

Desenvolvimento de plataforma WEB para planejamento de rotas na manutenção de parques eólicos offshore utilizando algoritmo da otimização da colônia de formigas: Estudo de Caso

Klebiano Kennedy da Silva Lima ^[1] e Matheus da Silva Menezes ^[2]

^[1] UFRSA - Universidade Federal Rural do Semi-Árido; klebyano@gmail.com

^[2] UFRSA - Universidade Federal Rural do Semi-Árido; matheus@ufersa.edu.br

Resumo: A energia eólica *offshore*, onde os aerogeradores são implementados no mar, ganharam grande interesse no Brasil e no mundo devido a busca por energia de fontes limpas, renováveis e com baixo impacto ambiental e social. Na construção e desenvolvimento de parques eólicos *offshore*, foram realizados diversos aprimoramentos para diminuir custos de implantação, mas a operação e manutenção (O&M) desses ativos ainda apresenta-se como um grande desafio para a implementação de projetos no Brasil e no exterior. Uma das formas mais eficazes de otimizar e, conseqüentemente, diminuir os custos de O&M é no agendamento e roteamento de atividades de manutenção, aprimorando o uso de recursos materiais e de pessoal qualificado. Este artigo apresenta a implementação de algoritmo de otimização da colônia de formigas aplicado para o roteamento de atividades de manutenção em aerogeradores eólicos em um complexo *offshore*, considerando características como trajeto e tempo estimado de manutenção. Para a visualização dos trajetos otimizados e seleção de parâmetros para execução da otimização, foi desenvolvida plataforma *web* e APIs (*application programming interfaces*, do inglês Interfaces de Programação de Aplicação) para comparação de resultados e tempo de execução.

Palavras-chave: energia eólica *offshore*; operação; manutenção; roteamento, algoritmo de colônia de formigas.

Abstract: Offshore wind energy, where wind turbines are deployed at sea, has gained significant interest in Brazil and around the world due to the pursuit of clean, renewable energy sources with low environmental and social impact. In the construction and development of offshore wind farms, various enhancements have been made to reduce implementation costs, but the operation and maintenance (O&M) of these assets still pose some of the greatest challenges for project implementation in Brazil and abroad. One of the most effective ways to optimize and consequently reduce O&M costs is in the scheduling and routing of maintenance activities, improving the use of material resources and qualified personnel. This article presents the implementation of an ant colony optimization algorithm applied for routing maintenance activities on offshore wind turbines in a offshore cluster, considering factors such as route and estimated maintenance time. For visualizing optimized routes and selecting parameters for optimization execution, a web platform and APIs (Application Programming Interfaces) were developed for result comparison and execution time assessment.

Key-words: offshore wind energy, operation, maintenance, routing, ant colony algorithm.

1. INTRODUÇÃO

No Brasil, a energia eólica *offshore* representa uma promissora fonte de energia renovável e com um potencial imenso a ser explorado. As vastas extensões do litoral brasileiro oferecem condições ideais para a implantação de projetos eólicos, sendo estimado uma geração total de, aproximadamente, 3TW e mais de 14.000 TWh de energia produzida anualmente, o que equivale a 20 vezes a capacidade total instalada do Brasil em 2020 [1]. Em 2023, mais de 75 projetos eólicos *offshore* aguardavam aprovação de licenciamento ambiental federal na costa brasileira pelo IBAMA (Instituto Brasileiro do Meio Ambiente e dos Recursos Naturais Renováveis), totalizando mais de 180 GW de geração esperada [2]. A construção e, posteriormente, a operação e manutenção desses ativos acarretam desafios cruciais para assegurar a sustentabilidade da geração e a disponibilidade energética.

A operação e manutenção (O&M) de parques eólicos apresentam parte significativa nos custos de energia gerada, o que pode inviabilizar a implantação de um determinado projeto. Em energia eólica *offshore*, esses custos são ainda maiores que os apresentados na eólica *onshore*, onde os parques eólicos são implementados em terra, representando cerca de 23% do custo total de investimento, principalmente devido as condições

ambientais de alta salinidade em que os aerogeradores estão instalados [3]. Para comparação, o O&M dos complexos eólicos *onshore* apresentam apenas 5% do investimento total, impactando menos no LCOE (*levelized cost of energy*, do inglês custo nivelado de energia). Outros parâmetros que aumentam os custos de implantação da energia *offshore* relacionados a manutenção são o uso de equipamento e pessoal altamente especializados como frotas de embarcações e técnicos treinados. Parâmetros como aumento da potência nominal do aerogerador também influenciam na maior necessidade de manutenção corretiva em diversos subsistemas dos aerogeradores, principalmente no sistema elétrico e eletrônico [4]. As condições climáticas em ambiente marítimo também adicionam mais desafios para a realização de atividades de manutenção, aumentando o *downtime* (do inglês, tempo de parada) do aerogerador e maior perda energética.

Múltiplos métodos foram pesquisados e testados em diferentes cenários considerando parâmetros como distância de trajeto, custo de peça de reposição, custo de combustível para tentar apresentar o melhor modelo de agendamento e roteamento de manutenção em parques eólicos *offshore* [5].

O problema de definir uma programação de manutenção e roteamento de aerogeradores para otimizar o atendimento com maior velocidade e menor tempo possui semelhança com um problema bastante conhecido de otimização, conhecido como o Problema do Caixeiro Viajante (PCV), onde o caixeiro deve seguir um determinado caminho visitando cada cidade uma única vez, com o menor custo ou distância possível para percorrer todas as cidades. No caso de um complexo eólico *offshore*, pode ser considerado apenas os aerogeradores em falha e com necessidade de manutenção imediata como conjunto a ser usado na execução da otimização.

