



UMA ANÁLISE TEÓRICA E COMPUTACIONAL DE ALGORITMOS DE ENXAME DE PARTÍCULAS NA SOLUÇÃO DE PROBLEMA DE DESPACHO ECONÔMICO

GIOVANNA SBOLDRIM PASCOLAT- g.pascolat@unesp.br
UNIVERSIDADE ESTADUAL PAULISTA – UNESP BAURU

LEONARDO NEPOMUCENO – leonardo.nepomuceno@unesp.br
UNIVERSIDADE ESTADUAL PAULISTA – UNESP BAURU

FABIANO BORGES DA SILVA- fabiano.borges@unesp.br
UNIVERSIDADE ESTADUAL PAULISTA – UNESP BAURU

ÁREA: 3. PESQUISA OPERACIONAL.
SUBÁREA: 3.1 – MODELAGEM, SIMULAÇÃO E OTIMIZAÇÃO

RESUMO: O DESPACHO ECONÔMICO É FUNDAMENTAL PARA A OPERAÇÃO EFICIENTE DE SISTEMAS DE GERAÇÃO DE ENERGIA. ESTE PROBLEMA TEM A PRESENÇA DE MÚLTIPLOS MÍNIMOS LOCAIS E PODE SER CARACTERIZADO POR UMA FUNÇÃO OBJETIVO NÃO DERIVÁVEL, O QUE TORNA SUA RESOLUÇÃO DESAFIADORA. ALGORITMOS BASEADOS EM HEURÍSTICAS TÊM SE MOSTRADO EFICAZES DEVIDO À SUA CAPACIDADE DE LIDAR COM ESSAS COMPLEXIDADES. ENTRE ESSES ALGORITMOS, O PARTICLE SWARM OPTIMIZATION (PSO) SE DESTACA POR SUA SIMPLICIDADE E EFICIÊNCIA COMPUTACIONAL. NO ENTANTO, O PSO SOFRE DE PROBLEMAS COMO A POSSIBILIDADE DE NÃO CONVERGÊNCIA E A INCAPACIDADE DE ESCAPAR DE MÍNIMOS LOCAIS. VAN DEN BERGH E ENGELBRECHT (2010) PROPUSERAM UMA SOLUÇÃO, QUE ABORDA O CONCEITO DE INTRODUIR PARTÍCULAS ALEATÓRIAS PARA O ENXAME, PARA TER UM ALGORITMO DE BUSCA GLOBAL. ESSE ALGORITMO FOI CHAMADO DE RANDOMISED PARTICLE SWARM OPTIMIZER (RPSO), ENTRETANTO ESSE ALGORITMO AINDA ENFRENTA DESAFIOS RELACIONADOS À TAXA DE CONVERGÊNCIA. POR OUTRO LADO, O COMPREHENSIVE LEARNING PARTICLE SWARM OPTIMIZER (CLPSO) TEM MOSTRADO ÓTIMOS RESULTADOS, MAS NÃO FOI ENCONTRADA NENHUMA PROVA DE CONVERGÊNCIA. NESTE TRABALHO, APRESENTAMOS UMA SOLUÇÃO HÍBRIDA QUE COMBINA OS PONTOS FORTES DO CLPSO E DO RPSO, POIS ASSIM TEMOS UM MÉTODO COM ÓTIMOS RESULTADOS PARA O DESPACHO ECONÔMICO E QUE TEM GARANTIA DE CONVERGÊNCIA.

PALAVRAS-CHAVES: DESPACHO ECONÔMICO; OTIMIZAÇÃO; PARTICLE SWARM OPTIMIZATION (PSO); CONVERGÊNCIA.

1. INTRODUÇÃO

O problema de despacho econômico é crucial para a operação eficiente de sistemas de geração de energia elétrica, pois visa minimizar os custos de produção de energia enquanto satisfaz as restrições operacionais e de demanda. A importância deste problema reside na necessidade de fornecer energia de maneira econômica, reduzindo os custos operacionais e, consequentemente, os custos para os consumidores finais. No entanto, a complexidade do problema aumenta devido à presença de múltiplos ótimos, decorrentes da complexidade da função objetivo que inclui termos senoidais, o que também inviabiliza o uso de métodos tradicionais de otimização que dependem de derivadas.

Algoritmos de otimização têm se mostrado ferramentas poderosas na resolução do problema de despacho econômico, especialmente aqueles baseados em heurísticas. Esses algoritmos são valiosos por sua capacidade de lidar com os múltiplos ótimos e a não diferenciabilidade da função objetivo. As heurísticas, em particular, oferecem uma abordagem prática para encontrar soluções de boa qualidade em tempo razoável, sem a necessidade de derivadas ou gradientes.

