = E_F \Sigma_{FM} \phi_t \tag{3.9}$$

Onde a seção de choque macroscópica média é definida por Σ_{FM} , o fluxo de nêutrons térmicos é dado por ϕ_t e E_F = 200 MeV = 3,2 x 10⁻¹¹ J é a energia liberada na fissão de um átomo de material físsil.

A determinação das temperaturas máximas no núcleo, que em geral ocorrem no elemento combustível, são o ponto principal da análise térmica de um reator nuclear, sobretudo com o objetivo de estipular limites seguros para o funcionamento do mesmo. Em um núcleo no qual a distribuição do combustível é uniforme, tal elemento combustível é aquele posicionado no centro. A variação do fluxo de nêutrons térmicos e a taxa de liberação de energia ao longo do elemento combustível central (direção z) são dadas por:

² Meio onde a condutividade térmica independe da direção

$$\phi(z) = \phi_{max} cos \frac{\pi}{L} z \tag{3.10a}$$

$$q^{\prime\prime\prime}(z) = q^{\prime\prime\prime}{}_{max} cos \frac{\pi}{L} z$$
(3.10b)

Onde L é a altura extrapolada do núcleo, definida como da soma da altura geométrica L do núcleo (que é igual à altura ativa dos elementos combustíveis) com a distância de extrapolação (LAMARSH & BARATTA, 2014).

No caso de um EC tipo placa, é possível assumir a condução de calor como unidimensional (apenas no eixo x). Desta maneira, a Eq. 3.8 é simplificada para a forma:

$$\frac{d^2T}{dx^2} = -\frac{q^{\prime\prime\prime}}{k} \tag{3.11}$$

Com isso, é possível calcular a diminuição total de temperatura do centro do combustível nuclear para a superfície externa do revestimento (desprezando qualquer variação de temperatura através da interface combustível nuclear/revestimento).

$$\Delta T_{EC} + \Delta T_R = aq^{\prime\prime\prime} \left(\frac{a}{2k_{EC}} + \frac{b}{k_R} \right)$$
(3.12)

Onde os índices EC e R referem-se respectivamente ao elemento combustível e ao revestimento e as constantes a e b são dimensões ao longo do comprimento do eixo x.

IV METODOLOGIA

Este capítulo descreve toda a metodologia existente na tese. Aqui são mostrados o PSO,(que é a metaheurística utilizada para a resolução do POR), os cálculos neutrônicos e termohidráulicos realizados após todo procedimento de recarga do IEA-R1, o código de otimização utilizando o PSO e suas validações.

4.1 Otimização por enxame de partículas (PSO)

O método de otimização por enxame de partículas – Particle Swarm Optimization (PSO) – é um dos mais clássicos algoritmos de enxame, servindo de base para outros métodos similares, como o Algoritmo de Colônia de Abelhas entre outros. Sua implementação relativamente simples, aliada aos bons resultados já comprovados na literatura fazem deste algoritmo uma boa ferramenta para a resolução do POR.

Neste tópico serão discutidas as características do método, bem como as modificações feitas para a adequação ao problema desta tese.

4.1.1 Definições básicas

O PSO é um algoritmo de otimização criado por EBERHART & KENNEDY (1995) voltado a resolução de problemas não-lineares, contínuos ou discretos. Sua criação se deu a partir de um dos vários estudos existentes no início da década de 1990 sobre o comportamento social de animais aplicados a computação. No caso, observou-se que bandos de animais (como peixes, abelhas e pássaros) possuíam uma vantagem competitiva para a sobrevivência de um determinado indivíduo caso o mesmo compartilhasse seu conhecimento com os demais membros do grupo (LEITE, 2017).

Uma analogia para o método é citada por WILSON (1975), onde se uma

pessoa jogar grãos de milho em um determinado local aberto, há a certeza de que em algum momento vários pássaros irão para lá se alimentar. Eles procuram comida de maneira aleatória pelas árvores e, quando encontram algo, avisam todo o bando.

Para MELO (2018), as suas principais características são a simplicidade de implantação e a não necessidade de um grande poder computacional para executá-lo.

A função básica do método, segundo LUZ et al (2016), é a de promover uma troca de informações entre os elementos do grupo, onde cada elemento é denominado como partícula e o grupo em questão, enxame. Essa troca permite o desenvolvimento de um método, segundo KENNEDY et al (2001), "poderoso, simples, de fácil implementação e eficiente".

4.1.2 Bases Sociocognitivas

A teoria sociocognitiva que dá suporte ao modelo ao PSO é relativamente simples. Nela, o processo de adaptação cultural se baseia em um componente de alto-nível, visto na formação de padrões por meio dos indivíduos na sua habilidade de resolver problemas, e um componente de baixo-nível, o comportamento dos indivíduos, que, segundo KENNEDY et al (2001), são baseados em três princípios: Avaliar, comparar e imitar.

Qualquer organismo vivo tem como uma de suas características de comportamento fundamentais a avaliação, seja ela do ambiente que o cerca ou de outros organismos vivos. A capacidade de determinar como positivo ou negativo certo estímulo é um requisito para o processo de aprendizagem, pois assim, o organismo será capaz de melhorar a avaliação média de seu ambiente e, a partir daí, poder se posicionar e atuar de forma mais adequada.

Na grande maioria de suas ações, os seres humanos se valem da comparação com seus semelhantes ou indivíduos de outras espécies para estipular um padrão de avaliação. Ao estabelecer uma escala de pontuação, um indivíduo consegue determinar onde ele está em relação a outros indivíduos. KENNEDY et al (2001) utiliza como base a teoria da comparação social, desenvolvida por Leon FESTINGER (1954), que descreve algumas das maneiras que as pessoas usam outras como um padrão para realizar uma autoavaliação.

A imitação é de fato uma maneira de aprendizagem muito efetiva. Entretanto, segundo o trabalho do zoólogo Konrad Lorenz,poucas são as espécies que possuem tal capacidade, limitando a apenas algumas pequenas variações de aprendizagem social para boa parte das espécies. O alemão concluiu que apenas seres humanos e algumas espécies de aves seriam capazes de realizar uma imitação real(LORENZ,1935).

No caso do PSO, um indivíduo faz a comparação entre si mesmo e seus vizinhos mais próximos, para identificar aquele que dentre todos os avaliados é o melhor e então passa a imitá-lo (ou conclui que é, por enquanto, o melhor, e permanece da maneira como está).

Os princípios de avaliação, comparação e imitação podem ser combinados, inclusive em seres sociais artificiais em programas de computador, possibilitando a estes uma adaptação a ambientes complexos, resolvendo problemas difíceis.

4.1.3 Descrição do algoritmo

Nesse método, o algoritimo define um espaço de busca multidimensional que será explorado pelo enxame. Cada indivíduo ou "partícula" deste enxame, devidamente gerado pelo usuário, move-se por meio do espaço de busca em iterações sucessivas, como se estivesse "voando", cooperando e competindo com outras partículas, se movimentando, de maneira a sempre guardar a melhor posição do espaço de busca que visitou, (ou seja, o melhor valor da função objetivo) e em direção ao melhor indivíduo de uma vizinhança topológica.

Uma partícula, neste método, representa uma solução para o problema que está sendo otimizado no momento. Ao descobrir um bom caminho que leve ao alimento, as outras partículas do enxame também são capazes de se deslocar para aquela direção, independentemente da distância que estejam do enxame. A figura 4.1 mostra o fluxograma do PSO original. Figura 4.1 – Fluxograma representando o funcionamento do PSO original.



Fonte: (LUZ et al, 2016 - adaptado)

Na versão original do método, o bando de pássaros é representado em um ambiente bidimensional. PARSOPOULOS & VRAHATIS (2010) mostra que, para assegurar a existência de certa "personalidade" individual das partículas, cada uma delas é dotada de um comportamento distinto e aleatório em que a ponderação destas duas informações é sorteada ao longo do processo, assumindo assim um valor diferente para cada uma das partículas e variando este valor ao longo do processo iterativo. Assim, considerando o movimento da partícula neste espaço e sua localização nele, é, então, caracterizada por suas coordenadas.

Sendo assim, define-se o vetor $\vec{x_i}$ como a posição de uma partícula i (i = 1, 2, ..., n), onde este vetor pode ser de qualquer dimensão. A variação de posição da partícula i é definida a partir do vetor velocidade, $\vec{v_i}$. Define-se o vetor velocidade como (LUZ et al, 2016):

$$\overrightarrow{v_l}(t) = \overrightarrow{x_l}(t) - \overrightarrow{x_l}(t-1)$$
(4.1)

Onde t é definido como um passo de tempo unitário e, consequentemente, t-1 denota o passo anterior.

A nova posição de cada partícula é definida a partir da soma de três vetores auxiliares. O primeiro deles é o vetor velocidade, já os outros dois vetores se referem a melhor posição já encontrada pela partícula e a melhor posição já encontrada pelo enxame, definidas respectivamente por $\vec{p_i}$ e $\vec{p_g}$.

Obviamente todos se originam na posição atual da partícula. A figura 4.2 mostra a composição de uma nova posição da partícula - x(t) - a partir dos demais vetores citados anteriormente.



Figura 4.2 – Representação da movimentação de uma partícula no PSO

Fonte: (Autor)

Assim, cada partícula terá sua posição ajustada de acordo com a influência de outra partícula mais bem posicionada. Cada uma delas possui um valor que mostra o quão próximo do resultado ideal ela está, sendo determinado de várias formas a depender do problema em questão. Portanto, cada problema possui uma função objetivo - ou fitness - que define esse valor (MELO,2018).

Considerando as condições citadas anteriormente para a determinação da velocidade da partícula, chega-se a seguinte expressão utilizada no PSO clássico:

$$\vec{v_{l}}(t) = \vec{v_{l}}(t-1) + c_{1}\left(\vec{p_{l}} - \vec{x_{l}}(t-1)\right)$$

$$+ c_{2}\left(\vec{p_{g}} - \vec{x_{l}}(t-1)\right)$$

$$\vec{x_{l}}(t) = \vec{x_{l}}(t-1) + \vec{v_{l}}(t)$$

$$(4.2b)$$

As constantes c_1 e c_2 são números aleatórios e contribuem para a autoexploração da partícula e com o movimento desta no sentido do deslocamento global do enxame, respectivamente.

Assim, conforme a descrição de LUZ et al(2016), as partículas se movem de maneira aleatória em torno de um ponto definido como a média ponderada das melhores posições, do indivíduo e do enxame, dado por:

$$\overrightarrow{x_{m,l}}(t) = \frac{c_1 \overrightarrow{p_l} + c_2 \overrightarrow{p_g}}{c_1 + c_2}$$
(4.3)

SHI & EBEHART (1998) buscaram balancear ao peso entre as buscas locais e globais inserindo um termo de inércia em 4.2a. Além deste termo, os autores inseriram duas outras variáveis aleatórias, definidas como r_1 e r_2 , atuando nos termos referentes às buscas local e global. Assim, 4.2a fica conforme descrito a seguir:

$$\overline{v_i}(t) = \omega \overline{v_i}(t-1) + c_1 r_1 \left(\overline{p_i} - \overline{x_i}(t-1) \right)$$

$$+ c_2 r_2 \left(\overline{p_g} - \overline{x_i}(t-1) \right)$$

$$(4.4)$$

O valor do termo de inércia (ω) tem a capacidade de influenciar na capacidade de exploração das partículas. LUZ et al(2016) estipula o intervalo $0.8 < \omega < 1.2$ como um bom balanço entre as buscas local e global.

