

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO  
INSTITUTO DE COMPUTAÇÃO  
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

JOÃO VIRGILIO DE CASTRO AVANCINI

CRIAÇÃO DE UM FRAMEWORK PARA AJUSTE FINO DE PARÂMETROS USANDO  
TESTES ESTATÍSTICOS

RIO DE JANEIRO

2022

JOÃO VIRGILIO DE CASTRO AVANCINI

CRIAÇÃO DE UM FRAMEWORK PARA AJUSTE FINO DE PARÂMETROS USANDO  
TESTES ESTATÍSTICOS

Trabalho de conclusão de curso de graduação apresentado ao Instituto de Computação da Universidade Federal do Rio de Janeiro como parte dos requisitos para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Profa. Carla Amor Divino  
Moreira Delgado

Co-orientador: Profa. Carolina Gil  
Marcelino

RIO DE JANEIRO

2022

## CIP - Catalogação na Publicação

A946c Avancini, João Virgílio de Castro  
Criação de um framework para ajuste fino de parâmetros usando testes estatísticos / João Virgílio de Castro Avancini. -- Rio de Janeiro, 2022.  
42 f.

Orientadora: Carla Amor Divino Moreira Delgado.  
Coorientadora: Carolina Gil Marcelino.  
Trabalho de conclusão de curso (graduação) -  
Universidade Federal do Rio de Janeiro, Instituto de Matemática, Bacharel em Ciência da Computação, 2022.

1. Otimização. 2. Ajuste fino de parâmetros. 3. Eficiência energética. 4. Energia limpa. 5. Energia eólica. I. Delgado, Carla Amor Divino Moreira, orient. II. Marcelino, Carolina Gil, coorient. III. Título.

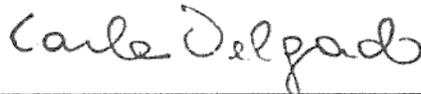
JOÃO VIRGILIO DE CASTRO AVANCINI

CRIAÇÃO DE UM FRAMEWORK PARA AJUSTE FINO DE PARÂMETROS USANDO  
TESTES ESTATÍSTICOS

Trabalho de conclusão de curso de graduação  
apresentado ao Instituto de Computação da  
Universidade Federal do Rio de Janeiro como  
parte dos requisitos para obtenção do grau de  
Bacharel em Ciência da Computação.

Aprovado em 11 de janeiro de 2022.

BANCA EXAMINADORA:



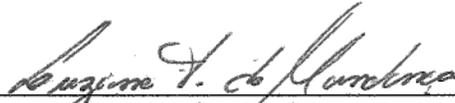
---

Carla Amor Divino Moreira Delgado, D.sC.  
Instituto de Computação - UFRJ



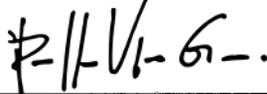
---

Carolina Gil Marcelino, D.sC.  
Instituto de Computação - UFRJ



---

Luziane Ferreira de Mendonça, D.sC.  
Instituto de Computação - UFRJ



---

Phillip Vilaça Gomes, PhD.  
Universidad Rey Juan Carlos

## **AGRADECIMENTOS**

Gostaria de agradecer meus pais, João Virgílio Avancini e Elizabeth Cristina Pereira de Castro Avancini, por sempre me apoiarem. Agradeço também a minha namorada, Camila Carvalho Ferreira, por estar ao meu lado durante toda essa caminhada, e aos amigos que fiz nesse caminho, dentro e fora da faculdade, e que sempre me ajudaram.

Gostaria de agradecer aos professores da UFRJ pelas inúmeras oportunidades de aprendizado, sobretudo a minha orientadora de iniciação científica, Carolina Gil Marcelino, minha orientadora do TCC, Carla Amor Divino Moreira Delgado, e a minha orientadora acadêmica Silvana Rossetto.

Um agradecimento ao CNPQ, por financiar esse projeto de pesquisa através do Programa Institucional de Bolsas de Iniciação Científica (PIBIC) e do Programa Institucional de Bolsas de Iniciação em Desenvolvimento Tecnológico e Inovação (PIBITI).

## RESUMO

Neste trabalho, foi criado um framework para realização do ajuste fino de parâmetros em uma meta-heurística evolutiva para problemas de eficiência energética. Foi realizado o ajuste fino no Canonical Differential Evolutionary Particle Swarm Optimization (C-DEEPSO), um algoritmo de otimização criado a partir da fusão de duas outras meta-heurísticas, o Differential Evolution (DE) e o Evolutionary Particle Swarm Optimization (EPSO). Esse ajuste fino foi efetuado pelo método de corrida iterativa com ajuda de um pacote chamado *iRace*. Neste método, são realizadas várias simulações do algoritmo com diferentes parâmetros e são identificadas as melhores configurações possíveis com a aplicação de testes de hipóteses. Em um primeiro momento, o framework foi testado em duas funções de benchmark, as funções de Rastrigin e Rosenbrock para 30, 50 e 100 dimensões. Uma vez validada a eficácia do framework, o mesmo foi aplicado a um problema de eficiência energética de uma usina eólica offshore, com foco na redução da perda de energia. Este problema de eficiência energética possui natureza dinâmica, sendo composto por 96 instâncias, que caracterizam a discretização de 24 horas divididas entre intervalos de 15 minutos. A partir dos resultados obtidos via framework, uma comparação com o método estado da arte, o Mean-Variance Mapping Optimization (MVMO) foi realizada com uso da ferramenta DSCTool. Os resultados mostraram que o algoritmo C-DEEPSO com seus principais parâmetros ajustados obtém em média resultados otimizados mais robustos que os apresentados pelo MVMO. É evidenciado que a aplicação do C-DEEPSO com parâmetros tunados no lugar do MVMO resultaria em uma redução de perdas de 70.02MWh, energia suficiente para abastecimento de cerca de 450 residências.

**Palavras-chave:** otimização; ajuste fino de parâmetros; eficiência energética; energia limpa; energia eólica.

## ABSTRACT

In this work, a framework was created that performs the fine tuning of parameters in an evolutionary metaheuristic at energy efficiency problems. This framework was used to perform the fine tuning at the Canonical Differential Evolutionary Particle Swarm Optimization (C-DEEPSO) algorithm, an optimization algorithm that was created by merging the Differential Evolution (DE) and the Evolutionary Particle Swarm Optimization (EPSO). This fine tuning was performed by the iterative racing method, by an R package called iRace. In this method, several simulations of the algorithm were performed with different parameters values and the best ones were identified by statistical hypothesis tests. This framework was tested in the Rastrigin and Rosenbrock functions, two benchmark functions, with 30, 50 and 100 dimensions. Once it was proven the efficiency of this framework, it was applied in a dynamic energy efficiency problem that simulates an offshore wind power plant (WPPs). This problem is composed of 96 instances, that represents 24 hours divided in 15 minutes intervals. The results obtained with our framework were compared with the state of art algorithm, the Mean-Variance Mapping Optimization (MVMO), using DSCTool, a web-service-based framework for statistical comparison. The results show that the C-DEEPSO algorithm with tuned parameters usually presents more robust results than the MVMO. It could decrease the energy loss of this system by 70.02MWh, enough energy to power 450 houses.

**Keywords:** optimization; fine tuning; energy efficiency; clean energies; wind power.

## LISTA DE ILUSTRAÇÕES

<b>Figura 1: Função de Rosenbrock</b>	<b>14</b>
<b>Figura 2: Função de Rastrigin</b>	<b>15</b>
<b>Figura 3: Processo básico de recombinação</b>	<b>17</b>
<b>Figura 4: Esquema do framework em função de benchmark</b>	<b>22</b>
<b>Figura 5: Esquema genérico de usina eólica <i>offshore</i></b>	<b>24</b>
<b>Figura 6: Esquema do controlador de usina eólica</b>	<b>24</b>
<b>Figura 7: Esquema do framework para o problema IEEE 41</b>	<b>27</b>
<b>Figura 8: Comparação do C-DEEPSO tunado com DEEPSO e MVMO</b>	<b>37</b>

## LISTA DE TABELAS

<b>Tabela 1: Métodos utilizados para resolução de problemas OPF</b>	<b>16</b>
<b>Tabela 2: Características do problema IEEE 41-BUS</b>	<b>25</b>
<b>Tabela 3: Resultado do ajuste fino na função de Rosenbrock</b>	<b>28</b>
<b>Tabela 4: Comparação do C-DEEPSO tunado e empírico em Rosenbrock</b>	<b>29</b>
<b>Tabela 5: Comparação do C-DEEPSO tunado e meta-heurísticas em Rosenbrock</b>	<b>29</b>
<b>Tabela 6: Resultado do ajuste fino na função de Rastrigin</b>	<b>31</b>
<b>Tabela 7: Comparação do C-DEEPSO tunado e empírico em Rastrigin</b>	<b>31</b>
<b>Tabela 8: Comparação do C-DEEPSO tunado e meta-heurísticas em Rastrigin</b>	<b>32</b>
<b>Tabela 9: Comparação do C-DEEPSO tunado e meta-heurísticas em problema OPF</b>	<b>33</b>

## LISTA DE SIGLAS

C-DEEPSO – *Canonical Differential Evolutionary Particle Swarm Optimization*

DE – Evolução Diferencial

EPSO – Otimização Evolutiva por Enxame de Partículas

OPF – Fluxo de Potência Ótima

WPPs – Usina de energia eólica

Mut – Fator de mutação

Com – Probabilidade de comunicação

DP – Desvio Padrão

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>10</b>
1.1	MOTIVAÇÃO E CONTEXTO	10
1.2	APRESENTAÇÃO DO TRABALHO	11
<b>2</b>	<b>FUNDAMENTAÇÃO</b>	<b>13</b>
2.1	OTIMIZAÇÃO MONO-OBJETIVO	13
2.2	META-HEURÍSTICAS EVOLUTIVAS E O C-DEEPSO	15
2.3	AJUSTE FINO DE PARÂMETROS	18
<b>3</b>	<b>TRABALHOS RELACIONADOS</b>	<b>20</b>
<b>4</b>	<b>AJUSTE FINO DO C-DEEPSO EM FUNÇÕES DE BENCHMARK E EM UM PROBLEMA OPF</b>	<b>22</b>
4.1	FUNÇÕES DE BENCHMARK	22
4.2	PROBLEMA DE FLUXO ÓTIMO DE POTÊNCIA: PRODUÇÃO EM UMA USINA EÓLICA	23
<b>5</b>	<b>EXPERIMENTOS E RESULTADOS</b>	<b>28</b>
5.1	FUNÇÃO DE ROSENBROCK	28
5.2	FUNÇÃO DE RASTRIGIN	30
5.3	PROBLEMA DE FLUXO ÓTIMO DE POTÊNCIA - IEEE41 USINA EÓLICA	33
<b>6</b>	<b>CONCLUSÃO</b>	<b>39</b>
	<b>REFERÊNCIAS</b>	<b>40</b>

## 1 INTRODUÇÃO

Este trabalho aborda o conceito de ajuste fino de parâmetros aplicados a uma meta-heurística evolutiva, o algoritmo *Canonical Differential Evolutionary Particle Swarm Optimization* (C-DEEPSO) (Marcelino e outros, 2016). Nele, foi desenvolvido um *framework* em que é realizada a análise do algoritmo C-DEEPSO com o foco em tornar este método o mais ajustado possível para a solução de determinado problema.

O *framework* proposto foi elaborado com o uso de inferência estatística a partir de testes de hipóteses para comparação dos resultados otimizados via C-DEEPSO. Esses testes compararam os resultados do algoritmo com diferentes combinações de parâmetros e informam o melhor conjunto de parâmetros para solução de determinado problema..

Para verificar a performance do *framework* funções objetivo a que se conhecem o ótimo foram utilizadas. Uma vez a etapa de validação da *framework* ter sido satisfatória, o *framework* foi então aplicado para solução de um problema real, que visa minimizar perdas de geração elétrica na produção renovável de energia. Assim, o estudo realizado neste trabalho auxilia na construção de um novo controlador de despacho elétrico otimizado para obter uma geração elétrica limpa e sustentável.

