

PAULO OLIVEIRA SIQUEIRA JUNIOR

**MODELO HÍBRIDO COM REDES NEURAIAS ARTIFICIAIS E ALGORITMOS
EVOLUCIONÁRIOS PARA OTIMIZAÇÃO DO CONSUMO DE COMBUSTÍVEL
EM EMBARCAÇÕES QUE UTILIZAM MOTOR DE COMBUSTÃO INTERNA A
DIESEL**

**MANAUS – AM
2021**

PAULO OLIVEIRA SIQUEIRA JUNIOR

**MODELO HÍBRIDO COM REDES NEURAS ARTIFICIAIS E ALGORITMOS
EVOLUCIONÁRIOS PARA OTIMIZAÇÃO DO CONSUMO DE COMBUSTÍVEL
EM EMBARCAÇÕES QUE UTILIZAM MOTOR DE COMBUSTÃO INTERNA A
DIESEL**

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia, Gestão de Processos, Sistemas e Ambiental - PPG.EGPSA, do Instituto de Tecnologia e Educação Galileo da Amazônia- ITEGAM, como parte dos requisitos necessários à obtenção do título de Mestre.

Orientador: Dr. Manoel Henrique Reis Nascimento

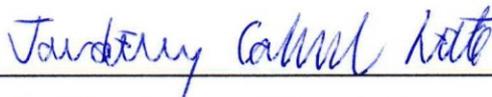
**MANAUS – AM
2021**

PAULO OLIVEIRA SIQUEIRA JUNIOR

**MODELO HÍBRIDO COM REDES NEURAS ARTIFICIAIS E ALGORITMOS
EVOLUCIONÁRIOS PARA OTIMIZAÇÃO DO CONSUMO DE COMBUSTÍVEL
EM EMBARCAÇÕES QUE UTILIZAM MOTOR DE COMBUSTÃO INTERNA A
DIESEL**

Esta dissertação foi julgada e aprovada para a obtenção do título de Mestre do Programa de Pós-Graduação em Engenharia, Gestão de Processos, Sistemas e Ambiental do Instituto de Tecnologia e Educação Galileo da Amazônia – ITEGAM.

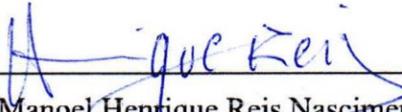
Manaus-AM, 27 de Julho de 2021.



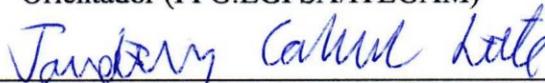
Prof. Dr. Jandecy Cabral Leite

Coordenador do PPG.EGPSA - ITEGAM

BANCA EXAMINADORA



Prof. Dr. Manoel Henrique Reis Nascimento
Orientador (PPG.EGPSA/ITEGAM)



Prof. Dr. Jandecy Cabral Leite
Examinador Interno (PPG.EGPSA/ITEGAM)



Prof. Dr. Carlos Alberto Oliveira de Freitas
Examinador Externo (UFAM)

**Dados Internacionais de Catalogação na Publicação (CIP)
Sistema de Biblioteca do ITEGAM**

Junior, Paulo Oliveira Siqueira, 2021 - Modelo Híbrido com Redes Neurais Artificiais e Algoritmos Evolucionários para Otimização do Consumo de Combustível em Embarcações que Utilizam Motor de Combustão Interna a Diesel / Paulo Oliveira Siqueira Junior - 2021. 105 f., il: Colorido

Orientador: Dr. Manoel Henrique Reis Nascimento

Dissertação: Instituto de Tecnologia e Educação Galileo da Amazônia, Programa de Pós Graduação em Engenharia, Gestão de Processos, Sistemas e Ambiental (PPG-EGPSA), Manaus - AM, 2021.

1. Modelo Híbrido Computacional 2. Algoritmos de Otimização 3. Redes Neurais Artificiais 4. Transporte Hidroviário 5. Otimização por Enxame de Partículas

CDD - 1001.ed.2021.3

AGRADECIMENTOS

A Deus, por sempre colocar pessoas incríveis na minha vida, as quais me fazem acreditar em um mundo melhor e me encorajam a prosseguir.

Aos meus pais, Paulo Oliveira Siqueira e Maria Erivalda de Sousa Oliveira, que sempre me apoiaram em todas as etapas da minha vida. À minha irmã Antônia Carla e aos meus familiares, por sempre estar presente na minha vida, por ter orgulho de mim e me incentivar a vencer os desafios encontrados.

Ao meu Orientador, Professor Doutor Manoel Henrique Reis Nascimento, por toda a paciência e ajuda durante a realização deste trabalho.

Aos meus amigos, Ítalo Rodrigo Soares Silva e Ricardo Silva Parente, que considero como se fossem meus irmãos, por sempre ter me dado apoio nesta jornada e nesse trabalho.

Aos professores do Programa de Pós-Graduação EGPSA e aos membros da banca, como Professor Doutor Jandecy Cabral Leite e o Professor Doutor Carlos Alberto Oliveira de Freitas, pelos ensinamentos e por todo conhecimento transmitido durante o curso de Mestrado.

Epígrafe

*“Se os fatos não se encaixam na teoria,
modifique os fatos”*

Albert Einstein

Dedicatória

À Deus, meus pais Paulo e Maria, a minha família, aos meus amigos e todas as pessoas que me ajudaram nessa jornada.

RESUMO

JUNIOR, Paulo Oliveira Siqueira. **MODELO HÍBRIDO COM REDES NEURAIAS ARTIFICIAIS E ALGORITMOS EVOLUCIONÁRIOS PARA OTIMIZAÇÃO DO CONSUMO DE COMBUSTÍVEL EM EMBARCAÇÕES QUE UTILIZAM MOTOR DE COMBUSTÃO INTERNA A DIESEL**. 2021. p. 105. Dissertação do programa de pós-graduação em Engenharia, Gestão de Processos, Sistemas e Ambiental (EGPSA), Instituto de Tecnologia e Educação Galileo da Amazônia (ITEGAM), Manaus, 2021.

O transporte hidroviário é o principal meio de locomoção na Região Norte, alimentando cidades por meio de embarcações, lanchas, navios e balsas com o transporte de mercadorias e/ou passageiros. Contudo, um dos problemas dessa modalidade de transporte é o custo com abastecimento, tendo em vista que a falta de tecnologias/recursos que permitam ou facilitem uma visão estratégica do negócio é uma realidade. O fluxo de transporte de passageiros concentra uma rotatividade de pessoas em média de 9 milhões, enquanto o transporte de cargas com aproximadamente 3 milhões, ambas distribuídas pela região norte. Fato que caracteriza uma demanda considerável no setor aquaviário de transporte, trazendo à tona a perspectiva desta pesquisa em estudar métodos de análise e apoio à tomada de decisão em função do consumo de combustível. A presente dissertação tem como objetivo apresentar os resultados acerca do desenvolvimento de um modelo de otimização do consumo de combustível considerando a velocidade ótima para embarcações de pequeno porte que operam de forma regular no Porto Fluvial de Manaus. Em virtude disso, a pesquisa cumpre com os objetivos de mapear as variáveis relativas as especificações da embarcação e motor a ser analisado, apresentar os métodos e resultados acerca do desenvolvimento do modelo híbrido computacional para otimização do consumo de combustível ao considerar como parâmetro de regulação a velocidade ótima para minimização da variável preditora e a distância do trajeto projetado para 3 cenários: Manaus à Itacoatiara, Manaus à Barcelos e Manaus à Parintins, determinar por meio de análise estatística de erros o melhor modelo de Rede Neural Artificial (RNA) ao considerar quantidade de neurônios, camadas ocultas, funções de ativação (Tangente hiperbólica, Sigmoide e Linear) e algoritmo de treinamento sendo este último 12 possibilidades cada um com distintos objetivos e estratégias de convergência, testar o modelo híbrido analisando o desempenho de 3 algoritmos de otimização (Enxame de Partículas, Algoritmo Genético e Recozimento Simulado) em função do custo computacional e a taxa de erro em cada

geração de elites, por fim, apresentar os resultados de simulação dos 3 cenários citados ao utilizar o modelo híbrido com o algoritmo vencedor em função dos requisitos citados. Para analisar os resultados providos das simulações, cenários e testes de aquisição dos melhores modelos, foi utilizado as técnicas estatísticas de Erro Médio Quadrático (MSE), Raiz do Erro Médio Quadrático (RMSE), Raiz do Erro Médio Quadrático Normalizado (NRMSE) e Erro Percentual Absoluto Médio (MAPE), os dados relativos a embarcação foram adquiridos por meio de reuniões e entrevistas com especialistas (proprietário da embarcação) para formalizar uma base de dados específica para o estudo delimitado nesta dissertação, 12 algoritmos de treinamento foram utilizados para a escolha da melhor RNA, de acordo com os resultados o Levenberg-Marquardt apresentou 100% de correlação entre as variáveis de saída e o Enxame de Partículas obteve o menor custo computacional em relação aos outros comprovando a eficácia do modelo híbrido computacional.

Palavras-Chave: Modelo Híbrido Computacional, Algoritmos de Otimização, Redes Neurais Artificiais, Transporte Hidroviário, Otimização por Enxame de Partículas.

ABSTRACT

JUNIOR, Paulo Oliveira Siqueira. **MODELO HÍBRIDO COM REDES NEURAIAS ARTIFICIAIS E ALGORITMOS EVOLUCIONÁRIOS PARA OTIMIZAÇÃO DO CONSUMO DE COMBUSTÍVEL EM EMBARCAÇÕES QUE UTILIZAM MOTOR DE COMBUSTÃO INTERNA A DIESEL**. 2021. p. 105. Dissertação do programa de pós-graduação em Engenharia, Gestão de Processos, Sistemas e Ambiental (EGPSA), Instituto de Tecnologia e Educação Galileo da Amazônia (ITEGAM), Manaus, 2021.

The waterway transport is the main means of locomotion in the North Region, feeding cities through boats, speedboats, ships and ferries with the transport of goods and/or passengers. However, one of the problems of this modality of transport is the cost of supply, considering that the lack of technologies/resources that allow or facilitate a strategic vision of the business is a reality. The flow of passenger transport concentrates an average turnover of 9 million people, while the cargo transport with approximately 3 million, both distributed throughout the northern region. This fact characterizes a considerable demand in the water transportation sector, bringing to light the perspective of this research to study methods of analysis and support for decision making on the basis of fuel consumption. This dissertation aims to present the results about the development of an optimization model of fuel consumption considering the optimal speed for small vessels that operate on a regular basis in the Manaus River Port. In view of this, the research meets the objectives of mapping the variables related to the specifications of the vessel and engine to be analyzed, present the methods and results about the development of the computational hybrid model for optimization of fuel consumption when considering as a parameter of regulation the optimal speed for minimizing the predictor variable and the distance of the projected path for 3 scenarios: Manaus to Itacoatiara, Manaus to Barcelos and Manaus to Parintins, determine by statistical error analysis the best model of Artificial Neural Network (ANN) when considering number of neurons, hidden layers, activation functions (hyperbolic Tangent, Sigmoid and Linear) and training algorithm being the latter 12 possibilities each with different objectives and convergence strategies, To test the hybrid model analyzing the performance of 3 optimization algorithms (Particle Swarm, Genetic Algorithm and Simulated Annealing) as a function of computational cost and error rate in each generation of elites, and finally, to present the simulation results of the 3 scenarios mentioned above when using the hybrid model with the winning algorithm as a function of the requirements mentioned above. To analyze the results provided by the simulations, scenarios and tests for the acquisition of the

best models, the statistical techniques of Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Normalized Root Mean Squared Error (NRMSE) and Mean Absolute Percentage Error (MAPE) were used, the data regarding the vessel were acquired through meetings and interviews with experts (vessel owner) to formalize a specific database for the study delimited in this dissertation, 12 training algorithms were used to choose the best ANN, according to the results the Levenberg-Marquardt presented 100% correlation between the output variables and the Particle Swarm obtained the lowest computational cost in relation to the others proving the effectiveness of the computational hybrid model.

Keywords: Computational Hybrid Model, Optimization Algorithms, Artificial Neural Networks, Waterway Transport, Particle Swarm Optimization.

LISTA DE FIGURAS

Figura 2.1 - Esquema de MCI de pistão.....	27
Figura 2.2 - Esquema de um MCI modelo SCANIA.	28
Figura 2.3 - Pistão dentro de um cilindro.	28
Figura 2.4 - Mínimos locais e global.....	30
Figura 2.5 - Fluxograma do algoritmo PSO.	34
Figura 2.6 - Fluxograma do algoritmo Recozimento Simulado.	35
Figura 2.7 - Comparativo de neurônio biológico com o artificial.....	37
Figura 2.8 - Arquitetura MLP.....	38
Figura 3.1 - Modelo de motor SCANIA.....	44
Figura 3.2 - Amostra de dados para treinamento da Rede Neural.....	45
Figura 3.3 - Exemplo de medida da inclinação.	46
Figura 3.4 - Fluxograma das etapas de processamento dos dados.	48
Figura 3.5 - Percurso de Manaus à Itacoatiara.	52
Figura 3.6 - Percurso de Manaus à Parintins.	52
Figura 3.7 - Percurso de Manaus à Barcelos.	52
Figura 4.1 - Fluxograma das etapas do algoritmo híbrido.....	53
Figura 4.2 - Fluxograma da etapa 1.....	54
Figura 4.3 - Fluxograma da etapa 2.....	55
Figura 4.4 - Fluxograma da etapa 3.....	56
Figura 4.5 - Total de Redes Neurais testadas.	58
Figura 4.6 - Gráficos de regressão da Rede Neural (500 dados).....	64
Figura 4.7 - Gráficos de regressão da Rede Neural (10000 dados).....	67
Figura 4.8 - Comportamento da otimização Manaus a Barcelos (500 dados).....	70
Figura 4.9 - Velocidade ideal alcançada pelo algoritmo (Cenário 1 com 500 dados).....	71
Figura 4.10 - Comportamento da otimização Manaus a Barcelos (10000 dados).....	72
Figura 4.11 - Velocidade ideal alcançada pelo algoritmo (Cenário 1 com 10000 dados)..	73
Figura 4.12 - Comportamento da otimização Manaus a Itacoatiara (500 dados).....	75
Figura 4.13 - Velocidade ideal alcançada pelo algoritmo (Cenário 2 com 500 dados).....	76
Figura 4.14 - Comportamento da otimização Manaus a Itacoatiara (10000 dados).....	78
Figura 4.15 - Velocidade ideal alcançada pelo algoritmo (Cenário 2 com 10000 dados)..	79
Figura 4.16 - Comportamento da otimização Manaus a Parintins (500 dados).	81
Figura 4.17 - Velocidade ideal alcançada pelo algoritmo (Cenário 3 com 500 dados).....	82

Figura 4.18 - Comportamento da otimização Manaus a Parintins (10000 dados).	84
Figura 4.19 - Velocidade ideal alcançada pelo algoritmo (Cenário 3 com 10000 dados)..	84
Figura 4.20 - Função objetiva do modelo calculado.	85
Figura 4.21 - Comportamento da curva de otimização - Manaus à Barcelos (Modelo calculado).....	87
Figura 4.22 - Ponto de otimização em função de 3 variáveis - Manaus à Barcelos (Modelo calculado).....	88
Figura 4.23 - Comportamento da curva de otimização - Manaus à Itacoatiara (Modelo calculado).....	89
Figura 4.24 - Ponto de otimização em função de 3 variáveis - Manaus à Itacoatiara (Modelo calculado).....	90
Figura 4.25 - Comportamento da curva de otimização - Manaus à Parintins (Modelo calculado).....	91
Figura 4.26 - Ponto de otimização em função de 3 variáveis - Manaus à Parintins (Modelo calculado).....	92
Figura 4.27 - Curva de distorção em função do objetivo e taxa de erro.....	95

LISTA DE GRÁFICOS

Gráfico 3.1 - Curva do SFOC.	47
Gráfico 3.2 - Curva de potência do motor.	47
Gráfico 4.1 - Levenberg-Marquardt com Sigmoide.	58
Gráfico 4.2 - Resultados com a Tangente hiperbólica.....	59
Gráfico 4.3 - Resultados com a função Linear.	60
Gráfico 4.4 - Comparativo entre Levenberg-Marquardt e Regularização Bayesiana (500 dados).....	63
Gráfico 4.5 - Comparativo entre Levenberg-Marquardt e Regularização Bayesiana (10000 dados).....	66
Gráfico 4.6 - Comparação de performance – 500 dados.	93
Gráfico 4.7 - Comparação de performance – 10000 dados.	94

LISTA DE TABELAS

Tabela 2.1 - Pseudocódigo do AG.....	31
Tabela 2.2 - Tipos de indivíduos.	31
Tabela 2.3 - Pseudocódigo do Recozimento Simulado.	36
Tabela 2.4 - Funções de Ativação.....	39
Tabela 3.1 - Características das Embarcações.....	40
Tabela 3.2 - Características de Máquinas das Embarcações	41
Tabela 4.1 - Resultados dos algoritmos de treinamento (Teste com 500 dados).	62
Tabela 4.2 - Resultados dos algoritmos de treinamento (Teste com 10000 dados).	65
Tabela 4.3 - Valores resultantes da simulação de Manaus a Barcelos (500 dados).	69
Tabela 4.4 - Valores resultantes da simulação de Manaus a Barcelos (10000 dados).	71
Tabela 4.5 - Valores resultantes da simulação de Manaus a Itacoatiara (500 dados).	74
Tabela 4.6 - Valores resultantes da simulação de Manaus a Itacoatiara (10000 dados). ...	77
Tabela 4.7 - Valores resultantes da simulação de Manaus a Parintins (500 dados).	80
Tabela 4.8 - Valores resultantes da simulação de Manaus a Parintins (10000 dados).	83
Tabela 4.9 - Resultantes da simulação calculada de Manaus à Barcelos.	86
Tabela 4.10 - Resultantes da simulação calculada de Manaus à Itacoatiara.	88
Tabela 4.11 - Resultantes da simulação calculada de Manaus à Parintins.	90

LISTA DE QUADROS

Quadro 2.1 - Abordagens utilizadas para minimizar o consumo de combustível.....	25
Quadro 2.2 - Variantes de PSO.	34
Quadro 3.1 - Relação entre as variáveis de entrada e saída.	41
Quadro 3.2 - Configurações da máquina de testes.	42
Quadro 3.3 - Algoritmos e parâmetros utilizados.	43

LISTA DE SIGLAS E ABREVIATURAS

AG	Algoritmos Genéticos
RS	Recozimento Simulado
GRASP	<i>Greedy Randomized Adaptive Search Procedure</i>
RNA	Redes Neurais Artificiais
IA	Inteligência Artificial
PSO	<i>Particle Swarm Optimization</i>
GA	<i>Genetic Algorithm</i>
SA	<i>Simulated Annealing</i>
FCVRP	<i>Fuel Capacitated Vehicle Routing Problem</i>
DCOP	<i>Distributed Optimization Problem</i>
PSO-iw	<i>PSO with inertia weight</i>
IILPSO	<i>Interswarm interactive learning strategy PSO</i>
ALCPSO	<i>PSO with an aging leader and challenger</i>
GLPSO	<i>Genetic learning PSO</i>
GAPSO	<i>Hybrid of Genetic Algorithm and PSO</i>
MLP	<i>Multilayer Perceptron</i>
TanH	Tangente Hiperbólica
ReLU	Unidade Linear Retificada
ELU	Unidade Linear Exponencial
Leaky ReLU	Unidade Linear Retificada com Vazamento
MSE	Erro Médio Quadrático
RMSE	Raíz do Erro Médio Quadrático
NMSE	Erro da Raíz Quadrática Média Normalizada
MAPE	Erro Percentual Médio Absoluto
GPU	Unidade de Processamento Gráfico
RAM	<i>Random Access Memory</i>
SSD	<i>Solid State Drives</i>
SFOC	Consumo Específico de Óleo Combustível
RPM	Rotações por Minuto
PW	Potência do Motor

SUMÁRIO

1	INTRODUÇÃO	21
1.1	JUSTIFICATIVA DA DISSERTAÇÃO	22
1.2	OBJETIVOS	23
1.2.1	Geral	23
1.2.2	Específicos	23
1.3	ESCOPO DO TRABALHO	24
2	REVISÃO BIBLIOGRÁFICA	25
2.1	TRABALHOS RELACIONADOS	25
2.2	MOTORES DE COMBUSTÃO INTERNA (MCI)	27
2.3	META-HEURÍSTICAS DA COMPUTAÇÃO	30
2.3.1	Otimização de Sistemas	30
2.3.2	Algoritmo Genético	31
2.3.2.1	Elitismo	32
2.3.2.2	Cruzamento	32
2.3.2.3	Mutação	32
2.3.3	Otimização por Enxame de Partículas (PSO)	32
2.3.4	Algoritmo do Recozimento Simulado (SA)	34
2.4	REDES NEURAIAS ARTIFICIAIS	37
3	MATERIAIS E MÉTODOS	40
3.1	MATERIAIS	40
3.1.1	Registro das Embarcações	40
3.1.2	Características de Máquinas	40
3.1.3	Variáveis Utilizadas no Modelo de RNA	41
3.1.4	Modelo de Análise Estatística	41
3.1.5	Algoritmos de Treinamento da Rede Neural	42
3.1.6	Recursos da Máquina de Teste	42
3.1.7	Parâmetros dos Algoritmos de Otimização	43

3.1.8	Ambiente de Desenvolvimento	43
3.2	MÉTODOS.....	43
3.2.1	Aquisição e Coleta de Dados.....	44
3.2.2	Seleção do Objeto de Estudo da Pesquisa	44
3.2.3	Seleção de Variáveis Significativas para o Modelo de Treinamento com RNA	45
3.2.4	Ajuste da Curva de SFOC e PW pelo Método do Coeficiente Angular	45
3.2.5	Processamento da Base de Dados.....	48
3.2.6	Procedimentos do Modelo de Otimização	50
3.2.7	Cenários de Simulação	51
4	RESULTADOS E DISCUSSÕES.....	53
4.1	MODELO DE OTIMIZAÇÃO BASEADO NO APRENDIZADO DE MÁQUINA... 53	
4.2	Configuração Inicial do Modelo de RNA.....	57
4.2.1	Modelo com a Função Linear	58
4.2.2	Modelo com a Função Tangente Hiperbólica	59
4.2.3	Modelo com a Função Sigmoide.....	60
4.2.4	Modelo Vencedor	61
4.3	Treinamento dos Modelos de RNA com 12 Algoritmos de Aprendizado	61
4.3.1	Teste com 500 Dados	62
4.3.2	Teste com 10000 Dados	65
4.4	SIMULAÇÕES DE CENÁRIOS	68
4.4.1	Cenário 1 (Manaus a Barcelos)	68
4.4.2	Cenário 1 (Manaus a Barcelos – 500)	68
4.4.3	Cenário 1 (Manaus a Barcelos – 10000)	71
4.4.4	Cenário 2 (Manaus a Itacoatiara)	73
4.4.5	Cenário 2 (Manaus a Itacoatiara – 500)	74
4.4.6	Cenário 2 (Manaus a Itacoatiara – 10000)	76

4.4.7	Cenário 3 (Manaus a Parintins)	79
4.4.8	Cenário 3 (Manaus a Parintins – 500)	80
4.4.9	Cenário 3 (Manaus a Parintins – 10000)	83
4.5	Modelo Calculado para Validação de Resultados	85
4.6	Resultados do Modelo Calculado	86
4.6.1	Cenário 1 (Manaus à Barcelos)	86
4.6.2	Cenário 2 (Manaus à Itacoatiara)	88
4.6.3	Cenário 3 (Manaus à Parintins)	90
4.7	Resultados Finais	92
4.7.1	Resultados Finais das Simulações	92
4.7.2	Resultados Finais do Modelo Calculado.....	94
5	CONCLUSÕES E SUGESTÕES	96
5.1	CONCLUSÕES	96
5.2	SUGESTÕES PARA TRABALHOS FUTUROS.....	96
	REFERÊNCIAS.....	98
	APÊNDICE A – ARTIGO PUBLICADO	105

CAPÍTULO 1

1 INTRODUÇÃO

Uma das atividades fluviais amplamente praticadas é o transporte de cargas e passageiros por embarcações (DELGADO-HIDALGO, RAINWATER e NACHTMANN, 2020), onde o fluxo de viagens se desenvolve de maneira ativa conforme as necessidades de viagem da embarcação. Contudo, um dos problemas dessa modalidade de transporte é o custo com abastecimento, tendo em vista que a falta de recursos que permitam uma visão estratégica do negócio é uma realidade (TAN, DURU e THEPSITHAR, 2020).

Considerando as características relativas à embarcação no que tange o tempo estimado de viagem, consumo específico de óleo combustível, velocidade média e a distância é possível propor medidas inteligentes para alcançar o mínimo esforço para o motor e conseqüentemente a redução do custo de combustível, tal como ocorre no modelo SCANIA de potência 315hp onde a produção de gases emissores prejudica o ambiente de acordo com o consumo relativo de combustível (HATAMI, HASANPOUR e JING, 2020).