PARSOPOULOS & VRAHATIS, (2010) estabelecem uma relação linearmente decrescente do termo de inércia com a iteração:

$$\omega = \omega(t) = \omega_{inicial} + \left(\omega_{final} - \omega_{inicial}\right) \left(\frac{t}{T_{max}}\right)$$
(4.5)

Onde t é a iteração atual, T_{max} a iteração final e $\omega_{final} < \omega_{inicial}$.

A Figura 5.3 ilustra a diversidade³, para um enxame de 20 partículas atualizadas com as equações de velocidade e posição definidas anteriormente, com $c_1 = c_2 = 2$, minimizando a instância bidimensional de um problema de teste proposto por PARSOPOULOS & VRAHATIS, (2010). A linha mais escura e contínua se refere ao uso do termo de inércia na equação da velocidade, i.e., utilizando a equação 4.4. Por sua vez, a linha mais clara e pontilhada trata da equação 4.2a, ou seja, independente de ω .

Obviamente, o uso do peso de inércia tem um efeito tremendo na diversidade do enxame, que quase desaparece após 300 iterações, em contraste com o caso de

³ É definida como o valor médio dos desvios padrão das partículas em uma direção (PARSOPOULOS & VRAHATIS, 2010)

fixação de velocidade simples, que retém quase os mesmos níveis de diversidade ao longo das pesquisas.



Figura 4.3 – Comparação da diversidade do enxame de partículas utilizando o termo de inércia (linha escura) ou não (linha clara)

Fonte: (PARSOPOULOS & VRAHATIS, 2010 - adaptado).

Por fim, o algoritmo 1 mostra o Pseudocódigo do método PSO com inércia em maiores detalhes, considerando, no caso, a equação 4.4 para a determinação da velocidade.

A posteriori será discutida a forma como o PSO foi inserido no código de otimização, de maneira a atender os requisitos exigidos para a resolução do POR.

```
Algoritmo 1: Pseudocódigo do PSO

1 início

2 Atribuição de valores para os parâmetros;
```

```
3
     i = 0;
4
     repita
5
      Inicializa \vec{x_i} com uma posição aleatória;
      Inicializa \vec{v_t} com uma posição aleatória;
6
      i = i + 1;
7
     até i = tamanho_do_enxame;
8
     t = 0;
9
    repita
10
11
      i = 0;
12
      repita
        Atualizar \vec{v} utilizando a equação 4.2a;
13
        Atualizar \vec{X}, utilizando a equação 4.2b;
14
15
        Atualizar \overrightarrow{p_i} \in \overrightarrow{p_g};
16
       i = i + 1;
17
      até i = tamanho_do_enxame;
18 | t = t + 1;
19 até t atingir itermax;
    Saída: Melhor \overrightarrow{p_a}
20 fim
```

Fonte: Autor.

4.1.4 O modelo de chaves aleatórias (Random keys) aplicado no PSO.

Em vários tipos de problemas de otimização combinatória existe a necessidade de se garantir que não exista nenhuma duplicidade de resultados. O POR é um desses casos, uma vez que um elemento combustível não pode ocupar duas posições no núcleo

Como o PSO por si só não consegue realizar um mapeamento para identificar os valores obtidos de modo a evitar o problema descrito, há a necessidade de se inserir no algoritmo um método que o faça. O presente trabalho utilizou o método chaves aleatórias - *Random keys* (RK)(BEAN,1994).

O RK mapeia um vetor composto de números reais em uma solução contendo números inteiros não repetidos. Desta forma, soluções possíveis podem ser formados em um problema de otimização (MENESES,2010). Para ilustrar tal método, seja o vetor A = $[0,14 \ 0,11 \ 0,92 \ 0,37]$. Uma vez mapeado por meio do RK, tornaria se então o vetor B = $[2 \ 1 \ 4 \ 3]$, pois o menor valor (0,11) encontra-se na segunda posição. Da mesma forma, o segundo menor valor é 0,14 e esta encontra-se na primeira posição e assim sucessivamente. Segundo MENESES (2010), esta é a abordagem chamada *Single Machine Scheduling Problem.* A figura 4.4 mostra como se comportam os vetores posição e velocidade utilizando o método RK.

Figura 4.4 –Vetores no RK: (a) Vetores posição e velocidade para um espaço de busca de 5 dimensões. (b) Novas posições como chaves aleatórias. (c) Vetor auxiliar contendo uma solução válida.



Fonte: (MENESES, 2010)

A função do método RK dentro do algoritimo PSO neste trabalho é o de decodificar os vetores de posição, fornecendo uma solução válida para ser avaliada pela fitness. Assim, obtém-se um espaço de busca com soluções válidas para o POR a partir de um espaço contínuo de busca, o que leva a geração de possíveis soluções para problemas combinatórios.

Para fins de simplificação, deste ponto em diante o algoritimo de otimização será referenciado juntamente com o método das chaves aleatórias, ou seja, PSORK.

4.2 Metodologia do grupo de cálculo do reator IEA-R1

A cada operação de recarga do núcleo do IEA-R1 uma nova configuração é criada. Sendo assim, se faz necessária a avaliação dos parâmetros neutrônicos e termo-hidráulicos gerados por esta nova composição dos EC.

O grupo de cálculo do IEA-R1 realiza essa avaliação por meio de uma

metodologia desenvolvida pela Divisão de Física de Reatores do Centro de Engenharia Nuclear (CEN) do IPEN (LARANJO et al, 2011). Neste tópico é feita a apresentação e discussão desta metodologia. A figura 4.5 mostra o diagrama de cálculo.

Apesar de extensa, como será mostrada em maiores detalhes adiante, apenas parte da metodologia de cálculo será integrada com o código de otimização ENXAME. Trata-se dos códigos 2DB, CITATION e COBRA, que serão descritos em maiores detalhes a seguir.



Figura 4.5 – Diagrama de cálculos neutrônicos e termo hidráulicos no IEA-R1

FONTE: (LARANJO et al, 2011).

4.2.1 2DB

O código 2DB (ou TWODB, como inclusive é representado em seus arquivos) foi desenvolvido num trabalho conjunto entre a Pacific Northwest Laboratories e a Los Álamos Scientific Laboratory (LITTLE & HARDIE JR, 1969), visando à solução da equação de difusão de nêutrons em duas dimensões e vários grupos de energia. Em 1986, segundo DOMINGOS (2010), foi atualizado para inclusão de capacidades adicionais de edição, melhoria dos métodos de cálculo e novas opções de entrada de dados.

Trata-se de um código bidimensional (x-y, r-z, r- θ , e geometria hexagonal) para uso em criticalidade rápida em reatores e análise de queima do elemento combustível. O código pode fornecer parâmetros como o K_{EFF}, o coeficiente de criticalidade efetivo, tempo de absorção, composição do reator, fluxo de nêutrons e a queima do material, este último utilizando gerenciamento "in-core".

4.2.2 CITATION

Em geral, os códigos de análise neutrônica resolvem a equação de difusão de nêutrons a multigrupo gerando informações relevantes para análise neutrônica do núcleo do reator, como K_{EFF}, distribuição de fluxo de nêutrons, densidade de potência, entre outros parâmetros.

O código CITATION foi desenvolvido pelo Laboratório Nacional de Oak Ridge - *Oak Ridge National Laboratory* (ORNL) - para análise do núcleo de reatores nucleares (FOWLER et al, 1972). Ele resolve problemas envolvendo a representação em diferenças finitas da equação de difusão de nêutrons, tratando casos com até três dimensões espaciais, com espalhamento arbitrário entre os grupos de energia, geometrias x-y-z, θ -r-z, hexagonal-z e trigonal-z, além de problemas estáticos, de evolução e gerência do combustível para análise de ciclos múltiplos.

O código ainda é capaz de fornecer resultados para perturbação em primeira ordem, desde que sejam fornecidos os dados microscópicos e as concentrações dos nuclídeos. Problemas de autovalores de fluxo neutrônico são resolvidos por iteração direta para determinar o K_{EFF} ou as densidades dos nuclídeos requeridas para um sistema crítico (DALLE, 1999). Foi desenvolvido em FORTRAN IV, sendo composto

por, aproximadamente, 30000 linhas de programação divididas entre 197 subrotinas (BELCHIOR JR, 1992).

A versão original do CITATION, feita pela ORNL, é de julho de 1972, sendo executada em mainframes. Foi transportado pela primeira vez para PC pela Energia Atômica do Canadá Limitada - Atomic Energy of Canadá Limited (AECL) - em outubro de 1988. A versão utilizada atualmente no IPEN, e, consequentemente, a utilizada neste trabalho, data de setembro de 1994.

PARRO (2018) salienta em seu trabalho que o 2DB é acionado antes e possui função semelhante ao CITATION, porém os dados fornecidos pelo CITATION são em modelo tridimensional, contra o modelo bidimensional fornecido pelo 2DB. A equipe de cálculo do CERPq, responsável pela metodologia dos cálculos do núcleo do reator IEA-R1, utiliza o CITATION de forma subsidiaria, pois o 2DB fornece resultados satisfatórios com uma menor solicitação computacional (menor quantidade de memória exigida) e requerendo menor complexidade e tempo gasto.

4.2.3 COBRA

O COBRA - *Coolant Boiling in Rod Arrays* - é um código utilizado no cálculo termo-hidraulico de reatores a água-leve. Sua primeira versão data de 1967, mas possui diversas atualizações. No IPEN, a versão utilizada é a COBRA 3-C/RERTR, desenvolvida pelo Argonne National Laboratory em 1980, que é uma versão modificada do programa COBRA 3-C/MIT para combustível tipo vareta de reatores a água pressurizada - Pressurized Water Reactor (PWR)(CHAO,1980) - em ambiente FORTRAN IV, assim como os demais códigos citados anteriormente (UMBEHAUN,2000).

O código calcula a distribuição de temperatura do elemento combustível em direção radial como função da densidade de potência e transferência de calor, bem como a entalpia e a queda de pressão, em regimes transiente e permanente, resolvendo assim as equações de conservação da massa, energia e momento da mecânica dos fluídos.

4.2.4 Descrição da metodologia

Neste tópico a metodologia de cálculo será descrita em maiores detalhes,

mostrando a integração entre os códigos, seus respectivos arquivos de saída gerados e as entradas dos códigos seguintes ao longo da sequência de cálculos.

Com base em YAMAGUCHI (1997), a geração de seções de choque do combustível é feita com o programa LEOPARD, utilizando o modelo da célula padrão (combustível, revestimento e moderador). No arquivo de entrada LEOPARD.DAT são fornecidos os dados da célula e então gerados os arquivos de saída LEOPARD.SAI e LEOPARD.OUT. As seções de choque são geradas com intervalos de aproximadamente 30 dias de queima até atingir 50 % de queima de U-235 e são gravadas no arquivo não formatado LEOPARD.BIN.

O programa LINXS converte este arquivo em formato compatível com o programa 2DB e grava no arquivo em formato binário LINXS.BIN. O número de grupos de energia é fornecido no arquivo de entrada LINXS.DAT.

Para as regiões onde não há material físsil, utiliza-se o programa HAMMER-TECHNION, pois o LEOPARD permite apenas a modelagem da célula padrão com região extra. As seções de choque macroscópicas dessas regiões são fornecidas ao programa 2DB através do arquivo TWODB.DAT juntamente com outros dados do reator.