Muitos algoritmos foram desenvolvidos e aplicados para encontrar a melhor solução para problemas de obtenção de rotas e agendamento na área da otimização combinatória, que é a disciplina matemática que se concentra na seleção das combinações ideais de elementos a partir de um conjunto finito. Alguns exemplos de algoritmos utilizados na solução desse tipo de problema são os algoritmos genéticos (AG), algoritmos meméticos, redes neurais artificiais e a otimização por colônia de formigas (OCF), onde para a otimização de rota de manutenção em parque eólico *offshore* podem ser considerados objetivos complexos ou simples, como a distância percorrida pela embarcação, custos e capacidade de pessoal e material, restrições sazonais e custos de disponibilidade de embarcações. A seleção das variáveis utilizadas na otimização depende da quantidade de dados disponíveis e necessidade de computação [6]. Neste trabalho, o algoritmo de otimização por colônia de formigas (OCF) foi implementado na otimização do roteamento e planejamento de manutenção em um complexo eólico na costa do nordeste brasileiro considerando dados de distância e tempo de manutenção como entrada. Para a visualização e testes realizados, foi implementada uma página Web interativa e API para consulta e interação do usuário com os dados. O objetivo desse trabalho é testar e validar o método proposto de otimização e roteamento, considerando os ganhos operacionais e tempo de resposta para a aplicação, assim como a visualização e fácil detecção de problemas por parte do usuário.

2. MATERIAIS E MÉTODOS

2.1. Otimização por colônia de formigas

A otimização por colônia de formigas é um método desenvolvido nos anos 1990 [7] pelo cientista da computação italiano Dr. Marco Dorigo que tenta replicar, de forma simplificada, um padrão da natureza ocorrido em formigueiros, onde uma colônia de formigas tende a encontrar a melhor rota até o alimento por meio da liberação de uma substância conhecida como feromônio. Com o aumento do feromônio em um determinado caminho ou circuito, as formigas seguintes seguem um caminho otimizado e os caminhos com menor passagem de indivíduos tendem a ser descartados devido a evaporação do feromônio liberado. O PCV foi o primeiro problema a ser aplicado o OCF, pois apresenta uma facilidade de aplicação da metáfora da colônia de formigas por ser um dos problemas mais estudados na área de otimização combinatória e ser NP-difícil, o que indica que para um número elevado de cidades a solução tem dificuldade para atingir um tempo computacional viável.

O PCV é um problema para encontrar a menor distância Hamiltoniana, onde cada vértice é visitado apenas uma vez no caminho fechado, em um grafo $G = (C, L)$, onde C é um conjunto de cidades e L é o conjunto de conexões entre os elementos de C . Segundo [7], a seleção da próxima cidade, do vértice i ao j , a ser visitada pode ser definida pela probabilidade dada pela equação (1).

$$p_{ij}^k = \frac{[T_{ij}^t]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} [T_{il}^t]^\alpha [\eta_{il}]^\beta} \quad (1)$$

Onde T_{ij}^t é a quantidade de feromônio presente na aresta entre os nós das cidades no tempo t , α e β são parâmetros que controlam o peso relativo entre o feromônio e a influência da rota na seleção, N_i^k é o conjunto

de nós não visitados, e η_{ij} é a variável com relação ao peso entre os nós, que pode ser definido como $\eta_{ij} = 1/J_{ci,cj}$, onde $J_{ci,cj}$ é a distância entre os nós i e j .

A taxa de atualização do feromônio pode ser definida pela equação (2):

$$T_{ij}^t = (1 - \rho)T_{ij}^t + \sum_{k=1}^m \Delta T_{ij}^k \quad (2)$$

Onde ρ é a taxa de evaporação do feromônio, e ΔT_{ij}^k é o valor depositado e atualizado de feromônio em uma determinada rota definido na equação (3) e m é o número de formigas utilizadas em cada iteração.

$$\Delta T_{ij}^k = \frac{Q}{d_k} \quad (3)$$

Onde Q é a constante de atualização do feromônio, e d_k é a distância total percorrida pela formiga k no intervalo i e j . O pseudocódigo demonstrando a implementação do algoritmo aplicado ao caso eólico *offshore* pode ser visualizado na Figura 1.

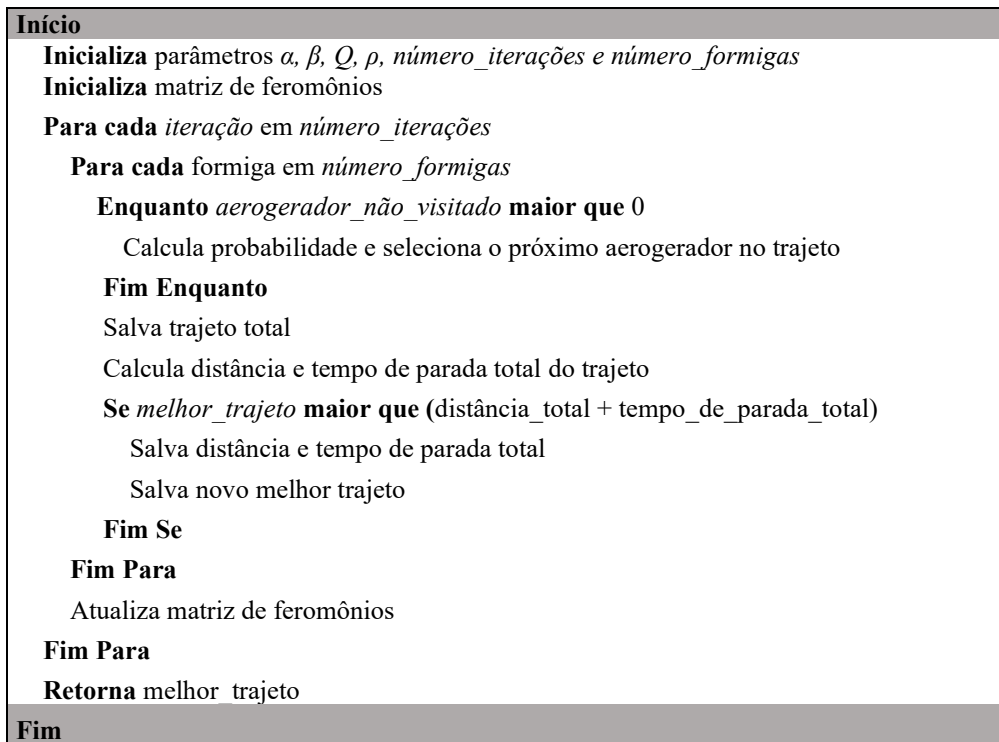


FIGURA 1. Pseudocódigo de implementação do algoritmo de colônia de formigas (autoria própria).