Com isso, uma das grandes dificuldades para mensurar ou analisar métodos que auxiliem na tomada de decisão é a falta de modelos inteligentes que otimizem ou realize projeções do melhor cenário estudado (MENZEL *et. al.*, 2020), dessa forma, surge a ideia da presente pesquisa ao contribuir com uma combinação de algoritmos para otimização e previsão do consumo de combustível que informe a melhor solução de velocidade para a embarcação.

Métodos heurísticos da computação são comumente utilizados em problemas de otimização onde não se conhece um algoritmo eficiente, apesar do esforço computacional para achar a melhor solução ainda é um procedimento viável quando combinada com outros métodos (HOSSEINIOUN *et. al.*, 2020).

Os meta-heurísticos são classificados da seguinte forma: Algoritmo Genético (AG) do inglês (*Genetic Algorithm - GA*), Recozimento Simulado (RS) do inglês (*Simulated Annealing - SA*), *Greedy Randomized Adaptive Search Procedure* (GRASP), Busca Tabu, Colônia de Formigas, Colônia de Abelhas e Lichtenberg Algorithm (CHEN *et. al.*, 2020; ABD ELAZIZ, EWEEES, OLIVA, 2020).

A função objetivo tem como finalidade fornecer meios matemáticos de solução do problema a ser otimizado, contudo, os modelos híbridos vêm sendo utilizados de forma assídua,

apresentando resultados satisfatórios no que diz respeito aproximação de resultados (GAO *et. al.*, 2020).

Dentre os métodos de combinação existentes se encontra o modelo computacional baseado em aprendizado de máquina, as Redes Neurais Artificiais (RNA) que são inspirados no sistema nervoso humano, com esse método bio-inspirado é possível realizar procedimentos de reconhecimento de padrões, predição e classificação (HUANG *et. al.*, 2020; FAGUNDEZ *et. al.*, 2020).

A aplicação dos métodos citados permite a combinação de algoritmos para otimização e previsão onde encontrar a melhor solução para o problema do consumo de combustível de embarcações que utilizam máquinas de combustão interna é o problema (FAGUNDEZ *et. al.*, 2020).

Dessa forma, a pesquisa tem como objetivo o desenvolvimento de um modelo computacional para otimização e previsão da melhor velocidade para as características do motor SCANIA 315hp considerando a redução do consumo de combustível. Como metodologia será utilizado a abordagem de aprendizado supervisionado de Redes Neurais Artificiais e Algoritmos de otimização para a minimização do consumo de combustível ao levar em consideração as características da máquina objeto de estudo e da embarcação, além disso o modelo é caracterizado inovador pelo uso de técnicas de Inteligência Artificial (IA).

1.1 JUSTIFICATIVA DA DISSERTAÇÃO

A problemática da pesquisa vem de encontro com a necessidade de otimizar custos com combustível por embarcações do porto fluvial de Manaus, considerando as que possuem motores de combustão interna, o custo com combustível pelas embarcações é variável e relativo à distância do trajeto de navegação.

As embarcações transportam cargas e passageiros para diversos municípios da Região Norte do Amazonas, cada localidade possui uma distância relativa que se for levada em consideração aumenta o consumo do motor se não for devidamente regrado.

É comum os proprietários de embarcações não terem uma base do quanto é gasto por cada viagem, nem tampouco o consumo diário, por esse motivo surge a presente pesquisa em desenvolver um modelo computacional de previsão e otimização que contribua com a minimização do consumo de combustível ao considerar a velocidade ideal para o trajeto simulado.

O estudo terá como relevância a implementação de um modelo híbrido para analisar e prever a melhor solução de velocidade da embarcação considerando algumas variáveis que minimizem o consumo de combustível, tendo em vista que a embarcação objeto de estudo possui um motor de combustão interna. Dentre as contribuições se destacam:

- Desenvolvimento de um modelo computacional híbrido;
- Analisar por meio de algoritmos de otimização a melhor solução para a velocidade da embarcação sem considerar variáveis relativas a influência do rio no desempenho da embarcação;

Além disso, a presente dissertação se limita a um estudo voltado a previsão e otimização da velocidade ideal do motor para reduzir custos com combustível, portanto são utilizadas as seguintes variáveis:

- Velocidade;
- Consumo de combustível;
- Distância.

Para o desenvolvimento do modelo híbrido será utilizado a técnica de previsão Redes Neurais Artificiais e Algoritmos de Otimização como:

- Algoritmo Genético;
- *Particle Swarm Optimization* (PSO);
- Recozimento Simulado.

O modelo híbrido proposto não leva em consideração variáveis como resistência da água, inclinação dos rios e peso da embarcação.

1.2 OBJETIVOS

1.2.1 Geral

Desenvolver um modelo híbrido de otimização do consumo de combustível a partir da velocidade ótima, utilizando Redes Neurais Artificiais e Algoritmos de otimização.

1.2.2 Específicos

- Mapear variáveis relativas as especificações da embarcação e motor a ser analisado;
- Desenvolver um modelo computacional híbrido de otimização do consumo de combustível considerando a velocidade ótima e a distância do trajeto.

- Determinar por meio de indicadores de erro o melhor modelo de RNA para substituir a função objetivo;
- Testar o modelo híbrido analisando o desempenho do algoritmo de otimização em função do custo computacional e taxa de erro.
- Analisar os resultados do modelo híbrido desenvolvido em 3 cenários.

1.3 ESCOPO DO TRABALHO

A presente dissertação é estruturada da seguinte forma:

- **Capítulo 1:** O capítulo apresenta as contextualizações iniciais acerca da pesquisa, enfatizando os objetivos que devem ser alcançados, a importância, justificativa e relevância do trabalho.
- **Capítulo 2:** O capítulo apresenta uma revisão bibliográfica acerca do tema, onde é apresentado tópicos relativos a motores de combustão interna, modelos de previsão utilizando IA e algoritmos de otimização; Neste capítulo é apresentado uma revisão bibliografia sobre o tema abordado;
- **Capítulo 3:** O capítulo aborda os principais materiais que foram utilizados na pesquisa enfatizando os métodos para se chegar no produto final que é o modelo computacional híbrido;
- **Capítulo 4:** O capítulo apresenta os principais resultados alcançados acerca das análises e testes realizados com a aplicação de modelos de previsão com RNA e algoritmos de otimização para a minimização do custo com combustível ao simular 3 cenários de viagem.
- **Capítulo 5:** Por fim são apresentadas as conclusões finais da pesquisa e as sugestões para trabalhos futuros.

CAPÍTULO 2

2 REVISÃO BIBLIOGRÁFICA

O presente capítulo tem como objetivo abordar uma revisão bibliográfica baseada em trabalhos relacionados ao tema proposto, enfatizando conceitos, técnicas e aplicações acerca de motores de combustão interna, algoritmos meta-heurísticos como: GA, PSO e SA. Por fim, é explorado sobre as técnicas de previsão utilizando RNA.

2.1 TRABALHOS RELACIONADOS

As diversas técnicas de IA são utilizadas para a previsão e otimização de distintos cenários e aplicações, não é diferente em relação ao consumo de combustível. Podem ser utilizadas algumas abordagens para se alcançar a minimização do consumo de combustível, como pode ser observado no Quadro 2.1.

Quadro 2.1 - Abordagens utilizadas para minimizar o consumo de combustível.

Autores	Abordagens	Título
(XAVIER <i>et al.</i> , 2010)	Algoritmos de otimização para o problema do Carteiro Chinês não Dirigido do inglês (<i>Undirected Chinese Postman</i>), no qual foi aplicado o algoritmo de seleção de melhor rota <i>Dijkstra</i> .	Heurística para modelagem e minimização do consumo de combustível para rotas de coleta de lixo.
(MARTINS <i>et al.</i> , 2018)	Aplicação de normas técnicas para o bom funcionamento do motor otimiza o consumo de combustível.	Otimização energética de um trator agrícola utilizando normas técnicas em operações de gradagem.
(ARRUDA JÚNIOR, 2014)	A forma como os motoristas conduzem o veículo possui relação com o gasto de combustível, logo a capacitação dos motoristas em técnicas de condução de veículos pode diminuir o consumo de combustível em cerca de 12% a 22% (conforme dados do estudo).	Desenvolvimento de um método para redução do consumo de combustível no transporte rodoviário de cargas: Capacitação, aplicações de torque e telemetria para veículos pesados.
(XIAO <i>et al.</i> , 2012)	Utilização de IA (Recozimento Simulado) para ajudar na otimização de rotas do Problema de Roteamento de Veículos com Capacidade de Combustível do inglês (<i>Fuel Capacitated Vehicle Routing Problem - FCVRP</i>).	<i>Development of a fuel consumption optimization model for the capacitated vehicle routing problem.</i>
(LEITE, 2009)	IA Distribuída com o <i>Framework</i> de Problema de Otimização Distribuída do inglês (<i>Distributed Optimization Problem - DCOP</i>).	Um esquema para redução do consumo de combustível em sistemas de condução férrea baseado em Otimização Distribuída de Restrição.

Fonte: AUTOR (2021).

Dentre as abordagens encontradas para a minimização de combustível, foram selecionadas cinco que podem ser elencadas como abordagens por: Rotas (XAVIER *et al.*, 2010), Aplicação de normas técnicas (MARTINS *et al.*, 2018), Técnicas de condução (ARRUDA JÚNIOR, 2014), IA+Rotas (XIAO *et al.*, 2012) e IA (LEITE, 2009).

Rotas: No artigo de Xavier *et al.* (2010), é utilizado uma abordagem baseada em rotas, utilizando o algoritmo *Dijkstra*, que é muito popular na teoria dos grafos, essa abordagem busca encontrar o menor caminho até o local de destino e assim minimiza o consumo de combustível. Este método de minimizar o custo de combustível é um dos mais utilizados, porém é uma ótima técnica apenas para veículos terrestres, para o estudo de caso não é viável, já que aborda rotas marítimas de curto e médio espaço, logo não é possível aplicar a solução de melhor rota para minimizar o consumo de combustível.

Aplicação de normas técnicas: O trabalho de MARTINS *et al.* (2018), busca diminuir o consumo de combustível de veículos automotores com aplicação de normas técnicas de bom uso dos motores, como a rotação e o peso utilizado no veículo.

Técnicas de condução: O consumo de combustível está ligado também a forma com que o condutor conduz o veículo, logo conduzir de forma correta sem forçar o motor e seguindo as normas técnicas também se mostra eficaz na minimização de combustível, no estudo de caso do artigo de Arruda Júnior (2014), aplicando esta técnica alcançou-se uma redução de 12% a 22% no consumo de combustível. Esta abordagem é interessante e está relacionada no trabalho proposto.

IA+Rotas: Neste trabalho de XIAO *et al.* (2012), é utilizada uma abordagem de escolha de melhor rota, no qual tem um diferencial, no qual é utilizado recozimento Simulado para ajudar na escolha da melhor rota. Como já explicado anteriormente, esta abordagem não é utilizada neste trabalho para a minimização do consumo de combustível.

IA: O trabalho de Leite (2009), utiliza Inteligência Artificial Distribuída para identificar a velocidade ideal a se percorrer pelos trilhos do trem, esta metodologia é muito interessante e serviu como base para o trabalho proposto, no qual utiliza-se PSO, AG e RS aplicados a RNA para prever a velocidade ideal para um determinado trajeto por navegação com balsas.

Logo entre as abordagens encontradas, a que melhor se adequou para a aplicação de minimização do consumo de combustível em balsas, foi a otimização por meio de IA. Esta escolhida para encontrar a velocidade média dos trajetos analisados neste trabalho, com o intuito de indicar ao condutor a melhor velocidade para obter o mínimo de gasto com combustível.

2.2 MOTORES DE COMBUSTÃO INTERNA (MCI)

O motor que é uma das invenções que causaram grandes impactos na sociedade, na economia e no meio ambiente, são máquinas que transformam a energia proveniente de reações químicas em energia mecânica (FAGUNDEZ *et. al.*, 2020; HATAMI, HASANPOUR e JING, 2020).

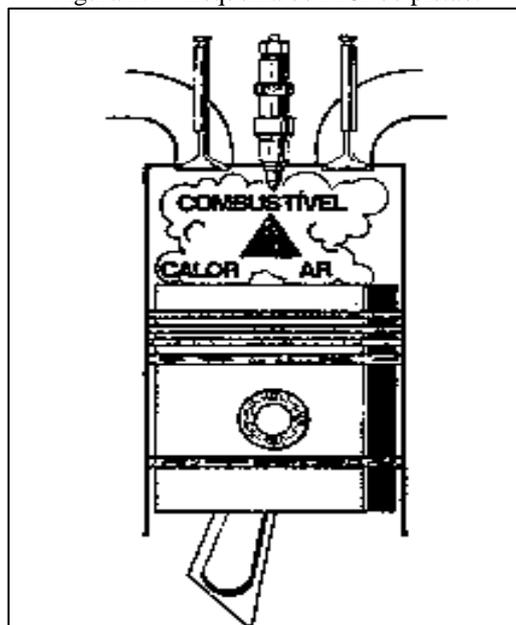
Estes motores que são comumente utilizados para propulsão de sistemas móveis também são utilizados em aplicações industriais como produção de petróleo, gás, compressão, exploração de pedreiras, reciclagem e geração de energia elétrica (SHEYKHI *et. al.*, 2020; BASKOV, IGNATOV e POLOTNYANSCHIKOV, 2020).

Esses motores movidos a diesel são caracterizados pela ignição por compressão, máquinas que impulsionam veículos pesados como caminhões, trens e navios são baseadas na propulsão diesel-elétrica (FAGUNDEZ *et. al.*, 2020; SILVA *et. al.*, 2019). O movimento alternativo do pistão dentro do cilindro é transformado em movimento rotativo através da biela e do virabrequim (SILVA *et. al.*, 2019).

Já nos motores 4 tempos onde é utilizado gases, um ciclo termodinâmico é completado a cada duas voltas no eixo, nesse momento ocorre admissão e compressão em uma volta e transferência de calor na segunda (BERTONI JUNIOR, 2020; SINGH, KUMAR e AGARWAL, 2020).

Os maiores motores de propulsão naval, a Diesel operam em 2 tempos com a utilização de apenas uma janela e uma válvula no cabeçote. A Figura 2.1 ilustra um esquema de motor a combustão interna de pistão.

Figura 2.1 - Esquema de MCI de pistão.



Fonte: Adaptado de DALPRÁ (2020).

O motor é dividido em partes fixas e móveis, as fixas são: bloco, cárter e cabeçote, os móveis são: árvore de manivelas, pistão, biela e comando de válvulas. (SINGH, KUMAR e AGARWAL, 2020). A seguir na Figura 2.2 é apresentado algumas partes componentes de uma máquina de combustão interna como forma de exemplificar o processo de composição e funcionamento do mesmo.

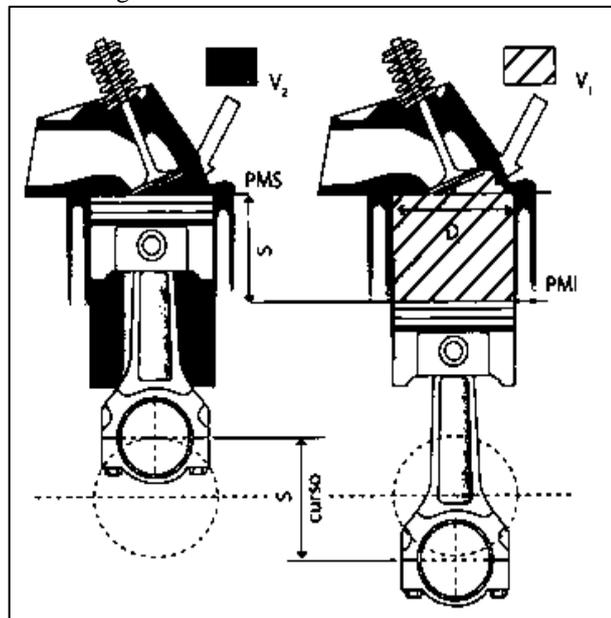
Figura 2.2 - Esquema de um MCI modelo SCANIA.



Fonte: Adaptado de SCANIA (2021).

O cilindro é a peça que recebe o movimento de expansão de gases, normalmente feita de alumínio ou liga de alumínio com um formato cilíndrico, a parte média é chamada de copo onde existem dois orifícios circulares para alojar o eixo do pistão que o une à biela (SINGH, KUMAR e AGARWAL, 2020; SILVA *et. al.*, 2019). A Figura 2.3 ilustra um pistão ou êmbolo dentro de um cilindro.

Figura 2.1 - Pistão dentro de um cilindro.



Fonte: BRUNETTI (2018).

Para entender a cilindrada do motor é necessário entender a cilindrada unitária também conhecido como volume deslocado útil (RUFINO *et.al.*, 2020; BRUNETTI, 2018), a cilindrada do volume é obtida pela Eq. (2.1):

$$V_u = \frac{\pi \cdot D^2}{4} S = V_1 - V_2 \quad (2.1)$$

Onde:

S = Curso do pistão;

V₁ = Volume de todo o cilindro;

V₂ = Volume da câmara de combustão;

D = Diâmetro do cilindro

A cilindrada total do motor é obtida pela Eq. (2.2):

$$V = V_u \cdot Z \quad (2.2)$$

Onde:

V_u = Volume unitário;

Z = cilindros que o motor possui

A taxa de compressão do motor é a relação volumétrica entre o volume de todo o cilindro e o volume da câmara de combustão (RUFINO *et.al.*, 2020; BRUNETTI, 2018), obtido pela Eq. (2.3):

$$T_c = \frac{V_1}{V_2} \quad (2.3)$$

Onde:

V₁ = Volume de todo o cilindro;

V₂ = Volume da câmara de combustão;

A potência de um motor é definida como o trabalho realizado em uma unidade de tempo (RUFINO *et.al.*, 2020; BRUNETTI, 2018), obtida através da Eq. (2.4):

$$\bar{W} = \frac{F \cdot d}{t} \quad (2.4)$$

Onde:

\bar{W} = potência expressa em cv, ps, hp ou watts;

F = intensidade da força;

d = distância entre o eixo e a força;

t = tempo

2.3 META-HEURÍSTICAS DA COMPUTAÇÃO

Uma meta-heurística pode ser compreendida como uma estratégia de busca não especificada para um determinado problema, este que tenta explorar eficientemente o espaço de busca, ou seja levam em consideração a vizinhança. Alguns autores classificam as meta-heurísticas em: Busca por entornos, Relaxação, Construtivas, Evolutivas (WANG *et. al.*, 2020).

Alguns dos algoritmos de otimização conhecidos por serem classificados como meta-heurísticos são: Algoritmos Genéticos, Recozimento Simulado, Greedy Randomized Adaptive Search Procedure, Busca Tabu, Colônia de Formigas, Colônia de Abelhas e Lichtenberg Algorithm (CHEN *et. al.*, 2020; ABD ELAZIZ, EWEES, OLIVA, 2020).

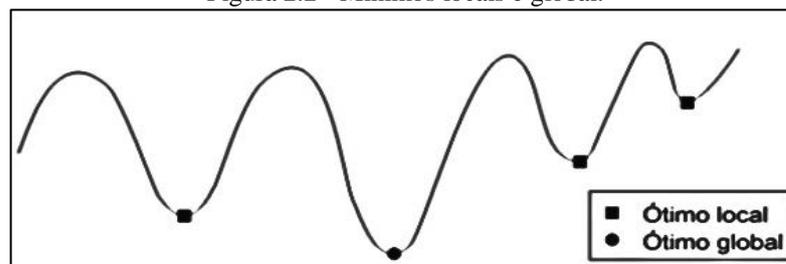
2.3.1 Otimização de Sistemas

A estratégia de busca de uma meta-heurística depende da metodologia de escapar dos mínimos locais e globais afim de explorar eficientemente o espaço de busca por soluções melhores (WANG *et. al.*, 2020; NASCIMENTO *et. al.*, 2017).

Seja f uma função com domínio S , caracterizada como função de custo ou função objetivo, tem-se um problema de minimização quando dado f se quer encontrar $s \in S$ tal que $f(s) \leq f(S), \forall s \in S$, se o objetivo é encontrar um $s \in S$ tal que $f(s) \geq f(S), \forall s \in S$, então se tem um problema de maximização (OSABA *et. al.*, 2020).

Dessa forma, um ponto ótimo global representa o ponto máximo ou mínimo correspondente a melhor solução do espaço de busca enquanto um ponto ótimo local representa o ponto máximo ou mínimo dentre um conjunto de pontos limitados (WANG *et. al.*, 2020; OSABA *et. al.*, 2020). A Figura 2.4 ilustra um exemplo de espaço de busca com mínimos locais e globais.

Figura 2.2 - Mínimos locais e global.



Fonte: Adaptado de SCHELLENBERG, LOHAN e DIMACHE (2020).

Problemas de otimização com restrições consideram limites para as variáveis do AG definidos por um número n_{restr} de funções $g_{nrestr} : \mathcal{R}^n \rightarrow \mathcal{R}$, que originam um subconjunto $F = \{\vec{x} \in M | g_j(\vec{x}) \geq 0 \forall j\} \subseteq M$, conhecido como conjunto de soluções viáveis de f (SCHELLENBERG, LOHAN e DIMACHE, 2020).

2.3.2 Algoritmo Genético

É uma classe de algoritmo que realiza procedimentos de busca baseado no conceito de seleção natural e sobrevivência do indivíduo mais apto, compostos por uma sequência de rotinas computacionais com o intuito de simular o comportamento evolucionário (WANG *et. al.*, 2020; NASCIMENTO *et. al.*, 2017). A Tabela 2.1 ilustra as etapas de algoritmo genético padrão.

Tabela 2.1 - Pseudocódigo do AG.

Pseudocódigo	
Function Genetic Algorithm(pop, objective) out: bestIndividual	
Inputs: definition pop; definition objective	
do	parentSelection(pop) recombination(pop) mutation(pop) evaluation(pop) apt(pop)
while don't hit stop	
return bestIndividual	

Fonte: AUTOR (2021).

População inicial: O AG inicializa uma população de forma aleatória, ou seja, por meio de sorteio computacional, este procedimento se torna essencial para a codificação das strings segundo os blocos de formação (HUI, ZENG e YU, 2020). A Tabela 2.2 apresenta um exemplo de populações em AG.

Tabela 2.2 - Tipos de indivíduos.

Indivíduo	População 1		População 2	
	1	0101100001101	17.1	7.9
2	0101111001101	21.3	8.1	
	
3	0101100001010	15.7	6.8	

Fonte: Adaptado de (HUI, ZENG e YU, 2020).

Seleção: Para a seleção dos indivíduos alguns algoritmos podem ser utilizados, estes que são (HUI, ZENG e YU, 2020):

- **Seleção Proporcional:** conhecido como Regra da Roleta, sorteia os indivíduos para cruzamento, onde a probabilidade de sorteio de um indivíduo i é dado por $p_i = \Phi(a_i) / \sum_{j=1}^u \Phi(a_j)$, onde u representa o tamanho da população e $\Phi: B^l \rightarrow \mathcal{R}^+$, a função de avaliação.
- **Escalonamento:** avalia-se cada indivíduo para recalculá-lo em $f' = a \cdot f + b$, onde f é a função original de avaliação e os coeficientes a e b podem ser escolhidos de outras maneiras.
- **Pressão Seletiva:** método empregado para avaliar o valor de avaliação dos indivíduos que varia de acordo com a pressão o que aumenta ou melhora a exploração do espaço de busca tendendo a convergência mais rápida.

• **Torneio:** sorteia dois indivíduos para cruzamento e seleciona para participar de um cruzamento aquele com melhor avaliação.

• **Ranking Linear:** Nesse método a população é ordenada de acordo com os valores de avaliação de cada indivíduo, onde a probabilidade p_i de sorteio do indivíduo é dada pela equação: $p_i(a_i^g) = 1/\lambda(n_{\max} - (n_{\max} - n_{\min}) \cdot \frac{i-1}{\lambda-1})$, onde $n_{\min} = 2 - n_{\max}$ e $1 \leq n_{\max} \leq 2$, a_i^g corresponde ao indivíduo i e λ é o número de indivíduos na população g .

2.3.2.1 Elitismo

O elitismo consiste em reintroduzir um indivíduo apto de uma geração para a próxima evitando a perda de informações importantes, há técnicas que determinam ou controlam a quantidade de reintroduções de um indivíduo para a próxima geração afim de escapar dos máximos locais (HUI, ZENG e YU, 2020; SCHELLENBERG, LOHAN e DIMACHE, 2020).

2.3.2.2 Cruzamento

O cruzamento se resume em uma troca de substrings entre dois indivíduos, este operador recombina soluções e favorece uma melhor exploração do espaço de busca. Existem alguns métodos para recombinação sendo eles: de um ponto simples, múltiplo e uniforme (HUI, ZENG e YU, 2020).