### 1.1 MOTIVAÇÃO E CONTEXTO

Após a revolução industrial, a emissão de gases de efeito estufa, como o dióxido de carbono (CO<sub>2</sub>), metano (CH<sub>4</sub>) e ozônio nitroso (N<sub>2</sub>O) se intensificou agudamente, provocando um cenário de mudança na temperatura da Terra. Ao mesmo tempo, a população mundial se multiplicou acentuadamente e a necessidade de produção de energia elétrica está cada vez maior. Esta energia elétrica é gerada principalmente por fontes não renováveis e poluentes, como a queima de carvão e gás natural, o que emite grande quantidade de dióxido de carbono na atmosfera, agravando o efeito estufa (Houghton e outros, 1990).

Com a assinatura do Acordo de Paris, acordos foram firmados entre países ao redor do mundo para redução dos gases de efeito estufa. A União Européia chegou a um acordo para diminuir em 55% a emissão de gases de efeito estufa até 2030, e chegar a neutralidade de emissões até 2050 (Nações Unidas, 2021).

Com isso, se torna cada vez mais necessário o uso de fontes de energia limpas e renováveis, como a energia eólica, solar e geotérmica. Essas fontes são alternativas que não emitem gases estufa e, portanto, seriam menos prejudiciais ao meio ambiente.

Estudos mostram que algoritmos evolutivos são bastante eficientes em problemas de otimização do despacho energético (Sakipour e Abdi, 2020). Através da melhora desses algoritmos, é possível obter melhores resultados nesses problemas, tendo portanto uma maior eficiência na geração de energia elétrica limpa e renovável.

Para melhora desses algoritmos, neste trabalho é proposto um *framework* para realização do controle ótimo de despacho elétrico de um sistema eólico. Este *framework* é criado através do acoplamento do *iRace*, uma ferramenta para realização do ajuste fino de parâmetros, com o *Canonical Differential Evolutionary Particle Swarm Optimization* (C-DEEPSO), um algoritmo evolutivo já utilizado anteriormente em problemas de despacho elétrico. Essa integração otimizaria a performance do algoritmo através do ajuste fino de seus parâmetros.

## 1.2 APRESENTAÇÃO DO TRABALHO

Neste trabalho foi proposto um *framework* para otimização do despacho elétrico em uma usina de energia eólica. A ideia do *framework* é realizar o ajuste fino dos parâmetros de um algoritmo para um problema de otimização. O ajuste fino é um método em que, por meio de diversas simulações, são identificados os melhores parâmetros de um algoritmo para um problema, fazendo com que este tenha os melhores resultados possíveis.

O objetivo deste trabalho é mostrar que, por meio do uso deste *framework*, é possível encontrar um conjunto de parâmetros que maximizam a performance de um algoritmo de otimização, de forma a torná-lo mais robusto.

Este *framework* foi criado com uma integração do algoritmo C-DEEPSO (Marcelino e outros, 2016) com uma ferramenta chamada *iRace* (López-Ibáñez e outros, 2016), um pacote em R criado para realização do ajuste fino de parâmetros de algoritmos (Gunawan e outros, 2008). A priori, para validação do *framework* construído, testes de performance foram realizados na solução de problemas os quais se conhece o valor ótimo. Assim, dada a relevância dos resultados obtidos, o *framework* foi utilizado em um problema de natureza

dinâmica. Este problema é conhecido como despacho elétrico e foi aplicado na geração de energia eólica. O despacho diário da energia foi discretizado em 96 instâncias de testes, cada uma correspondendo a um momento de 15 minutos, totalizando 24 horas.

Este trabalho está organizado da seguinte maneira: no Capítulo 2 é introduzida a fundamentação teórica do trabalho, explicando os conceitos-chaves necessários para o entendimento do desenho experimental. No Capítulo 3 são discutidos os trabalhos relacionados, com as semelhanças e diferenças de cada um. No Capítulo 4 é apresentada a construção do *framework* proposto. No Capítulo 5 é exibido o resultado do *framework* proposto e sua comparação com outros métodos. No Capítulo 6 são tiradas as conclusões e as propostas para trabalhos futuros desta pesquisa.

## 2 FUNDAMENTAÇÃO

Este capítulo aborda a fundamentação teórica do trabalho. O conceito de otimização mono-objetivo é abordado bem como o algoritmo de otimização C-DEESPO. Ao fim são discutidas as premissas para realização de ajuste fino de parâmetros e sua aplicação.

### 2.1 OTIMIZAÇÃO MONO-OBJETIVO

Problemas de otimização são problemas em que o objetivo é encontrar a melhor solução, seja ela a mínima ou a máxima, dentre todas as soluções possíveis. Eles podem ser contínuos ou discretos, de acordo com suas variáveis. Também podem possuir restrições, limitando seu domínio (Gaspar-Cunha e outros, 2012). Em geral possuem a estrutura:

$x^* = \min f(x)$  sujeito a:

$$g_1(x) \leq 0$$

$$g_2(x) \leq 0$$

...

$$g_m(x) \leq 0$$

$$j_1(x) = 0$$

$$j_2(x) = 0$$

...

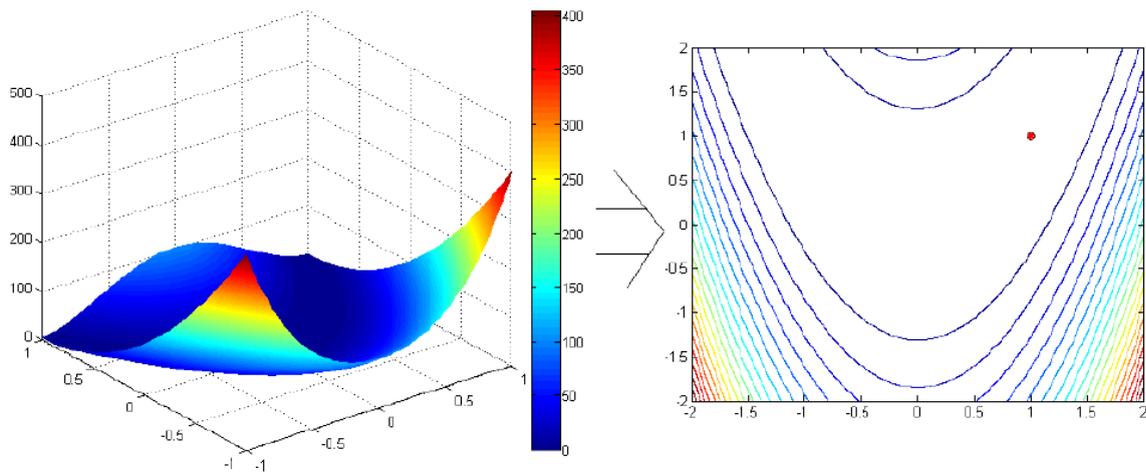
$$j_m(x) = 0$$

Na qual  $f(x)$  é a função objetivo cujo desejo é minimizar, sujeito às restrições de desigualdade  $g(x)$  e igualdade  $j(x)$ . Neste trabalho, foram utilizados dois problemas de otimização para testar o *framework* proposto, a função de Rosenbrock e a função de Rastrigin. Estas duas funções são funções de *benchmark* (Li e outros, 2013). Funções de *benchmark* são funções desenvolvidas onde já se tem um mínimo conhecido e são usadas para teste de algoritmos de otimização.

A função de Rosenbrock é uma função multimodal a partir de 4 dimensões, ou seja, possui mais de um mínimo. A Rosenbrock de  $n$  dimensões pode ser descrita pela Equação 1:

$$f(X) = \sum_{i=1}^{n-1} [100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2] \quad (1)$$

Com mínimo em  $f(X) = 0$  para  $x_i = 1, \forall i \in [1, n]$ . Em  $R^2$  a função de Rosenbrock possui um vale, conhecido como área da banana. Este vale é relativamente fácil de se encontrar, porém achar o mínimo dentro do vale se mostra uma tarefa bastante complicada. Este vale está representado em duas dimensões na Figura 1.

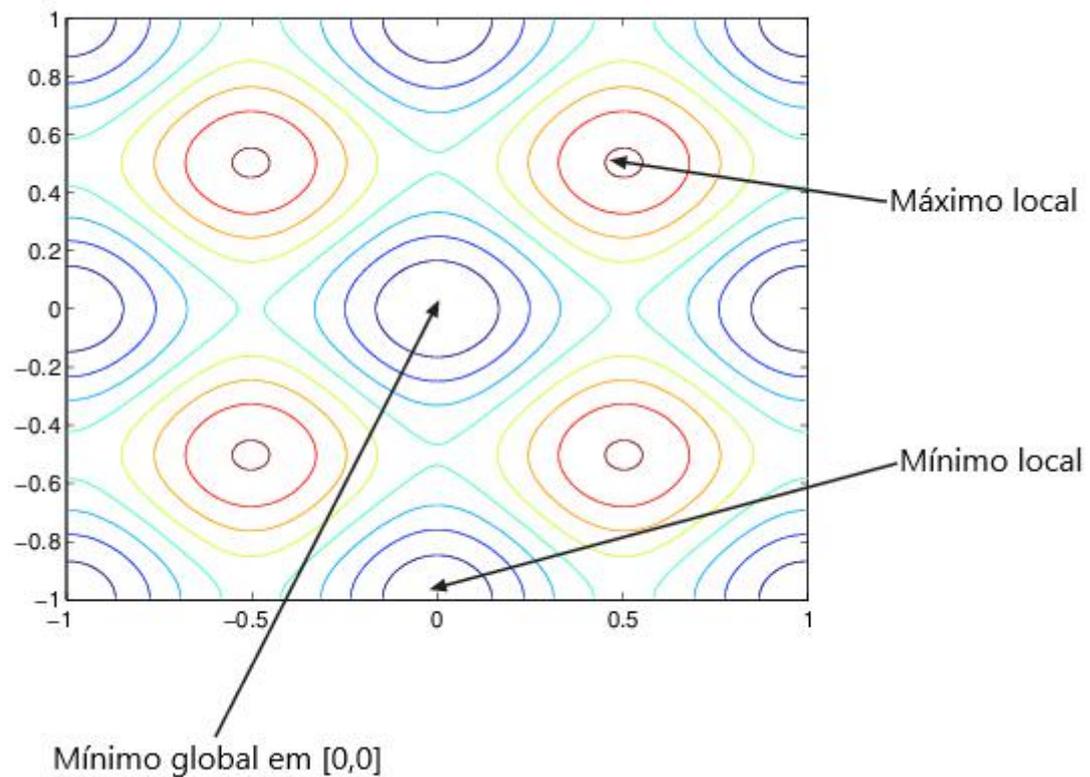


**Figura 1:** Função de Rosenbrock e vale da banana, retirada de Marcelino (2017)

A função de Rastrigin é uma função altamente multimodal. Ela possui vários mínimos regularmente distribuídos. Para  $n$  dimensões, ela possui seu ponto ótimo em  $f(X) = 0$  para  $x_i = 0, \forall i \in [1, n]$ . Sua generalização para  $n$  dimensões segue a fórmula da Equação 2:

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

Esta função é uma função com diversos vales, possuindo um grande número de mínimos e máximos locais, porém com apenas um mínimo global. Seus mínimos e máximos estão representados na Figura 2.



**Figura 2:** Função de Rastrigin em  $\mathbb{R}^2$  com seus diversos máximos e mínimos locais, retirada de *mathworks*

Para validar a proposta as funções de Rosenbrock e Rastrigin foram escolhidas devido sua facilidade de implementação e análise. Além disso, estas funções apresentam resultados vindos da literatura o que tornou a comparação de resultados com demais algoritmos possível. Além das funções de *benchmark*, este trabalho fez uso de um problema de fluxo ótimo de potência (OPF), que pode ser caracterizado como um mecanismo capaz de realizar os ajustes das variáveis de controle ótimas para operação de determinado sistema elétrico (Frank e outros, 2012). Em geral, problemas OPF visam otimizar fatores como custo ou perdas de energia via controle do fluxo energético em um sistema elétrico, sem violar os limites de energia ou limites dos equipamentos. A ideia é determinar os melhores valores para tensão, corrente e energia injetada em um sistema de energia elétrica. Esses valores podem resultar em diferentes soluções ótimas para o sistema, dependendo do requisitado. A modelagem matemática do problema OPF é apresentada no Capítulo 4 desta monografia.

## 2.2 META-HEURÍSTICAS EVOLUTIVAS E O C-DEEPSO

Ao longo do tempo vários algoritmos foram desenvolvidos para soluções de problemas de otimização. Especificamente a área de produção energética vem utilizando tanto soluções determinísticas quanto estocásticas para solução de OPF. A Tabela 1 sumariza as vantagens e desvantagens das técnicas.