Entre as diversas heurísticas disponíveis, o *Particle Swarm Optimization* (PSO) se destaca por sua simplicidade e eficiência computacional. O PSO simula o comportamento social de um grupo de partículas que se movem pelo espaço de busca em direção às melhores soluções conhecidas. A simplicidade de implementação e a capacidade de convergir rapidamente para soluções de alta qualidade tornam o PSO uma escolha popular para problemas de otimização, incluindo o despacho econômico.

Apesar de suas vantagens, o PSO sofre de problemas significativos, como a possibilidade de não convergência para o ótimo global e a incapacidade de escapar de mínimos locais. Essas limitações ocorrem porque, após várias iterações, as partículas tendem a se agrupar em torno de soluções locais, perdendo a capacidade de explorar novas regiões do espaço de busca. Isso pode resultar em soluções que não são as melhores possíveis, especialmente em problemas com múltiplos mínimos locais.

Para mitigar esses problemas, Van den Bergh e Engelbrecht (2010) propuseram o

Randomised Particle Swarm Optimizer (RPSO), que introduz partículas aleatórias no enxame para manter a diversidade e promover uma busca global mais eficaz. Embora essa abordagem melhore a capacidade do algoritmo de evitar mínimos locais, ela ainda enfrenta desafios relacionados à taxa de convergência. Em outras palavras, a inclusão de partículas aleatórias garante a convergência do problema, mas não se sabe, em termos práticos, quanto tempo levará para convergir.

Outra abordagem para o PSO é o *Comprehensive Learning Particle Swarm Optimizer* (CLPSO), que permite maior troca de informações entre as partículas. No CLPSO, cada partícula aprende a partir de outras partículas, o que aumenta a diversidade do enxame e melhora a exploração. Pavan (2023) observou que o CLPSO apresenta resultados muito promissores para o despacho econômico, sendo significativamente melhor que o PSO, ao remover o *global best* da equação e permitir que as partículas aprendam umas com as outras. No entanto, nada foi encontrado sobre a prova de convergência deste método.

Neste trabalho, apresentamos uma solução híbrida que combina os pontos fortes do CLPSO e do RPSO para resolver o problema de despacho econômico. Ao combinar essas estratégias, buscamos alcançar um método que não apenas produza ótimos resultados para o despacho econômico, mas que também tenha prova de convergência.

2. MODELO DE DESPACHO ECONÔMICO

O Problema de Despacho Econômico (PDE) é amplamente investigado e consiste em determinar o despacho ótimo de potência ativa P_i de cada unidade geradora i , buscando minimizar o custo total de produção, de tal forma que as potências geradas estejam dentro dos limites mínimo e máximo operacionais de cada unidade geradora dados por P_i^{min} e P_i^{max} respectivamente, a fim de suprir uma determinada demanda D .

Na formulação clássica, o PDE é modelado como um problema de otimização com função objetivo convexa e quadrática. Assim, tem-se que o PDE clássico é descrito pelo seguinte modelo:

$$\min \sum_{i=1}^{n_G} [a_i P_i^2 + b_i P_i + c_i]$$

sujeito a:

$$\sum_{i=1}^{n_G} P_i = D, \quad P_i^{\min} \leq P_i \leq P_i^{\max}, i = 1, \dots, n_G.$$

em que n_G é a quantidade de geradores do sistema, D é a demanda a ser atendida, P_i é a potência ativa gerada pelo gerador i e a_i , b_i e c_i são os coeficientes da função de custo do gerador i , cujos limites da capacidade de geração são dados por $P_i^{\min}$ e $P_i^{\max}$, nessa ordem.

Para tornar o modelo mais realista, é necessário considerar fatores adicionais que afetam os custos de produção. Nos geradores termelétricos, por exemplo, existem válvulas de admissão de calor cuja abertura parcial provoca perdas de energia através da passagem de vapor, aumentando os custos de produção. Quanto mais estrangulado o vapor, isto é, quanto menor a abertura da válvula, menor o rendimento da mesma. Para melhorar o rendimento global, o vapor é controlado por múltiplas válvulas parciais. Portanto, no momento de abertura de cada uma dessas válvulas parciais há uma alteração na curva de rendimento global e, por conseguinte, na curva de custos. Este fenômeno é chamado de efeito de ponto de carregamento de válvula, e sua inserção no problema de PDE resulta no problema de despacho denominado PDEPV.