2.2. Roteamento e planejamento da manutenção

Neste trabalho, foi utilizado o algoritmo de otimização por colônia de formigas definido nas equações anteriores para o roteamento e planejamento de manutenção em um complexo eólico com um total de 212 aerogeradores localizado no mar próximo a cidade de Areia Branca no Rio Grande do Norte e desenvolvido pela empresa Beta Energia. O complexo eólico foi selecionado a partir dos projetos em espera de aprovação ambiental do IBAMA em 2023 [2] e quando concluído, possuirá uma capacidade de geração de, aproximadamente, 3GW. O ponto de partida da embarcação para a manutenção dos aerogeradores foi selecionado na costa da cidade com maior proximidade aos ativos. Na Figura 2 é possível visualizar a posição dos aerogeradores e ponto de partida das embarcações (Doca) em mapa.

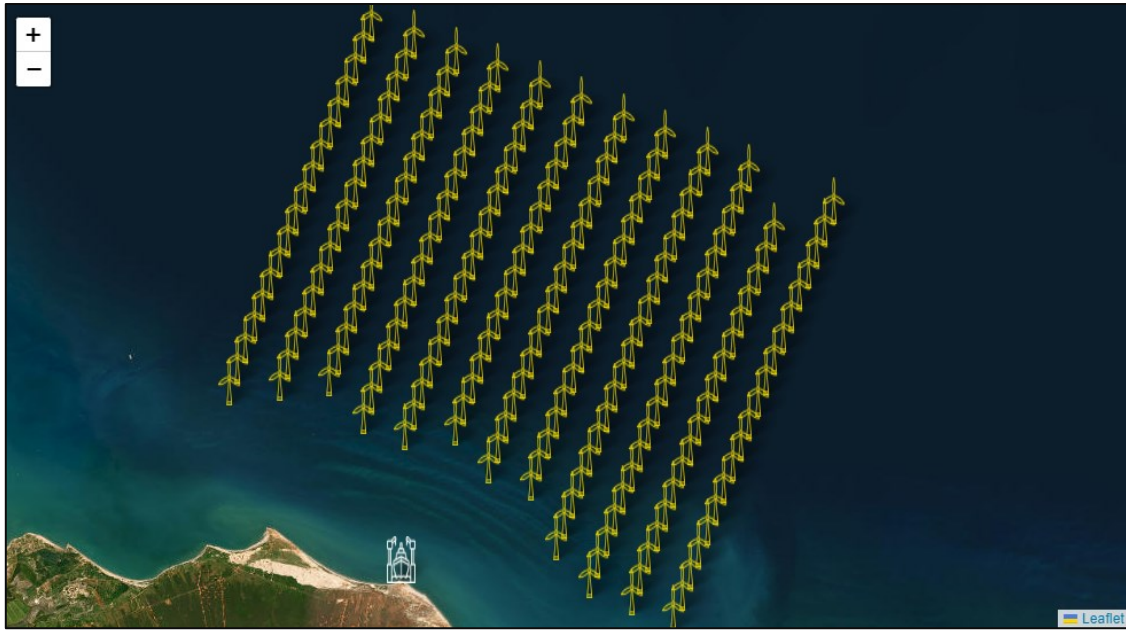


FIGURA 2. Posicionamento do complexo eólico e ponto de partida das embarcações, imagem construída com software de código aberto *Leaflet* versão 4.2.1 em biblioteca *React* versão 18.2.0 (autoria própria).

Segundo [8], a otimização da manutenção pode ser definida como um método para determinar o plano de manutenção mais eficaz e eficiente, que tem como objetivo atingir o melhor equilíbrio possível entre os custos da manutenção e as consequências se essa manutenção não for realizada. Atualmente, a manutenção em parques eólicos adota a combinação entre a manutenção corretiva ou reativa, onde a ação é tomada após a ocorrência da falha no equipamento e a manutenção preventiva, onde a ação de manutenção ou inspeção é realizada em intervalos predeterminados, em complementariedade a manutenção de oportunidade também é aplicada em diversos cenários, onde as tarefas corretivas e preventivas são realizadas em paralelo para reduzir visitas de equipes, possíveis perdas de produção e despesas de trajeto [9].

Para o planejamento e roteamento modelado neste trabalho, duas variáveis foram utilizadas para a minimização da função objetivo, distância percorrida pela embarcação e tempo de manutenção aproximado para cada atividade. Como forma de comparação entre modelos, foi utilizado o *solver* GLPK (*GNU linear programming kit*) na versão 5.0.0 para a obtenção de solução exata com método *dual simplex*. Como o problema a ser resolvido é NP-difícil, não existe garantia de solução pelos métodos exatos, para isso o tempo de processamento máximo definido no *software* GLPK foi de 10.000 segundos. O modelo implementado no GLPK pode ser visualizado na equação (4), e as restrições podem ser visualizadas nas equações 5, 6, 7 e 8, respectivamente.

$$\min \left(\sum_{j=1}^m \sum_{i=1}^m c_{ij} x_{ij} + \sum_{j=1}^m \sum_{i=1}^m t_{ij} x_{ij} \right) \quad (4)$$

$$\sum_{j=1}^m x_{ij} = 1 \quad \forall i = 1, \dots, m \quad (5)$$

$$\sum_{i=1}^m x_{ij} = 1 \quad \forall i = 1, \dots, m \quad (6)$$

$$\sum_{i \in k} \sum_{j \in k} x_{ij} \leq |K| - 1 \quad \forall i, j \quad (7)$$

$$x_{ij} = \{0,1\} \quad \forall i, j \quad (8)$$

Onde c_{ij} é a distância entre os nós, t_{ij} é o tempo de parada estimado entre os nós e a equação (7) é a restrição de sub-rotas.