**Tabela 1:** Métodos utilizados para resolução de problemas OPF, adaptada de Marcelino e outros (2016)

Método	Descrição	Vantagens	Desvantagens
Programação Linear	Uso de funções de custo lineares ou parcialmente lineares	Convergência rápida e garantida Menos esforço computacional Fácil tratamento de restrições de desigualdade	Baixa acurácia devido a linearização de funções e das restrições
Programação Quadrática	Uso de uma função objetivo quadrática Restrições lineares	Não precisa linearizar a função	Custo computacional pode ser alto
Meta-heurísticas	Algoritmo Genético, Algoritmo de Enxame de Partículas (PSO), Evolução Diferencial (DE) e Algoritmo de Colônia de Formigas (ACO)	Nenhuma linearização é requerida Pode ser usada em funções não-contínuas, não-convexas e não-diferenciáveis Fácil implementação	A solução obtida não tem garantia de ser a solução ótima

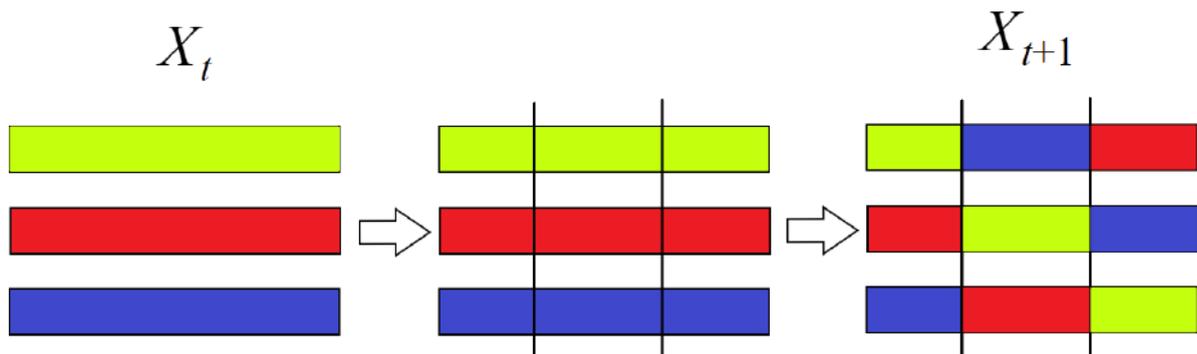
Em geral nem sempre os algoritmos de otimização determinísticos encontram a solução ótima em tempo viável. Neste caso, se torna usual recorrer ao emprego de meta-heurísticas para resolver problemas com alto número de variáveis, como o OPF. Meta-heurísticas são métodos capazes de encontrar soluções otimizadas, que atendem as premissas do problema em um tempo computacional adequado.

Meta-heurísticas evolutivas são um campo da computação evolutiva em que os algoritmos possuem uma inspiração biológica por trás, baseados em mecanismos de adaptação dos seres vivos. Estes algoritmos utilizam princípios originários da natureza, como recombinação, mutação e seleção, e através desses processos, buscam chegar a uma solução otimizada para um problema de otimização. Neste trabalho foram estudadas duas meta-heurísticas evolutivas, a Evolução Diferencial (DE) (Qin et al, 2009) e a Otimização Evolutiva por Enxame de Partículas (EPSO) (Miranda et al, 2002). Estas duas

meta-heurísticas foram combinadas e originaram o *Canonical Differential Evolutionary Particle Swarm Optimization* (C-DEEPSO), o algoritmo utilizado na criação do *framework*.

O C-DEEPSO é uma meta-heurística populacional. Em meta-heurísticas baseadas em populações, ao invés de existir apenas uma solução que vai sendo aprimorada com a execução do algoritmo, temos um conjunto de soluções, que chamaremos de  $X_t$ , e elas são modificadas e melhoradas a cada geração  $t$ . O C-DEEPSO combina a parte de recombinação do EPSO com a parte de seleção e adaptação do DE.

A recombinação, ou *crossover*, é um operador genético dos algoritmos. Ele realiza a combinação entre um par ou mais de indivíduos da população  $X_t$  e assim gera um novo indivíduo de  $X_{t+1}$ , a próxima geração dessa população. Em geral, os algoritmos genéticos possuem um processo como ilustrado na Figura 3. Nela, há uma população de 3 indivíduos que sofrem o *crossover* e geram outros 3 indivíduos diferentes, de uma nova população.



**Figura 3:** Processo básico de recombinação

Porém, no C-DEEPSO a recombinação é dada pela *Regra do Movimento*, vista nas Equações 3 e 4.

$$V_t = w_I^* \times V_{t-1} + w_A^* \times (X_{best} + F \times (X_r - X_{t-1})) + w_C^* \times C \times (X_{gb}^* - X_{t-1}) \quad (3)$$

$$X_t = X_{t-1} + V_t \quad (4)$$

Nas Equações 3 e 4,  $t$  denota a geração atual,  $V_t$  a velocidade do indivíduo,  $X_{t-1}$  é o indivíduo atual em que é realizado o *crossover*,  $X_{best}$  a melhor solução encontrada pelo indivíduo.  $w_I^*$ ,  $w_A^*$  e  $w_C^*$  são os pesos, referentes a inércia, assimilação e comunicação.

O termo  $X_r$  é um indivíduo diferente de  $X_{t-1}$  e obtido através de uma das estratégias de memória. A estratégia de memória pode assumir os valores  $S_g$ ,  $P_B$ ,  $S_g - rnd$  e  $P_B - rnd$ . Caso a estratégia de memória seja selecionada com o valor  $S_g$ , então  $X_r$  é o melhor indivíduo da geração anterior. Caso o valor selecionado seja  $P_B$ , é o melhor indivíduo de todas as gerações passadas. O valor de  $S_g - rnd$  indica uma recombinação de indivíduos da geração anterior e o valor de  $P_B - rnd$  indica uma recombinação de indivíduos de todas as gerações anteriores. Em todos os experimentos realizados neste trabalho, a estratégia escolhida foi a  $P_B - rnd$ . Ela foi escolhida devido a já ter sido utilizada no C-DEEPSO anteriormente e apresentar bons resultados em análises anteriores.

$C$  é uma matriz diagonal  $n \times n$  binária oriunda de uma distribuição de Bernoulli. Ela é criada a partir da geração de  $n$  números pseudo aleatórios entre 0 e 1. Se o número  $n_i$  for menor que uma taxa de comunicação  $P$ , também entre 0 e 1, informada pelo usuário na entrada do problema, então  $C_{i,i} = 1$ . Caso contrário,  $C_{i,i} = 0$ .

Na equação 5,  $X_{gb}^*$  é o melhor indivíduo encontrado na geração anterior que passa por uma mutação mostrada na Equação 5.

$$X_{gb}^* = X_{gb} [1 + \tau \times N(0, 1)] \quad (5)$$

Nela, o indivíduo é multiplicado por um número aleatório retirado de uma distribuição Gaussiana por um fator  $\tau$  chamado de fator de mutação, que varia de 0 a 1.

No C-DEEPSO, os pesos associados à inércia, assimilação e comunicação são modificados de acordo com a Equação 6. Nela, é usada uma Distribuição Gaussiana para modificar levemente a solução atual no espaço de busca.

$$w^* = w + \tau \times N(0, 1) \quad (6)$$

Basicamente, no C-DEEPSO, um novo indivíduo é criado através de uma mistura do indivíduo antigo com o melhor indivíduo da geração anterior e o melhor indivíduo já encontrado em todas as gerações. Os dois fatores decididos pelo usuário que mais influenciam nesses valores e no peso dessas combinações são a probabilidade de comunicação e o fator de mutação. Por este motivo, eles foram os parâmetros em que realizamos o nosso ajuste fino.

### 2.3 AJUSTE FINO DE PARÂMETROS

O ajuste fino de parâmetros é um processo em que métodos estatísticos podem ser aplicados para descobrir o melhor valor ou combinação de valores de um ou mais parâmetros. Estes melhores valores de um parâmetro servem para apenas um algoritmo em um determinado problema. No ajuste fino, primeiro é decidido em quais parâmetros será realizado este processo. No C-DEEPSO, foi decidido pelos parâmetros de taxa de mutação e probabilidade de comunicação pelo fato deles serem os componentes principais do algoritmo.

O ajuste fino do C-DEEPSO foi realizado com o *iRace*, um pacote do R para realização do ajuste fino de parâmetros (López-Ibáñez et al, 2016). Com este pacote, é possível escolher o número de vezes que o algoritmo seria rodado, o teste estatístico a ser utilizado para comparação dos resultados, o grau de confiança do nosso resultado, o tempo máximo de execução do *iRace*, entre outros fatores.

O *iRace* utiliza o método de corrida iterativa para realizar o ajuste fino de parâmetros. Nele, o algoritmo é rodado diversas vezes com diferentes configurações e seus resultados são armazenados. De tempos em tempos esses resultados são comparados com uso de testes estatísticos e as configurações de parâmetros que forem comprovadamente piores são descartadas. No final da corrida, quando restar só uma configuração ou caso não seja possível provar a diferença de resultado entre os parâmetros, é retornado os melhores conjuntos de parâmetros para o algoritmo.

### 3 TRABALHOS RELACIONADOS

Em geral, os trabalhos que tratam do ajuste fino de parâmetros em meta-heurísticas abordados aqui não apresentam muitos resultados de experimentos, realizando apenas uma análise teórica e reportando os benefícios desse processo. Trabalhos aplicando esse processo em problemas da área da energia são escassos na literatura. Em contrapartida, os benefícios de se aplicar ajuste fino de parâmetros em algoritmos de aprendizado de máquinas (Too e outros, 2019; Vrbancic e Podgorelec, 2020), otimização (Allawi e outros, 2019; Joshi e Bansal, 2020) e até mesmo alguns casos com uso de meta-heurística para otimizar parâmetros de redes neurais (Tien Bui e outros, 2021) já foram discutidos e estudados.

Na área de ajuste fino de parâmetros com o método de corrida iterativa em meta-heurísticas foram encontradas poucas pesquisas. Em Nannen e outros, (2008) foram citados os benefícios de se realizar o ajuste fino em algoritmos e as diferentes formas de se realizar esse processo. Em Stützle e López-Ibáñez, (2019) foram estudados alguns métodos para se realizar o ajuste fino em meta-heurísticas, entre eles o método de corrida iterativa utilizado pelo *iRace*, porém não realizaram um experimento com o método sendo aplicado em nenhum algoritmo para nenhum problema. Em Abdessalem e outros, (2018) foi realizado o controle de parâmetros em um algoritmo evolutivo através de técnicas de aprendizado de máquina, prática similar a realizada neste trabalho, porém no controle de parâmetros os melhores parâmetros do algoritmos são encontrados durante a execução do mesmo, ao invés de antes, como é feito no ajuste fino. Já Corazza e outros (2021) realizou o processo de ajuste fino em um algoritmo híbrido, oriundo do PSO, utilizando o *iRace* e comparou seus resultados com um ajuste fino utilizando o REVAC (Nannen e Eiben, 2007), um outro método para realização desse processo. Ele concluiu que o método do *iRace* se mostrou mais robusto que o método aplicado pelo REVAC.

Na literatura, é possível encontrar trabalhos que tratam dos benefícios do ajuste fino de parâmetros em algoritmos de aprendizado de máquina. Weerts e outros (2020) realizaram um estudo mostrando os benefícios dessa prática e fez uma análise da perda de performance em casos que os parâmetros foram escolhidos arbitrariamente. Em Friedrichs e Igel (2005) e Wang e outros (2019), meta-heurísticas foram utilizadas para realizar o ajuste fino de múltiplos parâmetros em algoritmos de aprendizado de máquina, ao invés do método utilizado no trabalho de corrida iterativa.

O uso de meta-heurísticas para a área de eficiência energética já existe uma ampla literatura. É possível ver o uso do PSO em um problema de OPF em Abido (2002). Já Joseph e outros (2014) propuseram um método utilizando o PSO para maximizar a capacidade de carga do sistema dentro das margens de estabilidade.