Utiliza-se então o código 2DB para simular o histórico de operação do reator em duas dimensões sem barras de controle e seus resultados são gravados no arquivo TWODB.OUT e no arquivo TWODB30.BIN.

Nas trocas de configurações, o ganho de reatividade é calculado com o programa CITATION em duas dimensões sem barras de controle com as seções de choque macroscópicas gravadas pelo 2DB, convertido para o arquivo formatado TDBCIT.DAT pelo programa TDBCIT1. Essas seções de choque são fornecidas ao programa CITATION por meio do arquivo CITATION.DAT juntamente com os dados do núcleo.

Os arquivos FORT.36 e FORT.37 correspondem a cálculos realizados pelo CITATION, porém em três dimensões, a partir do arquivo FORT.51. respectivamente. O programa DENS lê o arquivo FORT.37 e coloca a distribuição de densidade de potência num formato conveniente para análise termo-hidráulica do núcleo.

Com tal metodologia, o grupo de cálculo do IEA-R1 desenvolve relatórios

(RODRIGUES & IGLESIAS,2017) para cada nova configuração do núcleo. Nestes, estão presentes informações de:

- EC e elementos de controle (ECC) cujas posições foram alteradas;
- Ganho de reatividade em relação à configuração anterior;
- Buckling;
- k_{EFF}, coeficientes de reatividade (de temperatura do combustível e do moderador);
- Coeficientes de vazio;
- Parâmetros cinéticos, como as frações efeitivas de nêutrons atrasados (β) e do tempo de geração dos nêutrons prontos (λ);

Neste trabalho os dados dos relatórios fornecidos pelo grupo de cálculo do IEA-R1 servirão como *benchmark* na resolução do POR a partir do código de otimização.

Uma vez definida a metodologia empregada neste trabalho, é possível finalmente apresentar o código de otimização criado para a resolução do POR no IEA-R1, conforme será apresentado no capítulo seguinte.
V O CÓDIGO DE OTIMIZAÇÃO

O presente capítulo trata de mostrar o código de otimização criado para a resolução do POR no reator IEA-R1, suas características, a forma como se integra com os códigos pertencentes à metodologia de cálculos utilizada neste reator, sua validação por meio da utilização de funções de teste e os resultados obtidos por meio do código, bem como uma discussão destes.

5.1 Descrição do código

A justificativa para a utilização do PSO como metaheurística adequada para a resolução do POR se dá, como citado anteriormente, pela sua fácil implementação e o baixo custo computacional exigido. Em um primeiro momento, foi discutida a possibilidade da utilização do método ABC que foi descartada justamente por apresentar desvantagens nas duas caracteristicas citadas sobre o PSO.

É válido, entretanto, ressaltar que a metaheurística utilizada neste trabalho possui limitações, como a convergência prematura para um ponto crítico. Apesar disso, o PSO foi a metaheurística que atendeu melhor as necessidades operacionais deste trabalho, considerando o fato de ser pioneiro no IEA-R1.

O código criado neste trabalho para a resolução do POR se integra aos demais programas que compõem a metodologia de cálculos neutrônicos e termo-hidráulicos do reator IEA-R1 gerando novos arranjos para o núcleo do reator e procurando a solução ótima por meio do algoritmo do PSORK, isto é, utilizando a metaheurística do PSO acoplada ao método das Chaves Aleatórias, descrito no capítulo anterior. A figura 5.1 mostra o fluxograma do código de otimização sendo uma sub-rotina dos programas pertencentes a metodologia de cálculo do IEA-R1.

A entrada é constituída pelos arquivos de saída dos demais códigos que compõem a já citada metodologia de cálculo do núcleo. Tais códigos funcionarão como sub-rotinas do programa, gerando os parâmetros neutrônicos e termo-hidráulicos para as soluções fornecidas pelo código de otimização. A figura 5.2 mostra a interface com o usuário dos códigos da metodologia de cálculo do IEA-R1

já integrados ao código criado.





FONTE: Autor.

O usuário pode ou não selecionar a rotina de otimização e após isso define o tipo de cálculo a ser realizado, bem como o número de partículas e interações realizadas pelo código de otimização.

Assim, o espaço de soluções será gerado e buscará a melhor solução dentro do número de partículas e interações definidas pelo usuário. Tais soluções correspondem a novas configurações do núcleo do IEA-R1 e a melhor delas é exibida no arquivo de saída (TWODB.OUT).

Figura 5.2– Interface da metodologia de cálculo do IEA-R1 integrada ao código de otimização.



FONTE: Autor.

Além das partes que compõem o código, o mesmo possui algumas condições de contorno que devem ser consideradas para garantir uma maior eficácia do mesmo. São elas:

 O fluxo de nêutrons deve ter uma magnitude mínima desejável nos pontos de irradiação;

Não há alteração nas posições 229, 230, 242, 243 (marcadas em vermelho)
 e no elemento de irradiação central (EIBe);

5.2 A função objetivo

Neste sub-tópico é mostrado todo o processo de elaboração da função objetivo a ser otimizada neste trabalho.

Foram consideradas quatro grandezas na determinação da função objetivo: O fator de multiplicação efetivo (k_{EFF}), o fluxo de nêutrons (ϕ) o pico máximo de potência (Pic_{max}) e a variância do pico de potência (Var).

Vale ressaltar que este trabalho busca a maximização desta função objetivo e que para a resolução do POR aqui, além da maximização de k_{EFF} e ϕ , a minimização da variância se fez necessária para garantir a uniformidade do fluxo. Outro ponto importante é garantir que a nova configuração se mantenha dentro dos limites térmicos operacionais, i.e., que a temperatura do revestimento não ultrapasse os 95°C (UMBEHAUN,2016).

A partir das características acima citadas, buscou-se o desenvolvimento da função objetivo (f). Com base nos objetivos já citados anteriormente, i.e., a determinação de uma configuração que proporcione um fluxo de nêutrons maior e com uma distribuição mais estável, a 1^a versão obtida da função objetivo considerava a maximização de k_{EFF} e ϕ e a minimização do pico de potência (*Pic*).

Para a determinação da função, era desejável tornar o problema o mais simples possível, afim de evitar desajustes entre f e o código em si num primeiro momento. Com isso, foi elaborado uma função monoobjetivo, como mostrado a seguir.

$$f = \frac{k_{EFF} \cdot \Phi}{Pic} 10^{-13} \tag{5.1}$$

Observou-se que, no entanto, essa função não conseguia dar uma uniformidade maior ao fluxo, uma vez que ela minimizava o pico, mas não a diferença entre os picos de cada EC. Uma alternativa encontrada foi a de minimizar a variância dos picos, o que leva a função objetivo determinada na equação 5.2.

$$f = \frac{k_{EFF} \cdot \Phi}{Pic. Var} 10^{-13}$$
(5.2)

5.3 Função de teste

É comum o emprego de funções de referência com o pressuposto de que a dificuldade delas corresponde às encontradas em aplicações do mundo real. Por conta disso, elas são usadas para validar e comparar o desempenho de algoritmos de otimização. Há um número considerável delas na literatura, mas não há uma lista padrão ou conjunto de funções de teste que se deva seguir. O trabalho de YANG(2010) faz um apanhado de muitas dessas funções de teste.

Segundo Dieterich e Hartke (2012), qualquer nova abordagem para otimização global deve ser validada a partir de um conjunto de funções de referência. No entanto, para que os novos algoritmos de otimização sejam avaliados imparcialmente, é importante que essas funções possuam diversas propriedades, como:

- 1. Continuidade
- 2. Diferenciabilidade
- 3. Separabilidade, que é uma medida para se avaliar a dificuldade de uma determinada função. É definida, segundo PAIVA(2018) como $\frac{\partial f(\bar{x})}{\partial x_i} = g(\bar{x}_i)h(\bar{x})$
- 4. Escalabilidade, i.e, quando a função pode ser expandida.
- 5. Modalidade.

Tratando-se especificamente do PSO, trabalhos como os de ANGELINE (2005) e SHI & EBERHART (1999) utilizaram até 4 funções de teste para verificarem as suas formas de PSO. Neste trabalho, entretanto, como o método não é original apenas a função esfera será utilizada, dada a sua simplicidade de implementação em comparação as outras funções das referências citadas. No caso de outros trabalhos sobre o POR, WAINTRAUB (2009) utiliza funções de teste afim de validarem seus modelos.

A função esfera se trata de uma função simétrica, convexa, unimodal e contínua. Por não possuir restrições quanto ao número de variáveis consideradas, sua complexidade aumenta à medida que o número de parâmetros considerados aumenta. Analiticamente é dada por:

$$f(x) = \sum_{i=1}^{n} x_i^2$$
(5.4)

de maneira a encontrar o centro da mesma, que se encontra na origem do sistema de coordenadas. A tabela 5.1 mostra o teste considerando a variação de partículas e interações.

partículas	interações	х	у	Z
	10	0,6281	2,24E+00	0,291
10	100	2,81E-02	0,1277	0,1249
10	1000	5,09E-05	6,74E-06	5,45E-04
	10000	4,49E-21	4,22E-22	5,14E-21
	10	2,58E-01	2,90E-01	0,7319
100	100	3,08E-03	8,28E-03	1,10E-02
100	1000	3,24E-09	3,90E-11	3,98E-08
	10000	7,42E-24	4,11E-24	6,48E-24
	10	2,11E-01	1,96E-02	8,26E-02
1000	100	6,40E-04	3,67E-03	4,42E-03
1000	1000	5,89E-10	9,69E-09	1,46E-09
	10000	3,00E-24	6,46E-24	1,31E-23
	10	6,71E-03	2,36E-03	1,03E-02
10000	100	4,58E-04	2,86E-04	6,17E-04
10000	1000	5,08E-11	2,20E-11	5,11E-10
	10000	2,44E-22	5,03E-24	3,82E-24

Tabela 5.1 - Resolução para o problema da esfera com diferentes valores de partículas e interações.

FONTE: Autor.

O algoritmo funcionou dentro do esperado, convergindo para a origem do sistema de coordenadas à medida que os valores das partículas e, principalmente, das interações eram aumentados. Foram feitas simulações com valores maiores, chegando até a 100000 interações/partículas. Entretanto os resultados ficam limitados em decorrência da própria precisão do FORTRAN.

5.4 Resolução do POR utilizando o código de otimização

Esta seção mostra e discute os dados que compõem todo o processo de resolução do POR para o reator IEA-R1.

Sendo assim, foi feito um estudo acerca da função objetivo já determinada conforme processo mostrado no sub-item 5.2 e as grandezas que a compõem. Isso permite uma compreensão maior de todo o trabalho realizado pelo autor em busca da melhor configuração possível que seja capaz de atender aos requisitos para a resolução do POR para o IEA-R1.

Além disso, fatores determinantes como o tempo de cálculo para cada combinação de partículas e iterações, quantidade de EC trocados de posições também são avaliados e a queima do combustível também são analisados, apesar de não serem elementos presentes na otimização.

5.4.1 Estudo da Função objetivo

Como primeira análise, buscou-se observar o comportamento da função objetivo de maneira a garantir a convergência de suas soluções. Para isso, foram calculados casos em que houve apenas a variação do número de partículas e, em outros, o número de iterações. Neste caso, foi considerado um conjunto de partículas p = [20,30,40,50,60,70] e iterações t = [500,1000,1500]. A figura 5.3 mostra o comportamento da função objetivo em função do número de iterações.