Na formulação do modelo de PDEPV, temos as mesmas restrições do problema de PDE clássico, porém sua função objetivo possui um termo que corresponde ao módulo de uma

senoide, que representa os pontos de carregamento de válvula. Assim, tem-se que o PDEPV é descrito pelo modelo

$$\min \sum_{i=1}^{n_G} [a_i P_i^2 + b_i P_i + c_i + |e_i \sin(f_i(P_i^{\min} - P_i))|]$$

s.a:

$$\sum_{i=1}^{n_G} P_i = D, \quad P_i^{\min} \leq P_i \leq P_i^{\max}, i = 1, \dots, n_G,$$

em que a_i , b_i , c_i , e_i e f_i são os coeficientes da curva de custo de combustível do gerador i . Para a resolução do problema neste artigo, será adotado o modelo PDEPV em vez do PDE clássico.

3. DESCRIÇÃO DOS ALGORITMOS.

3.1 Particle Swarm Optimization (PSO)

O PSO é um algoritmo de otimização bioinspirado no comportamento coletivo de organismos, como bandos de pássaros em busca de alimento. Cada pássaro voa por uma parte diferente do espaço, representando possíveis soluções para um problema de otimização. Inicialmente, os movimentos dos pássaros são aleatórios. No entanto, ao longo do tempo, cada pássaro ajusta sua trajetória baseada em seu momento (ω), sua própria experiência (o melhor local que encontrou até agora, chamado de *personal best* (pbest)) e na experiência coletiva do enxame (o melhor local encontrado por qualquer pássaro, chamado de *global best* (gbest)). Esse comportamento permite que o enxame como um todo convirja para uma solução ótima.

Desde sua introdução por Kennedy e Eberhart (1995), o PSO tem sido amplamente utilizado em diversos campos devido à sua simplicidade e eficiência. Nesse algoritmo as partículas se movem em um espaço de busca n-dimensional, ajustando suas posições com base na melhor solução pessoal e na melhor solução global encontradas:

Ajuste de velocidade:

$$v_{ij,t+1} = \omega v_{ij,t} + c_1 r_{1,j,t} (y_{ij,t} - x_{ij,t}) + c_2 r_{2,j,t} y_{j,t}^* - x_{ij,t} \quad (1)$$

Ajuste de posição:

$$x_{ij,t+1} = x_{ij,t} + v_{ij,t+1}, \quad (2)$$

para $i, j \in Z$, $1 \leq i \leq h$ e $1 \leq j \leq n$. Aqui, h é o número total de partículas, n é a dimensão do problema, c_1 e c_2 são parâmetros, $r_{1,j,t}$ e $r_{2,j,t}$ são coeficientes estocásticos sorteados da

distribuição uniforme $U(0,1)$, $x_{ij,t}$ e $v_{ij,t}$ são a posição e a velocidade da partícula i no tempo t , $y_{ij,t}$ e $y_{j,t}^*$ é a melhor solução pessoal (pbest) e a melhor posição global até o momento (gbest) e ω é o peso da inércia.

Quando adotamos peso de inércia com decrescimento linear (com valores de ω^{max} e ω^{min}) os resultados tem se mostrado melhores, sendo assim, adotamos um ω que é atualizado iterativamente, ou seja, $\omega(k) = \omega^{max} - [\frac{\omega^{max}-\omega^{min}}{k^{max}}]k$.

3.2 Randomized Particle Swarm Optimization (RPSO)

O RPSO é uma variante do PSO que introduz partículas aleatórias no enxame para evitar a convergência prematura a mínimos locais. A cada iteração, no mínimo uma partícula é reinicializada aleatoriamente, garantindo uma melhor cobertura do espaço de busca.

O número exato de partículas a ser reinicializadas vai ser sorteado a cada iteração e é controlado pelo *srand* (que é sorteado entre *posIA* (parâmetro definido pelo usuário) e o número total de partículas (*nP*). Por exemplo, se *posIA* for definido como 19 e *nP* como 475, isso significa que a cada iteração um número aleatório entre 19 e 475 será sorteado para determinar quando uma partícula será reinicializada aleatoriamente. Se o número sorteado for 30, então todas as posições múltiplas de 30 serão reinicializadas, ou seja, redefinimos a posição de uma partícula específica apenas a cada *srand* (30) posições, permitindo que a partícula explore mais a região. Durante cada iteração, pelo menos uma partícula assume uma posição aleatória no espaço de busca. Essa abordagem permite uma exploração mais ampla do espaço de busca, aumentando as chances de escapar de mínimos locais. As equações de ajuste de posição e velocidade se mantêm a mesma.