Especificamente em controladores para eficiência energética em usinas eólicas, também é encontrado estudo na literatura. Kotur e Stefanov (2019) propuseram um controlador de fluxo ótimo de energia em um sistema *offshore* de usinas eólicas. Usando o sistema IEEE 14-bus para teste, utilizaram o método dos pontos interiores para resolver o problema. Um outro controlador para usinas eólicas foi proposto para maximizar a reserva de energia em Siniscalchi-Minna e outros (2019). Em Sakipour e Abdi (2020) nove algoritmos evolutivos foram estudados para serem usados em problemas de armazenamento de energia eólica. O principal objetivo era encontrar uma forma eficiente de armazenar energia eólica gerada para um microgrid. Os resultados mostraram que o *Grey Wolf Optimizer* (GWO) teve os melhores resultados para esse problema. Seok e Chen (2019) propuseram um modelo inteligente de coalizão de uma usina eólica, resolvido por programação matemática, de forma a maximizar os lucros. Eles concluíram que o modelo podia ser especialmente aplicado em usinas eólicas heterogêneas.

Em Pham e outros (2014), foi proposto um algoritmo de otimização chamado *Mean-Variance Mapping Optimization* (MVMO). Este algoritmo foi proposto para executar o controle de fluxo elétrico e reduzir a perda energética em uma usina eólica. Este é o algoritmo estado da arte de um dos problemas em foi testado o nosso *framework*.

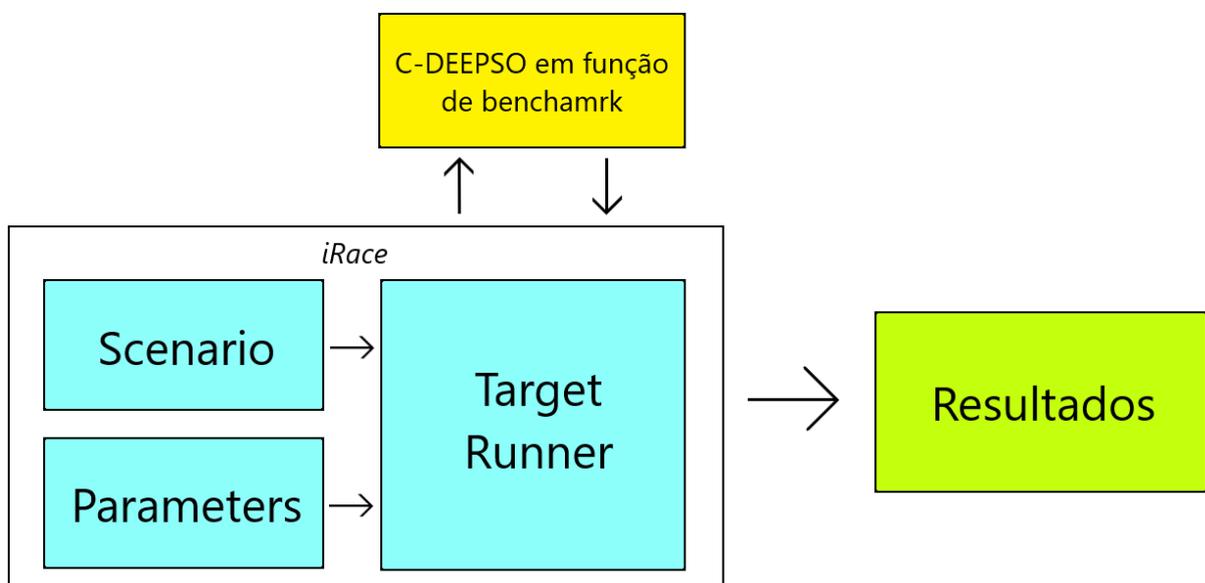
A abordagem realizada neste trabalho se diferencia das citadas anteriormente por unir diferentes etapas do problema. Os trabalhos vistos ou tratam do ajuste de parâmetros em algoritmos evolutivos ou estudam a aplicação do algoritmo evolutivo em problemas de despacho elétrico. Neste trabalho é realizado o processo de ajuste fino de parâmetros do algoritmo evolutivo e depois ele é aplicado em um problema de despacho elétrico, realizando uma conexão dos processos apresentados nos trabalhos estudados. No final, a solução encontrada por este algoritmo ajustado ainda é comparada com a solução encontrada pelo algoritmo estado da arte deste problema.

#### 4 AJUSTE FINO DO C-DEEPSO EM FUNÇÕES DE BENCHMARK E EM UM PROBLEMA OPF

Com a idéia de determinar se é possível encontrar um conjunto de parâmetros que melhore um algoritmo evolutivo e, assim, maximizar a sua performance, foi realizado a integração do *iRace* com o C-DEEPSO e criado, assim, um *framework* para realizar o ajuste fino dos parâmetros probabilidade de comunicação e fator de mutação. Este *framework* foi testado em diferentes problemas de otimização, a começar pelas funções de Rosenbrock e Rastrigin, duas funções de *benchmark*. Depois com algumas modificações, ele foi testado em um problema OPF de geração de energia eólica.

##### 4.1 FUNÇÕES DE BENCHMARK

Neste tipo de problema, o *iRace* foi utilizado para realizar o ajuste fino dos parâmetros probabilidade de comunicação e fator de mutação do C-DEEPSO. Foi realizada uma integração do *iRace* com o C-DEEPSO para aplicação nas funções de Rosenbrock e Rastrigin, com 30, 50 e 100 dimensões inicialmente, como representado na Figura 4.



**Figura 4:** Esquema do *framework* criado unindo *iRace* e C-DEEPSO em função de *benchmark*

Nesta integração, o *iRace* é composto de 3 arquivos principais, o *Scenario*, *Parameters* e *Target Runner*. O *Scenario* é o arquivo onde foram estabelecidas as configurações gerais do *iRace*. Nele, foi escolhido o tipo de teste estatístico utilizado para comparação dos resultados,

a confiança dos resultados, o grau de liberdade, o número máximo de testes e o tempo máximo de execução do *iRace*. O *Parameters* é o arquivo onde foram definidos quais parâmetros seriam tunados no algoritmo e o domínio desses parâmetros.

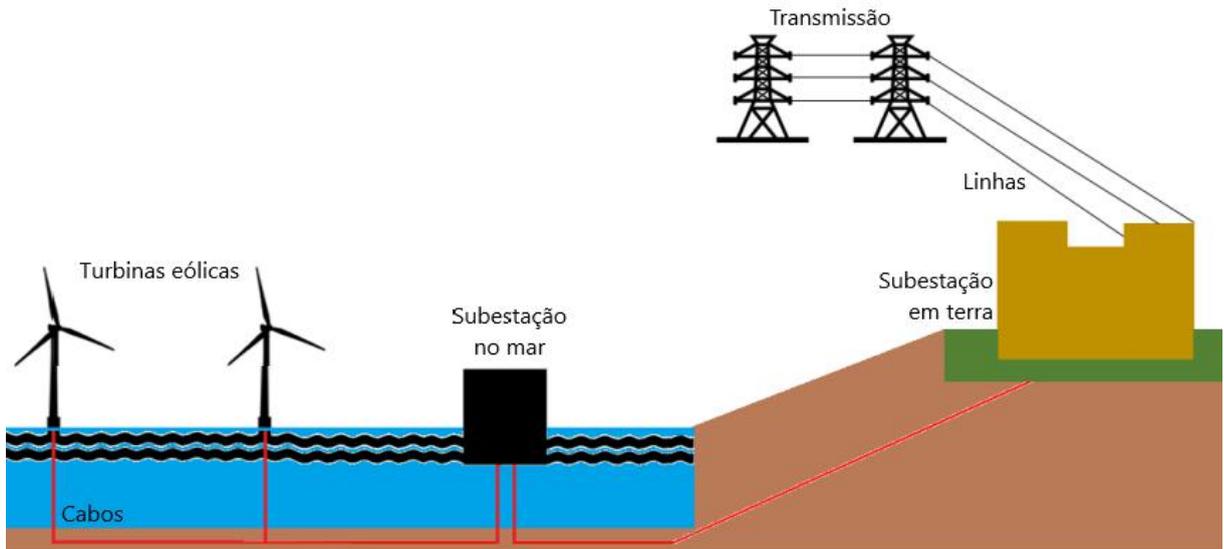
O *Target Runner* é o arquivo principal do *iRace*. Ele junta todas as informações estabelecidas no *Scenario* e no *Parameters* e manda essas informações para o C-DEEPSO rodar na função de Benchmark. Depois, pega o resultado do C-DEEPSO e armazena esse resultado. Após um número de simulações com diferentes configurações, ele realiza a comparação dos diferentes resultados utilizando o teste estatístico determinado. Ele então elimina os conjuntos de parâmetros que apresentaram resultados estatisticamente inferiores. No final, ele pega as melhores duplas de probabilidade de comunicação e fator de mutação e a média de seus resultados e armazena em um arquivo de resultados para o usuário.

Este *framework* criado foi aplicado nas funções de Rosenbrock e Rastrigin. No caso da função de Rosenbrock, como ela é uma função em que o algoritmo utilizado não encontra o mínimo nas dimensões 30, 50 e 100, a referência de resultado foi o menor valor encontrado após 10.000 avaliações de função. O valor mínimo da função de Rosenbrock é 0 quando todas as variáveis forem 1.

No caso da função de Rastrigin, como diversas vezes o C-DEEPSO encontrou o valor mínimo, além de fazer a comparação do menor valor encontrado, também foi analisada a velocidade com que o C-DEEPSO chegou nesse valor, através do número de gerações. O valor mínimo da função de Rastrigin é 0 quando todas as variáveis forem 0.

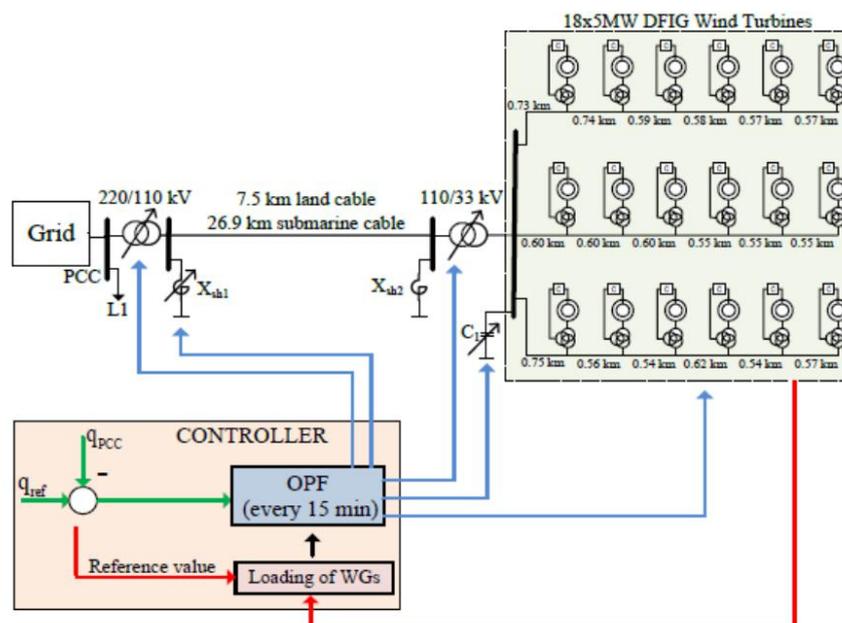
#### 4.2 PROBLEMA DE FLUXO ÓTIMO DE POTÊNCIA: PRODUÇÃO EM UMA USINA EÓLICA

Depois de validar o *framework* em funções de benchmark, ele foi adaptado para a realização do ajuste fino dos mesmos parâmetros, probabilidade de comunicação e fator de mutação, em um problema de fluxo ótimo de potência (OPF-like). O problema escolhido é um problema dinâmico de uma usina de energia eólica *offshore*. Usinas *offshore* são usinas onde a geração é realizada em um local distante da distribuição. Esta usina pode ser vista simplificada na Figura 5.



**Figura 5:** Exemplo genérico de uma usina *offshore* de energia eólica, adaptado de Marcelino e outros (2021)

O problema OPF para o qual nosso *framework* foi proposto é o sistema IEEE 41 (Erich e outros, 2014). É um problema de natureza dinâmica onde uma usina de energia eólica é simulada computacionalmente. Ele é composto de 96 instâncias e cada uma representa um intervalo de 15 minutos do sistema, o que equivale a 24 horas de despacho elétrico. O esquema elétrico de um típico sistema de controle para usina eólica *offshore* pode ser visto na Figura 6.