Figura 5.3 – Comportamento da função objetivo em relação ao número de iterações para 20, 30, 40, 50, 60 e 70 partículas



Fonte: Autor

É possível observar que no intervalo entre 20 e 50 partículas os valores da função objetivo estão na mesma ordem de grandeza, entre 300 e 420. Entretanto, os valores da função objetivo diminuem abruptamente em 60 e 70 partículas, independente do número de iterações, com valores inferiores a 70. Tal desempenho pode ser explicado pela rápida convergência para valores maiores de partículas utilizadas.

Para uma melhor compreensão, a figura 5.4 mostra apenas o intervalo entre 20 e 50 partículas.



Figura 5.4 – Comportamento da função objetivo em relação ao número de iterações para 20, 30, 40 e 50 partículas

Fonte: Autor

É possivel observar neste caso que, apesar de se situarem numa mesma ordem de grandeza, os valores obtidos para as funções objetivo aumentam com o número de partículas utilizadas dentro do intervalo $20 \le p \le 50$. A figura 5.5 mostra a análise em função do número de partículas.

Figura 5.5 – Comportamento da função objetivo em relação ao número de partículas para 500, 1000, 1500 iterações



Fonte: Autor

A figura 5.5 deixa mais evidente a convergência prematura para os casos com mais partículas utilizadas, conforme mencionado na literatura (LUZ et al,2016). Isso justifica a utilização dos dados entre 20 e 50 partículas, uma vez que a função objetivo para 60 e 70 partículas é consideravelmente menor.

Dando atenção ao intervalo de estudo considerado no parágrafo anterior, a função objetivo fornece os melhores resultados para 30 e 50 partículas, sempre mantendo a relação diretamente proporcional com o número de iterações. Desta forma, vale destacar que a função objetivo no caso de 30 partículas (414,818) foi ligeiramente superior ao caso obtido com 50 partículas (412,627). Entretanto, a diferença temporal entre os dois casos rodados foi de 9h (13,5h a 22,5h).

Outro estudo realizado foi a partir da semente utilizada para a geração do número aleatório que inicializa o código. Foram rodados quatro casos para uma única configuração (30 partículas e 1500 iterações) e os valores de cada função objetivo obtidos são mostradas na tabela 5.2.

Tabela 5.2 – Valores da função objetivo para o caso de 30 partículas e 1500 iterações com sementes distintas

Função objetivo
414,81
375,04
383,62
384,72

Fonte: Autor

5.4.2 Características das configurações obtidas.

Como definido e justificado anteriormente, foram rodados 12 casos de diferentes partículas e iterações, produzindo assim 12 configurações candidatas a melhor solução para o POR do IEA-R1.

Nesta seção são mostrados os resultados obtidos em cada uma dessas configurações, onde os sub-tópicos são divididos pelo número de partículas, i.e., 20, 30, 40 e 50 e em cada um deles são mostrados os casos variando o número de iterações.

Além de apresentar os valores nominais da função objetivo, valores como o K_{EFF}, fluxo, pico de potência, variância, o número de EC trocados de posição, a temperatura máxima no revestimento do EC e a queima também foram analisados.

5.4.2.1 – Características das configurações obtidas: Casos com 20 partículas

Neste sub-tópico são mostrados os resultados referentes às configurações obtidas por meio de três casos de partículas e iterações: 20 partículas e 500 iterações; 20 partículas e 1000 iterações e 20 partículas e 1500 iterações. A tabela 5.3 mostra os valores da função objetivo e das grandezas que a compõem.

	Função		fluxo		
iterações	objetivo	Keff (BOC)	(10 ¹⁴)	Pico	Variância
500	343,20	1,073399	1,1705	1,59	0,0545
1000	374,80	1,078547	1,1834	1,562	0,0658
1500	374,80	1,069812	1,1628	1,635	0,0442

Tabela 5.3 – Valores da função objetivo e as grandezas que a compõem para casos com 20 partículas

Fonte: Autor

Outra análise realizada foi observar o K_{EFF} ao longo de todo o ciclo de operação do reator. Como o IEA-R1 funciona 8h/dia durante três dias da semana, considerouse um dia de operação equivalente a uma semana corrida.

Deste modo, considerando a potência atual de 3,5MW⁴, foram analisados os valores para um dia, quatro dias (um mês), 12 dias (três meses), 24 dias (seis meses) e 48 dias (12 meses, aproximadamente). Na figura 5.6 são mostrados os valores do K_{EFF} em decorrência da queima do combustível.

Figura 5.6 – Comportamento do K_{EFF} decorrente da queima do combustível para 20 partículas



Fonte: Autor

⁴ Com base no fornecido pelo código CITATION

O comportamento do K_{EFF} em relação à queima é o mesmo em todos os casos, entretanto, cabe observar que o caso com 1500 partículas gerou um K_{EFF} no inicio de ciclo menor, o que levou a um menor valor para o mesmo no final do ciclo.

A próxima análise tratou sobre os posicionamentos dos EC de maior pico de potência em cada configuração. Com o objetivo de deixar mais fácil a identificação, a figura 5.7 mostra a forma de identificação dos elementos presentes no núcleo do IEA-R1 (RODRIGUES & IGLESIAS, 2017), onde a leitura é feita como se fosse uma matriz com posições denotadas pelas coordenadas (linha,coluna).

4					
5		ECC		ECC	
6			ElBe		
7		ECC		ECC	
8					
	3	4	5	6	7

Figura 5.7 – Esquema de posicionamento dos elementos no núcleo do IEA-R1.

Fonte: (RODRIGUES & IGLESIAS, 2017) – Adaptado pelo autor.

Deste modo, a figura 5.8 mostra o posicionamento do EC de maior pico de potência em cada uma das configurações obtidas nos casos com 20 partículas.

Figura 5.8 – Posições dos EC com maior valor de pico para 20 partículas



Fonte: Autor

Em todos os três casos, o maior pico foi encontrado no EC localizado na posição (6,4), que é a posição onde o maior pico costuma ser encontrado normalmente em cálculos desse tipo, o que leva a conclusão de que não foi possível encontrar um modelo diferente de configuração nos casos com 20 partículas.

Porém, para melhor observar como o código buscou novas configurações, foram avaliadas as mudanças de posições dos EC em cada caso. A tabela 5.4 mostra justamente o número de EC que, depois de estabelecida a nova configuração pelo código de otimização, foi colocado numa posição diferente à da configuração original para cada um dos números de iteração utilizados.

iterações	trocas de posição
500	19
1000	17
1500	19

Tabela 5.4 – Trocas de posição de EC nos casos de 20 partículas

Fonte: Autor

Com base na tabela 5.4, quase todos os EC são trocados de posição. Conforme será visto adiante, isso é uma característica que o código não conseguiu alterar.

Por fim, como última análise, são observadas as temperaturas máximas do revestimento dos EC, por meio do código COBRA. A temperatura limite do revestimento, segundo UMBENHAUM (2017) é de 95°C e é observada a maior temperatura possível no revestimento de maior densidade de potência. A tabela 5.5 mostra essas temperaturas para cada uma das iterações e a diferença dessas para o valor limite.

Tabela 5.5 – Temperaturas máximas no EC de maior densidade de potência e a comparação com a temperatura limite do material do revestimento nos casos de 20 partículas

iterações	T máx. revestimento(°C)	diferença do limite
500	95,2	0,2
1000	94,5	-0,5
1500	97,5	2,5

Fonte: Autor

Para 20 partículas, apenas o caso de 1000 iterações não teve uma temperatura superior ao limite. Importante dizer que este limite termo hidráulico é um fator determinante para a viabilidade da configuração determinada como solução do POR.

5.4.2.2 – Características das configurações obtidas: Casos com 30 partículas

Neste sub-tópico são mostrados os resultados referentes às configurações obtidas por casos com 30 partículas. A tabela 5.6 mostra os valores da função objetivo e das grandezas que a compõem.

	Função		fluxo		
iterações	objetivo	Keff (BOC)	(10 ¹⁴)	Pico	Variância
500	402,627	1,060928	1,1727	1,503	0,0423
1000	402,627	1,070038	1,1612	1,582	0,0509
1500	414,808	1,065663	1,1594	1,539	0,0330

Tabela 5.6 – Valores da função objetivo e as grandezas que a compõem para casos com 30 partículas

Fonte: Autor

Vale destacar aqui que o caso de 30 partículas e 1500 iterações foi o que apresentou o maior valor para a função objetivo. Esse caso será melhor estudado posteriormente.

Outro ponto interessante é o aumento no valor das funções objetivo quando comparadas aos casos de 20 partículas, porém com uma minimização dos valores do pico de potência, o que indica soluções mais interessantes para o POR e com uma garantia maior de respeito aos limites térmicos de segurança.

De maneira semelhante aos casos com 20 partículas, o comportamento do K_{EFF} é analisado ao longo de todo ciclo de operação anual do reator, conforme mostrado na figura 5.9.

Figura 5.9 – Comportamento do K_{EFF} decorrente da queima do combustível para 30 partículas



Assim como nos casos com 20 partículas, o comportamento do K_{EFF} em relação à queima mantém se o mesmo, mas dessa vez com valores maiores do K_{EFF} para 1500 iterações, superando inclusive os valores obtidos para 500 iterações. Novamente os maiores valores do K_{EFF} foram para t = 1000.

A seguir, a figura 5.10 mostra os posicionamentos dos EC de maior pico de potência em cada configuração obtida nos casos de 30 partículas.

Figura 5.10 – Posições dos EC com maior valor de pico para 30 partículas



Fonte: Autor

Diferentemente do caso com 20 partículas, os picos se distribuíram pelo núcleo nas posições (5,5) (t = 500), (6,4) (t = 1000) e (6,4) (t = 1500), o que mostra a tentativa do código em procurar configurações com padrões distintos ao da configuração do input.

A tabela 5.7 mostra a seguir a quantidade de EC trocados de posição na nova configuração gerada pelo código de otimização. Aqui, vale destacar que, apesar de ainda apresentarem um número grande de trocas de posições, os casos gerados para 30 partículas mostraram uma melhora nesse quesito, sendo em especial o caso de 30 partículas e 1500 iterações aquele que obteve o menor número de trocas de posições.

Tabela 5.7 – Trocas de posição de EC nos casos de 30 partículas

iterações	trocas de posição
500	19
1000	18
1500	14

Fonte: Autor

A análise final deste sub-tópico mostra, por meio da tabela 5.8 as temperaturas máximas no revestimento do EC de maior densidade de potência.

iterações	T máx. revestimento(°C)	diferença do limite
500	93,5	-1,5
1000	94,7	-0,3
1500	93,1	-1,9

Tabela 5.8 – Temperaturas máximas no EC de maior densidade de potência e a comparação com a temperatura limite do material do revestimento nos casos de 30 partículas

Fonte: Autor

Nestes três casos a temperatura limite foi respeitada, apesar de um aumento na mesma se comparada com a temperatura da configuração original usada como benchmark deste trabalho (91,2°C). Va le destacar que a menor temperatura das três calculadas foi a do caso de 30 partículas e 1500 iterações.

5.4.2.3 – Características das configurações obtidas: Casos com 40 partículas

Neste sub-tópico são mostrados os resultados referentes as configurações obtidas por casos com 40 partículas. A tabela 5.9 mostra os valores da função objetivo e das grandezas que a compõem.