De acordo com a WMEP (*Scientific Measurement and Evaluation Programme*, do inglês Programa de medição e avaliação científica) [4], em uma avaliação realizada por diversos agentes da indústria eólica entre 1989 e 2006, foi possível obter os parâmetros de confiabilidade MTTR (*mean time to repair*, do inglês tempo médio para reparo) e MTBF (*mean time between failure*, tempo médio entre falhas) para uma amostra de, aproximadamente, 1500 aerogeradores em cenários distintos. Na Tabela 1 é possível visualizar os valores de

Taxa de falha anual e Tempo de parada em dias para cada subsistema do aerogerador e para cada severidade de falha.

TABELA 1. Características de falhas em diferentes subsistemas [4].

<i>Subsistema</i>	<i>Severidade</i>	<i>Taxa de falha anual</i>	<i>Tempo de falha (dias)</i>
Sistema elétrico	Leve	0,45	0,17
Sistema elétrico	Grave	0,12	6,55
Controlador eletrônico	Leve	0,34	0,15
Controlador eletrônico	Grave	0,09	6,87
Sensores	Leve	0,2	0,16
Sensores	Grave	0,05	6,41
Sistema hidráulico	Leve	0,18	0,18
Sistema hidráulico	Grave	0,05	5,93
Sistema de orientação	Leve	0,13	0,16
Sistema de orientação	Grave	0,05	10,09
Cubo do rotor	Leve	0,12	0,18
Cubo do rotor	Grave	0,06	10,93
Freio mecânico	Leve	0,11	0,16
Freio mecânico	Grave	0,03	13,08
Pás eólicas	Leve	0,09	0,18
Pás eólicas	Grave	0,02	11,86
Multiplificadora	Leve	0,06	0,17
Multiplificadora	Grave	0,03	18,38
Gerador	Leve	0,07	0,15
Gerador	Grave	0,05	14,34
Suportes e invólucro	Leve	0,08	0,14
Suportes e invólucro	Grave	0,02	28,01
Sistema de transmissão mecânica	Leve	0,03	0,17
Sistema de transmissão mecânica	Grave	0,02	15,47

A obtenção dos dados de entrada e visualização dos dados de saída para o modelo *OCF* foi realizada usando plataforma *Web* desenvolvida com a biblioteca *React* 18.2.0 em linguagem de programação *Javascript*, o banco de dados selecionado para arquivar as tabelas de posicionamento geográfico, subsistema e tempo de parada foi o *PostgreSQL* versão 15.4.0, o acesso aos dados foi realizado empregando requisições *HTTP* (*HyperText Transfer Protocol*, do inglês Protocolo de Transferência de Hipertexto) para uma *API* desenvolvida em linguagem de programação *Python* versão 3.10.9 e biblioteca *FastAPI* versão 0.103.1, a computação dos parâmetros e do modelo *OCF* foi concebida em *Python* e para tratamento de dados de grande volume, a biblioteca *Pandas* versão 1.5.0 foi empregada. Na Figura 3 pode ser visualizada a página *Web* contendo os campos de entrada como aerogerador, subsistema e tipo de falha para a computação do algoritmo *OCF*. A Figura A1 evidencia a página completa desenvolvida.

Ant Colony

Turbine

BETA-01

▼

Subsystem

Electrical System

▼

Fault Type

Minor

▼

+

 ADD TO TABLE

N° of turbines

1

✕

 RANDOMIZE

≡

 CLEAR TABLE

TURBINE	SUBSYSTEM	FAULT TYPE	
BETA-19	Generator	Minor	

1-1 of 1

<

>

▶

 RUN ANTS

FIGURA 3. Plataforma Web desenvolvida para obtenção das entradas do algoritmo *OCF* (autoria própria).

O problema propõe a solução da melhor rota e planejamento minimizando distância e tempo de parada em máquina. Por exemplo, um aerogerador com severidade leve no sistema elétrico tem maior prioridade do que um aerogerador com severidade grave na multiplicadora, mesmo se a distância dessa for maior do que a primeira. Para este problema, foi considerado peso unitário para as variáveis minimizadas, distância e tempo de parada. Para o problema em questão serão realizadas 5 rotas distintas para a simulação, com 40, 20, 15, 10 e 5 aerogeradores cada selecionados de forma aleatória e com pelo menos um desses aerogeradores contendo uma falha com severidade do tipo grave. Os problemas exemplo utilizados nesse trabalho podem ser conferidos na Tabela 2.

TABELA 2. Problemas exemplos utilizados na comparação do algoritmo *OCF* (autoria própria).

<i>Problema</i>	<i>Número de aerogeradores</i>	<i>Aerogeradores</i>	<i>Aerogeradores com severidade do tipo grave</i>	<i>Subsistemas com severidade do tipo grave</i>
1	40	01, 05, 09, 18, 20, 25, 30, 33, 40, 45, 52, 55, 62, 65, 69, 74, 82, 84, 85, 89, 118, 120, 123, 126, 135, 137, 138, 141, 145, 149, 153, 158, 161, 172, 174, 182, 184, 198, 205, 212	05 18	Multiplicadora Sistema de transmissão mecânica
2	20	06, 22, 25, 39, 47, 48, 65, 68, 80, 107, 114, 121, 122, 124, 144, 167, 170, 172, 178, 183	47	Controlador eletrônico
3	15	12, 45, 56, 81, 109, 110, 118.	45	Controlador eletrônico

		131, 133, 136, 137, 174, 182, 188, 211	118	Multiplicadora
4	10	11, 15, 30, 73, 93, 108, 120, 129, 152, 201	11	Freio mecânico
5	5	36, 58, 120, 156, 201	120	Sistema hidráulico

3. RESULTADOS

Nessa seção, serão apresentados os resultados obtidos através dos experimentos realizados e resultados da seleção dos melhores parâmetros e resolução do roteamento dos problemas já apresentados. Os parâmetros utilizados na computação do algoritmo *OCF*, como número de formigas (m), α , β , ρ e seus valores são bastante estudados e apresentam diferentes objetivos para problemas de otimização distintos. Para este trabalho, os parâmetros de valor da matriz de feromônios inicial, constante de atualização do feromônio (Q) e número de iterações, foram considerados fixos em 1, 100 e 100, respectivamente. Para a seleção dos parâmetros m , α , β , ρ foi implementado o algoritmo de *grid search* com faixas e intervalos definidos em [10], para os conjuntos de parâmetros foi definida 20 iterações para cada problema exemplo da Tabela 2, os melhores parâmetros com menor tempo de execução e convergência foram selecionados, no total foram executadas 27.000 tarefas para a seleção dos parâmetros finais.