2.3.2.3 Mutação

Um algoritmo evolutivo é capaz de codificar uma probabilidade de mutação no indivíduo, desde que se tenha um valor diferente para cruzamento entre os indivíduos, ao longo da execução é esperado que o valor de mutação tenda para valores menores contribuindo para a convergência do mesmo ao encontrar o ponto ótimo em um espaço de busca (HUI, ZENG e YU, 2020).

2.3.3 Otimização por Enxame de Partículas (PSO)

Essa técnica metaheurística foi desenvolvida por Eberhart e Kennedy (1995) e Shi e Eberhart (1998). O algoritmo PSO seleciona um grupo de possíveis soluções (que no algoritmo é definida como partículas) com suas posições iniciadas de forma estocástica, no qual as partículas são configuradas a fim de se movimentarem no conjunto de soluções para alcançar a melhor solução para o problema (KUMAR *et al.*, 2017).

A técnica PSO surgiu na computação como forma de melhorar o processo de alcançar o alvo com uma convergência maior a partir da sociabilidade dos nós, imitando o comportamento

de animais que trabalham de forma coletiva como, formigas, abelhas e pássaros. Assim os indivíduos de uma população conversam entre os mesmos para alcançar o objetivo de forma mais rápida e eficiente (MARINI e WALCZAK, 2015; DUFOUR e NEVES, 2019; KUMAR *et al.*, 2017).

O PSO se baseia em três características dos pássaros e peixes, que são velocidade, localização e pouca memória (YANG, 2019). Os pássaros voam a uma determinada velocidade para alcançarem seus objetivos (alimento), enquanto que precisam de uma certa forma se organizarem e manterem um padrão no voo, para que não se separem um dos outros dificultando para os predadores, outro fator determinante para o funcionamento do algoritmo é a pouca memória, pássaros e peixes precisam se atualizar rapidamente da posição dos seus iguais, logo não podem armazenar uma posição muito antiga.

Com essas características básicas do comportamento dos animais que trabalham coletivamente é que o algoritmo foi concebido (YANG, 2019). De acordo com Arora (2012), os principais elementos utilizados no PSO são:

Partícula: Este é o nome dado ao indivíduo da população, como um peixe no cardume ou até mesmo uma abelha no enxame. Cada particular está localizado em um determinado ponto, que é uma solução em potencial do problema a ser solucionado.

Posição da partícula: É a localização das partículas nos eixos x e y do plano cartesiano.

Velocidade da partícula: Este termo referência a velocidade de convergência das partículas em relação ao objetivo a ser alcançado.

Líder do enxame: É a particular que se localiza o mais próximo da função objetivo.

Para melhor entendimento pode-se utilizar x_i e v_i nas Eq. (2.5) e (2.6) para exemplificar o funcionamento do algoritmo em relação a atualização das velocidades e localização das partículas (YANG, 2019).

$$v_i^{t+1} = v_i^t + \alpha \varepsilon_1 [g^* - x_i^t] + \beta \varepsilon_2 [x_i^* - x_i^t], \quad (2.5)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \Delta t, \quad (2.6)$$

Onde:

x_i = Localização da partícula.

v_i = Velocidade da partícula.

ε_1 = Número uniformemente distribuído no conjunto da população.

ε_2 = Número uniformemente distribuído no conjunto da população.

α = Parâmetro de aprendizagem.

β = Parâmetro de aprendizagem.

g^* = Melhor solução encontrada na iteração para todas as partículas.

x_i^* = Melhor solução individual por partícula.

Desde a criação do PSO em 1995, foram realizados diversos estudos que geraram diversas variantes do PSO, entre elas encontra-se os listados no Quadro 2.2.

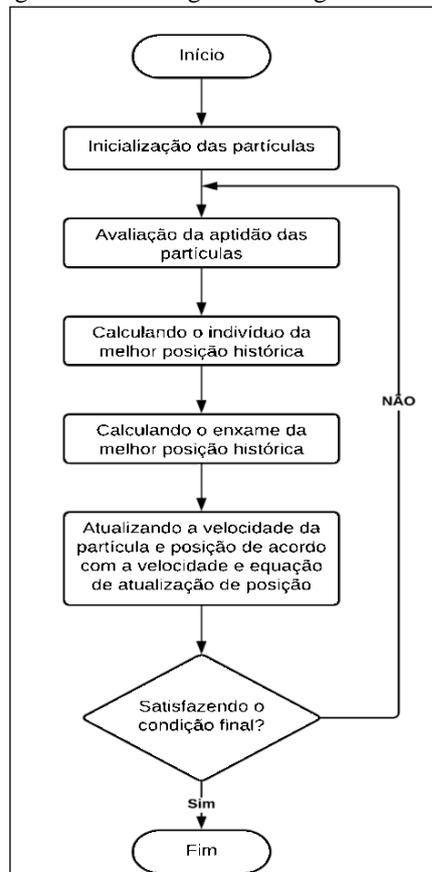
Quadro 2.2 - Variantes de PSO.

Nome	SIGLA	Autores
<i>PSO with inertia weight</i>	PSO-iw	(SHI e EBERHART, 1998)
<i>Inter swarm interactive learning strategy PSO</i>	IILPSO	(QIN <i>et al.</i> , 2015)
<i>PSO with an aging leader and challenger</i>	ALCPSO	(CHEN <i>et al.</i> , 2012)
<i>Genetic learning PSO</i>	GLPSO	(GONG <i>et al.</i> , 2015)
<i>Hybrid of Genetic Algorithm and PSO</i>	GAPSO	(NAKA <i>et al.</i> , 2003)

Fonte: Adaptado de PIOTROWSKI, NAPIORKOWSKI e PIOTROWSKA (2020); DEL VALLE, *et al.* (2008).

O algoritmo utilizado no trabalho foi o PSO original de Eberhart e Kennedy, publicado em 1995. No qual o fluxograma de funcionamento do algoritmo se encontra na Figura 2.5.

Figura 2.3 - Fluxograma do algoritmo PSO.



Fonte: Adaptado de WANG, TAN e LIU (2018).

2.3.4 Algoritmo do Recozimento Simulado (SA)

O algoritmo denominado de Recozimento Simulado foi proposto por Kirkpatrick, Gelatt e Vecchi em 1983 (KIRKPATRICK, GELATT e VECCHI, 1983). O Recozimento Simulado

foi baseado no comportamento físico do recozimento dos materiais e tem como finalidade a otimização (RAMUJANE, 2019).

O comportamento do recozimento dos materiais consiste em derreter o material (aço por exemplo) até que o mesmo atinja o estado líquido, isso faz com que os átomos se agitem e tenham uma alta taxa de mobilidade, após o derretimento, é feito um resfriamento gradativo no material, a fim de deixar os átomos menos agitados, se a comandando no material e alcançando uma melhor estabilidade e resistência (TIMANÁ-PEÑA, COBOS-LOZADA e TORRES-JIMENEZ, 2016; GONZALES, DOS SANTOS e NETO, 2018).

De acordo com Júnior (2018), o procedimento de têmpera dos metais possui três etapas básicas, sendo elas:

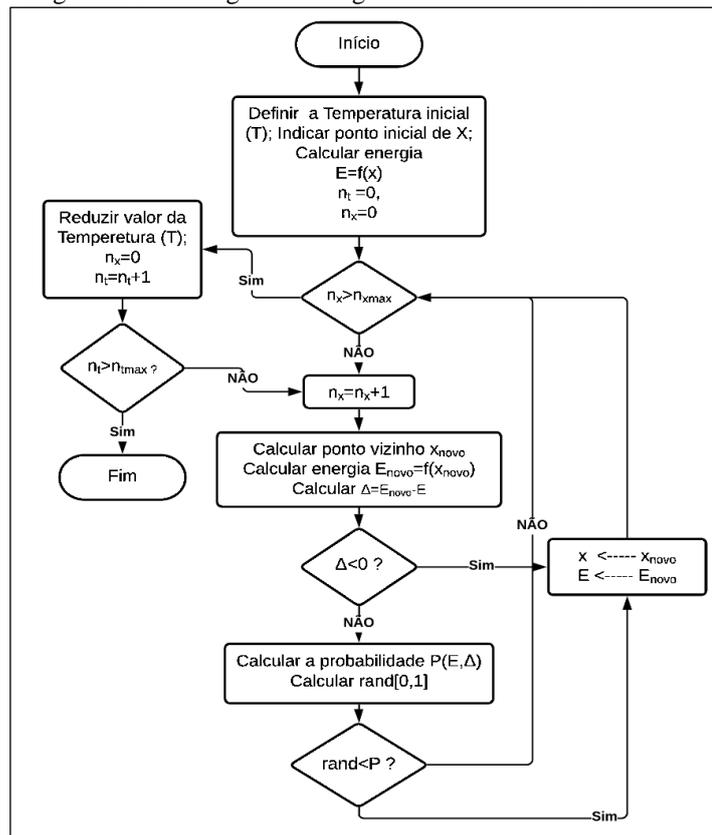
Etapa 1: Aquecimento do material a uma determinada temperatura.

Etapa 2: Manter a temperatura elevada para que aja o agitação das moléculas de átomos.

Etapa 3: Arrefecimento do metal, para controlar o tamanho dos cristais e fazer com que aja o acomodamento das moléculas de átomos.

O processo de funcionamento do Recozimento simulado ocorre conforme o fluxograma apresentado na Figura 2.6, no qual é definido uma temperatura T.

Figura 2.4 - Fluxograma do algoritmo Recozimento Simulado.



Fonte: Adaptado de LOBATO *et al.* (2012); JÚNIOR (2018).

A partir da definição da temperatura (T) inicial e o ponto inicial de X é possível calcular a energia em função de x, sendo que n_t e n_x devem ser iniciados em com o valor nulo (0). Vale ressaltar que a cada iteração do Recozimento Simulado é gerado uma nova possível solução que é comparada com a anterior, a fim de encontrar a melhor e fugir do máximo local e atingir o máximo global.

Caso a temperatura definida no início do algoritmo não satisfaça as condições de estar dentro do limite estabelecido, a temperatura é recalculada até que esteja dentro do limite máximo, quando esta condição é satisfeita, o próximo passo é calcular o ponto vizinho, a energia e o valor de deterioração (Δ).

Sendo a deterioração maior que zero, é calculada a probabilidade de deterioração e também é gerado um número aleatório entre 0 e 1, caso o número gerado aleatoriamente seja menor que a probabilidade de deterioração calculada, o valor de x e de E(energia) é adicionado e o algoritmo executa novamente desde a segunda etapa, que é a verificação dos parâmetros para identificar se a solução encontrada é ótima.

O algoritmo pode ser interrompido quando não há uma melhora na evolução das soluções ou quando o algoritmo alcança uma determinada quantidade de iterações especificada pelo utilizador do algoritmo.

De forma resumida e objetiva o algoritmo de Recozimento Simulado pode ser expresso conforme a Tabela 2.3.

Tabela 2.3 - Pseudocódigo do Recozimento Simulado.

Pseudocódigo
Elaboração das soluções candidatas
Definição dos valores dos parâmetros (t, r e k)
Selecionar uma solução inicial
Laço de repetição
Repita de 1 até k
Escolha aleatória de uma solução candidata vizinha (x^*)
Comparação da solução atual com a nova
$\Delta = (solução\ atual) x^* - (nova\ solução) x$
Se $\Delta \leq 0$ então faça
$x = x^*$
Senão se $probabilidade > \exp\left(\frac{-\Delta}{t\ (temperatura)}\right)$ faça
$x = x^*$
Reduza a temperatura: $t = t * r$
Crítérios de parada: Se atender ao critério de para estabelecido, PARAR, caso contrário repita o laço de repetição.

Fonte: Adaptado de BUENO (2009).

Onde:

t : Temperatura inicial;

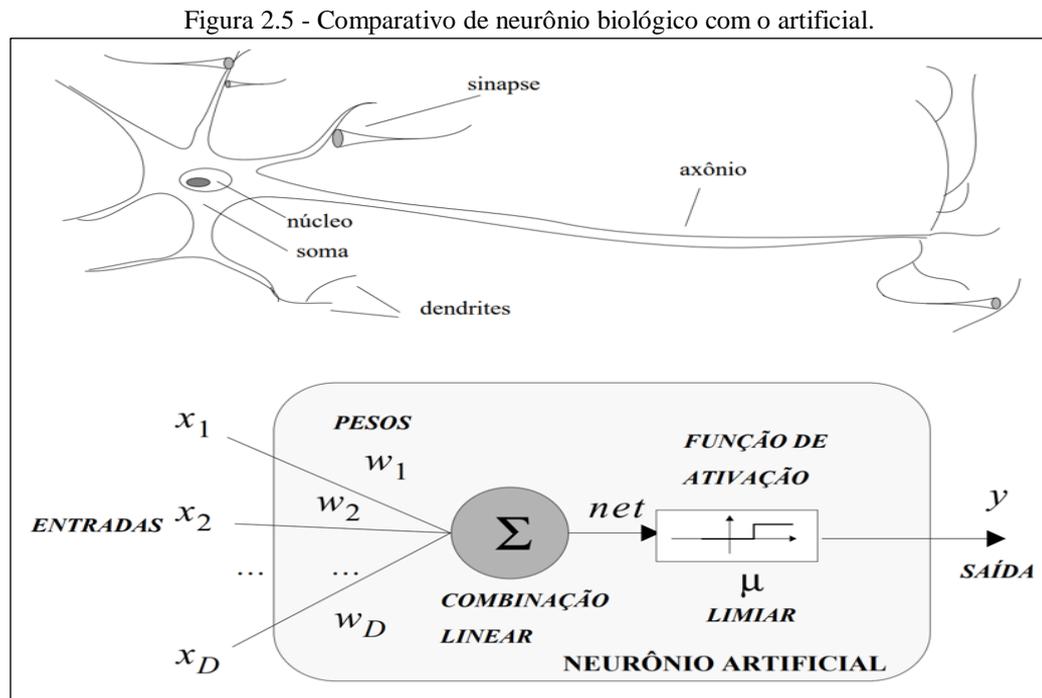
r : Fator de resfriamento, responsável pela velocidade de resfriamento do algoritmo (convergência até a solução ótima).

k : Número de iterações do algoritmo.

O Recozimento Simulado pode ser aplicado tanto a um problema de minimização quanto a problemas de maximização. Vale ressaltar que para o algoritmo RS o valor do parâmetro r é de extrema importância, pois se ele for muito baixo (próximo de 0), o resfriamento ocorrerá muito rápido, que resultara numa solução máxima local, caso seja muito alto (próximo de 1) haverá muitas iterações com soluções não tão satisfatórias (JÚNIOR, 2018).

2.4 REDES NEURAIS ARTIFICIAIS

São técnicas computacionais que apresentam um modelo matemático inspirado na estrutura neural humana, por meio de procedimentos matemáticos é possível guardar informações e gerar conhecimento (DADA *et. al.*, 2021). O comportamento de uma RNA vem das interações entre as unidades de processamento da rede e pode ter sua arquitetura comparada a um neurônio biológico, conforme a Figura 2.7.



Fonte: Adaptado de RAUBER (2005).

A RNA é formada por sinapses que passam por um processo de adaptação por meio de pesos, estes que podem assumir valores tanto positivo quanto negativo. A combinação linear tem a finalidade de somar os valores obtidos pelos pesos e a função de ativação limita a

amplitude dos valores de saída dos neurônios que estão entre 0 e 1 ou -1 e 1 (MARIO, DA SILVA FILHO e ABE, 2021).

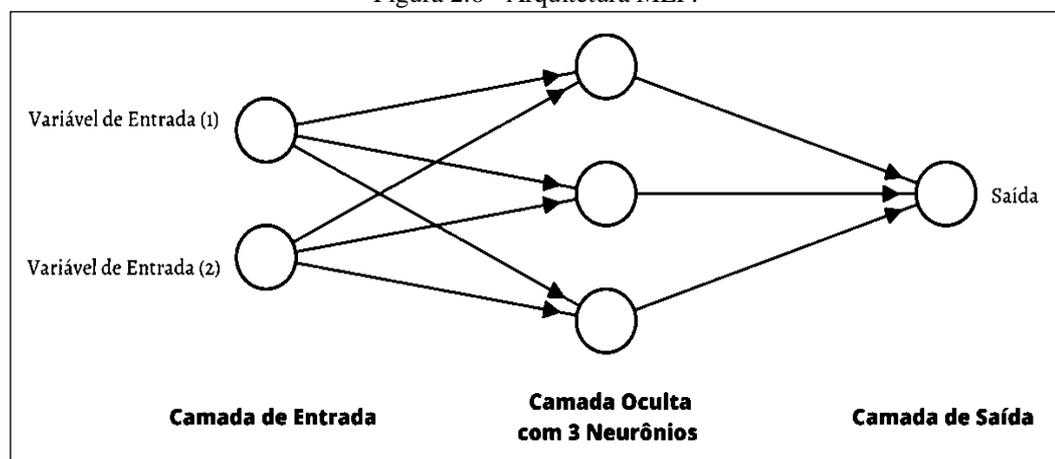
A RNA é classificada como um algoritmo de aprendizagem supervisionada que utiliza o treinamento baseado em dados históricos, afim de prever a um determinado período a frente, no qual se é utilizado variáveis de entrada e de saída para a previsão. O modelo mais simplório de RNA é conhecido como *perceptron* e foi proposto em 1958 por Frank Roseblatt, no qual a primeira aplicação do algoritmo ocorreu para o reconhecimento de imagens (GUAZZELLI, 1991; MARIO, DA SILVA FILHO e ABE, 2021).

A base do modelo de RNA apresentado na Figura 2.6 foi proposto por McCulloch e Pitts (1943), no qual contém uma saída que é resultante do somatório realizado pela combinação linear, que são influenciadas diretamente pelos pesos contidos nas sinapses (MCCULLOCH e PITTS, 1943). A partir daí surgiu outras variações de RNA, sendo as mais conhecidas o modelo *Perceptron* e o modelo *Multilayer Perceptron (MLP)*, sendo o primeiro constituído em 1958 por Rosenblatt, em que foi utilizado para a classificação binária (ROSENBLATT, 1958; BOUZON, 2021).

O modelo de MLP foi desenvolvido por Rumelhart, Hinton e Williams (1986), que trabalha tanto com classificação binária como também as não binárias (RUMELHART, HINTON e WILLIAMS, 1986; BOUZON, 2021).

A Figura 2.8 apresenta o modelo MLP, que diferentemente do modelo *perceptron*, possui uma camada oculta com mais de um neurônio.

Figura 2.6 - Arquitetura MLP.



Fonte: Adaptado de BOUZON (2021).

Conforme pode ser observado na Figura 2.8, a rede neural MLP faz parte da classe *feedforward*, que basicamente se propagam apenas em uma direção entre os nós da rede, Estas RNAs são constituídas de uma camada de entrada, com a possibilidade haver n camadas

ocultas, em que cada camada oculta pode conter x neurônios e uma camada de saída (RUMELHART, HINTON e WILLIAMS, 1986).

Contudo, para se obter o melhor desempenho nos processos de classificação, reconhecimento de padrão e predição é necessário que haja as configurações necessárias de arquitetura da RNA, tal como a definição da função de ativação, quantidade de camadas, quantidade de neurônios, taxa de aprendizado, quantidade de iterações e escolha do algoritmo de treinamento (DADA *et. al.*, 2021). A tabela 2 apresenta algumas funções de ativação utilizadas na configuração de camadas de neurônios.

Tabela 2.4 - Funções de Ativação.

Sigla	Função	Expressão
Sigmoide	Sigmoide	$\sigma = \frac{1}{1 + e^x}$
TanH	Tangente Hiperbólica	$\tanh(x) = 2\sigma(2x) - 1$ $\tanh'(x) = 1 - \tanh^2(x)$
ReLU	Unidade Linear Retificada	$ReLU(x) = \max\{0, x\}$, sendo $\begin{cases} 1, se x \geq 0 \\ 0, c. c \end{cases}$
ELU	Unidade Linear Exponencial	$ELU(x, \alpha) = \begin{cases} x, se x \geq 0 \\ \alpha(e^x - 1), se c. c \end{cases}$ $ELU'(x, \alpha) = \begin{cases} 1, se x \geq 0 \\ ELU(x, \alpha) + \alpha, se c. c \end{cases}$
Leaky ReLU	Unidade Linear Retificada com Vazamento	$LeakyReLU(x, \alpha) = \max\{\alpha x, x\}$ $LeakyReLU'(x, \alpha) = \begin{cases} 1, se x \geq 0 \\ \alpha, se c. c \end{cases}$

Fonte: Adaptado de KOÇAK e ÜSTÜNDAĞ ŞIRAY (2021).

Além disso é necessário definir a arquitetura da rede configurando a quantidade de camadas escondidas a serem utilizadas e quantidade de neurônios, com isso a necessidade de variação vai de acordo com o estado de convergência da simulação que pode ser analisado pelo Erro Médio Quadrático MSE (KOÇAK e ÜSTÜNDAĞ ŞIRAY, 2021).

Por fim a definição do algoritmo de treinamento que depende da formalidade do problema a ser resolvido, definição do conjunto de dados para treinamento da rede, bem como os dados alvo da simulação (BOOB, DEY e LAN, 2020). Em suma a escolha do algoritmo permite obter uma precisão na aproximação de funções no estado de treino e um tempo de convergência menor garantido uma acuracidade otimizada no aprendizado.

Para Mohammadi *et. al.* (2020) dentre os algoritmos de treinamento existentes se tem: Levenberg-Marquardt, Bayesian Regularization, Broyden-Fletcher-Goldfarb-Shanno Quasi-Newton, Resilient Backpropagation, Scaled Conjugate Gradient, Conjugate Gradient with Powell/Beale Restarts, Fletcher-Powell Conjugate Gradient, Polak-Ribière Conjugate Gradient, One Step Secant e Gradient Descent.

CAPÍTULO 3

3 MATERIAIS E MÉTODOS

Neste capítulo são apresentados os materiais necessários para o desenvolvimento da pesquisa, tendo em vista os elementos essenciais para aquisição dos modelos de RNA e simulações com Algoritmos de Otimização, os métodos utilizados para a solução de cada etapa consolidada, como aquisição de dados, modelagem de variáveis, treinamento das redes neurais e simulações com três algoritmos de otimização.

3.1 MATERIAIS

3.1.1 Registro das Embarcações

Os registros de embarcações são documentos com características específicas relativos à embarcação selecionada para o estudo, são apresentadas algumas informações referentes às 10 embarcações que realizam o transporte de cargas e passageiros no Porto Fluvial de Manaus na próxima subseção.

Tabela 3.1 - Características das Embarcações.

Nº	Nome	Comprimento	Lotação	Casco	Empresa
1	Fábio Júnior VI	16m	50	Ferro	GRAFA
2	Comandante Fabrício I	14,5m	40	Madeira	GRAFA
3	Reflexo	16,5m	55	Ferro	GRAFA
4	Arimatéia	17m	55	Madeira	GRAFA
5	Comandante Becil	15m	50	Madeira	Terminal AJATO
6	Expresso Moraes II	16m	55	Madeira	Terminal AJATO
7	Ajato IV	16m	55	Madeira	Terminal AJATO
8	Gisa	15m	15	Alumínio	Terminal AJATO
9	PUMA I	18m	60	Alumínio	Terminal AJATO
10	Vigiane II	14m	30	Alumínio	Terminal AJATO

Fonte: AUTOR (2021).

3.1.2 Características de Máquinas

As características das máquinas foram adquiridas por meio de documentos de especificação técnica referentes ao fabricante de cada máquina, outros fornecidos por especialistas que trabalham diariamente com as embarcações, algumas características são elencadas nesta seção.

Tabela 3.2 - Características de Máquinas das Embarcações.

Nº	Marca	Modelo	Potência	Núm. Cilindros	Peso	Tipo de pistão	Capacidade de óleo
1	SCANIA	Propulsão	315hp	5	950kg	Alumínio	31 a 36dm ³
2	YANMAR	Propulsão	115hp	N/O	N/O	N/O	N/O
3	SCANIA	Propulsão	367hp	N/O	N/O	N/O	N/O
4	SCANIA	Propulsão	315hp	5	950kg	Alumínio	31 a 36dm ³
5	SCANIA	Propulsão	367hp	N/O	N/O	N/O	N/O
6	SCANIA	Propulsão	367hp	N/O	N/O	N/O	N/O
7	YANMAR	Propulsão	250hp	4	334kg	N/O	N/O
8	YANMAR	Propulsão	250hp	4	334kg	N/O	N/O
9	SCANIA	Propulsão	600hp	N/O	N/O	N/O	N/O
10	YANMAR	Propulsão	250hp	N/O	N/O	N/O	N/O

Fonte: AUTOR (2021).