Tabela 5.9 – Valores da função objetivo e as grandezas que a compõem para casos com 40 partículas

	Função		fluxo		
iterações	objetivo	Keff (BOC)	(10 ¹⁴)	Pico	Variância
500	368,08	1,064503	1,1591	1,524	0,0367
1000	379,84	1,073031	1,1890	1,558	0,0650
1500	399,34	1,066549	1,1709	1,576	0,0330

Fonte: Autor

Conforme mostrado anteriormente na figura 5.5, os valores das funções objetivo nos casos com p = 40 são menores aos casos de p =30, o que, por si só já as excluem do processo de melhores soluções. Entretanto, vale observar que a variância, em especial para t = 500 e t = 1500 é consideravelmente minimizada, bem como os picos de potência se apresentam em valores abaixo de 1,6, o que garante que os limites de temperatura foram respeitados.

Na figura 5.11 o comportamento do K_{EFF} é analisado ao longo de todo ciclo de operação anual do reator, de maneira semelhante aos casos anteriores.



Figura 5.11 – Comportamento do K_{EFF} decorrente da queima do combustível para 40 partículas

Fonte: Autor

Nesta análise os valores de K_{EFF} para o caso com 1000 iterações novamente são os mais altos dentre os três valores de t estudados. Agora, porém, a diferença entre a curva de t =1000 e as demais é maior. Vale observar que o valor do K_{EFF} em BOC para 500 iterações foi mais alto do que o caso de 1500 iterações, mas contrariando os estudos de queima anteriores a queda do valor do K_{EFF} do caso t = 500 foi maior do que a queda no caso t = 1500.

Mantendo a sequência das análises, a figura 5.12 mostra os posicionamentos dos EC de maior pico de potência em cada configuração obtida nos casos de 40 partículas.



Figura 5.12 - Posições dos EC com maior valor de pico para 40 partículas

Fonte: Autor

Observa-se que neste caso, apesar da predominância da posição (6,4) em receber o EC de maior pico, em um dos casos (t = 1000) o maior pico ficou na posição (5,5). Isso reforça o argumento de que quão mais maximizada a função objetivo é, maiores as chances de serem encontradas novas configurações para o

núcleo, inclusive na sua distribuição de potência.

A análise do número de EC que tiveram suas posições trocadas em relação a configuração original para os casos com p = 40 é feita a seguir e os resultados são exibidos na tabela 5.10, onde apenas um EC não teve a sua posição alterada nas três simulações realizadas.

iterações	trocas de posição
500	19
1000	19
1500	19

Tabela 5.10 – Trocas de posição de EC nos casos de 40 partículas

Fonte: Autor

Por fim, as temperaturas máximas no revestimento são mostradas na tabela 5.11, onde é possível observar que as três temperaturas apresentam considerável disparidade. Em especial, no caso para 1000 iterações, a temperatura limite sequer é respeitada.

Tabela 5.11 – Temperaturas máximas no EC de maior densidade de potência e a comparação com a temperatura limite do material do revestimento nos casos de 40 partículas

iterações	T máx. revestimento(°C)	diferença do limite
500	92,7	-2,3
1000	95,8	0,8
1500	94,5	-0,5

Fonte: Autor

5.4.2.4 – Características das configurações obtidas: Casos com 50 partículas

Neste sub-tópico são mostrados os resultados referentes as configurações obtidas por casos com 50 partículas. A tabela 5.12 mostra os valores da função objetivo e das grandezas que a compõem.

	Função		fluxo		
iterações	objetivo	Keff (BOC)	(10 ¹⁴)	Pico	Variância
500	368,72	1,064015	1,1664	1,453	0,0473
1000	368,72	1,065419	1,1677	1,465	0,0486
1500	412,63	1,057811	1,1610	1,330	0,0310

Tabela 5.12 – Valores da função objetivo e as grandezas que a compõem para casos com 50 partículas

Fonte: Autor

Aqui a combinação formada por 50 partículas e 1500 iterações gerou o 2° maior valor das funções objetivo dos 12 casos analisados neste trabalho. Entretanto, a diferença entre os valores obtidos para os casos de 500 e 1000 iterações é maior se comparada aos casos para 30 partículas: o valor médio da função objetivo foi 383,352 \pm 19,516 para os casos de 50 partículas e de 406,687 \pm 5,414 para os casos de 30 partículas. Já os valores do pico apresentam-se consideravelmente menores.

O comportamento do K_{EFF} em durante o ciclo do reator é analisado na figura 5.13.



Figura 5.13 – Comportamento do KEFF decorrente da queima do combustível para 50 partículas

Fonte: Autor

Ao realizar a comparação entre todos os tipos de partículas do conjunto estudado neste trabalho, os casos com 50 partículas apresentam uma disparidade maior quando t = 1500. Para t = 500 e t = 1000, os valores de K_{EFF} ao longo de todo o ciclo são muito próximos, mas ainda tendo para 1000 partículas os melhores resultados, o que prova uma tendência do código, quando executado com 1000

partículas, privilegiar o KEFF.

Na figura 5.14 são mostrados os posicionamentos dos ECs de maior pico de potência em cada configuração obtida nos casos de 50 partículas. É possivel observar que, assim como os casos de 30 partículas, houve uma distribuição do posicionamento do EC com pico máximo. Para p = 50, os picos máximos foram localizados nas posições (6,4) (t = 500 e t = 1000) e (6,7) (t = 1500).





Fonte: Autor

Os números de EC que tiveram suas posições trocadas em relação a configuração original para os casos com p = 50 são mostrados exibidos na tabela 5.13. Aqui, a configuração que gerou menos alterações foi a de 50 partículas e 1000 configurações, com 17 trocas.

	Tabela 5.13 -	Trocas c	de posição	de EC nos	casos de 50	partículas
--	---------------	----------	------------	-----------	-------------	------------

iterações	trocas de posição
500	19
1000	17
1500	19

Fonte: Autor

As temperaturas máximas nos EC de maior densidade de potência nos casos com 50 partículas encerram esse ciclo de análises. Aqui é possível notar uma queda considerável no valor de T_{MAX} em todos os casos rodados, o que reforça a capacidade do código em encontrar soluções que minimizem cada vez mais o problema da temperatura limite.

iterações	T máx. revestimento(°C)	diferença do limite
500	89,9	-5,1
1000	91,2	-3,8
1500	88,3	-6,7

Tabela 5.14 – Temperaturas máximas no EC de maior densidade de potência e a comparação com a temperatura limite do material do revestimento nos casos de 50 partículas

Fonte: Autor

A seguir são feitas algumas conclusões sobre as análises realizadas com todas as 12 configurações obtidas. Assim já é possível identificar o comportamento do código, com base nas configurações e atestar a melhor solução encontrada dentre as 12 calculadas.

5.4.2.5 – Características das configurações obtidas: Conclusões preliminares

A partir do estudo feito com os resultados obtidos por todas as configurações geradas como melhores soluções para cada um dos 12 casos analisados, foi possível chegar a algumas conclusões sobre o comportamento do código e da resolução do POR para o IEA-R1:

- O código possui uma tendência de privilegiar a maximização do K_{EFF} em detrimento das outras grandezas nos casos em que t = 1000, independente do número de partículas.
- À medida que a função objetivo apresentou melhores valores, i.e., mais altos, o EC de maior pico de potência passou a ser encontrado em posições diferentes da tradicional posição (6,7).
- O código não encontrou nenhuma configuração cujos posicionamentos dos EC seja semelhante à configuração original (260). Um valor maximizado da função objetivo não forneceu nenhuma garantia nesse quesito, uma vez que ao compararmos os dois casos de melhor valor para a função objetivo, foram trocadas 19 posições para o caso de 50 partículas e 1500 iterações e 14 posições

para o caso de 30 partículas e 1500 iterações.

- Apesar do maior valor do fluxo ser encontrado no caso 40 partículas e 1000 iterações (1,1890x10¹⁴), os casos onde p = 50 tiveram maior convergência. O valor médio do fluxo foi de (1,1650 \pm 0,0027)x 10¹⁴ neste caso, enquanto nos demais os valores médios foram (1,1730 \pm 0,0027) x 10¹⁴ para p = 40, (1,1644 \pm 0,0055) x 10¹⁴ para p = 30 e (1,1722 \pm 0,0022) x 10¹⁴ para p = 20.
- As temperaturas e os picos máximos foram consideravelmente minimizados nos casos com 50 partículas. Para 30 partículas, a melhora foi em especial no caso com 1500 iterações.
- Os casos com 1500 iterações sempre obtiveram os melhores valores para a variância, tendo também para o caso 50 partículas e 1500 iterações o menor – e melhor – valor (0,0310). Os casos de 40 e 30 partículas obtiveram o mesmo valor para a variância (0,0330).

Uma vez feitas tais considerações, o autor então optou por fazer uma análise mais aprofundada de duas configurações: 30 partículas e 1500 iterações e 50 partículas e 1500 iterações e comparar seus resultados com a configuração 260 original, com o objetivo de observar se houve algum ganho e, por conseguinte, uma resolução para o POR no IEA-R1, mesmo que não definitiva.

Tais escolhas se justificam em dois pontos: Ambas as configurações produziram resultados relevantes se observados em separado, tendo maior destaque o caso de 50 partículas. Entretanto, foi o caso de 30 partículas que obteve o máximo valor para a função objetivo.

5.4.4 Estudo das melhores configurações obtidas pelo código de otimização

Nesta parte do trabalho foram analisadas as duas configurações citadas como melhores soluções no tópico anterior. Diferentemente das análises realizadas no tópico anterior, aqui as configurações calculadas serão comparadas com a configuração 260, que serviu de entrada para o código de otimização no sentido de averiguar se estas conseguiram gerar soluções para o POR do IEA-R1.

A configuração 260 é uma das mais recentes que possui apenas 2 steps de queima, o que torna os cálculos do 2DB mais simples e proporciona além de um ganho computacional, a possibilidade da utilização de valores maiores de partículas e iterações.

5.4.4.1 – Caso 30 partículas e 1500 iterações

Como inicio de estudo foi feita uma comparação entre o posicionamento dos EC das duas configurações, a 260 e a obtida pelo software quando a entrada é a configuração 260, mostradas na figura 5.15.

Figura 5.15 – Comparação entre as configurações 260: a) Original. b) Calculada com 30 partículas e 1500 iterações

а				b							
	3	4	5	6	7		3	4	5	6	7
	EC-218	EC-224	EC-216	EC-234	EC-231		EC-235	204 - MR	EC-216	EC-217	EC-234
4	12,870/14,06	9,664/10,59	27,126/29,31	3,829 /4,21	6,285/6,90	4	0,0/0,0	39,815/42,65	27,126/29,31	23,537/25,50	3,829 /4,21
	EC-214	ECC-229	EC-215	ECC-242	EC-217		EC-223	ECC-229	EC-215	ECC-242	EC-224
5	25,713/27,81	24,572/26,60	29,661/31,99	2,630/2,90	23,537/25,5	5	11,572/12,66	24,572/26,60	29,661/31,99	2,630/2,90	9,664/10,59
	EC-235	EC-207	EIBE	EC-206	EC-213		EC-231	EC-207	EIBE	EC-206	EC-214
6	0,0/0,0	40,281/43,13		35,420/38,05	28,088/30,3	6	6,285/6,90	40,281/43,13		35,420/38,05	25,713/27,81
	EC-223	ECC-243	EC-210	ECC-230	EC-225		EC-227	ECC-243	EC-218	ECC-230	EC-225
7	11,572/12,66	0,0/0,0	31,270/33,69	23,655/25,63	15,767/17,1	7	5,548/6,10	0,0/0,0	12,870/14,06	23,655/25,63	15,767/17,19
	EC-227	204 - MR	EC-232	EC-233	EC-226		EC-232	EC-213	EC-233	EC-210	EC-226
8	5,548/6,10	39,815/42,65	7,646/8,39	3,928/4,32	8,343/9,15	8	7,646/8,39	28,088/30,33	3,928/4,32	31,270/33,69	8,343/9,15

Fonte: Autor

A tabela 5.15 mostra uma comparação entre a configuração definida como a de melhor função objetivo e a configuração 260 do IEA-R1. Nela, são exibidos os dados de K_{EFF}, fluxo, pico de potência e variância.