3.3 Comprehensive Learning Particle Swarm Optimization (CLPSO)

Sabe-se que a componente social apresenta um grande impacto no fenômeno de convergência prematura do algoritmo, pois as partículas do enxame tendem a convergir facilmente para a posição gbest. O CLSPO aplica a aprendizagem abrangente periodicamente

e isso faz com que as partículas ajustem sua velocidade e posição baseando-se em uma memória compartilhada (atualizada com base no sucesso de outras partículas).

Nessa estratégia, as partículas aprendem utilizando informação da $p_{i,best}$ de outras partículas, tornando-as mais aptas a escaparem de possíveis mínimos locais do problema. Na variante CLPSO, a atualização das posições das partículas é realizada utilizando a equação (2). No entanto, a equação para o cálculo da velocidade (1) é modificada para

Ajuste de velocidade:

$$v_{ij,t+1} = \omega v_{ij,t} + c_1 r_{1j,t} (my_{ij,t} - x_{ij,t}) \quad (3)$$

Comparando-se as equações de velocidade (1) e (3) dos algoritmos PSO e CLPSO, verifica-se que o gbest é excluído da equação de atualização da velocidade e modifica-se o pbest $y_{ij,t}$ através da alteração da memória individual de cada partícula para $my_{ij,t}$. Aqui, $my_{ij,t} \in \mathbb{R}^{i \times j}$ é constituída pelas memórias individuais modificadas $my_{ij,t}$ de cada partícula do enxame.

Inicialmente, verifica-se a quantas iterações a partícula está sem obter progresso na otimização de sua função objetivo (gap_i), ou seja, se a função objetivo melhorar, o gap_i é zerado; caso contrário, incrementa-se gap_i em 1. Caso o gap_i da partícula i seja maior ou igual a r^{gap} (parâmetro definido pelo usuário) a partícula é então selecionada para passar pelo processo de aprendizagem abrangente e seu gap_i é zerado.

Se for selecionada, para cada uma das dimensões que compõem a partícula, sorteia-se um número aleatório proveniente de $U(0,1)$ e verifica-se se este é inferior ou igual a uma probabilidade de troca PT (parâmetro definido pelo usuário). Em caso positivo, duas outras memórias individuais das partículas do enxame são selecionadas aleatoriamente. Em seguida, compara-se qual destas partículas sorteadas possui a melhor avaliação da função objetivo do problema. A partícula sob o processo de aprendizado irá, então, receber a informação da respectiva dimensão da memória individual da partícula com a melhor avaliação. Em caso negativo, a memória modificada da partícula receberá, nesta dimensão, o próprio valor de sua memória individual original.

Uma vez executado este processo para todas as dimensões da partícula, verifica-se se a memória modificada resultante é igual a sua própria memória original. Caso seja, sorteia-se novamente outra memória individual dentre as partículas do enxame e uma dimensão, atribuindo-se, então, o novo valor à memória modificada. O CLPSO modifica o PSO para evitar a convergência prematura substituindo a dependência da melhor posição global (gbest) por uma

aprendizagem abrangente das melhores posições individuais (pbest) de outras partículas. Esse método melhora a exploração do espaço de busca e evita que todas as partículas convirjam prematuramente para o gbest.

3.4 Randomized Particle Comprehensive Learning Swarm Optimization (RPCLPSO)

Neste trabalho, propõe-se o RPCLPSO, o qual é uma variante híbrida do PSO que combina as estratégias do RPSO e do CLPSO para melhorar a capacidade do algoritmo de evitar a convergência prematura e explorar de forma mais eficiente o espaço de busca. Essa variante híbrida adota a estratégia do RPSO, onde partículas são reinicializadas aleatoriamente a cada iteração, e incorpora a estratégia de aprendizado abrangente do CLPSO.

A reinicialização aleatória proporciona uma cobertura ampla do espaço de busca, que garante que o método converge, enquanto o aprendizado abrangente permite que as partículas adaptem suas estratégias com base nas melhores experiências do enxame, que garante ótimos resultados. Essa combinação resulta em um método híbrido que tem garantia de convergência e ótimos resultados.

4. ANÁLISE TEÓRICA DOS ALGORITMOS

Para que algoritmos de busca aleatória, como o PSO e suas variantes, garantam a convergência ao ótimo global, certas hipóteses devem ser satisfeitas. Essas condições são essenciais para assegurar que o algoritmo explore eficientemente o espaço de busca e convirja para a solução ótima. Baseado no trabalho de Solis e Wets (1981) as seguintes hipóteses, (H1) e (H2) que serão discutidas a seguir, são fundamentais para a convergência de algoritmos de busca global.