**Figura 6:** Esquema do controlador de uma usina eólica retirado de Marcelino e outros (2016)

É possível ver nesta imagem os dois fatores  $X_{sh1}$  e  $X_{sh2}$ , que são continuamente ajustáveis, e o capacitor  $C_1$  que provém suporte auxiliar de energia.  $L_1$  representa o tanto de energia que é gerada pela usina eólica. O sistema considera que a usina vai prover a energia necessária ( $q_{ref}$ ) até atingir o poder de acoplamento comum ( $q_{pcc}$ ). As características gerais do problema IEEE 41 estão dispostas na Tabela 2.

**Tabela 2:** Características do problema IEEE 41-Bus, adaptada de Marcelino e outros (2016)

Sistema IEEE41 - WPP			
Composição do Sistema de Testes	Geradores		18
	Cabo Submarino		1
	Transformadores	T1 ( <i>Stepwise</i> )	2
		T2 (Tap fixo)	18
Composição do problema de otimização	Variáveis de otimização	Variáveis Contínuas	18
		Variáveis Discretas	1
		Variáveis Contínuas e de Stepwise	1
Restrições			123

Esse problema pode ser descrito pela função objetivo disposta na Equação 7:

$$\min P_{loss} = \sum_{k=1}^{N_L} G_K [U_i^2 + U_j^2 - 2U_i U_j \cos(\delta_i - \delta_j)] \quad (7)$$

Esta equação descreve a função objetivo adotada no sistema IEEE 41, onde o termo  $N_L$  representa o número total de linhas no sistema e  $G_K$  é a condução da linha K. Os valores  $U_i$  e  $U_j$  são as magnitudes das extremidades emissora e receptora da linha. Os termos  $\delta_i$  e  $\delta_j$  são os ângulos das tensões.

Para resolver este problema OPF, o algoritmo de otimização é usado como um controlador automático, que provê um funcionamento eficiente da usina eólica de acordo com a entrada das condições da potência reativa do sistema. A potência reativa é suscetível a falhas e perdas que devem ser minimizadas. Ajustes para minimizar essas perdas podem ser realizados nos pontos VAR. Pontos VAR são pontos de entrada para execuções de comandos

nas turbinas eólicas. As programações das entradas das turbinas são operadas pelo controlador da usina eólica através da alocação da potência reativa necessária igualmente para cada gerador.

A função objetivo está sujeito às seguintes restrições:

$$P_i = P_i^{gen} - P_i^{load} = \sum_{j=1}^n U_i U_j [G_{ij} \cos(\theta_i - \theta_j) + B_{ij} \sin(\theta_i - \theta_j)] \quad (8)$$

$$Q_i = Q_i^{gen} - Q_i^{load} = \sum_{j=1}^n U_i U_j [G_{ij} \sin(\theta_i - \theta_j) + B_{ij} \cos(\theta_i - \theta_j)] \quad (9)$$

As Equações 8 e 9 representam o balanceamento de energia ativa e reativa. Nelas,  $P_i$  representa a energia ativa gerada,  $Q_i$  é a potência reativa,  $U_i$  é a magnitude da voltagem e  $\theta_i$  é o ângulo da voltagem.

$$U_i^{min} \leq U_i \leq U_i^{max} \quad (10)$$

As inequações 10 representam as limitações para a voltagem dos barramentos (em kV), onde  $U_i$  corresponde a magnitude da voltagem dos barramentos e seus limites.

$$S_{ij}^{min} \leq S_{ij} \leq S_{ij}^{max} \quad (11)$$

As inequações 11 representam as limitações para o fluxo de potência aparente de uma ramificação. Nela,  $S_{ij}$  é o fluxo de potência aparente na extremidade de envio do circuito que conecta os barramentos i e j.

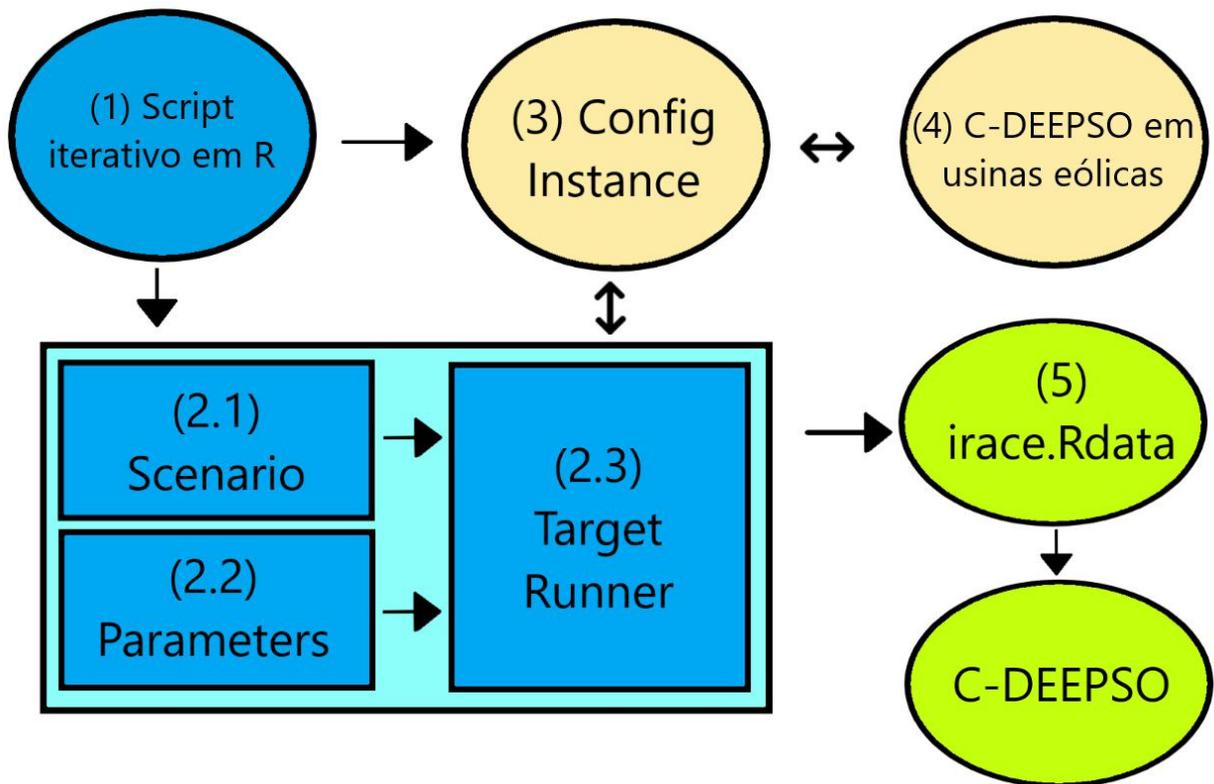
$$T_k^{min} \leq T_k \leq T_k^{max} \quad (12)$$

As inequações 12 representam as limitações para a mudança de *tap* do transformador, em que  $T_k$  corresponde aos vetores dos limites discretos de mudança de tap do transformador

$$Q_k^{min} \leq Q_k \leq Q_k^{max} \quad (13)$$

As inequações 13 representam as limitações para o shunt dos pontos VAR, onde  $Q_k$  são as variáveis discretas que representam os capacitores e reatores Neste trabalho as restrições descritas foram resolvidas via penalidade conforme Marcelino e outros (2021).

Para este problema, foi utilizada uma versão modificada do *framework* proposto, devido ao mesmo ter características dinâmicas e ser uma caixa preta. Esta versão modificada pode ser vista na Figura 7.



**Figura 7:** Esquema do *framework* criado unindo *iRace* e C-DEEPSO no IEEE 41

Nesta imagem, é possível ver a inserção de um novo *script* iterativo (1), que é responsável por iterar o *iRace*, chamando ele diversas vezes, cada vez para uma instância do problema. Ele envia as informações de qual instância está sendo realizado o processo de ajuste fino para um *script* de configuração de instância (3). Este *script* recebe a informação do *iRace* (2) e de (1) e configura o C-DEEPSO (4) de acordo com essas configurações. Então, ele realiza uma chamada para (4) e recebe o valor de resposta. Este valor é colocado no formato requisitado pelo *iRace* e enviado para (2.3) para realizar a análise dos dados. A cada iteração de (1), é gerado um arquivo *irace.Rdata* (5), que é o resultado do *iRace* com os melhores parâmetros para cada informação. No final são gerados 96 arquivos com esses resultados, então os melhores parâmetros são extraídos e salvos para uso futuro.

## 5 EXPERIMENTOS E RESULTADOS

Foi realizada inicialmente a aplicação do *framework* para o ajuste fino de parâmetros na otimização de problemas em que se conhece o ótimo, neste caso em funções de *benchmarks* para a compreensão na prática de como ele funciona. Assim foi possível validar como o *framework* gera os resultados e possibilitando obter uma estimativa de tempo de ação do *framework* para os problemas. Além disso, o trabalho visou explorar como se comportam os parâmetros do C-DEEPSO em diferentes tipos de problemas.

Estes parâmetros analisados foram o fator de mutação e probabilidade de comunicação. Essas duas taxas variam entre o intervalo de  $[0,1]$  e foram analisados os parâmetros com intervalo de 0.1. Isso produziu 9 fatores distintos de mutação e 9 de comunicação, sendo simuladas 31 vezes, totalizando 2511 resultados diferentes para comparação.

### 5.1 FUNÇÃO DE ROSENBRACK

Foi realizado o ajuste fino com o *framework* para o C-DEEPSO na função de Rosenbrock para as dimensões 30, 50 e 100. Neste experimento foi escolhido o Teste T de Student para realizar as comparações. O ajuste fino foi executado com 31 graus de liberdade e 95% de confiança. Os melhores parâmetros encontrados podem ser vistos na Tabela 3, junto com suas respectivas médias e desvio padrão. Em todos os casos, após a execução do algoritmo, testamos a normalidade da amostra gerada com o Teste de Kolmogorov-Smirnov. Em nenhum dos casos havia evidências suficientes para rejeitar a hipótese de que os dados seguem uma distribuição normal, com uma significância  $\alpha < 0.05$ .

**Tabela 3:** Resultados do ajuste fino de parâmetros do C-DEEPSO na função de Rosenbrock para 30, 50 e 100 dimensões

Dimensão	Mut.	Com.	Média	DP
30	0.1	0.5	15.8193	1.5743
50	0.5	0.6	36.9891	2.7556
100	0.4	0.3	90.4062	2.5476

Com esses resultados, foram realizadas comparações entre o PSO, DE, C-DEEPSO com parâmetros empíricos, hC-DEEPSO e C-DEEPSO com parâmetros tunados. O hC-DEEPSO é uma versão do C-DEEPSO modificada com uma abordagem híbrida (Marcelino e outros, 2018).

A primeira comparação foi entre o C-DEEPSO com parâmetros empíricos e parâmetros tunados, para 30, 50 e 100 dimensões. Os parâmetros empíricos utilizados foram 0.5 para mutação e 0.7 para comunicação. É possível ver essa comparação na Tabela 4:

**Tabela 4:** Comparação do C-DEEPSO tunado com o C-DEEPSO empírico na função de Rosenbrock com 30, 50 e 100 dimensões

Dimensão	Média com parâmetros tunados	Média com parâmetros empíricos	Melhora
30	15.8193	19.22	17.69%
50	36.9891	42.03	11.99%
100	90.4062	95.7	5.53%

Após isto, foi realizada uma comparação dos resultados dos outros algoritmos com o C-DEEPSO tunado, usando o teste T de Student para comparação. Os valores de todos os algoritmos são apresentados na Tabela 5. Na coluna comparação, é possível observar qual algoritmo se mostrou mais robusto entre o C-DEEPSO tunado e o algoritmo referente a linha.

**Tabela 5:** Comparação do C-DEEPSO tunado com hC-DEEPSO, C-DEEPSO empírico, DE e PSO na função de Rosenbrock

Dimensão	Algoritmo	Média	DP	Comparação
30	PSO	81.27	41.21	C-DEEPSO tunado
	DE	31.13	17.12	C-DEEPSO tunado
	C-DEEPSO empírico	19.22	2.01	C-DEEPSO tunado
	hC-DEEPSO	15.99	1.9	-
50	PSO	174.22	113.63	C-DEEPSO tunado

	DE	50.33	16.85	C-DEEPSO tunado
	C-DEEPSO empírico	42.03	2.23	C-DEEPSO tunado
	hC-DEEPSO	38.9	2.54	-
100	PSO	250.68	24.64	C-DEEPSO tunado
	DE	91.83	3.82	-
	C-DEEPSO empírico	95.7	1.76	C-DEEPSO tunado
	hC-DEEPSO	89.6	1.9	-

Por meio da análise de resultados do Teste T, com 31 graus de liberdade e 95% de confiança, foi possível verificar que o algoritmo C-DEEPSO tunado, quando comparado ao hC-DEEPSO, não identificou diferenças entre as médias testadas, uma vez que  $p\text{-valor} < 0.05$ . Assim, o C-DEEPSO tunado obtém soluções semelhantes a sua versão híbrida, o hC-DEEPSO.