3.1.3 Variáveis Utilizadas no Modelo de RNA

São as variáveis relativas à especificação técnica da embarcação com o uso de máquina de combustão interna. O Quadro 3.1 apresenta uma relação das variáveis de entrada e saída providas e configuradas para o modelo de RNA.

Quadro 3.1 - Relação entre as variáveis de entrada e saída.

Característica	Variável
Entrada	Distância em Milhas Náuticas (mni)
Entrada	Velocidade em quilômetros por hora (km/h)
Saída	Consumo em litros (L)

Fonte: AUTOR (2021).

3.1.4 Modelo de Análise Estatística

Para as análises estatísticas acerca dos erros encontrados em cada treinamento de possibilidades das Redes Neurais Artificiais foram utilizados 4 modelos matemáticos, expressos pelas Eq. (3.1), (3.2), (3.3) e (3.4).

$$MSE = \frac{\sum_{i=1}^n (ra_i - rs_i)^2}{n} \quad (3.1)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (ra_i - rs_i)^2}{n}} \quad (3.2)$$

$$NRMSE = \frac{RMSE}{\bar{O}} \quad (3.3)$$

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{ra_i - rs_i}{ra_i} \right| \quad (3.4)$$

Onde:

MSE = Erro Médio Quadrático;

$RMSE$ = Raíz do Erro Médio Quadrático;

$NMSE$ = Erro da Raíz Quadrática Média Normalizada;

$MAPE$ = Erro Percentual Médio Absoluto;

n = número de elementos do vetor de saída;

ra = resultado alvo;
rs = resultado simulado.

3.1.5 Algoritmos de Treinamento da Rede Neural

Os algoritmos de treinamento da RNA tiveram contribuição no ato de escolher o melhor estado de convergência da rede com uma taxa mínima de erro, considerando alguns parâmetros como: quantidade de neurônios, camadas, tipo de algoritmo e função de transferência. Estes que são apresentados a seguir:

- *Levenberg-Marquardt*,
- *Bayesian Regularization*,
- *Broyden-Fletcher-Goldfarb-Shanno Quasi-Newton*,
- *Resilient Backpropagation*,
- *Scaled Conjugate Gradient*,
- *Conjugate Gradient with Powell/Beale Restarts*,
- *Fletcher-Powell Conjugate Gradient*,
- *Polak-Ribière Conjugate Gradient*,
- *One Step Secant*,
- *Variable Learning Rate Gradient Descent*,
- *Gradient Descent with Momentum*;
- *Gradient Descent*.

3.1.6 Recursos da Máquina de Teste

Os recursos são provenientes da capacidade do hardware de uma máquina executar certos procedimentos computacionais como: processamento de dados e iterações de complexidade computacional elevada, portanto o uso desses recursos é necessário, dentre os quais se destaca: sem uso de recurso (Normal), uso de recurso (Somente Paralelo), uso de recurso (Paralelo e Unidade de Processamento Gráfico - GPU). O Quadro 3.2 apresenta uma relação das configurações da máquina simuladora.

Quadro 3.2 - Configurações da máquina de testes.

Configuração	Valor
<i>Random Access Memory</i> - RAM	16GB
Processador	I5 décima geração
Memória Cache L3	8MB
Tecnologia de armazenamento	<i>Solid State Drives</i> - SSD
Memória Secundária	500GB

Fonte: AUTOR (2021).

3.1.7 Parâmetros dos Algoritmos de Otimização

Os algoritmos utilizados para a otimização do problema são de suma importância, devido ao uso de recursos computacionais, é válido desenvolver mecanismos que avaliem o que obteve o menor tempo de resultado, ou o que se aproximou do mínimo local. Tendo em vista isso, o Quadro 3.3 apresenta os algoritmos e parâmetros utilizados.

Quadro 3.3 - Algoritmos e parâmetros utilizados.

Algoritmo	Método	Parâmetros
<i>Genetic Algorithm</i>	Otimização	Número de variáveis: 1 Limites inferiores: 30.868902 Limites superiores: 81.030869 Tamanho da população: 50 Máximo de gerações: 100 Gerações de elite: 20 Paralelismo: usados true e false
<i>Simulated Annealing</i>	Otimização	Valor inicial: 30.868902 Limites inferiores: 30.868902 Limites superiores: 81.030869
<i>Particle Swarm Optimization</i>	Evolucionário	Número de variáveis: 1 Limites inferiores: 30.868902 Limites superiores: 81.030869 Tamanho de partículas: 50 Paralelismo: usados true e false

Fonte: AUTOR (2021).

3.1.8 Ambiente de Desenvolvimento

Para o ambiente de desenvolvimento e simulações foi necessário o uso do *software MatLab* 2016a, esta ferramenta permite o uso de métodos, bibliotecas e funções previamente preparadas para aplicação em diversos problemas. Com esta ferramenta foi possível realizar o processamento de dados, as simulações e geração de resultados com gráficos e tabelas.

3.2 MÉTODOS

Para a realização desta pesquisa foi necessário realizar um estudo acerca do assunto, para isso, foram feitas visitas *in loco* e entrevistas com especialistas, dessa forma foi possível averiguar a viabilidade do estudo por meio da concessão de registros de viagens relativos ao Porto Fluvial de Manaus, onde o fluxo de embarcações que realizam o transporte fluvial é frequente, de acordo com as entrevistas as embarcações transportam cargas e passageiros para diversas localidades da região Norte do estado do Amazonas – Brasil. Os dados coletados de registros documentais e experiências dos proprietários e marítimos são de 2018 a 2019.

3.2.1 Aquisição e Coleta de Dados

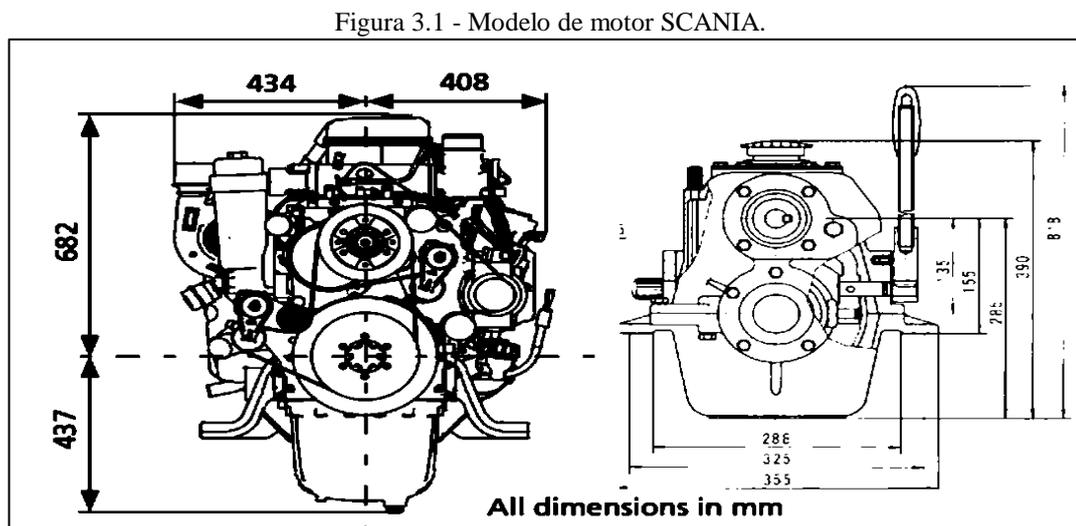
As entrevistas e visitas in loco serviram como métodos eficazes para coletar dados específicos e relativos à realidade do Porto Fluvial de Manaus ao considerar embarcações de pequeno e médio porte, ou seja, com um comprimento de 10 a 18m e capacidade de lotação entre 20 a 60 passageiros. Contudo, esta etapa teve como propósito levantar os elementos essenciais de apoio à pesquisa como: características de 14 embarcações e as características de seus motores.

3.2.2 Seleção do Objeto de Estudo da Pesquisa

Esta etapa consiste na escolha da embarcação que se adapta aos parâmetros da pesquisa como: especificações de máquinas com dados técnicos adquiridos em sites do fabricante do modelo, tamanho mínimo de 15m que ofereça uma estabilidade no consumo médio de combustível, capacidade de lotação mínima de 50 passageiros que permita ter uma variação no peso, tendo em vista o material do casco da embarcação, por fim o uso de máquina de combustão interna marinizada.

É válido ressaltar que mediante os parâmetros, a embarcação de número 1 foi a que atendeu aos requisitos, considerando que o modelo de motor é SCANIA e o site do fabricante disponibiliza material técnico de forma gratuita.

A Figura 3.1 ilustra um exemplo do modelo de máquina que é objeto de estudo da pesquisa, além da embarcação selecionada e suas características, é possível identificar no lado esquerdo um motor da fabricante SCANIA com dimensões em milímetros, no lado direito um modelo de reversor da fabricante ZF com dimensões em milímetros também.



Fonte: Adaptado de SCANIA (2020); ZF (2020).

3.2.3 Seleção de Variáveis Significativas para o Modelo de Treinamento com RNA

Pensando nos requisitos necessários para implementar um modelo computacional que permita a otimização do consumo de combustível foi analisado e escolhido as seguintes variáveis:

- Milhas náuticas;
- Distância em km;
- Tempo;
- Consumo em Litros

Para isso foi realizado uma análise temporal de 2018 a 2019 com registros de viagens e experiências dos especialistas, estas que foram adquiridas ao longo das entrevistas e visitas, o que permitiu popular uma base de dados generalizada para o estudo, somente depois de entender os processos logísticos da embarcação foi necessário realizar alguns procedimentos de processamento desses dados, e por fim utilizar no modelo de treinamento da RNA.

A Figura 3.2 ilustra uma amostra de dados utilizada para o processo de treinamento da Rede Neural, onde é considerado os parâmetros citados acima, a leitura desses dados pelo algoritmo é feita em 2 partes, primeiro a leitura do cabeçalho, a leitura do conjunto todo.

Contudo, não se limita a isso, o algoritmo é flexível na interpretação e leitura dos dados, onde é considerado a decisão do usuário em escolher ou determinar quais colunas (variáveis) ele utilizará.

Figura 3.2 - Amostra de dados para treinamento da Rede Neural.

CABEÇALHO		Distancia (NM)	Distancia (KM)	Velocidade (KM/H)	Tempo (H)	Consumo (L)	CONJUNTO TODO
VALORES		84,23326134	156	50,43206936	1,670232104	87,5135668	
		285,0971922	528	68,76048018	4,146236202	230,7206883	
		69,11447084	128	74,47122713	0,92806945	52,79959768	
		7,559395248	14	53,21027058	0,142066469	7,499554027	
		130,1295896	241	69,84089177	1,863229211	103,9660234	
		315,87473	585	62,12366616	5,084611865	277,6020708	
		226,2419006	419	49,89186357	4,53464522	237,0827501	
		144,1684665	267	53,55754573	2,69184229	142,2534181	
		54,53563715	101	43,10070503	1,26530731	55,98426111	
		4,319654428	8	63,20407774	0,068344553	3,74706163	
		134,4492441	249	31,40910823	4,280581386	107,5266594	
		111,7710583	207	47,07507622	2,374314973	120,4440953	
		132,8293737	246	43,10070503	3,081837607	136,3577053	
		307,2354212	569	33,26124238	9,237039846	256,0723979	
	173,3261339	321	32,72103659	5,297085667	143,3499196		

Fonte: AUTOR (2021).

3.2.4 Ajuste da Curva de SFOC e PW pelo Método do Coeficiente Angular

Durante o processo de geração dos gráficos em função das variáveis utilizadas foi necessário encontrar à medida que caracteriza a declividade entre dois pontos do eixo das

abscissas. As operações para determinar a inclinação da reta são feitas através dos pontos que delimitam a variação entre os eixos das coordenadas x e y, ou seja, a expressão que melhor explica isso é apresentado pela Eq. (3.5):

$$m = \frac{Y_B - Y_A}{X_B - X_A} = \frac{\Delta_y}{\Delta_x} \quad (3.5)$$

Onde:

m = coeficiente angular;

Y = coordenadas do eixo y;

X = coordenadas do eixo x;

Δ_y = variação em y;

Δ_x = variação em x

Para encontrar o ponto de encontro C em função do eixo x, onde um ponto de encontro A com coordenadas (x_A, y_A) e um ponto B de coordenadas (x_B, y_B) são fornecidos, foi necessário utilizar da Eq. (3.6).

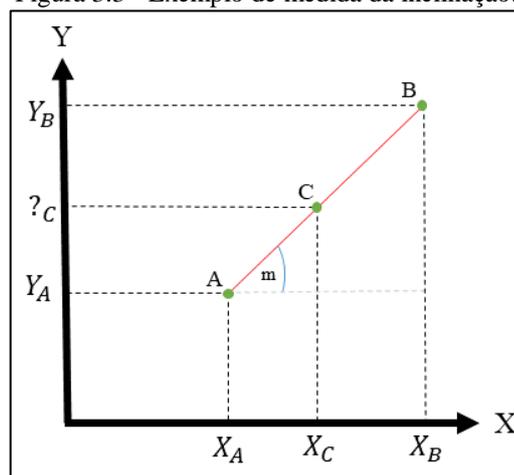
$$Y_C = (m * (X_C - X_A)) + Y_A \quad (3.6)$$

Onde:

Y_C = coordenada do eixo y;

A Figura 3.3 ilustra um exemplo de medida da inclinação de uma reta em relação ao eixo horizontal, onde m representa a taxa de declividade e a coordenada do eixo y no ponto C é o alvo a ser encontrado.

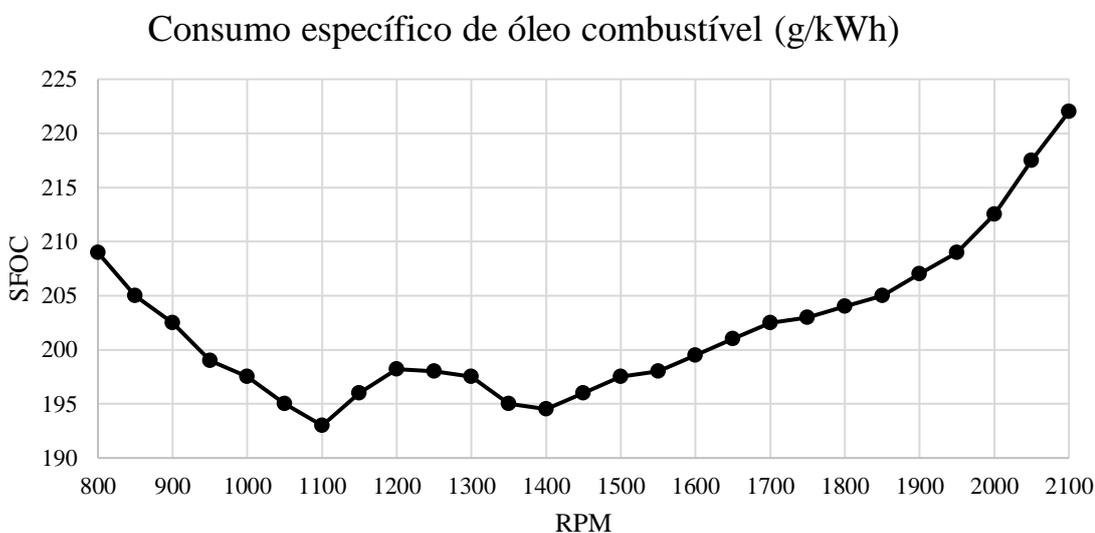
Figura 3.3 - Exemplo de medida da inclinação.



Fonte: Adaptado de SOARES (2019).

O Gráfico 3.1 mostra a curva do consumo específico de óleo combustível (SFOC) em g/kWh do motor SCANIA em função das Rotações por Minuto (RPM), por meio deste indicador, foi possível traçar outros pontos para estimar e determinar novos consumos, com isso a variável consumo em litros foi utilizada como saída no modelo da rede neural.

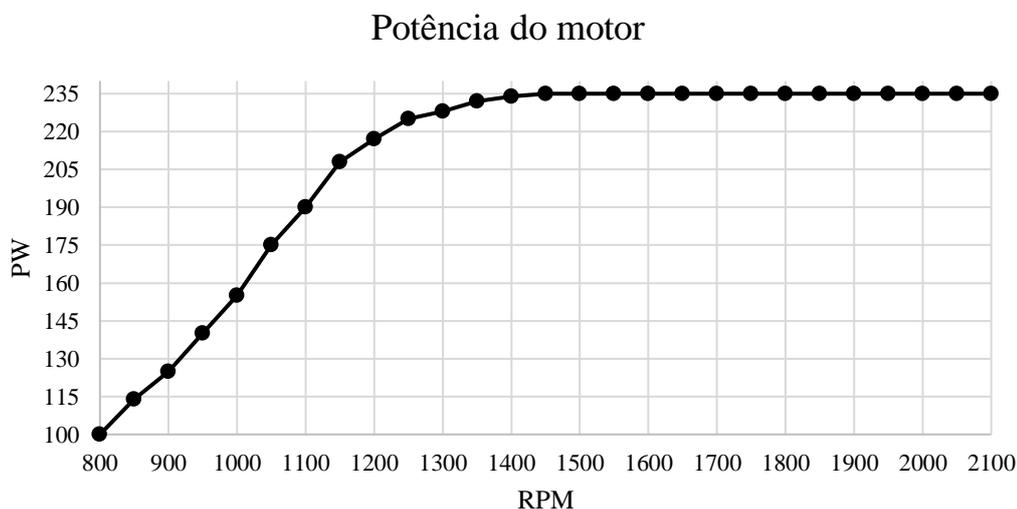
Gráfico 3.1 - Curva do SFOC.



Fonte: Adaptado de SCANIA (2021).

Considerando a curva do consumo específico, foi necessário analisar a curva de potência do motor (PW) em função do RPM, o Gráfico 3.2 mostra uma crescente evolução nas rotações por minuto em função da potência, ou seja, a velocidade angular do eixo principal da máquina tende a ser constante quando os pontos 220 e 230 do eixo Potência são alcançados, isso se refere a limitações físicas relacionadas à cambota e virabrequim.

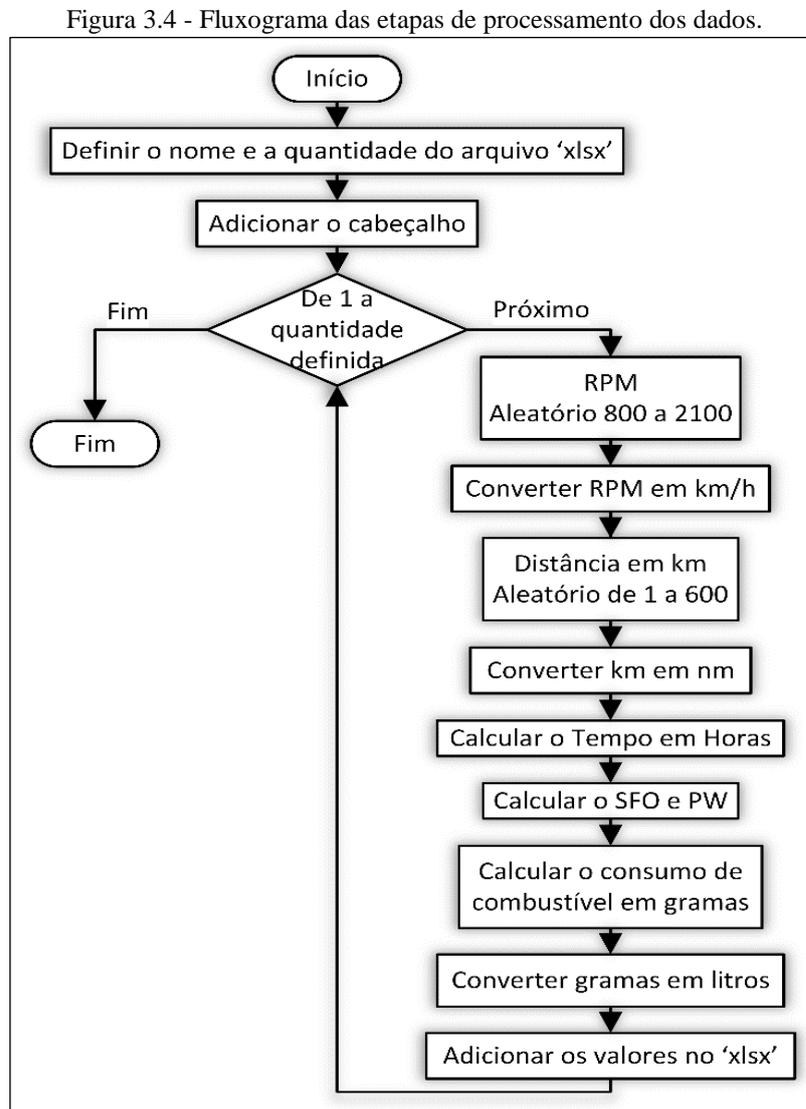
Gráfico 3.2 - Curva de potência do motor.



Fonte: Adaptado de SCANIA (2021).

3.2.5 Processamento da Base de Dados

Inicialmente são considerados alguns passos para o processamento desses dados e a sua utilização como dados experimentais para o algoritmo, a Figura 3.4 ilustra as etapas desse processo.



Fonte: AUTOR (2021).

Os procedimentos se resumem em calcular alguns parâmetros que servirão como dados de entrada do modelo de Rede Neural, onde são requisitos para a experimentação os seguintes passos:

- Geração de RPM aleatório entre 800 e 2100;
- Conversão de RPM em km/h;
- Conversão de km/h em milhas náuticas;
- Progressão de SFOC e PW;

- Consumo do combustível em gramas;
- Conversão de gramas em litro.

O primeiro passo consiste em gerar números aleatórios de RPM entre 800 e 2100, com esses valores é realizado a conversão para unidade em km/h. A Eq. (3.7) expressa o modelo de conversão.

$$V = \frac{RPM * PP * (1 - PS)}{GR * C} \quad (3.7)$$

Onde:

V = Velocidade em km/h;

RPM = Rotações por minuto;

PP = Propeller Pitch (polegadas);

PS = Propeller Slip (%);

GR = Gear Ratio;

C = valor constante 656 para converter polegadas por minuto de revolução para a velocidade do barco.

O próximo passo consiste em gerar um número aleatório para a distância em km e converter para milhas náuticas. A Eq. (3.8) expressa o modelo de conversão utilizado.

$$NM = \frac{km}{1.852} \quad (3.8)$$

Onde:

NM = distância em milhas náuticas;

km = distância em quilômetros.

O passo seguinte foi calcular o tempo em função da distância e velocidade, sendo expressa pela Eq. (3.9).

$$T = \frac{D}{V} \quad (3.9)$$

Onde:

T = tempo em horas;

D = distância em milhas náuticas;

V = velocidade em km/h

O passo seguinte é caracterizado pelo método do ajuste da curva por meio do coeficiente angular entre dois pontos da reta, ou seja, é calculado o SFOC e PW.

O próximo passo determina o consumo em gramas que é expressa pela Eq. (3.10).

$$CGR = PW * SFOC * T \quad (3.10)$$

Onde:

CGR = Consumo de combustível (gr);

PW = Potência do motor (kw)

SFOC = Consumo específico de óleo combustível (g/kwh);

T = tempo em horas

Mediante isso, é realizado a conversão dos valores encontrados em litro, sendo expresso pela Eq. (3.11).

$$CL = \frac{CGR}{860} \quad (3.11)$$

Onde:

CL = Consumo de combustível em litros;

CGR = Consumo de combustível em gramas;

O processo é repetido até atingir a quantidade de dados lidas pelo arquivo .xls, por fim, é gerado um novo arquivo com os dados processados.

3.2.6 Procedimentos do Modelo de Otimização

O modelo de otimização é dividido em 4 etapas, onde são:

- Leitura e organização dos dados;
- Configuração da Rede Neural;
- Configuração do algoritmo de otimização;
- Geração de Resultados.

A primeira etapa é onde são definidos dados que vão ser utilizados tanto na RNA quanto nos algoritmos de otimização;

A segunda consiste em determinar parâmetros da melhor configuração de RNA (quantidade de neurônios, camadas, funções de ativação e algoritmo de treinamento) e escolher qual modelo obteve a menor taxa de erro. Ainda nessa etapa, é realizado os procedimentos de treinamento da rede neural utilizando 12 algoritmos de treinamento, alternando as possibilidades com 3 funções de ativação: Sigmoide, Tangente Hiperbólica e Linear.

A terceira etapa consiste em avaliar 3 cenários de simulação, sendo eles:

- Manaus à Itacoatiara;
- Manaus à Parintins;
- Manaus à Barcelos

Dessa forma é possível determinar os parâmetros para o menor consumo de combustível para a embarcação e o motor objeto de estudo, de tal forma que a avaliação das configurações do Algoritmo Genético, Recozimento Simulado e Enxame de Partículas sejam específicas considerando trajetos com velocidades, distâncias e tempo diferentes.