Seja o problema de minimizar uma função $f: \mathcal{R}^n \rightarrow \mathcal{R}$ em um subconjunto $S \subseteq \mathcal{R}^n$ (espaço de busca). O objetivo é encontrar um ponto $x^* \in S$ que minimize f em S . A busca é guiada por um processo aleatório simples, atualizando posições com base em pontos aleatórios ξ e uma aplicação D , que sugere novas posições que não são piores que as atuais. A função de atualização do algoritmo deve garantir que a nova posição gerada por uma partícula, $D(x, \xi)$, tenha um valor de função objetivo não pior que o da posição atual, x . Formalmente,

Hipótese 1 (H1): Condição de Melhoria

$$f(D(x, \xi)) \leq f(x) \text{ e se } \xi \in S, f(D(x, \xi)) \leq f(\xi).$$

Isso implica que cada nova posição deve proporcionar uma melhoria ou, no mínimo, não piorar a solução atual. A sequência aleatória $\{x_k\}_{k=0}^{\infty}$ é gerada pelo algoritmo, onde x_k representa a posição de uma partícula no passo k . A convergência é alcançada se a sequência $f(x_k)_{k=1}^{\infty}$ é monótona (não crescente) e converge para o ínfimo (maior das cotas inferiores) de f em S .

Outra hipótese importante é a cobertura adequada do espaço de busca, que nos diz que a probabilidade de x_k não atingir um determinado subconjunto mensurável A de S com medida de Lebesgue positiva (“podendo ser área ou volume”) repetidamente deve ser zero. Em outras palavras, sendo $\mu_k(A) := P(x_k \in A)$ temos:

Hipótese 2 (H2): Cobertura Adequada

$$\prod_{k=0}^{\infty} (1 - \mu_k(A)) = 0.$$

Isso assegura que, com o tempo, todas as regiões do espaço de busca serão exploradas, prevenindo a possibilidade de o algoritmo ignorar sistematicamente regiões potencialmente ótimas.

Se as hipóteses (H1) e (H2) são satisfeitas, podemos garantir que a sequência gerada pelo algoritmo irá, com probabilidade 1, convergir para a região de otimalidade:

$$\lim_{k \rightarrow \infty} P(x_k \in R_{\epsilon}) = 1,$$

onde $R_{\epsilon} = \{x \in S / f(x) < \psi + \epsilon\}$, com ψ sendo o ínfimo essencial de f em S , e $\epsilon > 0$.

As hipóteses apresentadas fornecem uma base teórica sólida para o desenvolvimento e a análise de algoritmos de busca aleatória, incluindo o PSO e suas variantes. Essas condições ajudam a garantir que o algoritmo explore o espaço de busca de forma eficiente e que seja capaz de escapar de mínimos locais e convergir para a solução global ou uma região aceitável.

O PSO satisfaz a condição de melhoria (H1), pois a melhor posição global é sempre mantida, garantindo melhorias progressivas. No entanto, o PSO não satisfaz a hipótese de

cobertura adequada (H2). As análises já feitas na literatura, por exemplo, Clerc and Kennedy (2002), sugerem que as partículas no PSO convergem para um ponto, que é uma combinação linear do gbest (y^*) com o pbest (y):

$$(1 - \frac{c_2}{c_1+c_2})y + y^*, \text{ onde } a \in [0,1].$$

No entanto, a otimalidade desse ponto de convergência não pode ser garantida, indicando uma limitação do algoritmo PSO original. Por exemplo, considere o problema de otimização de minimizar a função objetivo $f(x, y) = (x - 3)^2 + (y - 3)^2$ que será resolvido utilizando o Algoritmo de Otimização por Enxame de Partículas (PSO). Seja fator de inércia (ω): 0.5 e coeficientes cognitivo (c_1) e social (c_2): ambos 2. Suponha que se utilizem duas partículas com os seguintes parâmetros (velocidade inicial, posição atual e *personal best*, respectivamente) iniciais:

- Partícula 1 (gbest): $v_1 = (0, 1)$, $x_1 = (1, 2)$ e $y_1 = (1, 3)$.
- Partícula 2: $v_2 = (0, 2)$, $x_2 = (1, 4)$ e $y_2 = (1, 5)$.

Para a partícula 1, temos o seguinte cálculo da nova velocidade (seja $r_1 = r_2 = 1$ para simplificar)

$$v_1 = 0.5 \cdot (0, 1) + 2 \cdot 1 \cdot ((1, 3) - (1, 2)) + 2 \cdot 1 \cdot ((1, 3) - (1, 2)) = (0, 4.5).$$

E com isso, sua nova posição será $x_1 = (1, 2) + (0, 4.5) = (1, 6.5)$.