Quando usado o Teste T para comparar os resultados do C-DEEPSO tunado com o C-DEEPSO empírico e o PSO, para os casos de 30, 50 e 100 dimensões, o teste apontou uma diferença nas médias, com  $p\text{-valor} > 0.05$ . O mesmo aconteceu na comparação do C-DEEPSO tunado com o DE para os casos de 30 e 50 dimensões. Já no caso de 100 dimensões, o Teste T não apresentou diferença entre as médias do C-DEEPSO tunado com o DE, com um  $p\text{-valor} < 0.05$ .

## 5.2 FUNÇÃO DE RASTRIGIN

Para o caso de Rastrigin, foram utilizadas as mesmas configurações que para o caso de Rosenbrock. Porém, além de comparar o resultado final, pelo C-DEEPSO ser capaz de diversas vezes encontrar o ponto ótimo da função de Rastrigin, também é comparada a velocidade em que o C-DEEPSO encontrou esse ponto ótimo. Esta velocidade foi medida por meio do número de gerações, visto que não era possível fazermos a medição através do número de avaliações de função neste caso devido à forma como o *iRace* é implementado, e o número de gerações nos provê uma boa estimativa da velocidade do algoritmo.

No caso do C-DEEPSO tunado, por ter encontrado o ponto mínimo da função de Rastrigin em todas as ocasiões, é mostrado na Tabela 6 também o número de gerações passadas até chegar neste valor, além dos melhores parâmetros.

**Tabela 6:** Resultados do ajuste fino de parâmetros do C-DEEPSO na função de Rastrigin para 30, 50 e 100 dimensões

Dimensão	Mut.	Com.	Média	DP	Média de gerações	DP de gerações
30	0.9	0.9	0	0	12.8	0.4
50	0.6	0.9	0	0	11.7	2.5
100	0.8	0.9	0	0	9.6	2.68

Foi comparada, então, a velocidade do C-DEEPSO tunado com o C-DEEPSO com parâmetros empíricos de 0.5 e 0.9, como na função de Rosenbrock. Os resultados foram comparados visto que nos dois casos foi encontrado o valor mínimo da função de Rastrigin em todos os testes realizados. Essa comparação pode ser vista na Tabela 7.

**Tabela 7:** Comparação da velocidade do C-DEEPSO tunado com o C-DEEPSO empírico na função de Rastrigin com 30, 50 e 100 dimensões

Dimensão	Média de gerações com parâmetros tunados	Média de gerações com parâmetros empíricos	Melhora
30	12.8	14.9	14.09%
50	11.7	11.8	0.85%
100	9.6	10.8	11.11%

Foram realizados então testes com o Teste T para mostrar estatisticamente se o C-DEEPSO tunado se apresentou mais rápido que o C-DEEPSO com parâmetros empíricos de 0.5 e 0.9. Este teste mostrou que, para 30 dimensões, o C-DEEPSO tunado apresentou ser um algoritmo mais rápido, porém para 50 e 100 dimensões não há evidências que provem uma diferença estatística entre os valores.

Foi comparado então, os resultados encontrados com os resultados de outros algoritmos, o PSO, DE e o hC-DEEPSO. Essa comparação pode ser vista na Tabela 8. Na coluna comparação, é possível ver qual algoritmo se mostrou melhor entre o C-DEEPSO tunado e o algoritmo referente a linha.

**Tabela 8:** Comparação do C-DEEPSO tunado com hC-DEEPSO, C-DEEPSO empírico, DE e PSO na função de Rastrigin

Dimensão	Algoritmo	Média	DP	Comparação
30	PSO	37.81	7.45	C-DEEPSO tunado
	DE	2.53	5.19	C-DEEPSO tunado
	C-DEEPSO empírico	0	0	-
	hC-DEEPSO	0	0	-
50	PSO	75.3	19.55	C-DEEPSO tunado
	DE	41.47	8.8	C-DEEPSO tunado
	C-DEEPSO empírico	0	0	-
	hC-DEEPSO	0	0	-
100	PSO	186.04	4.93	C-DEEPSO tunado
	DE	261.19	1.64	C-DEEPSO tunado
	C-DEEPSO empírico	0	0	-
	hC-DEEPSO	0	0	-

Por meio da análise de resultados do Teste T, com 31 graus de liberdade e 95% de confiança, não foi possível provar uma diferença estatística entre o C-DEEPSO tunado, o C-DEEPSO empírico e o hC-DEEPSO em todos os casos, visto que todos encontram o mínimo da função em todos os testes. O C-DEEPSO tunado se mostrou mais robusto que o PSO e o DE em todos os casos, com p-valor  $< 0.05$ .

### 5.3 PROBLEMA DE FLUXO ÓTIMO DE POTÊNCIA – IEEE 41 USINA EÓLICA

Depois que o *framework* foi devidamente testado nas funções de *benchmark*, ele foi aplicado para solução de um problema real de geração de energia elétrica por meio de uma usina eólica *offshore*. Neste problema, foi realizado o ajuste fino do C-DEEPSO porém com outro teste estatístico no *iRace*, o Teste de Friedman. Este teste foi escolhido visto que não foi provada a normalidade nas amostras de 31 execuções no caso do problema IEEE 41. O Teste T só pode ser usado em casos em que os dados seguem a distribuição normal.

Após o ajuste fino de parâmetros das 96 instâncias do problema, foi efetuada a comparação dos resultados do C-DEEPSO tunado com o DEEPSO (Miranda e Alves, 2013) e o MVMO (Pham e outros, 2014), o algoritmo estado da arte para este problema. É importante ressaltar que os algoritmos DEEPSO e MVMO foram testados em seus melhores conjuntos de parâmetros.

A comparação entre os algoritmos foi realizada utilizando uma ferramenta chamada *DSC Tool* (Eftimov e outros, 2020). O *DSC Tool* é uma ferramenta de comparação de resultados *pos-hoc*. Nesta tabela, foi marcado com (+) os casos em que o C-DEEPSO tunado se mostrou mais robusto que os outros algoritmos, com (=) os casos em que não apresenta evidências suficientes para diferenciar a média dos algoritmos e com (-) os casos em que se mostrou uma diferença na média dos algoritmos, porém com a média do C-DEEPSO tunado sendo menor. Essas comparações foram feitas utilizando o Teste de Friedman, com uma confiança de 95%. Essa comparação se encontra na Tabela 9 e no Figura 8..

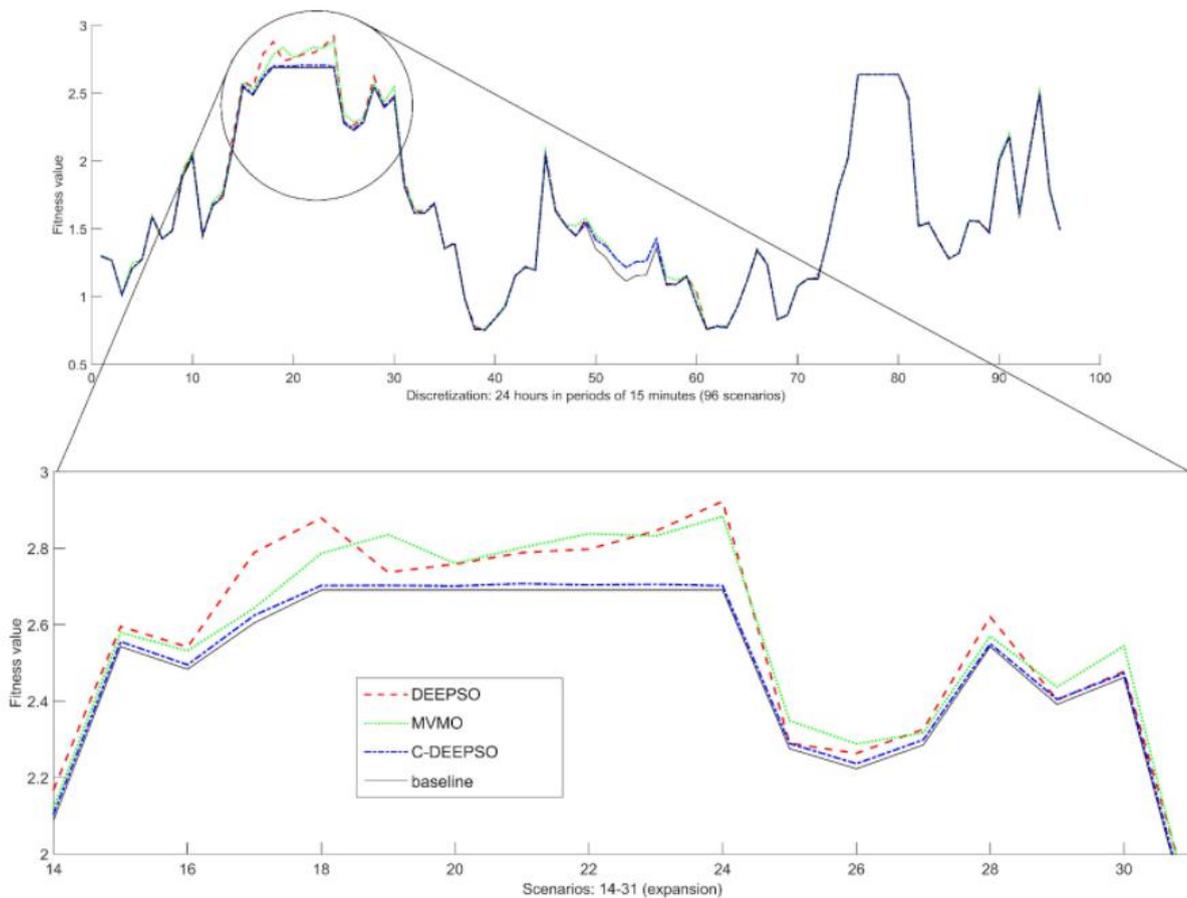
**Tabela 9:** Comparação do C-DEEPSO tunado com DEEPSO e MVMO e sua lista de parâmetros no problema IEEE 41

Instância	Comp.	C-DEEPSO tunado				DEEPSO		MVMO	
		Mut	Com.	Média	DP	Média	DP	Média	DP
1	(+)	0.7	0.5	<b>1.296144</b>	<b>0.006291</b>	1.300416	0.006464	1.299471	0.002149
2	(+)	0.3	0.4	<b>1.266966</b>	<b>0.000672</b>	1.27365	0.006849	1.270922	0.002452
3	(+)	0.5	0.2	<b>1.013735</b>	<b>0.003241</b>	1.018528	0.009137	1.017974	0.00618
4	(+)	0.9	0.5	<b>1.20609</b>	<b>0.001601</b>	1.210449	0.006677	1.214193	0.012536
5	(+)	0.1	0.3	<b>1.27449</b>	<b>0.00117</b>	1.277351	0.004858	1.277518	0.00302