Todos os exemplos implementados e calculados neste trabalho foram executados em um notebook Acer com processador Intel® Core™ i5 de 10ª geração, 20 GB de memória RAM e SSD de 1TB. O sistema operacional é o *Windows 11 Home Single Language*, versão 22H2 e o ambiente da linguagem de programação *Python 3.10.9*. Os códigos de computação não foram executados em paralelo, onde foi empregado apenas um núcleo para cada execução e funcionalidades assíncronas foram implementadas apenas em chamadas *HTTP* para o servidor em execução local. Os valores e intervalos utilizadas no algoritmo de *grid search* para cada parâmetro testado, podem ser conferidos na Tabela 3.

TABELA 3. Valores de parâmetros para o modelo *OCF* utilizado no algoritmo de *grid search* (autoria própria).

Parâmetro	Intervalo	Passo
m	1 a 10	2
α	1 a 10	2
β	0 a 2	1
ρ	0 a 1	0,5

Na Tabela 4, são demonstrados os três melhores parâmetros encontrados para cada problema analisado, assim como o tempo médio obtido de cada execução.

TABELA 4. Seleção dos três melhores parâmetros encontrados para cada um dos problemas definidos (autoria própria).

Problema	m	α	β	ρ	Heurística da Distância percorrida e tempo de parada	Tempo médio de execução (segundos)
1	5	10	2	0	9,6487	23,7565
	3	7	2	1	9,8567	12,5035
	10	10	2	1	9,9106	30,0329
2	10	5	2	1	5,0169	10,8672
	5	1	2	0,5	5,0220	8,0525
	5	7	2	0	5,0323	4,3825
3	3	7	2	1	5,2777	2,7958
	5	5	2	0	5,3164	2,6579
	7	10	2	1	5,3164	5,0452
4	1	1	2	0	4,6829	0,2830
	1	1	2	0,5	4,6829	0,4434
	1	3	2	1	4,6829	0,4593

5	1	3	0	0	4,6051	0,0785
	1	1	2	0,5	4,6051	0,0794
	1	3	2	0,5	4,6051	0,0796

Os parâmetros selecionados para aplicações generalizadas em todos os problemas de entrada solicitados pelo usuário final da plataforma *Web* foram $m = 3$, $\alpha = 5$, $\beta = 1,5$ e $\rho = 0,5$ para problemas menores ou iguais a 10 aerogeradores e $m = 8$, $\alpha = 5$, $\beta = 2$ e $\rho = 0,5$ para problemas maiores do que 11 aerogeradores. O controle adaptativo dos parâmetros [11] foi implementado para aumentar a experiência do usuário em relação ao tempo de espera da solução, esses parâmetros foram obtidos por meio da aplicação da média entre as melhores execuções em relação à heurística (distância percorrida e tempo de parada) e menor tempo de execução.

Na Tabela 5, é possível visualizar os resultados encontrados para os problemas da Tabela 2 e considerando os parâmetros gerais encontrados na Tabela 4 com o algoritmo *OCF* e o *solver* GLPK, assim como o tempo de execução para ambos os casos e a diferença percentual entre as soluções encontradas. Para cada problema foram executadas 200 iterações e realizado a média para a obtenção do valor demonstrado.

TABELA 5. Melhores soluções encontradas para o algoritmo *OCF* e nos problemas definidos (autoria própria).

Problema	Número de aerogeradores	Heurística da Distância percorrida e tempo de parada (<i>OCF</i>)	Tempo de execução <i>OCF</i> (segundos)	Heurística da Distância percorrida e tempo de parada (<i>GLPK</i>)	Tempo de execução <i>GLPK</i> (segundos)	Diferença entre soluções (%)
1	40	9,9479	81,9870	-	-	-
2	20	4,9081	18,5590	4,5845	7,1	7,05
3	15	5,2777	6,3867	5,2777	3,7	0
4	10	4,6829	1,9528	4,6829	0,5	0
5	5	4,6051	0,5042	4,6051	0,01	0

Na Tabela 5, é possível visualizar que o algoritmo *OCF* encontrou soluções aproximadas em quase todos os problemas propostos, com o tempo de execução superior em comparação com o *solver* GLPK. No problema 1 com 40 aerogeradores, o GLPK não alcançou a solução exata após 10.000 segundos de execução e por esse motivo não foi evidenciado na tabela, demonstrando que para a solução exata, o GLPK apresenta limitações para a quantidade de dados de entrada computados.

A Tabelas 6 apresenta os melhores resultados obtidos pelo algoritmo *OCF* para as 20 melhores soluções computadas para cada problema proposto. Para cada coluna, estão apresentados os dados da melhor solução encontrada para a heurística da distância percorrida e tempo de parada, a quantidade de vezes que esta solução foi encontrada (Qtd.Vezes), a média das melhores soluções encontradas (Méd. Solução), o tempo médio gasto pelo algoritmo em cada rodada (Méd. Tempo) e o desvio padrão entre as soluções (σ).

TABELA 6. Resultados, médias e desvios encontrados para as 20 melhores soluções nos problemas propostos (autoria própria).