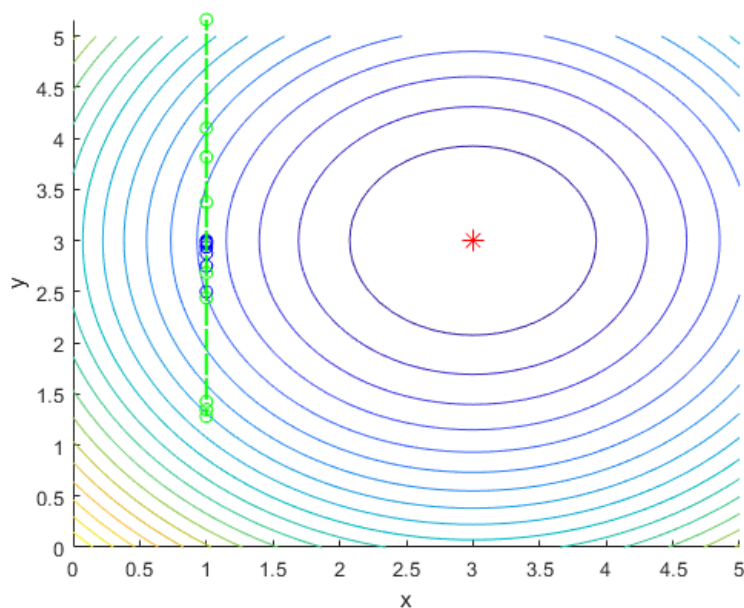
Para a partícula 2, temos o seguinte cálculo da nova velocidade:

$$v_2 = 0.5 \cdot (0, 2) + 2 \cdot 1 \cdot ((1, 5) - (1, 4)) + 2 \cdot 1 \cdot ((1, 3) - (1, 4)) = (0, 1).$$

E com isso, sua nova posição será $x_2 = (1, 4) + (0, 1) = (1, 5)$.

Nessa iteração, observa-se que as novas posições estão no eixo $x=1$. Isso é válido para qualquer iteração, ou seja, as partículas permanecem confinadas ao longo do eixo $x=1$, movendo-se apenas verticalmente e nunca saindo dessa trajetória. Sendo assim, elas nunca poderão encontrar o valor ótimo desse problema, que é o ponto (3, 3).

FIGURA 1 - Ilustração do exemplo de estagnação.



Fonte: Autoria própria.

Este exemplo ilustra um caso de estagnação, onde as partículas não conseguem encontrar o ótimo global da função objetivo. As partículas se movem apenas na direção de seus melhores valores locais e globais conhecidos, mas, devido à configuração dos parâmetros e das posições iniciais, podem não explorar suficientemente o espaço de busca, ou seja, para o PSO original, existe uma probabilidade positiva de não encontrar o ótimo do problema. Este exemplo destaca a importância de ajustar corretamente os parâmetros do PSO e considerar estratégias adicionais para evitar a estagnação.

O defeito do algoritmo PSO básico é que existem estados iniciais nos quais o algoritmo PSO original pode começar e que levarão a um estado estagnado em um número finito de passos. Para que o PSO satisfaça as condições necessárias para a convergência, o algoritmo deve gerar uma sequência de soluções que devem cobrir adequadamente o espaço de busca (Hipótese (H2)).

Para construir um algoritmo de busca global baseado em PSO, satisfazendo a condição (H2), é suficiente adicionar partículas aleatórias ao enxame. No método RPSO, isso é alcançado reiniciando a posição de pelo menos uma partícula, fazendo com que ela se torne uma partícula aleatória ao mover sua posição para um local aleatório no espaço de busca. A melhor posição global continua sendo mantida, portanto, o RPSO claramente satisfaz (H1). Durante cada iteração, pelo menos uma partícula assume uma posição aleatória no espaço de busca,

garantindo uma cobertura adequada do espaço de busca e satisfazendo (H2), caracterizando-o como um algoritmo de busca global.

O CLPSO mostra resultados muito bons na prática, ajustando as posições das partículas com base nas melhores posições individuais de outras partículas. Seguindo a lógica de testar as hipóteses, tudo indica que o CLPSO não satisfaz (H2), o que sugere que ele pode não convergir ao ótimo global de maneira garantida. Diante disso, o RPCLPSO foi criado para combinar as vantagens do RPSO e do CLPSO. Ele utiliza a estratégia de reinicialização aleatória do RPSO para satisfazer (H2), garantindo a convergência global, e incorpora o aprendizado abrangente do CLPSO, que tem mostrado bons resultados na prática. Embora o RPCLPSO garanta a convergência ao ótimo global, o problema é que a taxa de convergência não é definida. Não há uma medida clara de quanto tempo levará para alcançar a convergência, o que é um aspecto crítico a ser investigado em estudos futuros.