A última etapa é caracterizada pela geração de resultados, ou seja, gráficos com duas variáveis e três variáveis, indicadores, análises estatísticas e planilhas com os dados referentes aos treinamentos da rede neural e os modelos de otimização.

3.2.7 Cenários de Simulação

Em posse das melhores arquiteturas e configurações de redes neurais foi realizado o procedimento de avaliação e parametrização dos algoritmos de otimização para determinar o menor consumo de combustível com base na velocidade e distância do trajeto ao considerar os 3 cenários já citados.

Os dados da distância foram coletados da plataforma Rome2Rio que tem como finalidade o planejamento antecipado de viagens para qualquer local do mundo de forma segura e fácil, tendo como opções de transporte: voo, trem, ônibus, embarcação ou carro.

A Figura 3.5 ilustra o trajeto que deve ser realizado por uma embarcação saindo de **Manaus à Itacoatiara**, onde a distância é dada por 195,30 km e 105,46 milhas náuticas.

CAPÍTULO 4

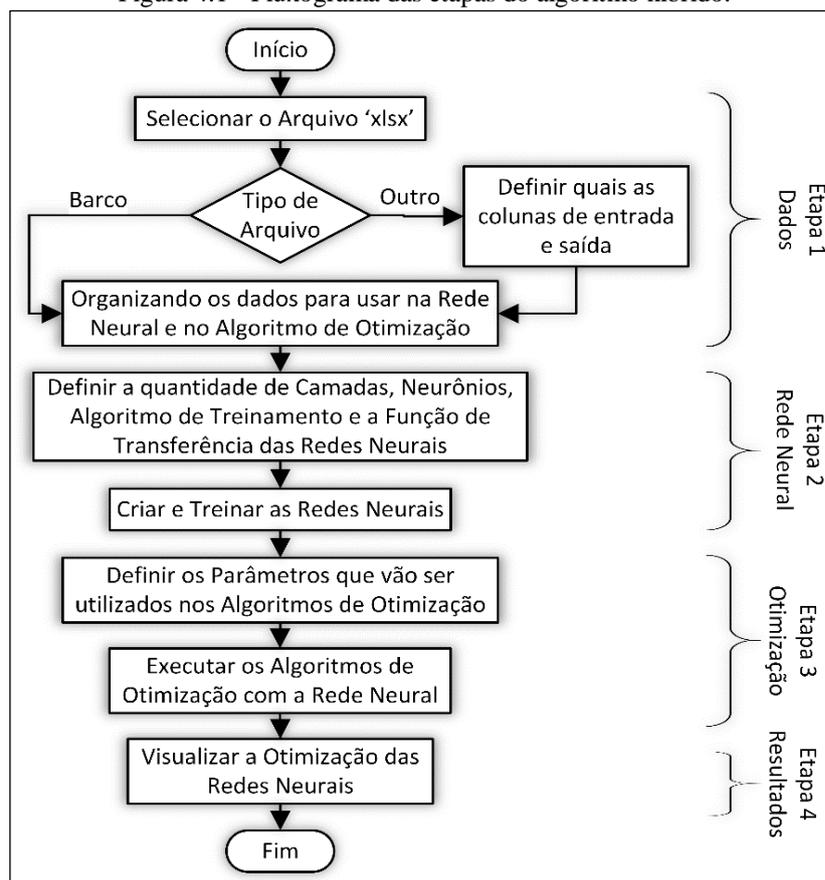
4 RESULTADOS E DISCUSSÕES

Nesta seção são apresentados os resultados acerca do modelo de otimização utilizando meta-heurísticas da computação baseado no aprendizado de máquina por meio das técnicas de Redes Neurais Artificiais. As subseções abordam sobre o modelo desenvolvido que é objetivo geral da dissertação, resultados acerca das melhores configurações (quantidade de neurônios, camadas, funções de ativação e algoritmo de treinamento) de RNA, resultados acerca da análise dos algoritmos de otimização considerando parâmetros, técnicas, característica do mesmo e população, por fim, os resultados acerca das simulações.

4.1 MODELO DE OTIMIZAÇÃO BASEADO NO APRENDIZADO DE MÁQUINA

Conforme já mencionado anteriormente na seção de métodos do capítulo 4, o modelo de otimização contempla alguns passos característicos do algoritmo que implementa os modelos de otimização e previsão, sendo eles representados e ilustrados pela Figura 4.1.

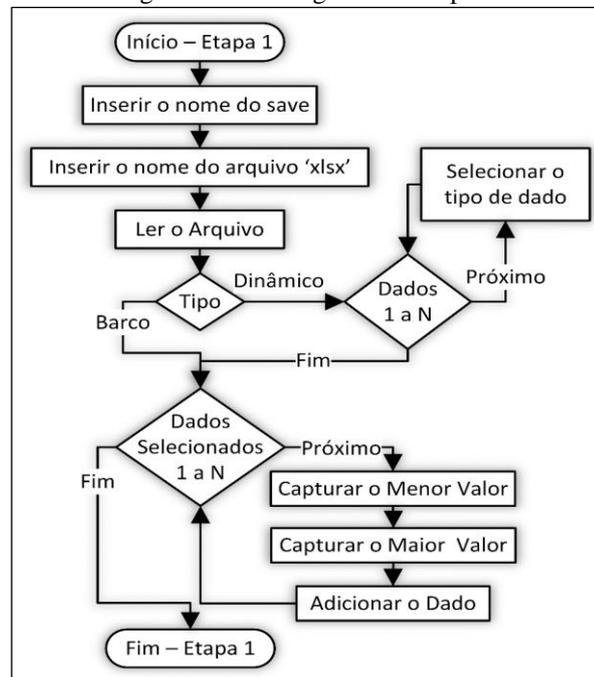
Figura 4.1 - Fluxograma das etapas do algoritmo híbrido.



Fonte: AUTOR (2021).

Etapa 1: O algoritmo é inicializado, o primeiro passo consiste em configurar os parâmetros de leitura da base de dados que serão utilizados no modelo de RNA e no algoritmo de otimização. Esta primeira etapa pode ser esclarecida pela Figura 4.2.

Figura 4.2 - Fluxograma da etapa 1.

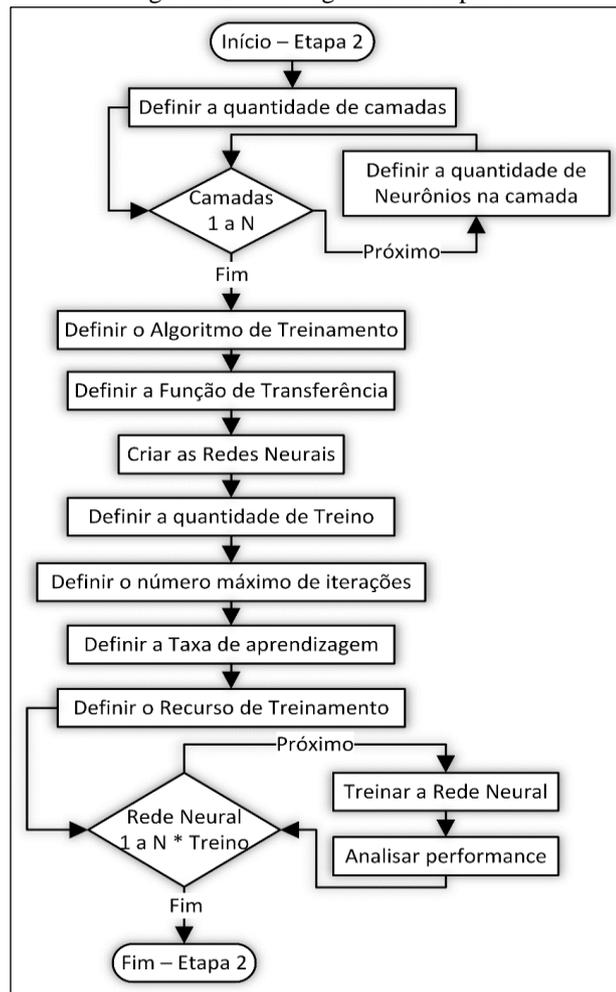


Fonte: AUTOR (2021).

- **Passo 1.1:** Inserir o nome do arquivo que será salvo com uma nova base de dados processada;
- **Passo 1.2:** Insere o nome do arquivo referente aos dados históricos da embarcação que será analisado e processado;
- **Passo 1.3:** O algoritmo determina se o histórico de dados é do tipo barco onde o histórico é previamente configurado com as colunas e tuplas que devem ser lidas ou do tipo dinâmico onde o usuário decide quais colunas devem ser lidas pelo algoritmo;
- **Passo 1.4:** O algoritmo realiza a leitura do arquivo mediante o seu tipo, após isso, ele encontra os mínimos e máximos referente a série histórica, estes dados servirão de parâmetros para o modelo de otimização;

Etapa 2: Nesta etapa o algoritmo realiza o procedimento de configuração, parametrização, treinamento, validação e análise de testes das melhores arquiteturas de RNA que serão utilizadas no modelo de otimização. Este processo pode ser melhor compreendido por meio da Figura 4.3.

Figura 4.3 - Fluxograma da etapa 2.



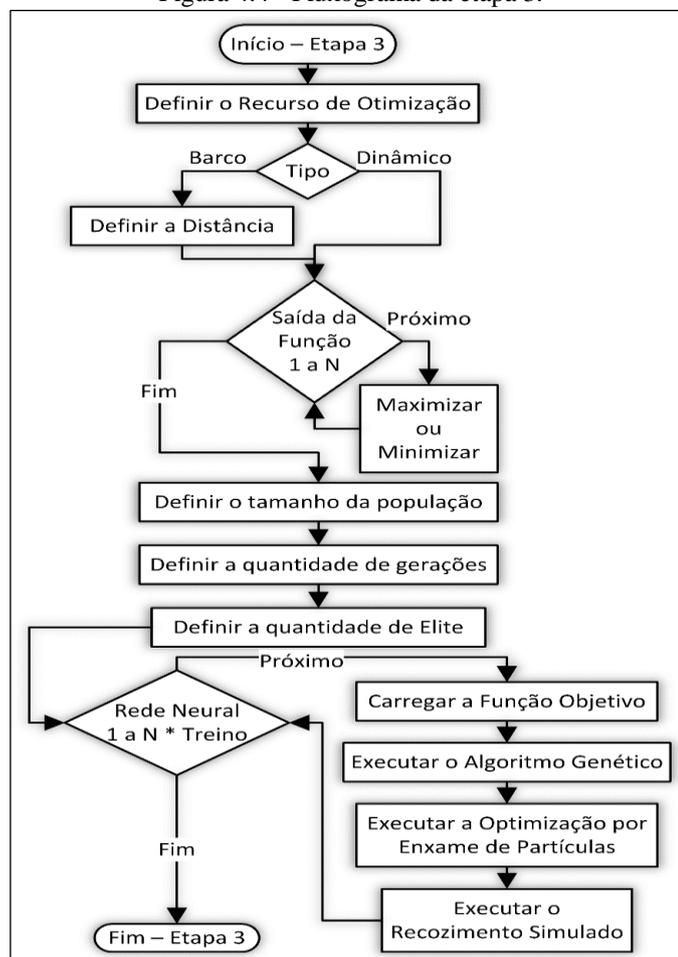
Fonte: AUTOR (2021).

- **Passo 2.1:** é definido o número de camadas ocultas que serão utilizadas no processo de treinamento da rede neural;
- **Passo 2.2:** o algoritmo entra em processo de iteração;
- **Passo 2.3:** é definido a quantidade de neurônios em cada camada oculta;
- **Passo 2.4:** é definido qual dos 12 algoritmos de treinamento será utilizado;
- **Passo 2.5:** é definido qual das 3 funções de transferências serão utilizados;
- **Passo 2.6:** é definido a quantidade de iterações que deverão ocorrer para treinar a rede e encontrar o estado ótimo local de convergência;
- **Passo 2.7:** é definido o número estipulado máximo de iterações;
- **Passo 2.8:** é definido a taxa de aprendizagem do algoritmo;
- **Passo 2.9:** é definido o recurso do treinamento, sendo ele: Sem recurso, Paralelo ou Paralelo com GPU;
- **Passo 2.10:** O algoritmo inicializa o processo de treinamento;

- **Passo 2.11:** As estruturas de dados referentes ao processo de treinamento da RNA são guardados e são analisados a performance com base na análise de erros;
- **Passo 2.12:** O algoritmo determina qual o melhor modelo de RNA para ser utilizado no modelo de otimização.

Etapa 3: Tendo as estruturas de dados referente ao melhor modelo de RNA, o algoritmo inicializa o processo de parametrização dos algoritmos de otimização utilizando o modelo de RNA como substituto da função objetiva, considerando que o modelo matemático ou melhor, o resultado disso é provido pelo modelo de previsão. Este processo pode ser representado e ilustrado por meio da Figura 4.4.

Figura 4.4 - Fluxograma da etapa 3.



Fonte: AUTOR (2021).

- **Passo 3.1:** é definido pelo usuário se será utilizado computação paralela ou não;
- **Passo 3.2:** o algoritmo solicita a parametrização da base de dados, se for do tipo “barco” será solicitado a distância da viagem, com a finalidade de configurar o modo em que o algoritmo interpretará a função objetiva, caso não seja do tipo “barco”, será “dinâmica”,

neste momento o algoritmo irá solicitar do usuário se o problema deve ser maximizado ou minimizado;

- **Passo 3.3:** Após isso, o algoritmo irá solicitar do usuário alguns parâmetros de definição na otimização como: tamanho da população ou de partículas e as gerações de elite do algoritmo genético;
- **Passo 3.4:** o algoritmo faz o carregamento da função objetiva;
- **Passo 3.5:** o algoritmo de otimização Genético é executado mediante a função objetiva;
- **Passo 3.6:** o algoritmo evolucionário Enxame de Partículas é executado mediante a função objetiva;
- **Passo 3.7:** o algoritmo de otimização por Recozimento Simulado é executado mediante a função objetiva;
- **Passo 3.8:** para cada iteração ou geração a função objetiva é chamada, de tal forma que o modelo matemático seja substituído pelo resultado da previsão;
- **Passo 3.9:** A etapa 3 é finalizada.

Etapa 4: A última etapa consiste nos procedimentos de geração dos resultados acerca dos treinamentos de cada modelo de RNA, resultados das otimizações com os 3 algoritmos testados e comparados, geração de gráficos 2D, 3D e planilhas com dados estatísticos.

4.2 Configuração Inicial do Modelo de RNA

Os testes foram realizados no laboratório de matrizes *MatLab® 2016a*, onde foi possível usufruir de funções e bibliotecas previamente prontas para aplicação em problemas ou algoritmos.

Inicialmente foi realizado um teste com 100 dados para se obter a quantidade de neurônios, recursos e a função de transferência que deve ser utilizado nos testes com 12 algoritmos de treinamento na Rede Neural, atendendo ao requisito de se ter a menor taxa de erro com base no MSE, RMSE, NRMSE e MAPE.

No primeiro teste se utilizou o algoritmo *Levenberg-Marquardt*, testado com um intervalo de neurônios que varia em cada camada oculta, na primeira camada oculta se utilizou de um intervalo entre 1 a 20 neurônios, já na segunda camada oculta foi utilizado um intervalo entre 1 a 40 neurônios.

Contudo, as possibilidades não se limitaram apenas nas variações de neurônios nas camadas ocultas, há também as possibilidades entre os recursos, sendo eles: sem recurso, paralelo ou paralelo com GPU. Além disso, há as possibilidades entre as funções de transferências que são 3: Sigmoide, Tangente hiperbólica e Linear. A Figura 4.5 ilustra um exemplo do total de possibilidades que foram adquiridos ao longo dos treinamentos utilizando os modelos de Redes Neurais Artificiais totalizando 7.200 redes neurais.

Figura 4.5 - Total de Redes Neurais testadas.

Camada 1	Camada 2	Recurso	Função
1	1	1	1
.	.	.	.
.	.	.	.
20	40	3	3

} 20 x 40 x 3 x 3 = 7.200 Redes Neurais

Recurso:

1. Normal
2. Paralelo
3. Paralelo e GPU

Função de Transferência:

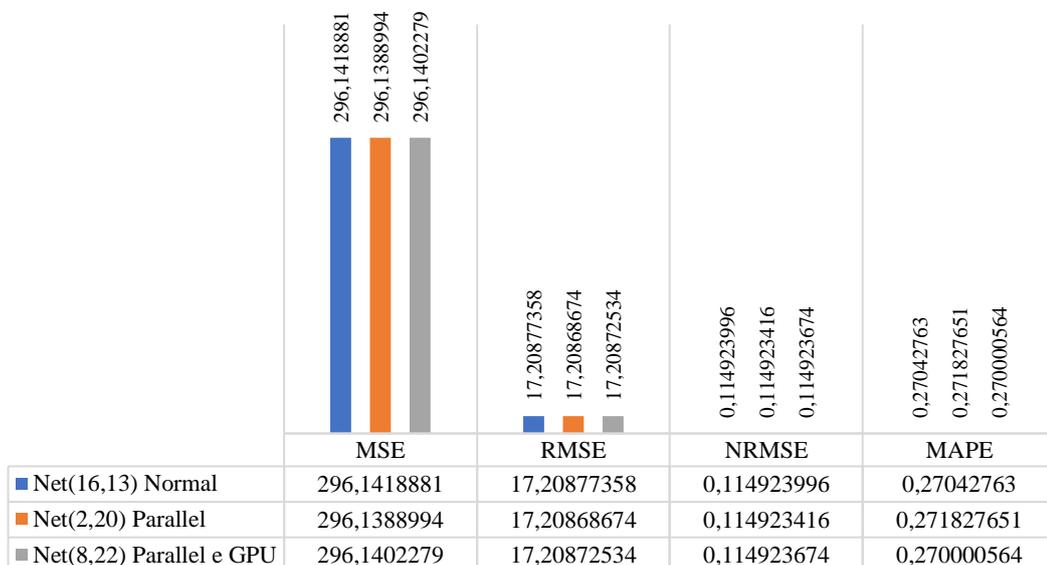
1. purelin
2. tansig
3. logsig

Fonte: AUTOR (2021).

4.2.1 Modelo com a Função Linear

O Gráfico 4.1 apresenta os resultados referentes as possibilidades citadas anteriormente, nele é possível identificar as colunas com MSE, RMSE, NRMSE e MAPE, separados pelos recursos utilizados. Este gráfico é referente à função de transferência Sigmoide.

Gráfico 4.1 - Levenberg-Marquardt com Sigmoide.



Fonte: AUTOR (2021).

A coluna que apresenta os resultados do MSE indica que o menor valor é obtido ao utilizar o modelo com recurso paralelo, caracterizada pelo uso da função Linear, com 2 neurônios na primeira camada oculta e 20 neurônios na segunda camada oculta.

A segunda coluna que apresenta os resultados do RMSE indica que o menor valor é obtido ao utilizar o modelo com recurso paralelo, caracterizada pelo uso da função Linear, com 2 neurônios na primeira camada oculta e 20 na segunda camada oculta.

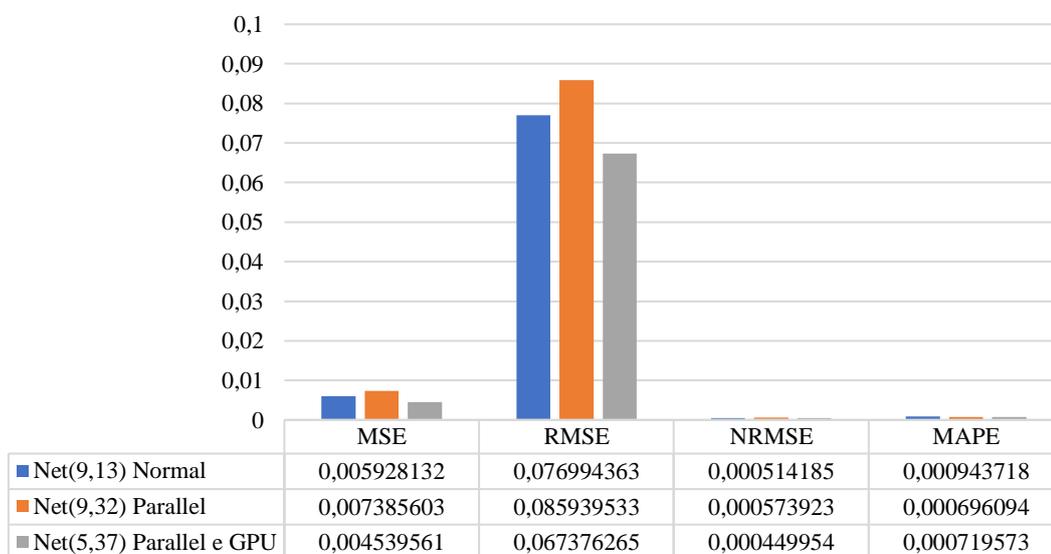
A terceira coluna que apresenta os resultados do NRMSE indica que o menor valor é obtido ao utilizar o modelo com recurso paralelo e GPU, caracterizada pelo uso da função Linear, com 8 neurônios na primeira camada oculta e 22 neurônios na segunda camada oculta.

A quarta e última coluna que apresenta os resultados do MAPE indica que o menor valor é obtido ao utilizar o modelo com recurso paralelo e GPU, caracterizada pelo uso da função Linear, com 8 neurônios na primeira camada oculta e 22 neurônios na segunda camada oculta.

4.2.2 Modelo com a Função Tangente Hiperbólica

O Gráfico 4.2 apresenta os resultados referentes as possibilidades com o uso da função Tangente hiperbólica nas camadas ocultas da rede neural, nele é possível identificar as colunas com MSE, RMSE, NRMSE e MAPE, separados pelos recursos utilizados.

Gráfico 4.2 - Resultados com a Tangente hiperbólica.



Fonte: AUTOR (2021).

O primeiro conjunto de colunas que apresenta a análise estatística com MSE, indica que o menor valor obtido ao utilizar o modelo com recurso paralelo e GPU, caracterizada pelo uso

da função tangente hiperbólica, com 5 neurônios na primeira camada oculta e 37 neurônios na segunda camada oculta.

O segundo conjunto de colunas que apresenta a análise estatística com RMSE, indica que o menor valor é obtido ao utilizar o modelo com recurso paralelo e GPU, caracterizada pelo uso da função tangente hiperbólica, com 5 neurônios na primeira camada oculta e 37 neurônios na segunda camada oculta.

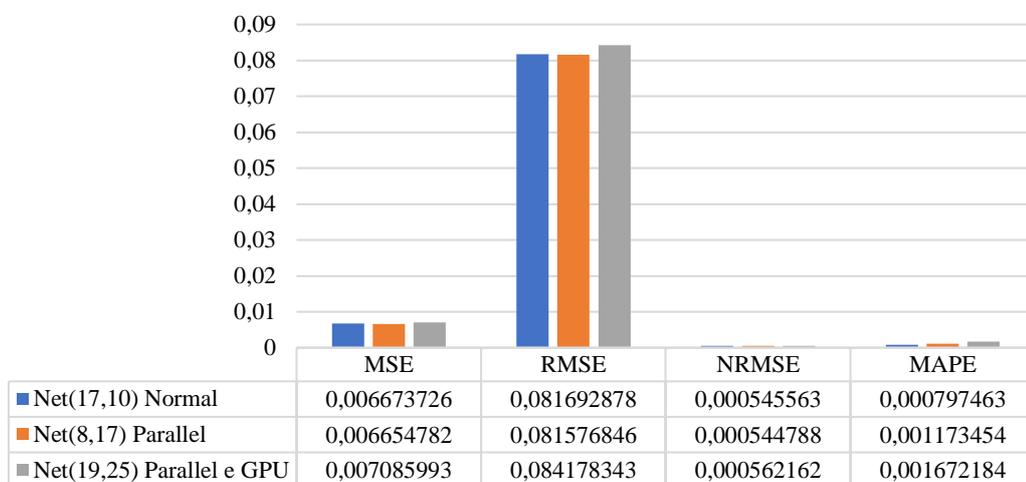
O terceiro conjunto de colunas que apresenta a análise estatística com NRMSE, indica que o menor valor é obtido ao utilizar o modelo com recurso paralelo e GPU, também caracterizada pelo uso da função tangente hiperbólica, projetando 5 neurônios na primeira camada oculta e 37 neurônios na segunda camada oculta.

O quarto e último conjunto de colunas que apresenta a análise estatística com MAPE, indica que o menor valor é obtido ao utilizar o modelo com recurso paralelo, projetando uma arquitetura com 9 neurônios na primeira camada oculta e 32 neurônios na segunda camada oculta.

4.2.3 Modelo com a Função Sigmoide

O Gráfico 4.3 apresenta os resultados referentes as possibilidades com o uso da função Linear nas camadas ocultas da rede neural, nele é possível identificar as colunas com MSE, RMSE, NRMSE e MAPE, separados pelos recursos utilizados.

Gráfico 4.3 - Resultados com a função Linear.



Fonte: AUTOR (2021).