6	(+)	0.4	0.4	<b>1.589284</b>	<b>0.001323</b>	1.590727	0.002382	1.595016	0.002961
7	(+)	0.6	0.9	<b>1.4272</b>	<b>0.002118</b>	1.430635	0.003885	1.431654	0.003422
8	(+)	0.5	0.4	<b>1.4864</b>	<b>0.002283</b>	1.490068	0.002738	1.491214	0.001765
9	(+)	0.7	0.5	<b>1.8886</b>	<b>0.006691</b>	1.920608	0.025138	1.912675	0.017726
10	(+)	0.6	0.7	<b>2.03698</b>	<b>0.006763</b>	2.059875	0.021007	2.068124	0.024814
11	(+)	0.9	0.8	<b>1.44932</b>	<b>0.001774</b>	1.460489	0.019159	1.466775	0.028831
12	(+)	0.2	0.1	<b>1.673235</b>	<b>0.002128</b>	1.687921	0.00919	1.696736	0.020164
13	(+)	0.1	0.1	<b>1.7389</b>	<b>0.006</b>	1.75454	0.009227	1.765304	0.019683
14	(+)	0.6	0.8	<b>2.10522</b>	<b>0.006694</b>	2.138487	0.02316	2.131292	0.015799
15	(+)	0.6	0.5	<b>2.55552</b>	<b>0.009603</b>	2.576392	0.015664	2.583475	0.017772
16	(+)	0.7	0.3	<b>2.495002</b>	<b>0.008118</b>	2.52295	0.022583	2.526806	0.015743
17	(+)	0.7	0.4	<b>2.624494</b>	<b>0.012738</b>	2.763067	0.063241	2.752978	0.049748
18	(+)	0.7	0.3	<b>2.7023</b>	<b>0.0155</b>	2.837714	0.049846	2.832812	0.065309
19	(+)	0.6	0.4	<b>2.702778</b>	<b>0.013871</b>	2.838913	0.054696	2.832115	0.05962
20	(+)	0.9	0.3	<b>2.701027</b>	<b>0.018485</b>	2.829544	0.062866	2.836847	0.037857
21	(+)	0.5	0.1	<b>2.7075</b>	<b>0.0225</b>	2.826354	0.054087	2.824264	0.052248
22	(+)	0.7	0.3	<b>2.7038</b>	<b>0.0204</b>	2.840033	0.041991	2.8105	0.043337
23	(+)	0.8	0.4	<b>2.7057</b>	<b>0.01741</b>	2.84924	0.052075	2.813017	0.060382
24	(+)	0.9	0.3	<b>2.701787</b>	<b>0.016174</b>	2.824618	0.060398	2.830624	0.038445
25	(+)	0.5	0.7	<b>2.289114</b>	<b>0.001733</b>	2.320457	0.026687	2.328435	0.025966
26	(+)	0.9	0.8	<b>2.2362752</b>	<b>0.008451</b>	2.279125	0.018528	2.273852	0.022695
27	(+)	0.9	0.5	<b>2.298706</b>	<b>0.000826</b>	2.332162	0.019854	2.326763	0.016357
28	(+)	0.7	0.5	<b>2.550038</b>	<b>0.005145</b>	2.584022	0.024599	2.588382	0.025195
29	(+)	0.9	0.7	<b>2.404554</b>	<b>0.004696</b>	2.434756	0.025572	2.435477	0.016747
30	(+)	0.6	0.7	<b>2.472782</b>	<b>0.004116</b>	2.504222	0.019501	2.510542	0.021636
31	(+)	0.9	0.9	<b>1.821617</b>	<b>0.009539</b>	1.837128	0.013342	1.845527	0.031906
32	(+)	0.5	0.3	<b>1.619</b>	<b>0.00318</b>	1.632586	0.012615	1.638392	0.024243
33	(+)	0.4	0.7	<b>1.61528</b>	<b>0.002049</b>	1.620574	0.005501	1.621487	0.002227
34	(=)	0.1	0.8	<b>1.68682</b>	<b>0.00261</b>	1.688831	0.003948	1.690077	0.002368
35	(=)	0.3	0.9	<b>1.3567</b>	<b>0.0022</b>	1.35859	0.003192	1.359256	0.00347
36	(+)	0.7	0.8	<b>1.3935</b>	<b>0.0016</b>	1.396066	0.003154	1.397859	0.003853
37	(+)	0.9	0.9	<b>0.984648</b>	<b>0.001103</b>	0.989143	0.008027	0.986252	0.002255

38	(+)	0.5	0.9	<b>0.7588865</b>	<b>0.001457</b>	0.765184	0.015427	0.761642	0.002352
39	(-)	0.4	0.8	<b>0.754861</b>	<b>0.001483</b>	0.760265	0.012586	0.757799	0.003082
40	(=)	0.3	0.9	<b>0.837967</b>	<b>0.000849</b>	0.840169	0.007057	0.84189	0.005286
41	(+)	0.4	0.4	<b>0.926908</b>	<b>0.013581</b>	0.927213	0.007331	0.92718	0.003881
42	(+)	0.8	0.5	<b>1.152199</b>	<b>0.000312</b>	1.155496	0.005161	1.155485	0.002304
43	(+)	0.9	0.7	<b>1.217169</b>	<b>0.001247</b>	1.220947	0.006493	1.221542	0.004354
44	(+)	0.8	0.4	<b>1.196379</b>	<b>0.001243</b>	1.201751	0.009862	1.199745	0.002207
45	(+)	0.7	0.8	<b>2.03498</b>	<b>0.006113</b>	2.06263	0.018563	2.07081	0.025702
46	(+)	0.6	0.1	<b>1.632232</b>	<b>0.00267</b>	1.644981	0.009954	1.655683	0.025572
47	(+)	0.9	0.6	<b>1.52315</b>	<b>0.001937</b>	1.535093	0.010909	1.552997	0.033933
48	(+)	0.8	0.3	<b>1.449133</b>	<b>0.00123</b>	1.464665	0.021088	1.466773	0.021983
49	(+)	0.5	0.8	<b>1.55468</b>	<b>0.000492</b>	1.573228	0.014294	1.593472	0.038212
50	(+)	0.7	0.9	<b>1.41897</b>	<b>0.000114</b>	1.42171	0.004049	1.429751	0.015539
51	(+)	0.8	0.9	<b>1.372518</b>	<b>0.000059</b>	1.37299	0.000638	1.379473	0.009745
52	(-)	0.8	0.9	<b>1.280373</b>	<b>0.000035</b>	1.280345	0.0000392	1.280361	0.0000504
53	(+)	0.7	0.8	<b>1.216041</b>	<b>0.000018</b>	1.216045	0.0000526	1.216075	0.0000743
54	(=)	0.7	0.9	<b>1.257851</b>	<b>0.000021</b>	1.257835	0.0000225	1.257843	0.0000272
55	(=)	0.6	0.9	<b>1.261069</b>	<b>0.000033</b>	1.261064	0.0000458	1.261064	0.0000453
56	(+)	0.6	0.7	<b>1.425776</b>	<b>0.000095</b>	1.427878	0.002874	1.435186	0.016387
57	(+)	0.8	0.5	<b>1.089644</b>	<b>0.0013</b>	1.100627	0.019237	6.477143	29.84215
58	(+)	0.8	0.7	<b>1.09408</b>	<b>0.00575</b>	1.107108	0.024072	1.12865	0.037902
59	(+)	0.4	0.3	<b>1.150634</b>	<b>0.002753</b>	1.162567	0.025166	1.18516	0.032974
60	(+)	0.9	0.5	<b>0.942244</b>	<b>0.001247</b>	0.967602	0.041148	485.9867	2700.31
61	(+)	0.9	0.4	<b>0.759019</b>	<b>0.000856</b>	0.768336	0.019544	0.764196	0.009798
62	(+)	0.9	0.6	<b>0.7825</b>	<b>0.0079</b>	0.785943	0.010336	0.788255	0.01326
63	(+)	0.9	0.5	<b>0.7726</b>	<b>0.00102</b>	0.781982	0.018497	0.779951	0.012635
64	(+)	0.9	0.5	<b>0.919392</b>	<b>0.00066</b>	0.924948	0.016801	0.923803	0.0095
65	(+)	0.9	0.9	<b>1.11817</b>	<b>0.000672</b>	1.12153	0.004301	1.12021	0.002402
66	(+)	0.9	0.5	<b>1.342056</b>	<b>0.001249</b>	1.344351	0.002214	1.344502	0.001449
67	(+)	0.4	0.9	<b>1.236029</b>	<b>0.001281</b>	1.23958	0.004837	1.239307	0.001269
68	(+)	0.4	0.7	<b>0.831671</b>	<b>0.000856</b>	0.838725	0.014454	0.834139	0.003636
69	(+)	0.5	0.7	<b>0.864264</b>	<b>0.004413</b>	0.868957	0.010561	0.865998	0.002148

70	(+)	0.5	0.8	<b>1.073961</b>	<b>0.000588</b>	1.076649	0.004521	1.075708	0.001544
71	(+)	0.5	0.7	<b>1.131685</b>	<b>0.000458</b>	1.13391	0.003072	1.134448	0.004129
72	(+)	0.7	0.7	<b>1.129945</b>	<b>0.00045</b>	1.132922	0.003447	1.132384	0.002756
73	(=)	0.5	0.1	<b>1.419485</b>	<b>0.001599</b>	1.424401	0.00346	1.42228	0.002085
74	(+)	0.9	0.6	<b>1.780371</b>	<b>0.004054</b>	1.789148	0.02496	1.783307	0.003791
75	(+)	0.8	0.8	<b>2.011265</b>	<b>0.000332</b>	75.50458	409.1587	2.022251	0.002981
76	(+)	0.9	0.5	<b>2.63726</b>	<b>0.0000002</b>	1269600	5761404	2.637258	0.000000037
77	(=)	0.2	0.2	<b>2.637259</b>	<b>0.00000004</b>	923281.7	5118422	2.637258	0.000000025
78	(+)	0.3	0.4	<b>2.637259</b>	<b>0.00000043</b>	646388.5	3598924	7085962	39438.28
79	(=)	0.8	0.3	<b>2.637258</b>	<b>0.00000033</b>	20074.95	111757.9	2.637258	0.000000030
80	(=)	0.3	0.4	<b>2.637259</b>	<b>0.00000045</b>	2.637258	0.000000074	2.637258	0.000000030
81	(+)	0.7	0.2	<b>2.461037</b>	<b>0.001241</b>	2.47379	0.03726	2.463748	0.006993
82	(+)	0.8	0.8	<b>1.520083</b>	<b>0.001706</b>	1.522664	0.003022	1.524768	0.00223
83	(+)	0.4	0.5	<b>1.544517</b>	<b>0.001797</b>	1.546459	0.002979	1.549251	0.001964
84	(+)	0.6	0.9	<b>1.407016</b>	<b>0.00218</b>	1.409471	0.002567	1.409581	0.001261
85	(+)	0.1	0.9	<b>1.27947</b>	<b>0.00194</b>	1.282229	0.003461	1.281927	0.001248
86	(+)	0.6	0.9	<b>1.320293</b>	<b>0.0028</b>	1.323782	0.003268	1.322905	0.00182
87	(+)	0.1	0.5	<b>1.558535</b>	<b>0.001353</b>	1.560763	0.002548	1.563525	0.00138
88	(+)	0.1	0.9	<b>1.552623</b>	<b>0.00177</b>	1.554608	0.002166	1.557436	0.002001
89	(+)	0.4	0.5	<b>1.472417</b>	<b>0.001241</b>	1.47573	0.003377	1.476867	0.001657
90	(+)	0.2	0.7	<b>2.006577</b>	<b>0.002587</b>	2.011905	0.006837	2.022072	0.006272
91	(+)	0.8	0.7	<b>2.180685</b>	<b>0.000385</b>	2.192417	0.012609	2.198064	0.005255
92	(=)	0.3	0.6	<b>1.61232</b>	<b>0.0018</b>	1.617162	0.002416	1.622292	0.003239
93	(+)	0.8	0.9	<b>2.05536</b>	<b>0.0059</b>	2.064203	0.010235	2.071865	0.00516
94	(+)	0.6	0.9	<b>2.48991</b>	<b>0.0043</b>	2.516119	0.023582	2.512312	0.007086
95	(+)	0.6	0.8	<b>1.7829</b>	<b>0.0015</b>	1.784854	0.003648	1.79233	0.005024
96	(+)	0.8	0.4	<b>1.497641</b>	<b>0.00156</b>	1.500072	0.002245	1.502958	0.002782



**Figura 8::** Comparação do C-DEEPSO tunado com DEEPSO e MVMO

É possível notar que, em 87.5% dos casos, o C-DEEPSO tunado apresentou uma diferença estatística entre as médias, comprovado pelo Teste de Friedman. Nesses casos o valor médio apresentado pelo C-DEEPSO tunado indica que, aplicar a solução encontrada por ele pode trazer benefícios em relação aos demais algoritmos. Em outros 10.4%, não foi possível identificar diferença entre as médias com o Teste de Friedman. Já nos outros 2%, o Teste de Friedman apresentou uma diferença entre a média do C-DEEPSO tunado e do MVMO, porém a média do MVMO se mostrou menor, trazendo mais benefícios ao utilizarmos a solução dada por este método.

É possível notar que no gráfico, todas as meta-heurísticas não foram capazes de encontrar soluções próximas ao baseline entre os cenários 45 e 65. Possivelmente, as soluções ficam presas em uma bacia de atração pouco promissora. Como uma tarefa futura, para buscar sanar este problema, um operador de busca local pode ser incorporado ao C-DEEPSO visando encontrar soluções mais eficientes.