A combinação das hipóteses (H1) e (H2) é essencial para garantir que um algoritmo de busca aleatória convirja para a solução ótima global. Enquanto o RPSO satisfaz ambas as hipóteses e oferece garantias teóricas de convergência, o CLPSO, embora eficaz na prática, não possui uma prova formal de convergência conhecida. A criação do RPCLPSO representa um esforço para unir a eficácia prática do CLPSO com as garantias teóricas do RPSO, proporcionando um algoritmo híbrido com potencial para melhor desempenho. No entanto, a taxa de convergência do RPCLPSO permanece uma área de pesquisa em aberto.

5. RESULTADOS

Os experimentos foram realizados em um conjunto de problemas de despacho econômico PDEPV, abrangendo diferentes cenários com 3, 13, 19 e 40 geradores, com demanda 850 MW, 2520 MW, 2908 MW e 10500 MW, respectivamente. Para cada método e cada número de geradores, ajustamos os parâmetros de acordo com as características do problema. Para encontrar os melhores valores para esses parâmetros, utilizamos o pacote Optuna em Python, desenvolvido pela equipe *Preferred Networks, Inc.* Esse pacote permite uma busca eficiente no espaço de hiper parâmetros, otimizando a função objetivo ao longo do tempo.

Os parâmetros otimizados de modo geral foram: nP (número de partículas), alpha (utilizado no cálculo da velocidade das partículas para o primeiro passo do algoritmo), ω^{max} , ω^{min} , c_1 , c_2 e NITER (número de iterações). Para o RPSO especificamente temos: PT, (probabilidade de

troca) e o r^{gap} . Para o CLPSO, temos o posIA. Após a busca, obtivemos os parâmetros otimizados para diferentes números de geradores, que são resumidos na tabela a seguir:

TABELA 1 - Parâmetros otimizados dos métodos para diferentes números de geradores.

Geradores	Método	Parâmetros									NITER
		nP	Alpha	ω^{max}	ω^{min}	c_1	c_2	PosIA	PT	r^{gap}	
3	PSO	480	0.1	0.65	0.4	1.4	1.05	-	-	-	2680
	RPSO	475	0.09	0.7	0.25	1.4	1.95	18.6	-	-	2700
	CLPSO	490	0.09	0.99	0.25	1.8	-	-	0.5	4	1250
	RPCLPSO	100	0.08	0.85	0.15	1.1	-	-	0.8	5	3000
13	PSO	205	0.02	0.8	0.4	1.5	1.95	-	-	-	2795
	RPSO	285	0.02	0.6	0.1	1.25	1.7	54.9	-	-	2320
	CLPSO	400	0.1	0.9	0.4	1.5	-	-	0.9	2	1500
	RPCLPSO	400	0.1	0.8	0.4	1	-	-	0.9	2	2000
19	PSO	310	0.07	0.97	0.35	1.9	1.12	-	-	-	2280
	RPSO	420	0.08	0.65	0.25	1.9	1.35	32.4	-	-	2380
	CLPSO	480	0.06	0.9	0.35	1	-	-	0.55	5	2140
	RPCLPSO	140	0.07	0.65	0.2	1	-	-	0.9	4	2150
40	PSO	400	0.1	0.9	0.4	1.5	1.5	-	-	-	1500
	RPSO	400	0.1	0.9	0.4	1.5	1.5	2	-	-	1500
	CLPSO	400	0.1	0.9	0.4	1.5	-	-	0.9	2	1500
	RPCLPSO	320	0.08	0.7	0.15	1.5	-	-	0.3	1	1795

Fonte: Autoria própria.

Os melhores resultados observados nas dissertações de Pavan (2023) e Silva (2014) e os resultados obtidos neste experimento (o número de execuções foi 100 vezes cada estudo) estão resumidos nas tabelas a seguir, onde são apresentados o menor valor obtido, a média e o desvio padrão dos resultados para cada algoritmo e número de geradores:

TABELA 2 - Melhores resultados, segundo Pavan (2023) e Silva (2014).

Geradores	Algoritmo	Menor Valor	Média	Desvio Padrão
3	QPSO	8234.07	-	-
13	CLPSO-SQP	24169.9177	24228.3647	69.1505
19	CLPSO-SQP	16945.6023	16952.9393	10.9538
40	CLPSO-SQP	121476.6736	121496.0788	35.3006

Fonte: Autoria própria.

TABELA 3 - Comparação dos resultados para diferentes algoritmos e número de geradores.