Problema	Número de aerogeradores	Heurística da Distância percorrida e tempo de parada	Qtd.Vezes	Méd. Solução	Méd. Tempo (segundos)	σ
1	40	9,9479	1	10,2826	58,1464	0,2024
2	20	4,9081	1	5,2160	15,8830	0,1140
3	15	5,2777	2	5,3885	11,0724	0,0579
4	10	4,6829	10	4,6870	2,0984	0,0041
5	5	4,6051	20	4,6051	0,6247	0

Na Tabela 6 é possível notar que a solução mínima da Tabela 5 não é encontrada em todas as iterações mesmo utilizando parâmetros de entrada iguais, isso é devido à randomização ocorrida na seleção de cidades e, consequentemente, na probabilidade utilizada em cada uma dessas escolhas. Também é possível verificar que para um número superior de dados de entrada (número de aerogeradores), o algoritmo *OCF* tem menor eficiência e maior divergência entre soluções encontradas.

Na Figura 4, pode ser visualizada os resultados encontrados para o Problema 1 com 40 aerogeradores e a melhor rota selecionada considerando a minimização da distância e tempo de parada. Os problemas 2, 3, 4 e 5 podem ser visualizadas nas Figuras A2, A3, A4 e A5, respectivamente.



FIGURA 3. Exemplo de melhor trajeto encontrado para Problema 1 com 40 aerogeradores com algoritmo *OCF* (autoria própria).

4. CONCLUSÃO

A energia eólica *offshore* apresenta um grande potencial no Brasil para a década de 2030, contando com diversas propostas de implementação de complexos eólicos em grande parte da costa brasileira, esses projetos futuros necessitarão de amplos investimentos para sua implementação e para o mantimento de uma operação com baixos custos e, consequentemente, custos mais baixos de energia, fazendo com que esses projetos se tornem duradouros. Com uma grande parte dos custos sendo de O&M, a maior eficiência no planejamento e roteamento de atividades nesses complexos se torna fundamental para a sua viabilidade.

Neste artigo, o algoritmo de otimização por colônia de formigas (*OCF*) foi aplicado para otimizar e encontrar o melhor trajeto em um complexo eólico de larga escala, contendo no total 212 aerogeradores na costa do estado do Rio Grande do Norte. As variáveis para a minimização selecionadas foram a distância e tempo de parada em cada aerogerador, para isso foram modelados 5 problemas com uma quantidade de aerogeradores distintos e com diferentes tipos de falhas e severidades. Os resultados obtidos para cada problema definido, bem como os tempos de execução para cada cenário, foram comparados com o *solver* GLPK para validação do modelo e para confirmar os benefícios da otimização. Todos os trajetos selecionados e testados foram observados com uma plataforma *Web* desenvolvida para utilização, visualização e otimização do tempo de possível verificação para cada usuário do sistema.

Para melhorias futuras, podem ser realizadas avanços no processamento do algoritmo *OCF* adicionando o paralelismo em alguns processos e a utilização de linguagens de programação de alto desempenho como C++, Golang e Rust em tarefas principais. Para trabalhos futuros, a adição de mais variáveis para a minimização e processamento do *OCF* podem melhorar a aplicação da plataforma em situações mais próximas da realidade do O&M de múltiplos complexos eólicos em paralelo, variáveis como número de embarcações, pessoal e material disponível, condições climáticas favoráveis, horas trabalhadas possíveis e custos operacionais podem ser utilizados para uma melhor modelagem do sistema. Sendo assim, esse trabalho conseguiu atingir os objetivos de testar e validar o *OCF* na otimização e roteamento, considerando os ganhos operacionais e tempo de resposta para a aplicação, assim como a visualização e detecção de problemas por meio do usuário ou equipe na plataforma *Web* desenvolvida.

AGRADECIMENTOS

Agradeço a Universidade Federal Rural do Semi-Arido (UFERSA) pelo apoio e ao Prof. Dr. Matheus da Silva Menezes pelo suporte na modelagem desse trabalho.

REFERÊNCIAS

- [1] AZEVEDO, Sylvester Stallone Pereira de et al. Assessment of Offshore Wind Power Potential along the Brazilian Coast. *Energies*, v. 13, n. 10, p. 2557, 2020.
- [2] Mapas de projetos em licenciamento - Complexos Eólicos Offshore. Disponível online: <https://www.gov.br/ibama/pt-br/assuntos/laf/consultas/mapas-de-projetos-em-licenciamento-complexos-eolicos-offshore> (acesso em 20/08/2023).
- [3] REN, Zhengru et al. Offshore wind turbine operations and maintenance: A state-of-the-art review. *Renewable and Sustainable Energy Reviews*, v. 144, p. 110886, 2021.
- [4] FAULSTICH, Stefan; HAHN, Berthold; TAVNER, Peter J. Wind turbine downtime and its importance for offshore deployment. *Wind energy*, v. 14, n. 3, p. 327-337, 2011.
- [5] ZHANG, Zhen You. Scheduling and routing optimization of maintenance fleet for offshore wind farms using Duo-ACO. *Advanced materials research*, v. 1039, p. 294-301, 2014.
- [6] STOCK-WILLIAMS, Clym; SWAMY, Siddharth Krishna. Automated daily maintenance planning for offshore wind farms. *Renewable Energy*, v. 133, p. 1393-1403, 2019.
- [7] DORIGO, Marco; DI CARO, Gianni. Ant colony optimization: a new meta-heuristic. In: *Proceedings of the 1999 congress on evolutionary computation-CEC99* (Cat. No. 99TH8406). IEEE, 1999. p. 1470-1477.
- [8] SHAFIEE, Mahmood; SØRENSEN, John Dalsgaard. Maintenance optimization and inspection planning of wind energy assets: Models, methods and strategies. *Reliability Engineering & System Safety*, v. 192, p. 105993, 2019.
- [9] BAKIR, Ilke; YILDIRIM, Murat; URSAVAS, Evrim. An integrated optimization framework for multi-component predictive analytics in wind farm operations & maintenance. *Renewable and Sustainable Energy Reviews*, v. 138, p. 110639, 2021.
- [10] WONG, Kuan Yew et al. Parameter tuning for ant colony optimization: A review. In: *2008 International Conference on Computer and Communication Engineering*. IEEE, 2008. p. 542-545.
- [11] HAO, Zhi-Feng; CAI, Rui-Chu; HUANG, Han. An adaptive parameter control strategy for ACO. In: *2006 International Conference on Machine Learning and Cybernetics*. IEEE, 2006. p. 203-206.