No final de um dia, caso o C-DEEPSO tunado fosse utilizado no lugar do MVMO, existiria uma minimização das perdas de geração de 2.34MWh. Numa projeção mensal, essa redução seria em torno de 70.02MWh por dia, energia suficiente para abastecer cerca de 450 casas a mais por mês. A solução otimizada e bem como a implementação da *framework* e o tutorial de instalação e configuração do iRace estão disponíveis em: <https://github.com/jvcavancini/C-DEEPSO>. De posse dessa documentação é possível replicar os experimentos e adequar o *framework* para outro problema de otimização.

## 6 CONCLUSÃO

Neste trabalho, foi proposto um *framework* que realiza o ajuste fino em algoritmos de otimização. Ele foi testado com o algoritmo C-DEEPSO, porém pode ser utilizado em qualquer algoritmo de otimização. É possível ver como o ajuste fino de parâmetros pode melhorar o desempenho do algoritmo em que é aplicado. Os resultados se apresentaram bastante satisfatórios tanto em funções de benchmark como no problema real de despacho elétrico abordado aqui. No caso de funções de benchmark, a melhora chegou a ser superior a 15% em alguns casos de teste quando comparado ao algoritmo com parâmetros empíricos. No problema IEEE 41, um problema dinâmico e de difícil resolução devido às várias restrições, foi apresentada uma diminuição comprovada de 6% na perda de energia durante a geração, em comparação ao algoritmo estado da arte, o MVMO.

Além dos benefícios vistos na melhora dos resultados, este framework desenvolvido utilizando o C-DEEPSO pode ser facilmente replicado para diferentes algoritmos em demais problemas, aumentando a eficiência em outros casos. Futuramente, seria interessante o estudo de outras formas mais velozes de realizar o ajuste fino ao invés do método da corrida iterativa. A corrida iterativa, ainda que nos dê resultados muito bons, não escala bem quando a quantidade ou o domínio dos parâmetros aumenta. Com uma nova técnica, computacionalmente mais rápida, seria possível ver uma melhora ainda maior dos resultados obtidos através do ajuste fino de um número maior de parâmetros.

Um outro estudo que pode dar continuidade a este trabalho é o estudo do controle de parâmetros em algoritmos de aprendizado de máquinas. Esses parâmetros podem inclusive, com técnicas de aprendizado, aprender qual combinação de parâmetros é a melhor em cada caso e mudar o valor dos parâmetros diversas vezes durante a execução do algoritmo.

Finalmente, como contribuição ao estado da arte, este trabalho gerou uma publicação em uma revista internacional indexada da área de sustentabilidade. A publicação pode ser acessada via DOI: <https://doi.org/10.3390/su132111924>.

## REFERÊNCIAS

- MARCELINO, C.G.; AVANCINI, J.V.C.; DELGADO, C.A.D.M.; WANNER, E.F.; JIMÉNEZ-FERNÁNDEZ, S.; SALCEDO-SANZ, S.; Dynamic Electric Dispatch for Wind Power Plants: A New Automatic Controller System Using Evolutionary Algorithms. **Sustainability**. 2021; 13(21):11924. <https://doi.org/10.3390/su132111924>
- MARCELINO C.G.; ALMEIDA P. E. M.; WANNER E.F.; CARVALHO L. M.; MIRANDA V.; Fundamentals of the C-DEEPSO algorithm and its application to the reactive power optimization of wind farms, **2016 IEEE Congress on Evolutionary Computation (CEC)**, 2016, pp. 1547-1554, doi: 10.1109/CEC.2016.7743973.
- MARCELINO, C.G.; ALMEIDA, P.E.M.; WANNER, E.F. et al. Solving security constrained optimal power flow problems: a hybrid evolutionary approach. *Appl Intell* 48, 3672–3690 (2018). <https://doi.org/10.1007/s10489-018-1167-5>
- GASPAR-CUNHA, A.; TAKAHASHI, R.; ANTUNES, C.H.; **Manual de Computação Evolutiva e meta-heurística**. 2012. DOI: <http://dx.doi.org/10.14195/978-989-26-0583-8>
- LÓPEZ-IBÁÑEZ M.; DUBOIS-LACOSTE, J.; CÁCERES, L.P.; BIRATTARI, M.; STÜTZLE, T.; The irace package: Iterated racing for automatic algorithm configuration, *Operations Research Perspectives*, Volume 3, 2016, Pages 43-58, ISSN 2214-7160, <https://doi.org/10.1016/j.orp.2016.09.002>.
- GUNAWAN, A.; LAU, H.; LINDAWATI,. (2011). **Fine-Tuning Algorithm Parameters Using the Design of Experiments Approach**. 278-292. 10.1007/978-3-642-25566-3\_21.
- NANNEN, V.; SMIT, S.K.; EIBEN, A.; (2008). **Costs and Benefits of Tuning Parameters of Evolutionary Algorithms**. 10.1007/978-3-540-87700-4\_53.
- LI, X.; TANG, K.; OMIDVAR, M.N.; YANG, Z.; QIN, K. (2013). **Benchmark Functions for the CEC'2013 Special Session and Competition on Large-Scale Global Optimization**.
- FRANK, S.; STEPONAVIČĚ, I.; REBENNACK, S.; (2012). **Optimal power flow: a bibliographic survey I. Energy Systems**. 3. 10.1007/s12667-012-0056-y.
- NELDER, J.; MEAD, R.; (1965). A Simplex Method for Function Minimization. **The Computer Journal**, Volume 7, Issue 4, January 1965, Pages 308-313, <https://doi.org/10.1093/comjnl/7.4.308>
- QIN, K.; HUANG, V.; SUGANTHAN, P.; (2009). Differential Evolution Algorithm With Strategy Adaptation for Global Numerical Optimization. **Evolutionary Computation, IEEE Transactions on**. 13. 398 - 417. 10.1109/TEVC.2008.927706.
- MIRANDA, V.; FONSECA, N.; EPSO-evolutionary particle swarm optimization, a new algorithm with applications in power systems; **IEEE/PES Transmission and Distribution Conference and Exhibition**, 2002, pp. 745-750 vol.2, doi: 10.1109/TDC.2002.1177567.

TOO, E.C.; LI Y.; SAM N.; LIU Y.; A Comparative Study of Fine-tuning Deep Learning Models for Plant Disease Identification. **Computers and Electronics in Agriculture** 161 (2019): 272-79.

VRBANCIC, G.; VILI P.; Transfer Learning With Adaptive Fine-Tuning. **IEEE Access** 8 (2020): 196197-96211.

ALLAWI Z.T.; IBRAHEEM I.K.; HUMAIDI A.J.; **Fine-Tuning Meta-Heuristic Algorithm for Global Optimization**. *Processes*. 2019; 7(10):657. <https://doi.org/10.3390/pr7100657>

JOSHI, S. K.; BANSAL, J. C.; Parameter tuning for meta-heuristics, **Knowledge-Based Systems**, Volume 189, 2020, 105094, ISSN 0950-7051, <https://doi.org/10.1016/j.knosys.2019.105094>.

TIEN BUI, D., ABDULLAHI, M.M., GHAREH, S. et al. Fine-tuning of neural computing using whale optimization algorithm for predicting compressive strength of concrete. **Engineering with Computers** 37, 701–712 (2021). <https://doi.org/10.1007/s00366-019-00850-w>

STÜTZLE T., LÓPEZ-IBÁÑEZ M. (2019) Automated Design of Metaheuristic Algorithms. In: Gendreau M., Potvin JY. (eds) Handbook of Metaheuristics. **International Series in Operations Research & Management Science**, vol 272. Springer, Cham. [https://doi.org/10.1007/978-3-319-91086-4\\_17](https://doi.org/10.1007/978-3-319-91086-4_17)

BEN ABDESSALEM, R.; NEJATI, S.; BRIAND, L.C.; STIFTER, T.; "Testing Vision-Based Control Systems Using Learnable Evolutionary Algorithms," **2018 IEEE/ACM 40th International Conference on Software Engineering (ICSE)**, 2018, pp. 1016-1026, doi: 10.1145/3180155.3180160.

CORAZZA, M.; DI TOLLO, G.; FASANO, G.; PESENTI, R; (2021). A novel hybrid PSO-based metaheuristic for costly portfolio selection problems. **Annals of Operations Research**, 304(1-2), 109–137. doi:10.1007/s10479-021-04075-3

NANNEN, V.; EIBEN, A. E.; "Efficient relevance estimation and value calibration of evolutionary algorithm parameters," **2007 IEEE Congress on Evolutionary Computation**, 2007, pp. 103-110, doi: 10.1109/CEC.2007.4424460.

WEERTS; HILDE J.P.; MUELLER, A.C.; VANSCHOREN, J.;. **Importance of tuning hyperparameters of machine learning algorithms**. 2007.07588 (2020).

FRIEDRICHS F.; IGEL C.; Evolutionary tuning of multiple SVM parameters, **Neurocomputing**, Volume 64, 2005, Pages 107-117, ISSN 0925-2312, <https://doi.org/10.1016/j.neucom.2004.11.022>.

WANG, Y.; HAOXIN, Z.; ZHANG, G.; (2019). cPSO-CNN: An efficient PSO-based algorithm for fine-tuning hyper-parameters of convolutional neural networks. **Swarm and Evolutionary Computation**. 49. 114-123. 10.1016/j.swevo.2019.06.002.

ABIDO, M.A.; Optimal power flow using particle swarm optimization, **International Journal of Electrical Power & Energy Systems**, Volume 24, Issue 7, 2002, Pages 563-571, ISSN 0142-0615, [https://doi.org/10.1016/S0142-0615\(01\)00067-9](https://doi.org/10.1016/S0142-0615(01)00067-9).

KOTUR, D.; STEFANOV, P.; Optimal power flow control in the system with offshore wind power plants. **Int. J. Electr. Power Energy Syst.** 2019, 105, 142–150

SAKIPOUR R.; ABDI H.; Optimizing Battery Energy Storage System Data in the Presence of Wind Power Plants: A Comparative Study on Evolutionary Algorithms. **Sustainability.** 2020; 12(24):10257. <https://doi.org/10.3390/su122410257>

SINISCALCHI-MINNA, S.; BIANCHI, F.; DE-PRADA-GIL, M.; OCAMPO-MARTINEZ, C. A wind farm control strategy for power reserve maximization. *Renew. Energy* 2019, 131, 37–44.

SEOK, H.; CHEN, C. An intelligent wind power plant coalition formation model achieving balanced market penetration growth and profit increase. *Renew. Energy* 2019, 138, 1134–1142.

JOSEPH S. et al., "PSO based controller algorithm for optimal allocation & setting of fuel cell in a wind — PV integrated power system for maximizing loadability," **2014 International Conference on Advances in Green Energy (ICAGE)**, 2014, pp. 1-6, doi: 10.1109/ICAGE.2014.7050135.

PHAM, H.V.; RUEDA, J.L.; ERLICH, I.; "Online Optimal Control of Reactive Sources in Wind Power Plants," in **IEEE Transactions on Sustainable Energy**, vol. 5, no. 2, pp. 608-616, April 2014, doi: 10.1109/TSTE.2013.2272586.

ERLICH, I.; LEE, K.; RUEDA, J.; WILDENHUES, S.; **Competition on Application of Modern Heuristic Optimization Algorithms for Solving Optimal Power Flow Problems**; Technical Report; WGMHO: Atlanta, GA, USA, 2014.

MIRANDA V.; ALVES, R.; "Differential Evolutionary Particle Swarm Optimization (DEEPSO): A Successful Hybrid," **2013 BRICS Congress on Computational Intelligence and 11th Brazilian Congress on Computational Intelligence**, 2013, pp. 368-374, doi: 10.1109/BRICS-CCI-CBIC.2013.68.

EFTIMOV, T.; PETELIN, G.; KOROŠEC, P.; DSCTool: A web-service-based framework for statistical comparison of stochastic optimization algorithms, **Applied Soft Computing**, Volume 87, 2020, 105977, ISSN 1568-4946, <https://doi.org/10.1016/j.asoc.2019.105977>.

HOUGHTON, J.T.; JENKINS, G.J.; EPHRAUMS, J.J.; **Climate change**. United Kingdom: N. p., 1990. Web.

NATIONS, U.; **Paris Agreement**. Available online: <https://unfccc.int/process-and-meetings/the-paris-agreement/the-parisagreement> (accessado em 14 de dezembro de 2021)