O primeiro conjunto de colunas que apresenta a análise estatística com MSE, indica que o menor valor é obtido ao utilizar o modelo com recurso paralelo, projetando uma arquitetura com 8 neurônios na primeira camada oculta e 17 neurônios na segunda camada oculta.

O segundo conjunto de colunas que apresenta a análise estatística com RMSE, indica que o menor valor é obtido ao utilizar o modelo com recurso paralelo, projetando uma arquitetura com 8 neurônios na primeira camada oculta e 17 neurônios na segunda camada oculta.

O terceiro conjunto de colunas que apresenta a análise estatística com NRMSE, indica que o menor valor é obtido ao utilizar o modelo com recurso paralelo, projetando também uma arquitetura com 8 neurônios na primeira camada oculta e 17 neurônios na segunda camada oculta.

O quarto e último conjunto de colunas que apresenta a análise estatística com MAPE, indica que o menor valor é obtido ao utilizar o modelo sem recurso, projetando uma arquitetura com 17 neurônios na primeira camada oculta e 10 neurônios na segunda camada oculta.

4.2.4 Modelo Vencedor

De acordo com os resultados apresentados nos Gráficos 4.1, 4.2 e 4.3 o modelo com a função tangente hiperbólica utilizando o algoritmo *Levenberg-Marquardt*, projetando uma arquitetura com 5 neurônios na primeira camada oculta e 37 na segunda camada oculta, como recurso é utilizado a computação paralela e a GPU, este modelo obteve a menor taxa de erro e melhor performance para utilizar nos testes com 12 algoritmos de treinamento.

4.3 Treinamento dos Modelos de RNA com 12 Algoritmos de Aprendizado

Esta etapa consiste na execução do treinamento das redes neurais considerando 12 algoritmos de treinamento e as configurações do modelo vencedor na etapa anterior. Foram realizados 2 testes com: 500 e 10000 conjuntos de dados.

Isso foi necessário para verificar o comportamento e a eficiência do algoritmo ao avaliar conjuntos de dados maiores e confirmar quais os melhores resultados ao treinar os 12 algoritmos para cada modelo de rede neural.

4.3.1 Teste com 500 Dados

O primeiro teste a ser realizado foi utilizar 500 dados para treinar os 12 algoritmos de aprendizado, este treinamento utiliza as configurações validadas na subseção anterior, com 5 neurônios na primeira camada oculta e 37 na segunda camada, além de utilizar a função de transferência Tangente hiperbólica.

A Tabela 4.1 apresenta o resultado dos treinamentos realizados para cada algoritmo avaliado, onde a menor taxa de erro é alcançada pelo algoritmo Levenberg-Marquardt em um tempo estimado de 00:00:03 segundos, e taxas de erro MSE, RMSE, NRMSE e MAPE respectivamente 0.23, 0.48, 0.003 e 0.014.

Cada algoritmo obteve um valor diferente na avaliação estatística para determinar a menor taxa de erro, isso é explicado pelo fato de cada algoritmo possuir uma característica diferente de organizar o espaço amostral de dados no ato de buscar os melhores resultantes para a função objetiva, com isso ele determina pesos de forma randômica, afim de testar diferentes cenários de evolução ao longo das épocas.

Em virtude deste algoritmo ter usufruído de 72 iterações para avaliar o seu estado de convergência, ou seja, reduzir a curva de erro em função da explosão do gradiente, ele se destaca em modelagens de variáveis empíricas quando a correlação entre estas variáveis são maiores e desempenham maior influência na variável preditora, de tal forma que a evolução do aprendizado é maior que um algoritmo genético.

Tabela 4.1 - Resultados dos algoritmos de treinamento (Teste com 500 dados).

Item	Algoritmo de Treinamento	Iterações	Tempo	MSE	RMSE	NRMSE	MAPE
1	Levenberg-Marquardt	72	00:00:03	0,23	0,48	0,003	0,014
2	Regularização Bayesiana	1000	00:00:32	0,28	0,53	0,004	0,005
3	Polak-Ribière Conjugate Gradient	212	00:00:04	14,00	3,74	0,026	0,144
4	Fletcher-Powell Conjugate Gradient	159	00:00:03	22,53	4,75	0,033	0,185
5	Scaled Conjugate Gradient	110	00:00:03	67,99	8,25	0,057	0,353
6	Conjugate Gradient with Powell/Beale Restarts	56	00:00:03	112,16	10,59	0,073	0,608
7	Resilient Backpropagation	96	00:00:03	242,52	15,57	0,108	0,232
8	BFGS Quasi-Newton	1000	00:00:27	268,10	16,37	0,113	0,770
9	One Step Secant	65	00:00:03	350,65	18,73	0,130	0,512
10	Gradient Descent with Momentum	1000	00:00:05	78845,81	280,79	1,943	4,455
11	Variable Learning Rate Gradient Descent	1000	00:00:05	136624,85	369,63	2,558	18,921
12	Gradient Descent	1000	00:00:06	272370,83	521,89	3,611	13,932

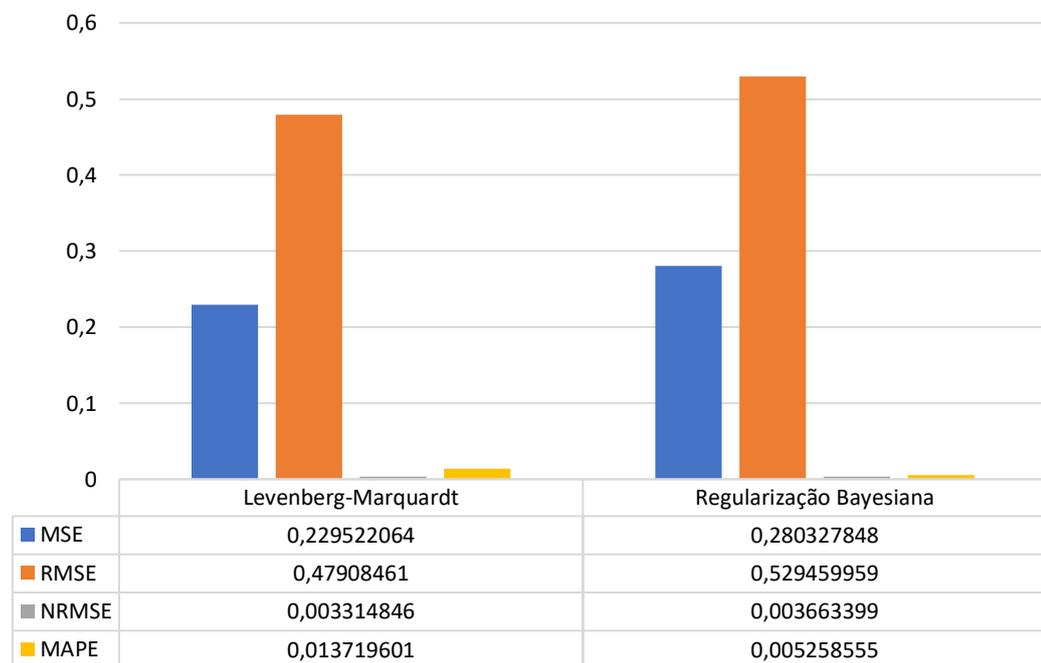
Fonte: AUTOR (2021).

De acordo com a Tabela 4.1 o segundo algoritmo que obteve a menor taxa de erro foi a Regularização Bayesiana onde foi possível alcançar um tempo estimado de 00:00:32 segundos em um intervalo de 1000 iterações, os erros MSE, RMSE, NRMSE e MAPE são respectivamente 0.28, 0.53, 0.004 e 0.005.

Este algoritmo utiliza o modelo de Bayes, ou seja, os processos estocásticos são validados pelo conceito da máxima verossimilhança, que é método utilizado na Inteligência Computacional para mineração de dados e aferição de padrões.

Em função disso, o Gráfico 4.4 apresenta uma relação comparativa entre os dois algoritmos com as menores taxas de erros: MSE, RMSE, NRMSE e MAPE, onde é possível afirmar que o RMSE representado pela primeira coluna é bem próximo entre ambos, variando entre 0,47 e 0,53, o MSE representado pela segunda coluna entre eles também é próximo, variando entre 0,23 e 0,28.

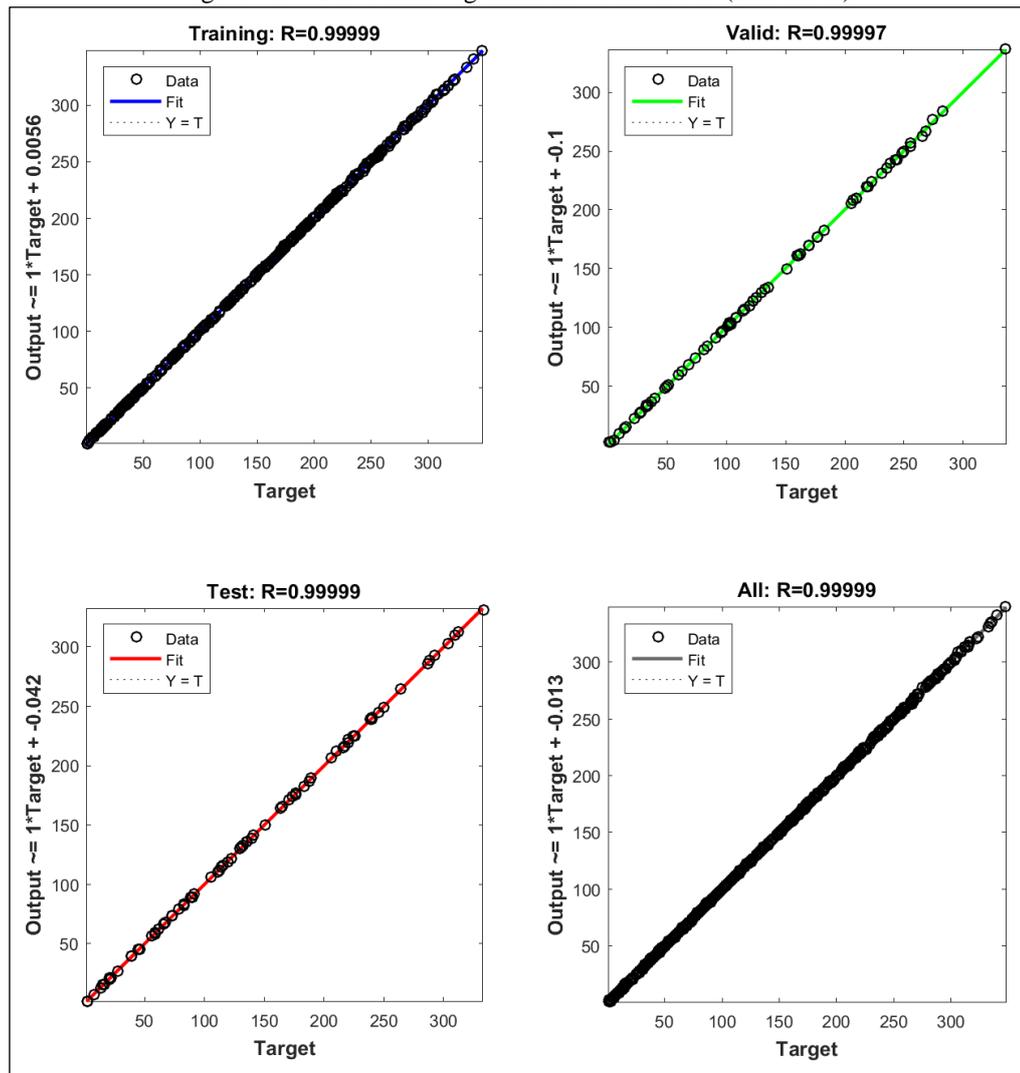
Gráfico 4.4 - Comparativo entre Levenberg-Marquardt e Regularização Bayesiana (500 dados).



Fonte: AUTOR (2021).

Contudo, para se obter confiança no modelo é de suma importância demonstrar a correlação em que existe entre a função de saída observada e a função de saída prevista, pensando nisto, a Figura 4.6 ilustra 4 gráficos que representam respectivamente a Regressão do treinamento, Regressão da validação, Regressão do teste e Regressão mediana entre as três etapas.

Figura 4.6 - Gráficos de regressão da Rede Neural (500 dados).



Fonte: AUTOR (2021).

O primeiro gráfico representado pelo treinamento obteve 99% de correlação entre a variável observada e prevista, evidenciando confiabilidade no modelo, o segundo, terceiro e quarto gráfico também apresentam uma taxa de 99% de correlação, fato característico de uma regressão e ajuste da rede neural precisa.

É válido ressaltar que o “All” da regressão significa “Todos” evidenciando uma média apurada para o modelo que válida 99% de confiabilidade na previsão e correlação entre a variável observada e preditora.

Conforme mencionado anteriormente especificamente no Capítulo 2, o Levenberg-Marquardt converge mais rápido ao considerar variáveis de entrada empíricas, ou seja, modelos de análise para comparar a aproximação de funções, que é muito utilizado no campo da matemática. É válido ressaltar também que este modelo tende a convergir mais rápido que um algoritmo genético simples.

4.3.2 Teste com 10000 Dados

O próximo teste consiste em utilizar 10000 dados para avaliar a confiabilidade do modelo ao aumentar a quantidade da base de dados, de tal forma que seja possível analisar o comportamento da previsão com 12 algoritmos de treinamento da rede neural.

Cada algoritmo obteve um resultado diferente em relação as taxas de erros: MSE, RMSE, NRMSE e MAPE e estimação de tempos, fato verídico e característico da diferença dos modelos matemáticos ao serem aplicados em métodos ou abordagens análogas que objetivam uma solução, dependendo do modelo o custo computacional pode ser maior ou não, principalmente ao se levar em consideração a inicialização de seus pesos e bias.

Considerando este fator, é válido ressaltar que a taxa de aprendizagem vem de encontro com o número de camadas aplicadas e funções de transferências, por isso, o primeiro teste para encontrar a melhor configuração foi necessário, de tal forma que nesta etapa da seleção do melhor modelo o custo computacional seria menor em relação ao primeiro.

Contudo, a Tabela 4.2 apresenta um cenário com diferentes valores para os indicadores de erros, onde é possível afirmar que o algoritmo de número 1 (Levenberg-Marquardt) obteve a menor taxa de erro em relação ao MSE, RMSE, NRMSE e MAPE, respectivamente com os seguintes valores: 0.01, 0.10, 0.0007 e 0.0011.

Em virtude deste algoritmo possuir uma maior chance de encontrar o melhor estado de convergência na rede neural ele obteve um tempo estimado de 00:01:21 segundos, o valor não caracteriza o menor tempo, contudo, as taxas de erros são menores, que é o critério de parada e seleção para ao modelo.

Tabela 4.2 - Resultados dos algoritmos de treinamento (Teste com 10000 dados).

Item	Algoritmo de Treinamento	Tempo	MSE	RMSE	NRMSE	MAPE
1	Levenberg-Marquardt	00:01:21	0,01	0,10	0,0007	0,0011
2	Regularização Bayesiana	00:01:30	0,02	0,16	0,0011	0,0022
3	Scaled Conjugate Gradient	00:00:09	0,85	0,92	0,0064	0,0207
4	Conjugate Gradient with Powell/Beale Restarts	00:00:13	1,82	1,35	0,0094	0,0273
5	Resilient Backpropagation	00:00:07	4,45	2,11	0,0146	0,0509
6	Fletcher-Powell Conjugate Gradient	00:00:07	5,42	2,33	0,0161	0,0534
7	One Step Secant	00:00:08	30,38	5,51	0,0382	0,1291
8	Polak-Ribière Conjugate Gradient	00:00:08	46,41	6,81	0,0472	0,0973
9	BFGS Quasi-Newton	00:00:41	299,33	17,30	0,1199	0,5322
10	Variable Learning Rate Gradient Descent	00:00:08	43090,53	207,58	1,4384	6,0327
11	Gradient Descent	00:00:07	71645,55	267,67	1,8548	5,9351
12	Gradient Descent with Momentum	00:00:07	223577,48	472,84	3,2765	8,4546

Fonte: AUTOR (2021).

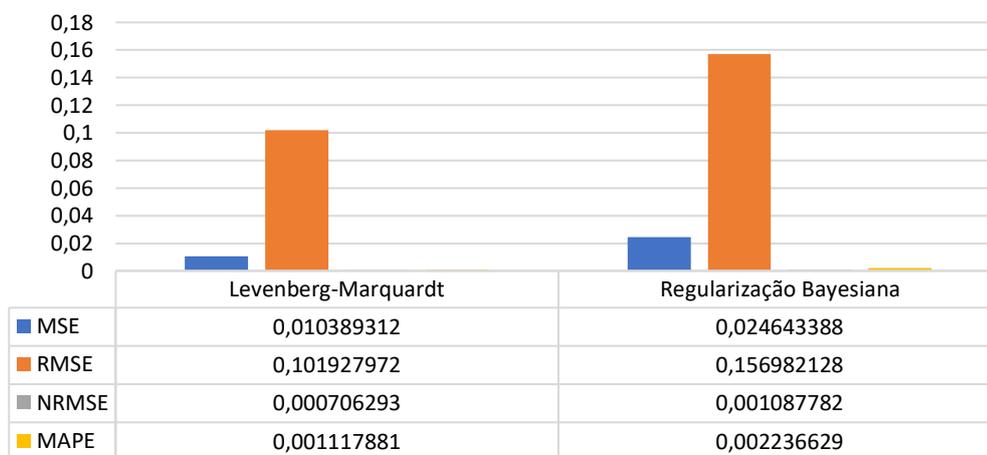
O segundo algoritmo denominado Regularização Bayesiana obteve um tempo estimado de 00:01:30, considerado o segundo melhor mediante a Tabela 4.2, os indicadores de erro MSE, RMSE, NRMSE e MAPE possuem respectivamente: 0.02, 0.16, 0.0011, 0.0022 como taxas de erro.

O terceiro algoritmo denominado Gradiente Conjugado Escalado obteve um tempo estimado de 00:00:09, e as taxas de erros são respectivamente: 0.85, 0.92, 0.0064 e 0.0207. Os demais algoritmos tendem a ter valores variados para o tempo estimado, e as taxas de erros aumentam do primeiro ao décimo segundo.

O Gráfico 4.5 apresenta uma relação comparativa entre os dois melhores algoritmos avaliados neste cenário de 10000 dados, onde o Levenberg-Marquardt foi o melhor e a Regularização Bayesiana foi o segundo melhor. Em função disso a comparação é dada pelas suas taxas de erros: MSE, RMSE, NRMSE e MAPE.

Neste gráfico é possível afirmar que a taxa de MSE caracterizada pela primeira coluna, sofre variações em um intervalo de 0,01 e 0,02, a taxa de RMSE caracterizada pela segunda coluna também sofre variações em uma escala de 0,10 a 0,15, a taxa de NRMSE caracterizada pela terceira coluna também sofre variações em uma escala de 0,0007 a 0,001, por fim, a última taxa MAPE caracterizada pela quarta coluna também sofre variação em uma escala de 0,001 a 0,002.

Gráfico 4.5 - Comparativo entre Levenberg-Marquardt e Regularização Bayesiana (10000 dados).



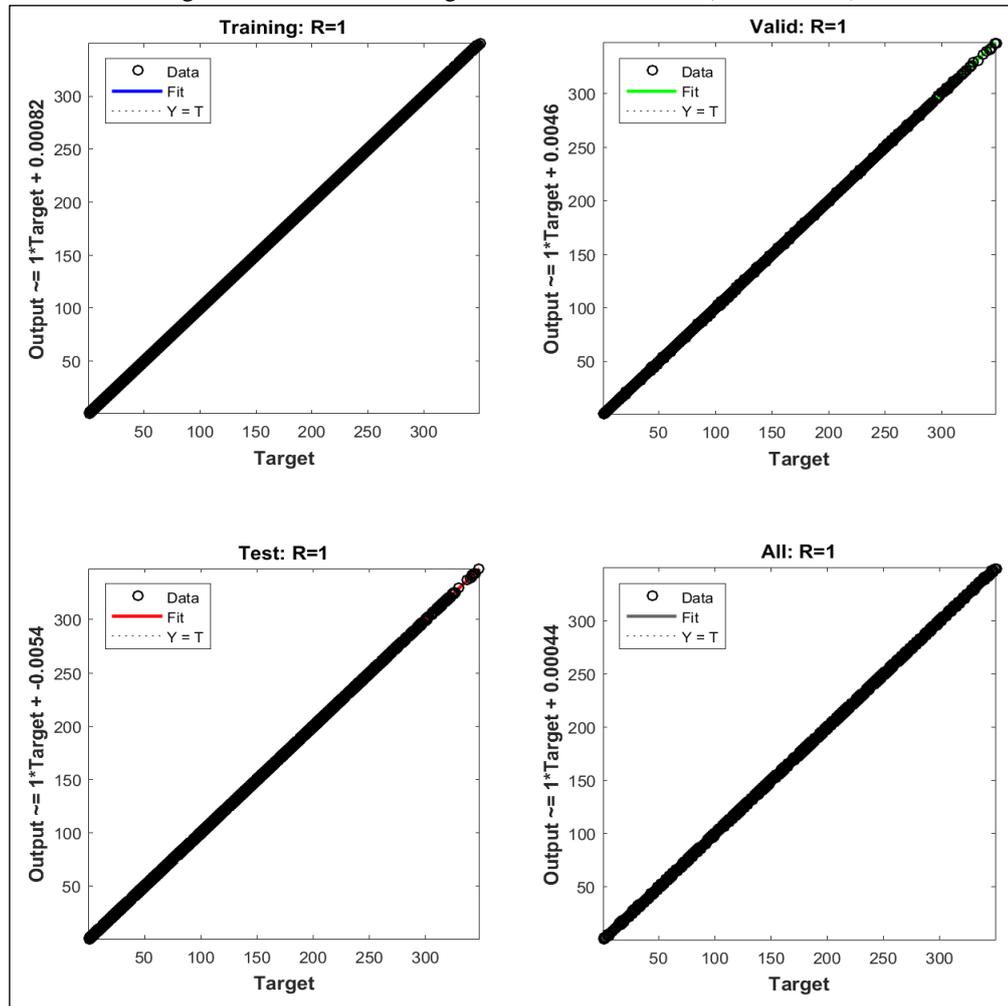
Fonte: AUTOR (2021).

Tendo em vista que foi possível encontrar o melhor algoritmo com as menores taxas de erro em relação aos outros 11 algoritmos avaliados, se faz necessário evidenciar a importância e os valores de correlação entre as variáveis de saída observada e prevista, de tal forma que a

regressão entre eles seja próxima de 100%, caracterizando eficiência e confiabilidade no modelo de previsão utilizando o modelo de rede neural.

A Figura 4.7 ilustra um cenário com 4 gráficos resultantes do teste de regressão entre 3 etapas: treinamento, teste e validação, estes são métodos precisos para se avaliar o comportamento e ajuste de uma rede neural em função de seus dados e valores de observação.

Figura 4.7 - Gráficos de regressão da Rede Neural (10000 dados).



Fonte: AUTOR (2021).

O primeiro gráfico representado pelo treinamento, alcançou 100% de correlação entre a observação e a previsão, fato caracterizado pelo encontro de seus pontos em função das coordenadas x e y, a reta de tendência é totalmente coberta pelos pontos de encontro, evidenciando valores exatos e iguais para cada variável correlacionada.

O segundo gráfico representado pela validação, alcançou também 100% de correlação entre a observação e a previsão, evidenciando a confiabilidade e correlação entre seus valores.

O terceiro gráfico representado pelo teste, alcançou 100% de correlação entre a observação e a previsão, evidenciando 100% de certeza no encontro de seus valores, ou seja, em função dos dados inferidos na rede neural, a saída prevista terá 100% de acerto conforme o estado e resultado do ajuste do modelo.

Por fim, o quarto gráfico representa a média entre eles, ou seja, o todo da regressão, que neste caso indica 100% de correlação entre as etapas comentadas.

4.4 SIMULAÇÕES DE CENÁRIOS

Nesta seção é apresentado os resultados das simulações realizadas para cada cenário, sendo os cenários baseados nos percursos dos municípios de Manaus a Barcelos, Manaus a Itacoatiara e Manaus a Parintins. Sendo que a rede neural utilizada foi a que se sobressaiu nos testes realizados, com o algoritmo de treinamento Regularização Bayesiana.

4.4.1 Cenário 1 (Manaus a Barcelos)

Este cenário se divide em duas fases, a primeira foi utilizada 500 dados para treinamento da rede neural e a segunda 10 mil dados.

4.4.2 Cenário 1 (Manaus a Barcelos – 500)

Para o cenário de Manaus a Barcelos com 500 dados de entrada na rede neural, alcançou-se os valores contidos na Tabela 4.3, na qual possui o ID que identifica o algoritmo, os dados de consumo de combustível em litros, velocidade média ideal em km/h, tempo de execução do algoritmo, tipo de computação utilizada e o algoritmo de otimização utilizado em conjunto com a rede neural.