Tabela 5.15 – Valores de grandezas otimizadas no caso 30 partículas e 1500 iterações comparadas com a configuração 260 do reator IEA-R1.

	config. 260	calculada
K _{EFF} (BOC)	1,06289	1,065663
pico	1,429	1,539
fluxo (10 ¹⁴)	1,1558	1,1594
variância	0,0343	0,0330

Fonte: Autor

É possível notar que, para esta configuração, apenas o pico possui valores piores se comparados aos da configuração 260. Vale observar que é aqui analisado o inverso do pico e da variância, uma vez que um dos objetivos deste POR é o de minimizar estas duas grandezas individualmente.

A figura 5.16 ilustra isso com um gráfico de radar dos valores normalizados, com o intuito de mostrar qual das duas configurações ocupa a maior área (uma vez que se trata de um problema de maximização da função objetivo).O gráfico compara as grandezas mostradas na tabela 5.15 com os valores normalizados..

Figura 5.16 – Desempenho da função objetivo no POR comparado a configuração 260 do reator IEA- R1 para o caso p = 30, t = 1500.



Fonte: Autor

Com base na figura 5.16, é possível notar que a configuração calculada tem ganho na variância (3,8%) e pequenos ganhos no K_{EFF} (0,26%) e no fluxo (0,31%), mas perda no pico (7,14%). Ressalta-se que o código atingiu a otimização necessária e a configuração calculada não viola os limites de temperatura.

Continuando a análise, a figura 5.17 mostra a comparação entre as distribuições da densidade de potência em 2D e 3D para as duas configurações estudadas aqui. Nos gráficos a seguir os valores para o EIBe (posição central) são desconsiderados.

As duas configurações têm distribuições semelhantes, tendo picos menores nas regiões externas do núcleo, como nas posições (6,3) e (7,3). Além disso, todo o entorno do pico máximo (localizado em ambos na posição (6,4)) estava acima de um.

Entretanto, destaca-se, a maior diferença entre o pico máximo e a região do entorno para o caso calculado, uma vez que o pico máximo encontrado foi maior.



Figura 5.17 – Distribuição da densidade de potência em 2D e 3D para a configuração 260 (superior) e a calculada (inferior)

Fonte: Autor

Em seguida é feita uma comparação entre as distribuições de temperatura os EC de maior densidade de potência, conforme mostrado na figura 5.18.

Já era conhecido o fato que a temperatura máxima do EC de maior densidade de potência na configuração 30 partículas e 1500 iterações possuía um maior valor em relação à mesma medida na configuração 260. Entretanto, a figura 5.18 mostra que as temperaturas se distribuem igualmente ao longo do EC, não ultrapassando o valor limite de 95°C. Figura 5.18 – Comparação entre as distribuições de temperatura no EC de maior densidade de potência nas configurações 260 e calculada com 30 partículas e 1500 iterações e a temperatura limite.



Fonte: Autor

A comparação final foi feita observando-se o comportamento do K_{EFF} ao longo do ciclo de operação do reator, conforme mostrado pela figura 5.19. Os intervalos de queima utilizados foram os mesmos feitos nas análises dos itens 5.4.3.

Figura 5.19 – Comparação do Comportamento do K_{EFF} decorrente da queima do combustível entre as configurações 260 e calculada para 30 partículas e 1500 iterações.



Fonte: Autor

Não houve mudança no perfil da queima, i.e., o K_{EFF} encontrado na configuração de 30 partículas em BOC, por ser maior do que o K_{EFF} da configuração 260 sob a mesma condição, obteve um valor maior após 48 dias (1152h) de queima.

5.4.4.2 – Caso 50 partículas e 1500 iterações

A comparação entre a configuração 260 utilizada como entrada do código de otimização e no caso de 50 partículas e 1500 iterações foi feita exatamente da mesma forma a realizada no item 5.4.4.1. Deste modo, a figura 5. 20 mostra a disposição dos ECs nas duas configurações estudadas neste tópico e na sequência a tabela 5.16 exibe os dados de K_{EFF}, fluxo, pico de potência e variância para as duas configurações.

Figura 5.20 – Comparação entre as configurações 260: a) Original. b) Calculada com 50 partículas e 1500 iterações

		а			b
3	4	5	6	7	3 4 5 6
EC-218	EC-224	EC-216	EC-234	EC-231	EC-227 EC-225 EC-210 EC-223
12,870/14,06	9,664/10,59	27,126/29,31	3,829 /4,21	6,285/6,90	5,548/6,10 15,767/17,19 31,270/33,69 11,572/12,66
EC-214	ECC-229	EC-215	ECC-242	EC-217	EC-206 ECC-229 EC-207 ECC-242
25,713/27,81	24,572/26,60	29,661/31,99	2,630/2,90	23,537/25,5	35,420/38,05 24,572/26,60 40,281/43,13 2,630/2,90
				0	
EC-235	EC-207	EIBE	EC-206	EC-213	EC-217 EC-215 EIBE EC-214
0,0/0,0	40,281/43,13		35,420/38,05	28,088/30,3	23,537/25,50 29,661/31,99 25,713/27,81
				3	
EC-223	ECC-243	EC-210	ECC-230	EC-225	EC.218 ECC.243 EC.234 ECC.230
11 572/12 66	0.0/0.0	21 270/22 60	22 655/25 62	15 767/17 1	12 970/14 06 0.0/0.0 2 920 /4 21 22 655 /25 62
11,572/12,00	0,070,0	31,270/33,09	23,033/23,03	13,707,17,1	12,870/14,00 0,070,0 3,825/4,21 23,033/23,03
			n	3	
EC-227	204 - MR	EC-232	EC-233	EC-226	204 - MR EC-232 EC-224 EC-231
5.548/6.10	39.815/42.65	7.646/8.39	3.928/4.32	8.343/9.15	39.815/42.65 7.646/8.39 9.664/10.59 6.285/6.90
-,, 0,20		.,	-,, .,	-,, -, -, -, -, -, -, -, -, -, -, -,	

Fonte: Autor

Tabela 5.16 – Valores de grandezas otimizadas no caso 50 partículas e 1500 iterações comparadas com a configuração 260 do reator IEA-R1.

	config. 260	calculada
K _{EFF} (BOC)	1,06289	1,05781
pico	1,429	1,330
fluxo (10 ¹⁴)	1,1558	1,161
variância	0,0343	0,0310

Fonte: Autor

Assim como na análise anterior, a configuração 260 obteve vantagem sobre

a calculada em apenas uma grandeza. Entretanto, ao invés do pico, foi o K_{EFF} que não obteve valores melhores.

Para uma melhor compreensão, a figura 5.21 mostra o gráfico em formato de radar com os valores das grandezas normalizados.

Figura 5.21 – Desempenho da função objetivo no POR comparado a configuração 260 do reator IEA- R1 para o caso p = 50, t = 1500.



Fonte: Autor

Fica claro que a área ocupada no caso da configuração calculada é consideravelmente maior do que o da configuração 260, fato esse que ratifica esta solução como a melhor das encontradas, mesmo esta tendo um valor para a função objetivo ligeiramente menor do que o do caso de p = 30, t = 1500.

A perda do K_{EFF} em relação à configuração original é pequena (0,48%) e os ganhos de pico (6,93%) e variância (9,62%) são expressivos. O fluxo calculado possui pequena vantagem em relação ao da configuração original (0,45%).

As distribuições de densidade de potência das duas configurações são exibidas em 2D e 3D na figura 5.22, observando que aqui a disposição das configurações possui certa diferença, uma vez que os picos na configuração 260 localizam-se de maneira mais isolada, enquanto os da configuração calculada aglutinam-se nas regiões centrais formando uma espécie de "planalto". Além do ponto mencionado acima, é importante observar que esses picos possuem valores menores se comparados aos poucos picos da configuração 260, o que acarreta em uma economia maior do combustível no aspecto geral.

Os picos de menor intensidade continuam a localizar-se nos EC mais externos, como no caso das posições (4,3),(8,3),(4,7) e (8,7). Para os casos de 30 partículas e 1500 iterações, bem como a configuração 260 original, tal tendência se mantém. Entretanto, vale notar que o gradiente entre o pico dessas posições mais externas citadas e as posições de maiores picos é consideravelmente maior nas outras configurações se comparada ao caso p = 50, t = 1500.

A figura 5.23 mostra a análise do perfil de temperatura ao longo do EC para as duas configurações comparadas. Assim como no item 5.4.4.1, já era conhecido o fato da temperatura máxima da configuração calculada (no caso, p =30 e t =1500) respeitar os limites térmicos.



Figura 5.22 – Distribuição da densidade de potência em 2D e 3D para a configuração 260 (superior) e a calculada (inferior)

Fonte: Autor

Entretanto, neste item, além de respeitar o limite de temperatura, o perfil calculado para o caso p = 50, t = 1500 possui valores menores aos da configuração 260, o que é um claro reflexo da efetiva minimização do pico pelo código de otimização.





Fonte: Autor

De maneira semelhante ao apresentado no item 5.4.4.1, não há nenhuma alteração no comportamento do K_{EFF} ao longo do ciclo de operação do reator. Porém, no caso mostrado pela figura 5.24, os menores valores são os da configuração calculada.

Figura 5.24 – Comparação do Comportamento do K_{EFF} decorrente da queima do combustível entre as configurações 260 e calculada para 50 partículas e 1500 iterações.



Fonte: Autor

VI CONCLUSÕES E COMENTÁRIOS

O capítulo final desta tese trata das conclusões obtidas com este trabalho e sugestões para futuras melhorias e/ou trabalhos semelhantes.

Após análise de todos os resultados, pode ser concluído que:

- O código é de otimização é capaz de resolver o POR para o IEA-R1 conforme previsto. Algumas configurações obtidas como melhores soluções apresentam vantagens em relação à configuração original.
- Apesar dos ganhos, se faz necessário implantar pesos nas grandezas avaliadas. Os dados mostram que o K_{EFF} tem uma relevância maior na fitness do que as demais grandezas. Esse fato explica o porquê da configuração p =50, t = 1500 possuir um valor da função objetivo ligeiramente menor do que a da configuração p = 30, t = 1500.
- O código é eficaz na busca de soluções de padrões de posicionamento diferentes ao padrão de carregamento vigente, levando, inclusive, ao pico máximo se localizar na região mais externa do núcleo.
- O fato acima leva o autor a concluir que o padrão atual de carregamento dos EC no núcleo do IEA-R1 pode ser melhorado com uma grande reformulação do posicionamento dos EC.
- Apesar disso, poderia ser interessante para um ponto de vista mais prático a inserção de uma restrição sobre o número de trocas de posições do EC, visto que o código não encontrou por si só configurações novas com poucas trocas de posições.
- As melhores soluções não ferem e algumas inclusive melhoram os padrões de segurança termo hidráulicos, mesmo com aumento do fluxo.