APÊNDICE A1

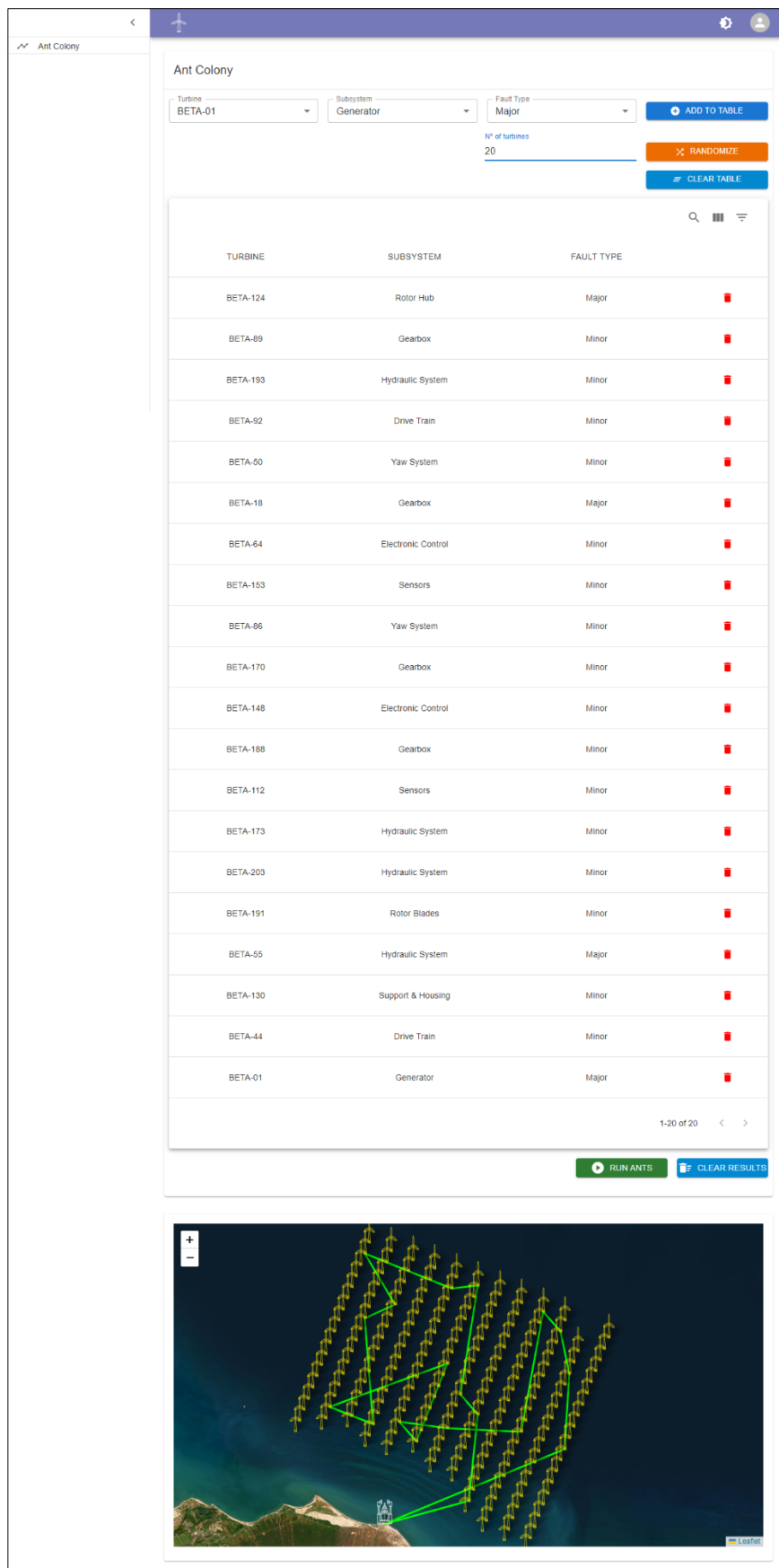


FIGURA A1. Plataforma Web desenvolvida com exemplo de rota calculada com algoritmo OCF (autoria própria).

APÊNDICE A2

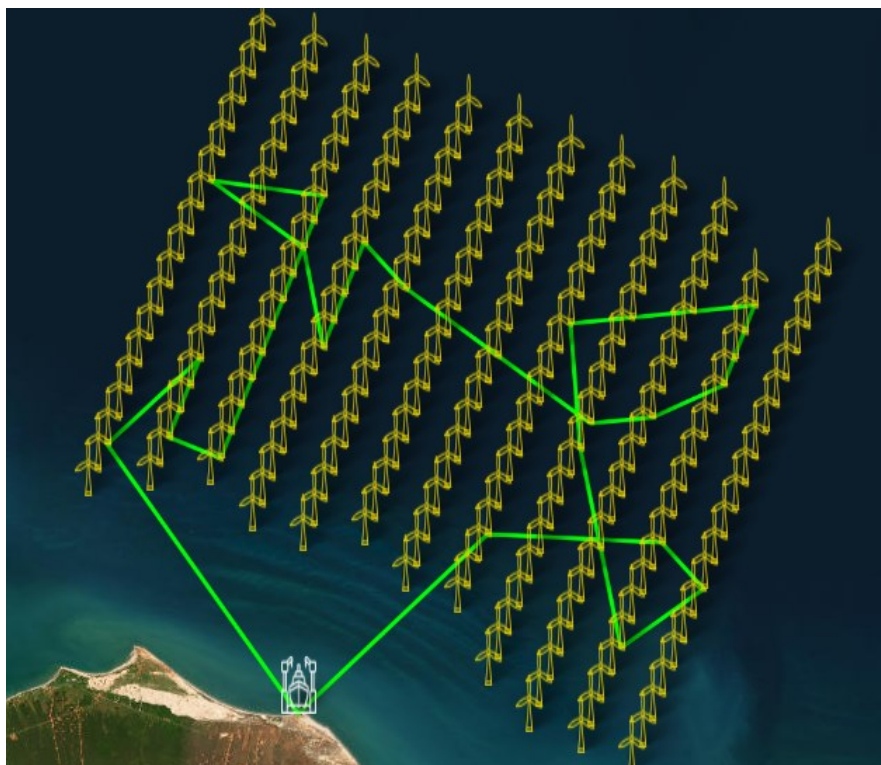


FIGURA A2. Exemplo de melhor trajeto encontrado para Problema 2 com 20 aerogeradores com algoritmo *OCF* (autoria própria).