Vale ressaltar que devido a velocidade média encontrada ser muito próxima em todas as configurações testadas, o que é levado em consideração para definir o algoritmo mais eficiente é o tempo de convergência dado em segundos.

A velocidade média ideal alcançada para este cenário foi de 80,81 aproximadamente, que gasta 178,1123 litros de combustível neste percurso, é necessário evidenciar que este gasto de combustível é para o percurso de Manaus a Barcelos somente, e não de Manaus a Barcelos e Barcelos a Manaus.

Os algoritmos utilizados foram GA, PSO e SA, formando um algoritmo híbrido entre Rede Neural e Algoritmo de Otimização.

Tabela 4.3 - Valores resultantes da simulação de Manaus a Barcelos (500 dados).

ID	Algoritmo	Paralelo	Tempo	Velocidade (km/h)	Consumo (L)
1	GA	TRUE	00:00:20	80,80689	178,1123
2	GA	FALSE	00:00:25	80,8069	178,1123
3	PSO	TRUE	00:00:09	80,80684	178,1123
4	PSO	FALSE	00:00:11	80,80685	178,1123
5	SA	FALSE	00:00:38	80,80493	178,1123

Fonte: AUTOR (2021).

Na Tabela 4.3 pode-se observar que a configuração de ID 1, que utilizou GA com computação paralela obteve um resultado melhor que a configuração de ID 2, que não possui paralelismo, com um tempo de convergência de 20 segundos para o ID 1 e 25 segundos para o ID 2.

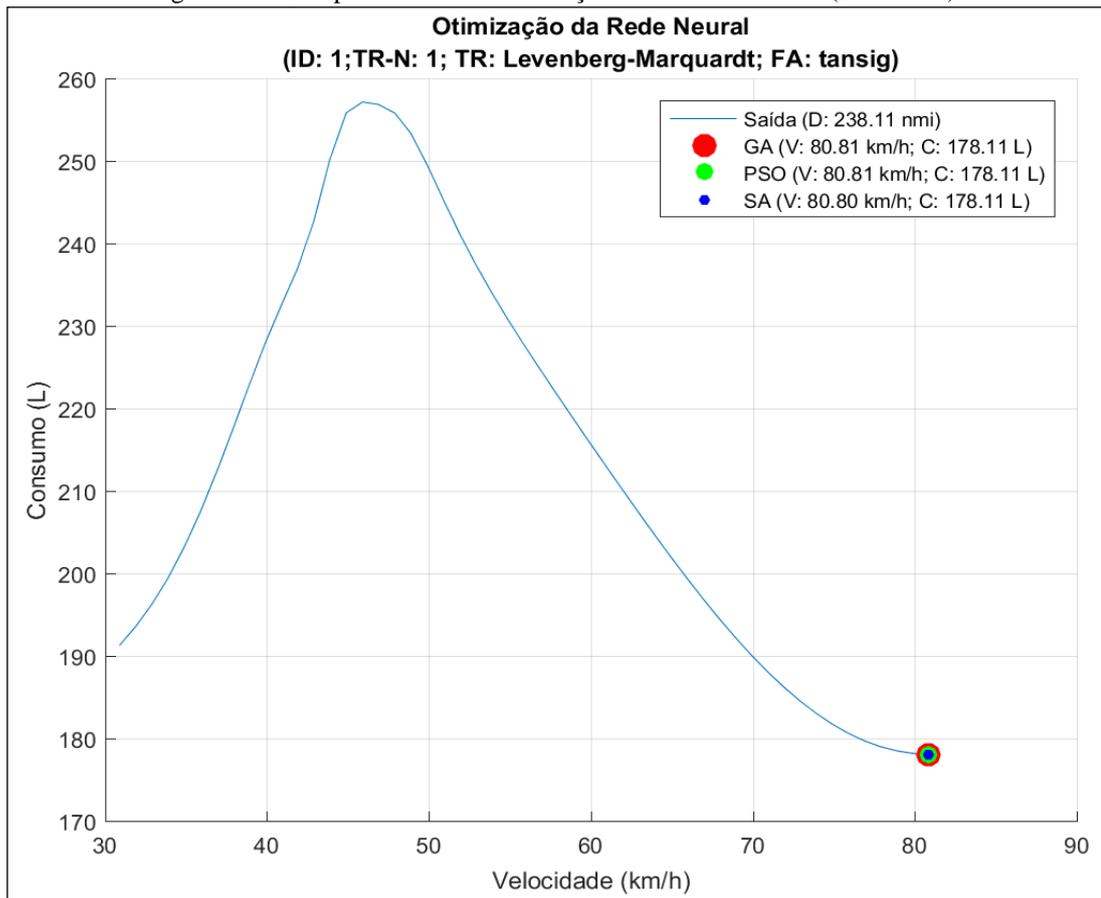
Para o ID 3 é utilizado o algoritmo PSO com paralelismo, que alcançou 9 segundos no tempo de convergência, já o ID 4, que também utiliza PSO, porém sem paralelismo, atingiu o tempo de 11 segundos.

Já o ID 5 composto pelo algoritmo SA sem paralelismo, obteve um resultado de tempo de convergência de 38 segundos.

Analisando a Tabela 4.3, é possível afirmar que o algoritmo com a pior eficiência neste cenário foi o SA com 38 segundos gastos para a convergência, em quarto lugar ficou o algoritmo GA com sem paralelismo (ID 2) com o tempo de 25 segundos, em terceiro está o GA com paralelismo (ID 1) que obteve um tempo gasto de 20 segundos, enquanto que o segundo lugar ficou ocupado pelo PSO sem paralelismo (ID 4), com 11 segundos de convergência. Logo o vencedor que obteve o melhor tempo foi o PSO com paralelismo (ID 3) com um tempo de convergência de 9 segundos.

Na Figura 4.8 é apresentado o comportamento da otimização com GA, PSO e SA, indicando que os valores encontrados da velocidade ideal são muito similares, e que por este motivo adotou-se o tempo de convergência como fator principal para escolher o algoritmo de otimização mais eficiente, já que para o consumo de combustível que é o alvo a ser minimizado foi próximo para todos também, porém no gráfico não se pode verificar estes valores tão próximos e sim iguais, devido as casas decimais, que por serem tão longas foram desconsideradas, assim como não é considerado para efeito de análise e comparação a velocidade ideal alcançado por cada algoritmo, e adota-se para este cenário o valor de 80,81 km/h para todos os algoritmos testados (GA, PSO e SA).

Figura 4.8 - Comportamento da otimização Manaus a Barcelos (500 dados).



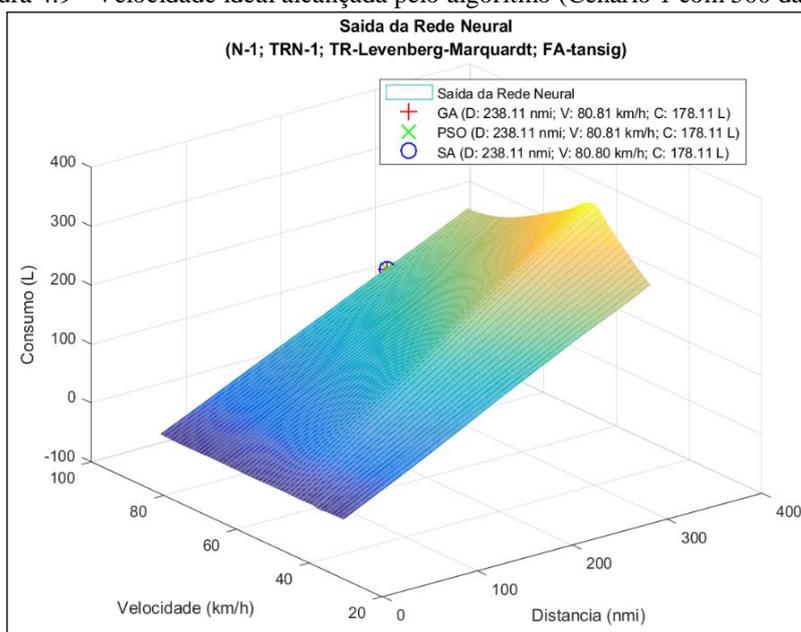
Fonte: AUTOR (2021).

Na Figura 4.8 é apresentado o comportamento da otimização pelos algoritmos testados levando em consideração o consumo em litros e a velocidade medida em km/h, no qual pode-se observar que o momento em que o motor consome mais combustível é quando a velocidade está em 46 km/h aproximadamente, consumindo um total de 258 litros de combustível aproximadamente e atinge a velocidade ótima em torno de 80,81 km/h, quando o consumo de combustível é de 178,11 litros, considerando o percurso total de Manaus a Barcelos que é de 440,98 km ou 238,11 milhas náuticas.

Nota-se que o algoritmo de treinamento na rede neural foi o *Levenberg-Marquardt* com a função de ativação *tansig*.

Na Figura 4.9 é mostrada a relação entre distância medida em milhas náuticas e velocidade medida em km/h em função do consumo de combustível em litros, que é o resultado que se obtém através da rede neural.

Figura 4.9 - Velocidade ideal alcançada pelo algoritmo (Cenário 1 com 500 dados).



Fonte: AUTOR (2021).

Na Figura 4.9 é apresentado um gráfico de malha em que é possível identificar que a velocidade ideal alcançada pelo algoritmo híbrido é de aproximadamente 80,81 km/h para uma distância de 238,11 milhas náuticas ou 440,98 km, na qual tem um gasto de combustível de 178,11 litros.

4.4.3 Cenário 1 (Manaus a Barcelos – 10000)

Para o cenário de Manaus a Barcelos com uma quantidade de dados de 10 mil na entrada da rede neural, foi possível alcançar os valores que estão contidos na Tabela 4.4, no qual possui os valores indicados do ID de cada algoritmo testado, a sigla de cada algoritmo, o tipo que indica se o algoritmo utiliza ou não a computação paralela, o tempo de convergência para cada algoritmo, a velocidade encontrada por cada algoritmo e o consumo de combustível em litros.

A velocidade ótima alcançada pelos algoritmos forma muito próximas uma da outra sendo de 81,03 km/h aproximadamente, por este fato, é utilizado o tempo de convergência para identificar o melhor algoritmo.

Tabela 4.4 - Valores resultantes da simulação de Manaus a Barcelos (10000 dados).

ID	Algoritmo	Paralelo	Tempo	Velocidade (km/h)	Consumo (L)
1	GA	TRUE	00:00:25	81,0308689	178,1810183
2	GA	FALSE	00:00:22	81,03086889	178,1810183
3	PSO	TRUE	00:00:07	81,0308689	178,1810182
4	PSO	FALSE	00:00:09	81,0308689	178,1810182
5	SA	FALSE	00:00:21	81,03086888	178,1810183

Fonte: AUTOR (2021).

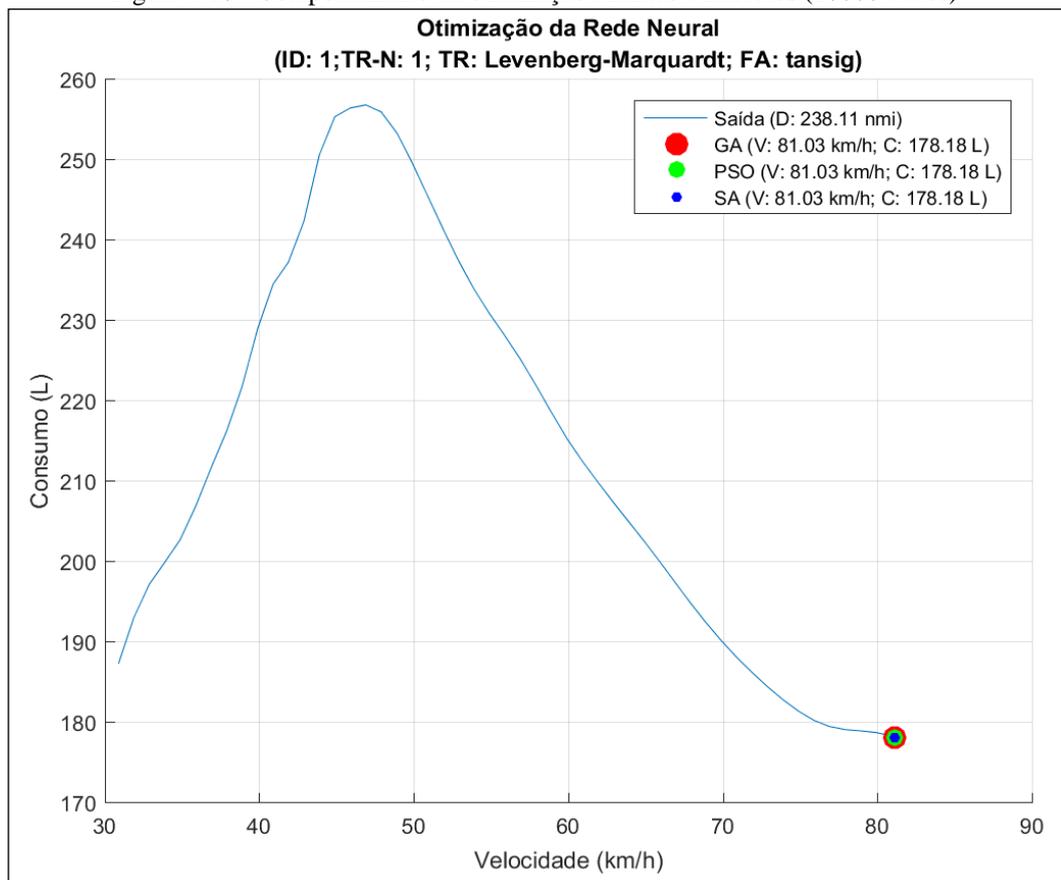
Analisando a Tabela 4.4 é possível aferir que o GA que possui computação paralela identificado pelo ID 1, atingiu um tempo de convergência de 25 segundos, enquanto que o mesmo GA sem paralelismo (ID 2) atingiu um tempo de 22 segundos.

Em relação ao algoritmo PSO com paralelismo (ID 3) teve um tempo de convergência de 7 segundos, enquanto que o PSO sem computação paralela (ID 4) obteve uma convergência de 9 segundos. Já o SA que não possui computação paralela (ID 5), alcançou a convergência em 21 segundos.

Com esta análise é possível identificar que o algoritmo que teve o pior resultado foi o GA com paralelismo (ID 1), em quarto lugar ficou o GA sem paralelismo (ID 2), em terceiro o SA sem computação paralela (ID 5), já o PSO alcançou os melhores resultados, o segundo lugar ficou ocupado pelo PSO sem paralelismo (ID 4) e o primeiro com paralelismo (ID 3).

Na Figura 4.10 é apresentado o gráfico com a relação entre consumo em litros e velocidade em km/h, observando a Figura 4.10 pode-se verificar que todos os algoritmos (GA, PSO e SA) alcançaram praticamente o mesmo valor para a velocidade.

Figura 4.10 - Comportamento da otimização Manaus a Barcelos (10000 dados).

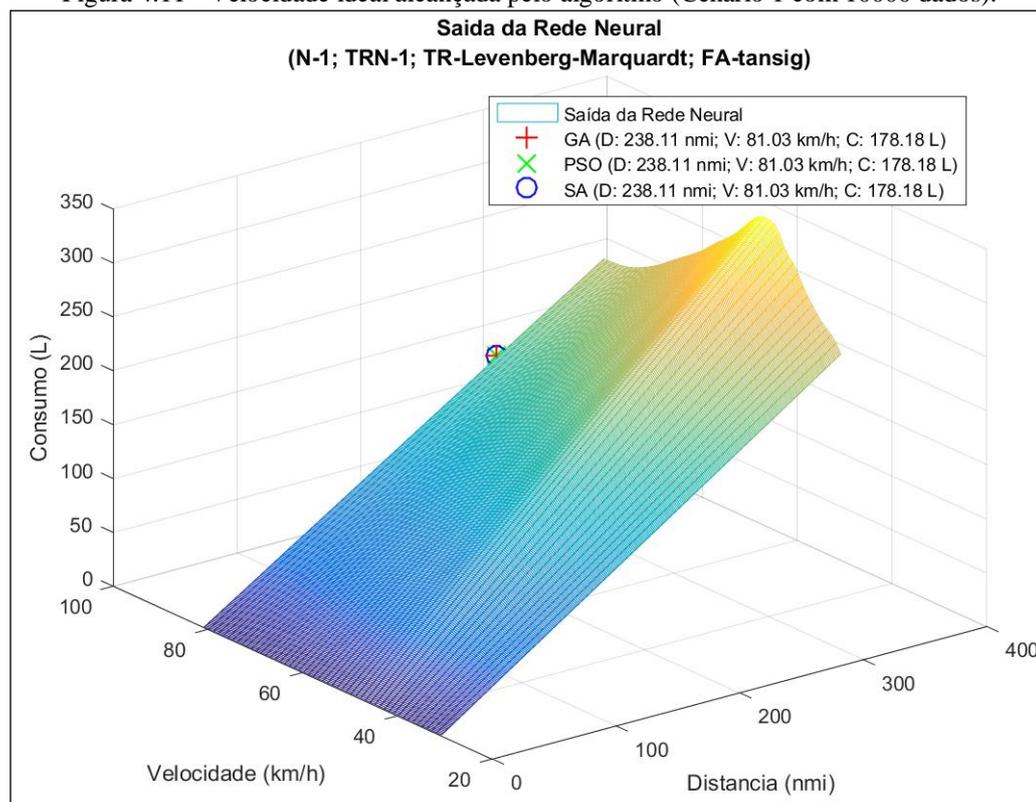


Fonte: AUTOR (2021).

Na Figura 4.10 nota-se que no momento em que os algoritmos alcançam a velocidade ideal que é de aproximadamente 81,03 km/h, o consumo é de 178,18 litros de combustível, sendo o ponto ótimo alcançado pelos algoritmos, o pior ponto pelo comportamento da curva de consumo x velocidade, é o ponto que corresponde a 46 km/h no eixo x e 256 litros de combustível no eixo y.

Na Figura 4.11 é apresentado um gráfico de superfície com a relação entre Distância e Velocidade em função do consumo de combustível.

Figura 4.11 - Velocidade ideal alcançada pelo algoritmo (Cenário 1 com 10000 dados).



Fonte: AUTOR (2021).

A Figura 4.11 é a representação 3D do gráfico que se encontra na Figura 4.10, que tem como diferencial o incremento da distância em milhas náuticas. Tanto o GA, como o PSO e SA alcançaram o mesmo ponto de otimização no consumo de combustível, sendo os valores alcançados para o consumo 178,18 litros a uma velocidade média de 81,03 km/h.

4.4.4 Cenário 2 (Manaus a Itacoatiara)

O cenário 2 se divide em duas etapas de quantidade de dados utilizados para treinar a rede neural, sendo que na primeira etapa foi utilizada 500 dados para o treinamento da rede neural e a segunda 10 mil dados.

4.4.5 Cenário 2 (Manaus a Itacoatiara – 500)

O cenário 2 é caracterizado pelo percurso de Manaus a Itacoatiara, com uma distância de 195,30 km ou 105,46 milhas náuticas. Na Tabela 4.5 é mostrado os valores de:

ID: Item responsável por identificar o algoritmo na tabela.

Algoritmo: Sigla do algoritmo que está sendo apresentado.

Paralelo: Indicativo que permite identificar se o algoritmo analisado foi executado com computação paralela ou não.

Tempo: Mostra o tempo em segundos que o algoritmo gastou para convergir até o resultado.

Velocidade: Mostra a velocidade resultante do algoritmo em km/h.

Consumo: Apresenta os valores alcançados pelo algoritmo do menor consumo que pode ser gasto caso a velocidade média encontrada seja obedecida.

Vale ressaltar que os valores alcançados por cada algoritmo em relação a velocidade e ao consumo foram idênticos, devido a esta característica os algoritmos foram analisados com base no tempo de convergência que cada um levou para alcançar o resultado da melhor velocidade média. Os valores da velocidade foram de aproximadamente 78,68 km/h, caso a embarcação efetue o percurso com esta velocidade média o consumo de combustível será de aproximadamente de 79,16 litros.

Tabela 4.5 - Valores resultantes da simulação de Manaus a Itacoatiara (500 dados).

ID	Algoritmo	Paralelo	Tempo	Velocidade (km/h)	Consumo (L)
1	GA	TRUE	00:00:20	78,68125	79,16012
2	GA	FALSE	00:00:24	78,68123	79,16012
3	PSO	TRUE	00:00:10	78,68122	79,16012
4	PSO	FALSE	00:00:10	78,68124	79,16012
5	SA	FALSE	00:01:04	78,68134	79,16012

Fonte: AUTOR (2021).

O algoritmo GA com paralelismo identificado por (ID 1) atingiu a velocidade média ideal em 20 segundos de execução. Já o GA sem computação paralela (ID 2) alcançou o mesmo resultado com um tempo gasto de 24 segundos.

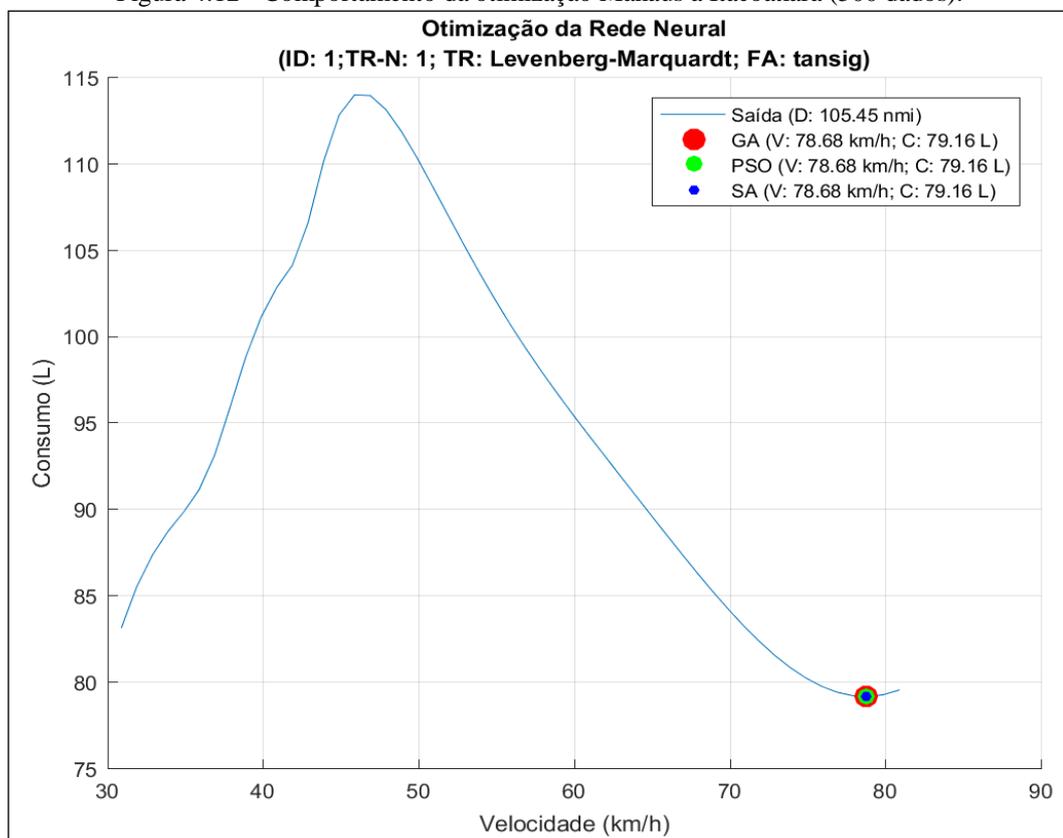
O algoritmo PSO obteve o resultado com uma convergência de 10 segundos com computação paralela (ID 3), já o mesmo algoritmo sem paralelismo (ID 4) gastou os mesmos

10 segundos para convergir até encontrando o valor de 78,68 km/h para a velocidade e 79,16 litros em consumo de combustível. O SA que não utiliza computação paralela (ID 5) atingiu 1 minuto e 4 segundos para convergir até a velocidade ótima.

Analisando os valores encontrados pelos algoritmos de otimização, pode-se classifica-los em posições de pior para melhor, sendo o pior algoritmo o SA (ID 5), que ficou muito distante dos concorrentes com 1 minuto e 4 segundos. Em terceiro lugar ficou o GA de ID 2, com 24 segundos de convergência, o segundo lugar ficou ocupado pelo GA de ID1, que obteve 20 segundos gastos para atingir o resultado ótimo. Em primeiro ficaram empatados os algoritmos PSO (ID 3 e 4) com 10 segundo de execução para alcançar o melhor valor entre velocidade e consumo.

Na Figura 4.12 é apresentado o gráfico do comportamento das duas variáveis (velocidade e consumo), que estão relacionadas pelo eixo x e y.

Figura 4.12 - Comportamento da otimização Manaus a Itacoatiara (500 dados).