- Mesmo com uma diminuição do K_{EFF} na melhor solução encontrada (p = 50, t = 1500) a operação do reator não seria prejudicada, conforme mostrado na análise de queima da figura 5.23.
- Com a criação deste código é estabelecido também um primeiro benchmark para este tipo de problema no IEA-R1. É perfeitamente possível que os resultados sejam melhorados utilizando outras meta-heurísticas puras ou em formato híbrido.
- Conforme mostrado em anexo, o código não possui uma interface amigável.
 Fica como sugestão a criação de uma interface que facilite a interação homem-maquina.
- Não foi feito, por razões explicadas ao longo desta tese, simulações por longos períodos. A utilização de um computador exclusivo para a resolução do POR, mesmo sem uma configuração robusta, pode permitir a determinação de soluções consideravelmente melhores.
- Uma sugestão para aprimoramento do trabalho é uma função objetivo da forma $f = \alpha K_{EFF} + \beta \phi + \frac{\gamma}{Pico} + \frac{\delta}{Var}$ onde $\alpha, \beta, \gamma, e \delta$ seriam as constantes que definiriam os pesos de cada uma das grandezas avaliadas.

Referências Bibliográficas

AHMAD, A.; AHMAD, S. Optimization of fuel loading pattern for a Material Test Reactor using Swarm Intelligence. **Progress in Nuclear Energy**,v.103, p.p. 45-50, 2018.

BÄCK, T.; HOFFMEISTER, F.; SCHWEFEL, H. A Survey of Evolution Strategies. **Proceedings of the fourth international conference on genetic algorithms.** 1991.

BAPTISTA, R.P. Otimização de múltiplos ciclos da recarga de reatores tipo PWR usando algoritimos genéticos. **Tese de Doutorado**, UFRJ, Rio de Janeiro, 2009.

BARCELLOS, J.C. Algoritmos Genéticos Comparativos: Um estudo comparativo. **Dissertação de mestrado**. USP, São Paulo, 2000.

BEAN, J.C. Genethic Algorithms and Random Keys for sequencing and optimization. **ORSA Journal of Computing**, 6(2),1994.

BELCHIOR JR, A. Cálculo de harmônicos estáticos bidimensionais com o código CITATION. **Dissertação de mestrado**. Instituto de Pesquisas Energéticas e Nucleares – IPEN, São Paulo, 1992.

BENI, G.; WANG, J. Swarm Intelligence. **Proceedings seventh anual meeting of the robotics society of Japan. RSJ Press**. pp 425-428. Toquio, 1989.

BOUNABEAU, E.; DORIGO, M.; THERAULAZ, G. Swarm Intelligence: From natural to artificial systems. Nova Iorque, **Oxford University Press**, 1999.

BUTENKO, S.; PARDALOS, P. Numerical methods and optimization: An introduction. **CRC Press**, 2008.

CASTRO,L.; BOCCATO,L.; VON ZUBEN, F.; ATTUX, R. Algoritmos Genéticos: Introdução. **Notas de aula da disciplina IA707 – Computação Evolutiva**. FEE-UNICAMP, 2016.

DALLE, H. Cálculo neutrônico do reator TRIGA IPR-R1 utilizando WINSD4 e CITATION. **Dissertação de mestrado**. Centro de Desenvolvimento da Tecnologia Nuclear – CDTN, Belo Horizonte, 1999.

DASGUPTA, D. An Overview of Artificial Immune Systems and their applications. **Springer**, Alemanha, 1993.

DeCHEINE, M. D.; FELTUS, M. A. Fuel management optimization using genetic algorithms and expert knowledge. **Nuclear Science and Engineering**, pp. 188-196, 1996.

DO, B.Q.; NGUYEN, L.P. Application of a genetic algorithm to the fuel reaload optimization for a ressearch reactor. **Applied mathematics and computation**, v.187, p.p. 977-988, 2007.

DOMINGOS, D.B. Cálculos Neutrônicos, Termo-hidráulicos e de Segurança de um Dispositivo para Irradiação de Miniplacas (DIM) de Elementos Combustíveis Tipo Dispersão. **Dissertação de Mestrado,** IPEN, São Paulo, 2010.

EBEHART, R.; KENNEDY, J. A New Optimizer Using Particle Swarm Theory. **Proceedings** of the Sixth International Symposium on MHS'95, pp. 39-43. IEEE, 1995.

FEOKTISTOV, V. Differential evolution. **Springer US**, 2006.

FERREIRA, J.M. Problemas de Otimização Combinatória para União Explícita de Arestas. **Tese de Doutorado,** UFG, Goiania, 2018.

FESTINGER L. A. Theory of Social Comparison Processes. **Human Relations**.;7(2):117-140, 1954

FOWLER, T. B.; VONDY, D. R.; CUNNIGHAM, G. W. Nuclear reactor core analysis code: CITATION. Oak Ridge, USA: Oak Ridge National Laboratory, **ORNL-TM-2496**, Rev. 2, Suppl. 3, 1972.

GABRIEL, P.H.R.; DELBEM, A.C.B. Fundamentos de Algoritmos Evolutivos. **Notas de Aula**, ICMC-USP, São Carlos, 2008.

GALPERIN,A.; NISSAN, E. Application of a heuristic search method for generation of fuel reload configurations. **Nuclear Science and Engineering**, v.99, n.4, pp. 343-352, 1988.

GAMBARADELLA, L. M.; DORIGO, M. Ant-Q: A reinforcement Learning aproach to traveling salesman problem. pp. 252-260. **Proceedings of ML-95, Twelfth Intern. Conf. on Machine Learning**, 1995.

GENEZINI, F.; FERNANDES, A.J.; ZAHN, G.S.; The IEA-R1 62 years of operation: Experiences and lessons learned. **Proceedings of of an International Conference -Research Reactors: Addressing Challenges and Opportunities to Ensure Effectiveness and Sustainability**. Buenos Aires, 2019.

GENEZINI, F. Comunicação privada, 2022.

GOLDBARG, M.; GOLDBARG, E.; LUNA, H.P. Otimização Combinatória e meta-heurísticas: Algoritimos e aplicações. **Elsevier**, 2016.

GOLDBERG, D.E. Genetic Algorithms in search, optimization and machine learning. **Addison Wesley**, 1989.

HASHIMOTO,K. Técnicas de otimização combinatória multiobjetivo aplicadas na estimação do desempenho elétrico de redes de distribuição. **Tese de Doutorado**. EPUSP, São Paulo, 2004.

HERNANDES, F.; YAMAKAMI, A.; TAKAHASHI, M.T.; VERDEGAY, J.L. Um algoritmo para o problema de fluxo maximo em redes com incertezas. **XXXVIII Simpósio Brasileiro de Pesquisa Operacional**, Goiânia, 2006.

HOPFIELD, J.J. Artificial neural networks in **IEEE Circuits and Devices Magazine**, vol. 4, no. 5, pp. 3-10, 1988.

HOLLAND, J.H. Adaptation in natural and artificial systems. **University of Michigan Press**, 1975.

HOLTZ, G.C.C. Traçado automático de envoltórias de esforços em estruturas planas utilizando um algorítmo evolucionário. **Dissertação de Mestrado**. PUC-RJ, Rio de Janeiro, 2005.

HWANG, F.K.; RICHARDS, D.S. Steiner Tree Problems. **Networks**, v.22, n.1, p. 55-89, 1992.

JOÃO, T.G. Proposta de novas configurações para o núcleo do reator IEA-R1 do IPEN/CNEN com combustíveis de alta densidade de Urânio. **Tese de Doutorado**, IPEN, São Paulo, 2016.

KELLERER, H.; PFERSCHY, U.; PISINGER, D. Knapsack Problems. **Springler**, 2004.

KENNEDY, J.; EBERHART, R.; SHI,Y. Swarm Intelligence. **Morgan Kaufmann Publishers**, 2001.

KHOSHAHVAL, F.; ZOLFAGHARI, A.; MINUCHEHR, H.; SEDIGHI, M. PWR fuel management optimization using continuous particle swarm intelligence. **Annals of nuclear energy**, v.37, p.p. 1263-1271, 2010.

KIRKPATRICK, S.; GELATT, C., & VECCHI, M. **Otimizing by Simulated Anneling. Science**, v.220, n. 4598, 671-680, 1982.

KROPACZEK, D.; TURINSKY, P. J. In-Core Nuclear Fuel Management Optimization for Pressurized Water Reactors Utilizing Simulated Annealing, **Nuclear Technology**, v.95, n.1, 9-32, 1991.

LAMARSH, J.R.; BARATTA, A.J. Introduction to Nuclear Engineering – Pearson New International Edition, 3^a edição. **Pearson Education Limited**, 2014.

LARANJO, G.S.; CONTI, T.N.; SANTOS, T.A.; FEDORENKO, G.G.; MAIO, M.F.; CASTRO, V.A. Implementation of the optmization for the methodology of the neutronic calculation and thermo-hydraulic in IEA-R1 Reactor. **International Nuclear Atlantic Conference (Inac 2011).** Belo Horizonte, 2011.

LEITE, V.C. Otimização por enxame de partículas aplicado à estrutura mecânica da mola da grade espaçadora do elemento combustível de um reator nuclear. **Dissertação de Mestrado**, UFRJ, 2017.

LIBÓRIO, F. Algoritmos exatos e heurísticos para os problemas de Steiner e de Conexão de terminais com número restrito de roteadores e elos. **Monografia**. UFRPE, Garanhuns, 2019.

LIMA, A. Recarga de reatores nucleares utilizando redes conectivas de colônias artificiais. **Tese de doutorado**. UFRJ, Rio de Janeiro, 2005.

LINDEN, R. Algoritmos Genéticos – 3ª edição. Editora Ciência Moderna, 2012.

LITTLE, W.; HARDIE JR, R. W. 2DB user's manual - revision I. Battelle Pacific Northwest Laboratory, **BNWL-831**. EUA, 1969.

LORENZ, K. Der Kumpan in der Umwelt des Vogels. **Journal Für Ornithologie**, 83(2), 137–213, 1935.

LUZ, E.F.P.; BECCENERI, J.C.; STEPHANY, S.; VELHO, H. F.; NETO, A.J.; Otimização por Enxame de partículas (*Particle Swarm Optimization*). In: NETO, A.J. et al.(Org), Inteligência Computacional aplicada a problemas inversos de transferência radiativa. Rio de Janeiro, **Editora UERJ**, pp. 139 -153, 2016.

KARABOGA, D. An idea based on honey bee swarm for numerical optimization. **Technical Report TR 06**, Erciyes University, Turquia, 2005.

KASHI, S.; MINUCHEHR, A.; POURSALEHI, N.; ZOLFAGHARI, A. Bat algorithm for the fuel arrangement optimization fuel core. **Annals of Nuclear Energy**, v.64, p.p. 144-151, 2014.

KHOSHAVAL, F.; ZOLFAGHARI, A.; MINUCHEHR, M.; SADIGHI, M.; NOROUZI, A. PWR fuel management optimization using continuous particle swarm intelligence. **Annals of Nuclear Energy**, v.37, p.p. 1263-1271, 2010.

MACHADO, L. Otimização de recarga do Combustível nuclear por Agentes Artificiais. Rio de Janeiro, RJ. **Tese de Doutorado**, UFRJ, 2001.

MACHADO, M.D. Algoritimo Evolucionário PBIL Multi-objetivo aplicado ao problema da recarga de reatores nucleares. **Tese de Doutorado**, UFRJ, 2005.