APÊNDICE A3



FIGURA A3. Exemplo de melhor trajeto encontrado para Problema 3 com 15 aerogeradores com algoritmo *OCF* (autoria própria).

APÊNDICE A4

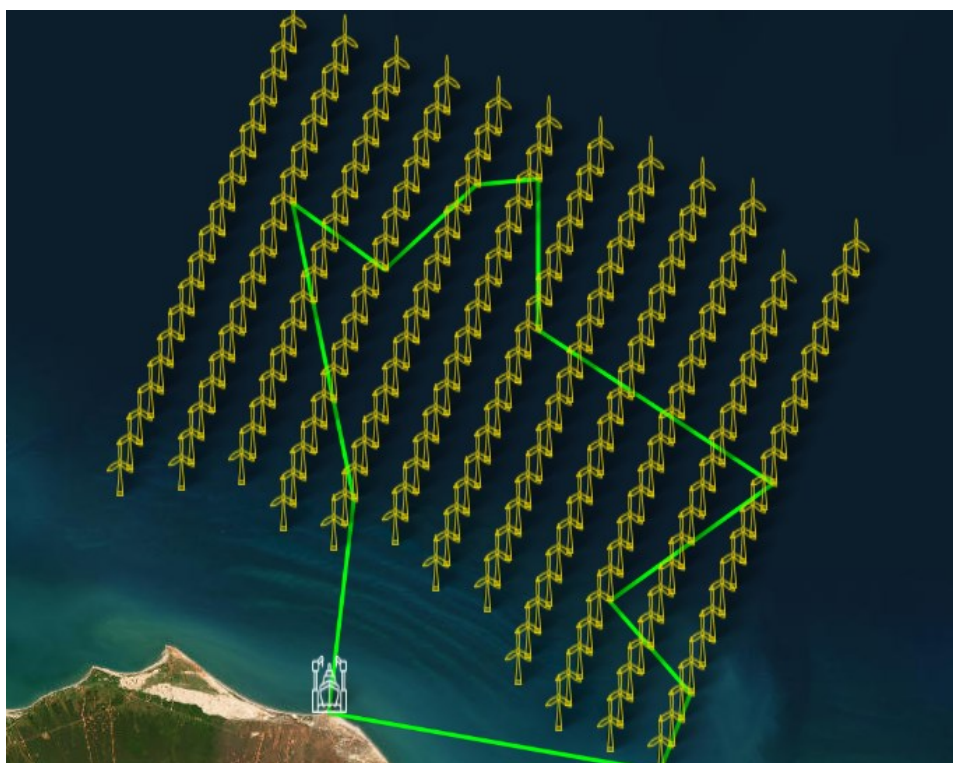


FIGURA A4. Exemplo de melhor trajeto encontrado para Problema 4 com 10 aerogeradores com algoritmo *OCF* (autoria própria).

APÊNDICE A5

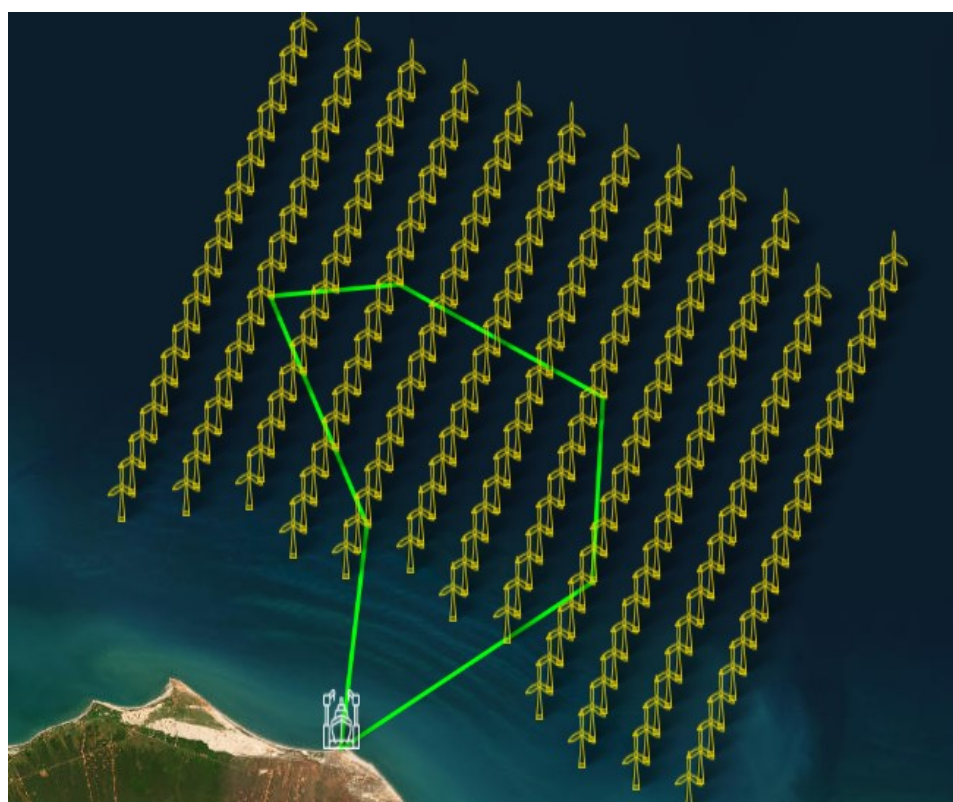


FIGURA A5. Exemplo de melhor trajeto encontrado para Problema 5 com 5 aerogeradores com algoritmo *OCF* (autoria própria).