Geradores	Algoritmo	Menor Valor	Média	Desvio Padrão
3	PSO	8233.889	8237.105	4.920
	CLPSO	8233.889	8233.901	0.025
	RPSO	8233.889	8233.889	0.000
	RPCLPSO	8233.889	8233.932	0.063
13	PSO	24216.754	24460.105	143.160
	CLPSO	24169.956	24260.158	33.314
	RPSO	24218.111	24342.355	97.745
	RPCLPSO	24170.504	24272.084	11.266
19	PSO	16945.524	16968.846	17.718
	CLPSO	16947.103	16960.564	2.093
	RPSO	16948.751	16984.876	35.456
	RPCLPSO	16950.670	16994.956	23.162
40	PSO	122366.693	123743.719	778.176
	CLPSO	121508.784	121569.145	63.001
	RPSO	122758.343	124609.353	748.452
	RPCLPSO	121468.617	121529.425	31.764

Fonte: Autoria própria.

6. CONSIDERAÇÕES FINAIS

Em suma, nossa análise comparativa dos métodos de otimização PSO, RPSO, CLPSO e RPCLPSO revelou uma série de discussões valiosas sobre o desempenho dessas técnicas no Problema de Despacho Econômico com ponto de carregamento de válvula (PDEPV). O PSO original, embora amplamente utilizado e capaz de fornecer soluções próximas ao ótimo, demonstrou limitações significativas ao enfrentar mínimos locais, destacando a necessidade premente de métodos mais robustos, derivados do PSO.

Introduzindo reinicializações aleatórias para evitar mínimos locais, o RPSO, infelizmente, não alcançou melhorias significativas em relação ao PSO em nossas análises. Por outro lado, o CLPSO consistentemente superou o PSO e o RPSO, especialmente em cenários com um maior número de geradores. Sua capacidade de promover maior diversidade nas soluções mostrou-se altamente vantajosa em problemas com múltiplos mínimos locais, como os encontrados no PDEPV.

No entanto, o destaque de nossa investigação é o RPCLPSO, que não só conseguiu evitar mínimos locais de forma eficaz, mas também alcançou resultados comparáveis aos melhores valores reportados na literatura. Por exemplo, para o caso de 40 geradores, a resolução do RPCLPSO foi ligeiramente melhor que a melhor encontrada na literatura (segundo Pavan

(2023)). Esta solução híbrida promissora integra o melhor dos métodos PSO e CLPSO, demonstrando sua eficácia em otimizar o Problema do Despacho.

É importante ressaltar que, embora métodos como o RPSO possam ter uma prova de convergência, isso não necessariamente garante uma convergência rápida ou eficiente na prática. A taxa de convergência é um fator crucial a ser considerado ao avaliar o desempenho de algoritmos de otimização, e nossos resultados destacam a importância de se escolher abordagens que ofereçam um equilíbrio entre eficácia e eficiência computacional. Ao levar em conta essas discussões, esperamos que este estudo contribua para o avanço contínuo no estudo teórico de métodos heurísticos.

Além disso, um próximo passo promissor em nossa pesquisa seria uma investigação sobre a taxa de convergência, em termos matemáticos. Compreender quanto tempo leva para cada algoritmo convergir permitirá avaliar sua eficácia em diferentes cenários. Essa análise mais detalhada poderia ser fundamental para o desenvolvimento de estratégias de otimização mais eficientes e adaptáveis em futuras aplicações.

REFERÊNCIAS

EBERHART, R.; KENNEDY, J. Particle swarm optimization. In: **Proceedings of the IEEE international conference on neural networks**. v. 4, p. 1942-1948, 1995.

SILVA, D.N. **Método primal-dual previsor-corretor de pontos interiores e exteriores com estratégias de correção de inércia e suavização hiperbólica aplicado ao problema de despacho econômico com ponto de carregamento de válvula e representação da transmissão**. Dissertação de mestrado, UNESP, Bauru, 2014.

PAVAN, R. **Método híbrido de enxame de partículas de aprendizagem abrangente com branch-and-bound e programação quadrática sequencial para resolução de problemas do fluxo de potência ótimo com variáveis discretas**. Dissertação de mestrado, UNESP, Bauru, 2023.

SOLIS, F.J.; WETS, R.J.-B. Minimization by random search techniques. **Mathematics of Operations Research**, v. 6, n. 1, p. 19-30, 1981.

VAN DEN BERGH, F.; ENGELBRECHT, A.P. A convergence proof for the particle swarm optimiser. **Fundamenta Informaticae**, v. 105, n. 4, p. 341-374, 2010.