MAIORINO, J. R. The utilization and operational experience of IEA-R1 Brazilian research reactor. **International Symposium on Resarch Reactor Utilization**, **Safety and Management** (p. 1). Vienna: International Atomic Energy Agency, 1999.
MARTINEZ, J.; SANTOS, S. Métodos Computacionais de Otimização. IMPA, 2ed, 2020.

MAZROU, H.; HAMADOUCHE, M. Development of a supporting tool for a optimal fuel management in research reactors using artificial neural networks. **Nuclear engeneering and design**, v.236, p.p 255-266, 2006.

MEDEIROS, J. Enxame de partículas como ferramenta de otimização em problemas complexos de engenharia nuclear. **Tese de Doutorado**, UFRJ, Rio de Janeiro, 2005.

MEDINA, I.R. Algoritmos Bioinspirados: Una revisión según sus fundamentos biológicos. **Trabalho de conclusão de curso**. Universidade de Granada, Espanha, 2014.

MELO, L.A.M. Comparação de algorítimos de enxame de partículas para a otimização de problemas de larga escala. **Dissertação de Mestrado**, UFG & EMC, 2018.

MÉNDEZ-DIAS, I. ZABALA, P. A branch-and-cut algorithm for graph coloring. **Discrete Applied Mathematics**, V.154, p.p 826-847, 2006.

MÜLLER-MERBACH, H. Heuristics and their design: A survey. **European Journal of Operational Research**,v.1,n.8, p.p. 01-13, 1981.

NETO, A. D.; BECCENERI, J.; VELHO, H. C. **Inteligência computacional aplicada a problemas inversos em transferência radiativa**. Rio de Janeiro: EDUFRJ, 2006.

NICOLIS, G.; PRIGOGINE, I. Self-Organization in Non-Equilibrium Systems. Nova Iorque, **Wiley & Sons**, 1977.

NILSSON, N. J. Principals of artificial intelligence. **Elsevier**, 1982.

OLIVEIRA, I. Inteligência de Enxames aplicada ao problema de otimização de recargas de reatores nucleares a água pressurizada. **Tese de Doutorado**, UFRJ, Rio de Janeiro, 2013.

OZISIK, M.; Heat Conduction – 2nd edition. John Wiley & Sons, 1993.

PAIVA, F. Otimização por enxame de partículas usando uma adaptação de serendipidade. **Editora IFRN**, Natal, 2018.

PARRO, D.P. Desenvolvimento e simulação de um programa computacional para cálculos neutrônicos e termo-hidráulicos do reator de pesquisas IEA-R1. **Dissertação de Mestrado**. Instituto de Pesquisas Energéticas e Nucleares – IPEN. São Paulo, 2018.

PARSOPOULOS, K.; VRAHATIS, M. Particle Swarm Optimization and Intelligence: Advances and Applications. **Information Science Reference**, 2010.

POON, P.W.; PARKS, G.T. Optimizing PWR reload core Designs. **Parallel Problem Solving from Nature 2**, Elsevier Science Publishers B.V., pp. 371- 380, 1992.

POURSALEHI, N.; ZOLFAGHARI, A.; MINUCHEHR M, A. Development of a high order and multi-dimensional nodal code, ACNEC3D, for reactor core analysis. **Ann. Nucl. Energy**, v.55, pp. 211-224, 2013.

PRESTES, A.N. Uma análise experimental de abordagens heurísticas aplicadas ao problema do caixeiro viajante. **Dissertação de mestrado**. UFRN, Natal, 2006.

RODRIGUES, R.D.F. Times assíncronos para a resolução de problemas de otimização combinatória com múltiplas funções objetivo. **Dissertação de mestrado**.UNICAMP, Campinas, 1996.

RODRIGUES, A.C. I. Estudo e projeto de novos cestos com boro para o armazenamento de elementos combustíveis queimados do reator IEA-R1. **Dissertação de Mestrado**, IPEN, 2016.

RODRIGUES, V.G.; IGLESIAS, A.C. Mudança de de configuração no núcleo do reator IEA-R1: Configuração 260 (4,5 MW). **Relatório Técnico**. Grupo de Cálculo – CRPq, IPEN, São Paulo, 2017.

SANTOS-OLIVEIRA, R.; CARNEIRO-LEÃO, A.M. dos A. História da radiofarmácia e as implicações da Emenda Constitucional n. 49. **Revista Brasileira de Ciências Farmacêuticas**, v. 44, p. 377-382, 2008.

SHE YANG, X.; GANDOMI, A. Bat Algorithm: A Novel Approach for Global Engineering Optimization. **Engineering Computations**, Vol. 29 , pp. 464-483, 2012.

SHI, Y.; EBERHART,R. A Modified Particle Swarm Optimizer.**Atas da IEEE International Conference on Evolutionary Computation**. Universidade de Nagoya, pp 69-73, 1998.

SILVA, M.T. Hibridização de métodos exatos e heurísticos para a minimização do atraso ponderado no escalonamento de tarefas em máquinas paralelas. **Dissertação de mestrado**, UFAM, Manaus, 2018.

TANKER, E.; TANKER, A. Aplication of a genetic algorithm to core reload pattern optimization. **Proceedings of the International Conference on Reactor Physics on Reactor Computations**, pp. 559 – 563, 1994.

TERREMOTO, L. A. Fundamentos de Tecnologia Nuclear - **Notas de aula**. São Paulo, 2004.

UMBEHAUN, P.E. Metodologia para análise termo-hidráulica de reatores de pesquisa tipo piscina com combustível tipo placa. **Dissertação de Mestrado**. Instituto de Pesquisas

Energéticas e Nucleares - IPEN. São Paulo, 2000.

UMBEHAUN, P.E. Desenvolvimento de um combustível instrumentado para o reator de pesquisa IEA-R1. **Tese de Doutorado**, IPEN, São Paulo, 2016.

WAINTRAUB, M. Algoritmos paralelos de otimização por enxame de partículas em problemas nucleares. **Tese de Doutorado**, UFRJ, Rio de Janeiro, 2009.

WANG, C.R.; ZHOU, C.L.; MA, J.W. An improved artificial fish-swarm algorithm and its application in feed-forward neural networks. **2005 International conference on machine learning and cybernetics.** p.p. 2890-2894. IEEE, 2005.

WILSON, E. Sociobiology: The New Synthesis. Belknap Press, 1975.

YADAV, R.S.; GUPTA, H.P. Optimization studies of fuel loading pattern for a typical Pressurized Water Reactor (PWR) using particle swarm method. **Annals of nuclear energy**, v.38, p.p. 2086-2095, 2011.

YAMAGUCHI, M.; UMBEHAUM, P.E.; FANARO, L.C. Análise Neutrônica e termo-hidráulica do experimento de operação contínua de 48 horas a 5 MW do reator IEA-R1. XI Encontro de Física de Reatores e Termo-hidráulica, Poços de Caldas, 1997.

YAMAMOTO, A. A Quantitative Comparison of Loading Pattern Optimization Methods for In-Core Fuel Management of PWR. **Journal of Nuclear Science and Technology**, v. 34, n. 4, pp. 339-347, 1997.

YANG, X.-S. Firefly Algorithms for Multimodal Optimization. Lecture Notes in Computer Science, 169–178, 2009.

ANEXOS

ANEXO I: MANUAL DE UTILIZAÇÃO DO CÓDIGO DE OTIMIZAÇÃO

Este manual desceve de maneira sucinta como operar o código de otimização, uma vez que não há uma interface amigável ao usuário e a forma de trabalhar as entradas e saídas de arquivos com o FORTRAN 90 é diferente se comparado a linguagens de programção mais modernas.

Antes de operar o arquivo que contém o código de otimização, é necessário colocar a configuração do núcleo a ser otimizada no arquivo de entrada do 2DB (inp1twodb.dat), conforme mostrado na figura A.1.

Figura A.1 – Configuração a ser otimizada no arquivo de entrada do 2DB.

🧾 inp1twodb - Bloco de Notas														×
Arquivo Editar Formatar Exibir Ajuda														
<pre>'=15** ARRAY BUCKLING MODIFIERS==(NZONES OR NZONES*NGPS)====================================</pre>														^
17**														
'=17** ARRAY BURNUP IN %FISSILE=(NXTAPE)====================================														
'* CONFIGURACAO 260 -IV- Set/17 - Queima - Real														
26.885	5 24.3	324 23.	972 26	.838										
24.571	1 2.0	528 0.	000 23	.654										
12.870	9.6	664 27.	126 3	.828	6.284									
25.712	2	29.	660	2	3.536									
0.000	a 40.2	280	35	.419 2	8.086									
11.571	1	31.	269	1	5.767									
5.548	3 39.8	814 7.	645 3	.927	8.341									
13.989 1	14.358	11.399	12.122	28.461	29.878	3.425	5.045	3.736	3.524					
2.700	2.736			2.482	2.329			2.428	2.181					
0.000	0.000	1.280	1.124			1.532	1.282	0.579	0.575					
0.866	0.712			2.022	1.797			0.809	0.679					
0.156	0.156	0.298	0.240	0.231	0.220	0.258	0.210	0.191	0.181					
•														

Fonte: Autor

O código encontra-se em um arquivo executável chamado auto+inpcobra. Ao abri-lo, a tela mostrada na figura A.2 aparecerá mostrando as entradas que o usuário deverá informar:

- Se o usuário pretende utilizar o código de otimização. Caso não queira, código apenas aplicará a antiga metodologia de cálculos já existente sobre a configuração inserida no arquivo de entrada do 2DB.
- Se devem ser realizados cálculos 2D (via 2DB) ou 3D (via CITATION).
 Entretanto o código de otimização não rodará, mesmo que selecionado, todas as configurações utilizando o citation, uma vez que o custo computacional tornaria-se impraticável para o computador utilizado.

Figura A.2 – Tela de entrada do código de otimização.



Fonte: Autor

- O número de partículas utilizadas no PSO
- Se existem ou não miniplacas no núcleo.
- O número de iterações desejado
- A semente (*Seed*) para a geração aleatória de soluções.

Após a rotina concluída, varios arquivos são gerados. Um deles é o inp3twodb.dat, que é a entrada gerada para a melhor configuração obtida. Após a determinação da melhor configuração, o programa rodará esta em separado aplicando toda a metodologia de cálculo do IEA-R1 (incluíndo o CITATION). Caso o programa rode sem a opção do código de otimização, a entrada do 2DB será justamente esse arquivo.

Os outros arquivos de saída a serem utilizados são o out1twodb.dat, que é a saída de fato do 2DB e o DENS, que é uma saída do CITATION que fornece as densidades de potência radial e axial.

Para a verificação termohidráulica, basta acionar o executável cbieaw e consultar a saída do COBRA (cbout.out). Importante frisar que o pico máximo de potência ja é inserido no arquivo de entrada (cbdat.dat).

INSTITUTO DE PESQUISAS ENERGÉTICAS E NUCLEARES Diretoria de Pesquisa, Desenvolvimento e Ensino Av. Prof. Lineu Prestes, 2242 – Cidade Universitária CEP: 05508-000 Fone/Fax(0XX11) 3133-8908 SÃO PAULO – São Paulo – Brasil http://www.ipen.br

O IPEN é uma Autarquia vinculada à Secretaria de Desenvolvimento, associada à Universidade de São Paulo e gerida técnica e administrativamente pela Comissão Nacional de Energia Nuclear, órgão do Ministério da Ciência, Tecnologia, Inovações e Comunicações.