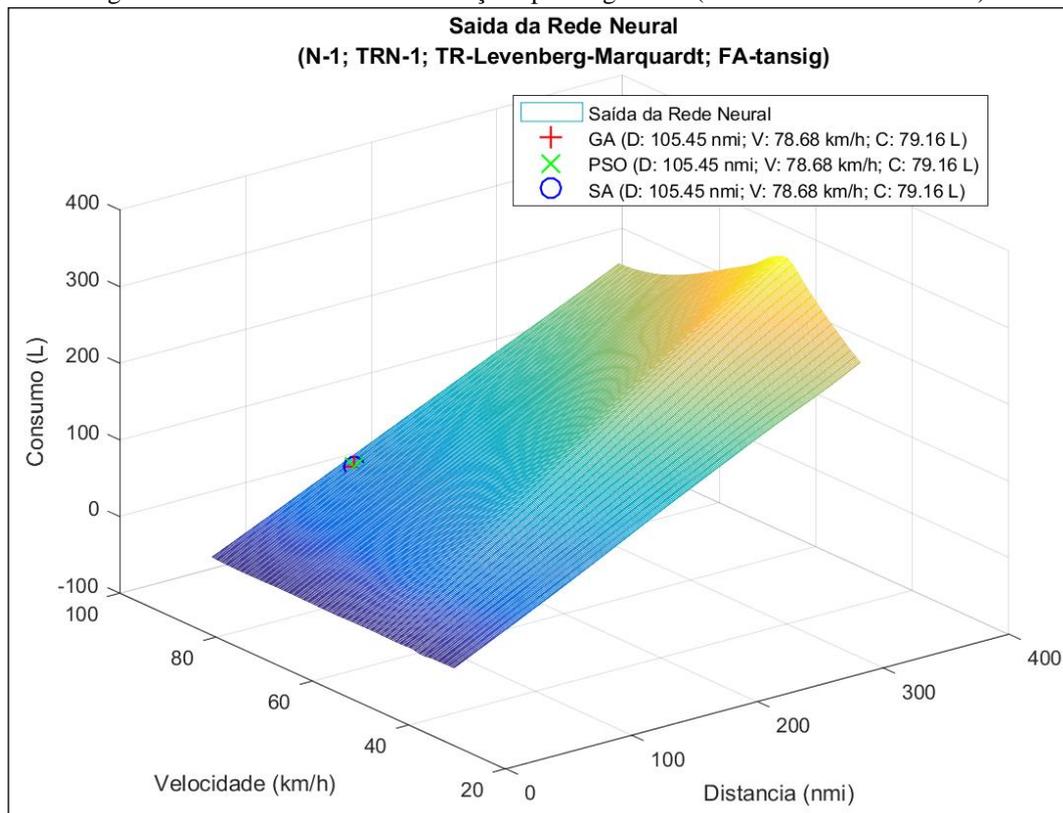
Fonte: AUTOR (2021).

Na Figura 4.12 é possível notar que o consumo é baixo (83 litros aproximadamente) quando a velocidade fica em torno de 32 km/h, e aumenta gradativamente ao longo da aceleração até atingir o máximo de consumo de combustível que é aproximadamente quando a velocidade atinge 47 km/h, sendo o consumo de 114 litros, levando em consideração a distância

estabelecida do percurso que é de 195,30 km. A partir de 48 km/h aproximadamente, o consumo diminui até atingir seu menor ponto, que é alcançado no momento em que a velocidade atinge o valor de aproximadamente 78,68 km/h, neste momento o consumo fica em torno de 79,16 litros.

A Figura 4.13 apresenta um gráfico de malha que relaciona a distância, velocidade e consumo, nesta figura é possível verificar que quanto maior é a distância de um percurso, maior é gasto de consumo de combustível, com isso pode-se deduzir que encontrar a velocidade em que o motor atinge a menor taxa de consumo, é viável para viagens com um longo percurso, que é o caso deste cenário analisado (Manaus a Itacoatiara) e dos outros cenários também (Manaus a Barcelos e Manaus a Parintins).

Figura 4.13 - Velocidade ideal alcançada pelo algoritmo (Cenário 2 com 500 dados).



Fonte: AUTOR (2021).

Na Figura 4.13 pode-se notar que os três algoritmos utilizados para encontrar a velocidade ótima, tiveram o mesmo resultado, diferindo apenas a velocidade (tempo) com que convergiram.

4.4.6 Cenário 2 (Manaus a Itacoatiara – 10000)

Na Tabela 4.6 é mostrado seis colunas com os seguintes parâmetros:

ID: Item que identifica um algoritmo e conseqüentemente suas configurações na tabela, bem como os resultados correlacionados com o mesmo.

Algoritmo: Nesta coluna é apresentado as siglas dos algoritmos utilizados nos testes.

Paralelo: Coluna que mostra a configuração do método de execução do algoritmo, podendo ele ser executado com computação paralela ou não.

Tempo: Mostra o tempo em segundos da convergência do algoritmo, item que é utilizado para comparação de performance dos algoritmos de otimização.

Velocidade: Coluna que indica os valores encontrados da velocidade ideal pelos algoritmos de otimização, valores estes que são utilizados para se obter um menor gasto no consumo de combustível nos testes realizados.

Consumo: Item que apresenta os valores de consumo de combustível conforme a velocidade ideal.

Neste cenário os valores encontrados pelo algoritmo híbrido proposto com rede neural e algoritmos de otimização, são muito próximos um do outro em termo de velocidade ideal e consumo de combustível, logo pelo fato de haver estes valores muito próximos, o item que diferencia a performance dos algoritmos é o tempo de convergência para encontrar os resultados de velocidade e consumo.

Os valores da velocidade encontrados pelos algoritmos foram de aproximadamente 81,03 km/h para a velocidade e 78,98 litros para o consumo de combustível. Todos os valores podem ser visualizados na Tabela 4.6.

Tabela 4.6 - Valores resultantes da simulação de Manaus a Itacoatiara (10000 dados).

ID	Algoritmo	Paralelo	Tempo	Velocidade (km/h)	Consumo (L)
1	GA	TRUE	00:00:21	81,03086567	78,97697597
2	GA	FALSE	00:00:24	81,03086888	78,97697577
3	PSO	TRUE	00:00:07	81,0308689	78,97697576
4	PSO	FALSE	00:00:09	81,0308689	78,97697576
5	SA	FALSE	00:00:21	81,0308671	78,97697588

Fonte: AUTOR (2021).

Na Tabela 4.6 é possível observar que o algoritmo de otimização GA com computação paralela (ID 1) levou 21 segundos para convergir até a velocidade ótima. Já o mesmo algoritmo, porém utilizando o recurso de paralelismo (ID 2) atingiu o resultado da velocidade ótima com um custo de 24 segundos de convergência.

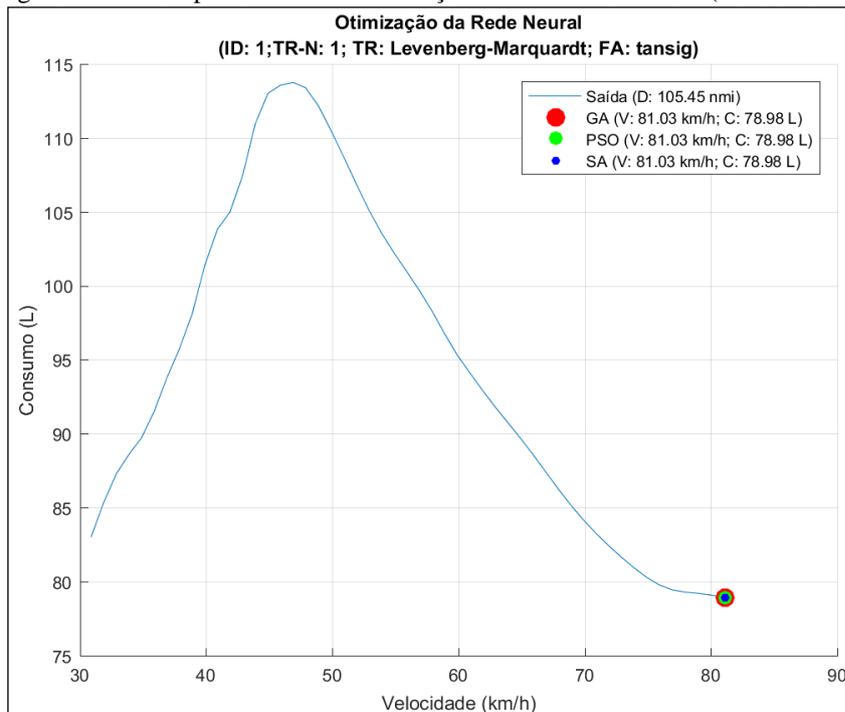
A convergência do PSO com computação paralela (ID 3) gastou 7 segundos, enquanto que o mesmo algoritmo, porém sem a utilização do recurso de paralelismo (ID 4) demorou 9 segundos para convergir até a velocidade ideal de 81,03 km/h aproximadamente.

O algoritmo SA (ID 5) que não utiliza computação paralela, obteve um resultado de convergência que demorou cerca de 21 segundos para encontrar a velocidade ótima em função do consumo.

Com estes resultados apresentados na Tabela 4.6, pode-se aferir que o GA sem paralelismo (ID 2) foi o que apresentou a pior convergência, já o GA com paralelismo (ID 1) juntamente com o algoritmo de otimização SA (ID 5), gastaram 21 segundos para alcançar o valor da velocidade ideal, tendo os mesmos ficado empatados na terceira colocação entre os algoritmos testados. Em segundo encontra-se o PSO sem computação paralela (ID 4) com um tempo de convergência de 9 segundos e na primeira posição ficou o PSO com a utilização do recurso de paralelismo (ID 3), alcançando o menor tempo de convergência entre os algoritmos testados, sendo de 7 segundos gastos para encontrar a velocidade ótima.

Na Figura 4.14 é mostrado o gráfico do comportamento do consumo de combustível em relação a velocidade.

Figura 4.14 - Comportamento da otimização Manaus a Itacoatiara (10000 dados).



Fonte: AUTOR (2021).

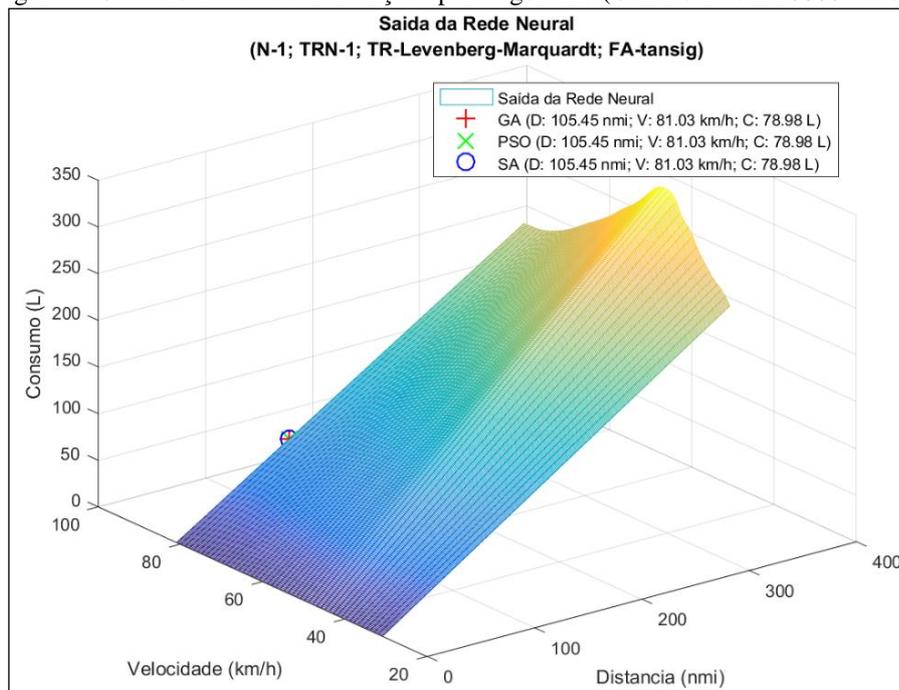
Na Figura 4.14 pode-se notar que o comportamento do consumo em relação a velocidade existe um aumento gradual do consumo quando a velocidade se encontra a 83 km/h

aproximadamente até 47 km/h aproximadamente, sendo o consumo de combustível mais alto quando a velocidade está em 47 km/h que consome aproximadamente 113 litros de combustível.

A partir do ponto 47 km/h aproximadamente, o consumo diminui de forma gradual até atingir o ponto ótimo encontrado pelos algoritmos de otimização, que é a velocidade a 81,03 km/h que consome em torno de 78,98 litros de combustível.

Na Figura 4.15 é apresentado o gráfico de superfície em 3D que relaciona a distância do percurso, a velocidade e o consumo.

Figura 4.15 - Velocidade ideal alcançada pelo algoritmo (Cenário 2 com 10000 dados).



Fonte: AUTOR (2021).

Na Figura 4.15 pode-se observar que os algoritmos alcançaram o mesmo ponto ótimo da velocidade, levando em consideração a distância para estimar o consumo que é gasto se a velocidade ideal for obedecida.

Vale ressaltar que o algoritmo encontra velocidade ideal, a fim de estimar a quantidade de combustível em litros que a embarcação gasta para percorrer determinada distancia, que neste cenário é de 105,45 milhas náuticas.

4.4.7 Cenário 3 (Manaus a Parintins)

O cenário 3 está dividido em duas etapas, uma com 500 dados de entrada para treinar a rede neural e outro com 10000 dados.

4.4.8 Cenário 3 (Manaus a Parintins – 500)

O cenário 3 que tem como base o percurso de Manaus a Parintins, com uma distância de 433.75 km ou 234,21 milhas náuticas. Na Tabela 4.7 é apresentado os valores dos seguintes itens:

ID: Item que serve para identificar os algoritmos testados.

Algoritmo: Item que possui a sigla do algoritmo testado.

Paralelo: Item que tem por finalidade indicar se o algoritmo analisado foi executado com o recurso de computação paralela ou se não foi aplicado este recurso.

Tempo: Item que serve de comparação para determinar a performance dos algoritmos testados, no qual o tempo é mostrado em segundos.

Velocidade: Item que exibe o valor da velocidade encontrada pelos algoritmos de otimização.

Consumo: Mostra os valores estimados do consumo de combustível em relação a velocidade encontrada pelos algoritmos de otimização.

Como pode ser observado na Tabela 4.7, os resultados obtidos pelos algoritmos de otimização (GA, PSO e SA) foram muito próximos um do outro, devido à esta característica o item que é utilizado para identificar qual algoritmo teve o melhor desempenho, é o tempo de convergência.

Os valores alcançados pelos algoritmos para a velocidade ficaram em torno de 80,74 km/h e para o consumo os resultados foram de aproximadamente 175,27 litros, estes valores podem ser encontrados na Tabela 4.7.

Tabela 4.7 - Valores resultantes da simulação de Manaus a Parintins (500 dados).

ID	Algoritmo	Paralelo	Tempo	Velocidade (km/h)	Consumo (L)
1	GA	TRUE	00:00:21	80,74405	175,267
2	GA	FALSE	00:00:24	80,744	175,267
3	PSO	TRUE	00:00:07	80,74407	175,267
4	PSO	FALSE	00:00:12	80,74407	175,267
5	SA	FALSE	00:00:28	80,74296	175,267

Fonte: AUTOR (2021).

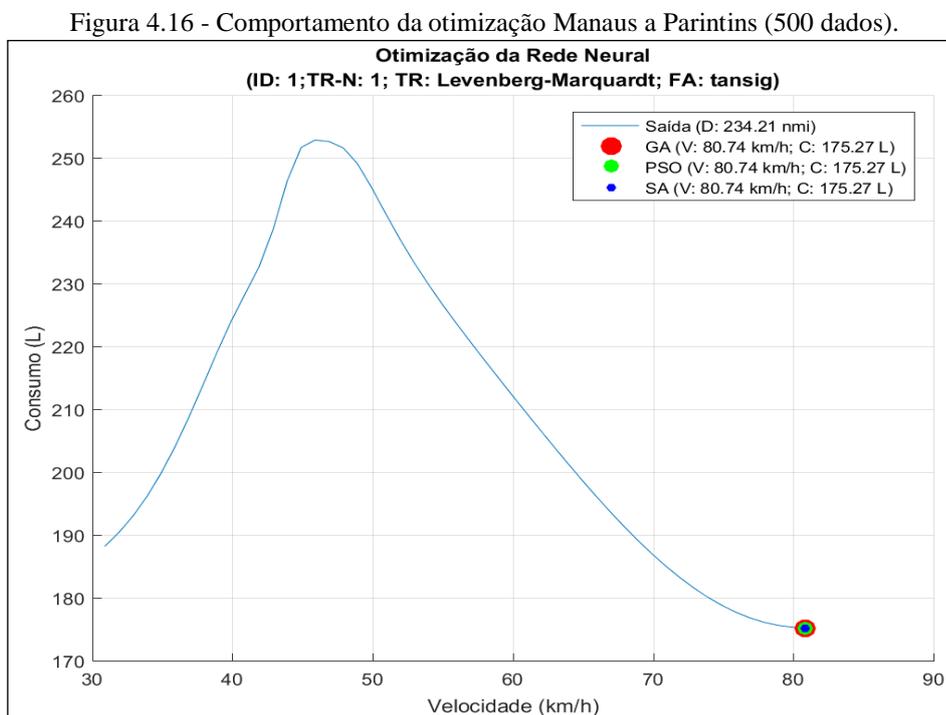
Na Tabela 4.7 pode ser notado que o algoritmo GA com o recurso de computação paralela (ID 1) gastou um tempo de 21 segundos para convergir até ao ponto ideal da velocidade, já o GA (ID 2) sem o recurso de paralelismo, obteve um tempo de convergência de 24 segundos.

O algoritmo PSO com paralelismo (ID 3) teve um custo computacional de 7 segundos para atingir o melhor ponto de minimização da velocidade, enquanto que o mesmo algoritmo sem o recurso de computação paralela (ID 4) alcançou um tempo de convergência de 12 segundos. Já o algoritmo SA que não possui o recurso de paralelismo (ID 5) gastou 28 segundos para conseguir atingir o melhor ponto de minimização da velocidade em relação ao consumo de combustível.

De forma a elencar por posição de melhor performance que foi utilizado o tempo de convergência para averiguar este processo, com a análise da Tabela 4.7 pode-se afirmar que o SA (ID 5) foi o pior entre os algoritmos testados pois obteve um custo computacional para a otimização de 28 segundos, ficando assim na quinta e última colocação. Já o GA (ID 2) conseguiu ser o quarto colocado, com 24 segundos, em terceiro ficou o GA (ID 1) com 21 segundo de custo computacional. Em segundo lugar ficou o PSO (ID 4) com 12 segundos para convergir, e o primeiro foi o PSO (ID 3) com apenas 7 segundos para atingir o ponto de otimização.

Com isso pode-se verificar que o algoritmo PSO com (ID 3) e sem (ID 4) o recurso de computação paralela foram os que tiveram melhores performances, mostrando uma eficiência superior aos demais algoritmos testados (GA e SA).

Na Figura 4.16 é apresentado o comportamento da relação entre o consumo e a velocidade.



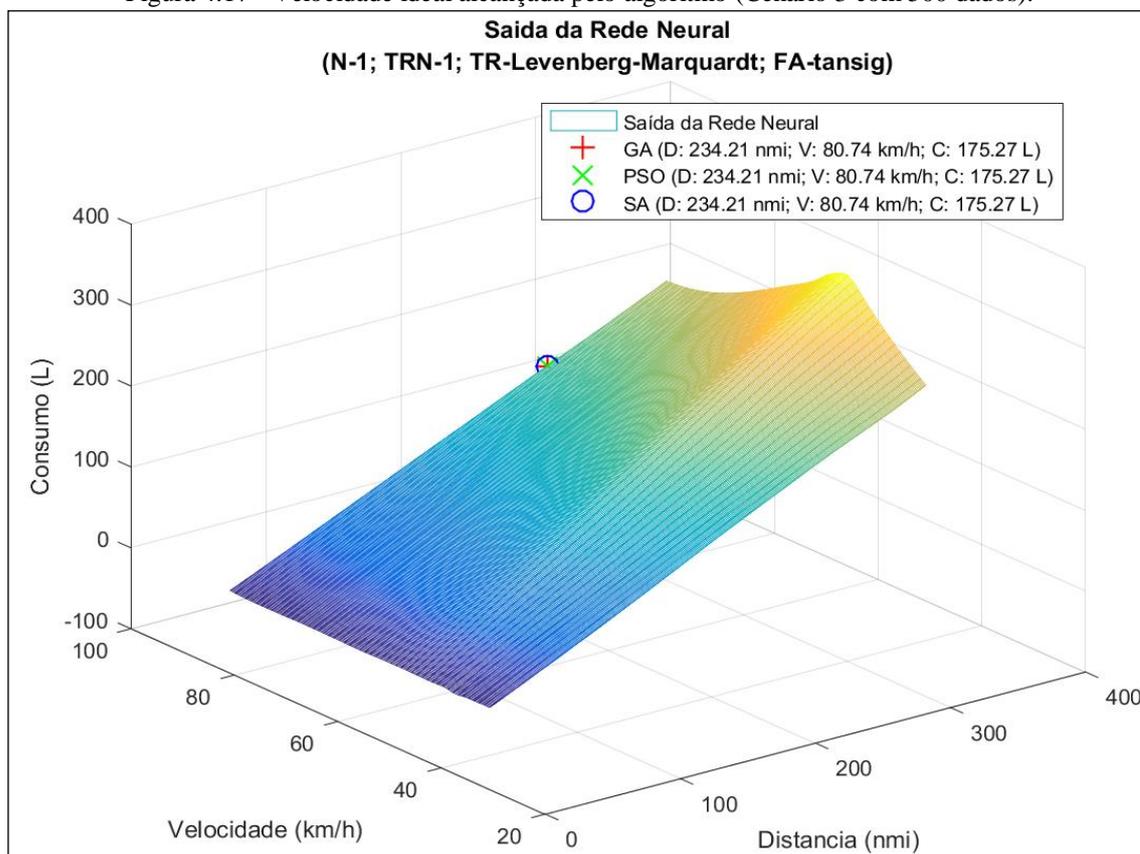
Fonte: AUTOR (2021).

Na Figura 4.16 é possível observar que o consumo de combustível tem ligação direta com a velocidade, no qual pode-se notar que quando a velocidade é de 31 km/h aproximadamente, o consumo de combustível fica em torno de 198 litros para o percurso analisado neste cenário (Manaus a Parintins), deste ponto citado ao ponto x(47 km/h) e y(253 litros), há uma elevação gradativa no consumo conforme a velocidade aumenta.

Do ponto x(47 km/h) ao ponto de otimização x(80,74 km/h) há uma diminuição drástica no consumo de combustível, sendo que quando a velocidade média de 80,74 km/h é obedecida, o consumo de litros para o trajeto do cenário 3 é de 175,27 litros aproximadamente.

Na Figura 4.17 é mostrado o gráfico de superfície no qual correlaciona a distância em milhas náuticas, a velocidade em km/h e o consumo em litros.

Figura 4.17 - Velocidade ideal alcançada pelo algoritmo (Cenário 3 com 500 dados).



Fonte: AUTOR (2021).

Na Figura 4.17 pode-se observar que os algoritmos de otimização alcançaram o mesmo ponto de minimização no consumo de combustível, encontrando a velocidade média em 80,74 km/h e o consumo de 175,27 litros em uma distância de 234,21 milhas náuticas.

4.4.9 Cenário 3 (Manaus a Parintins – 10000)

Na Tabela 4.8 é apresentado alguns itens como a velocidade encontrada pelos algoritmos, o consumo de combustível para a distância de 433.75 km, o recurso de computação paralela e o tempo para a convergência dos algoritmos de otimização.

Devido aos algoritmos neste cenário assim como nos outros terem alcançados resultados bastantes similares, é utilizado o tempo de convergência de cada um deles para indicar a performance de cada um. Os valores encontrados pelos algoritmos em relação a velocidade ficam em torno de 81,03 km/h e o consumo é de aproximadamente 175,26 litros.

Tabela 4.8 - Valores resultantes da simulação de Manaus a Parintins (10000 dados).

ID	Algoritmo	Paralelo	Tempo	Velocidade (km/h)	Consumo (L)
1	GA	TRUE	00:00:20	81,03086889	175,2613814
2	GA	FALSE	00:00:30	81,03086869	175,2613815
3	PSO	TRUE	00:00:07	81,0308689	175,2613814
4	PSO	FALSE	00:00:09	81,0308689	175,2613814
5	SA	FALSE	00:00:38	81,0308689	175,2613814

Fonte: AUTOR (2021).

Na Tabela 4.8 pode-se observar que o algoritmo GA utilizando o recurso de computação paralela (ID 1) possui um tempo de convergência de 20 segundos, em contrapartida o mesmo algoritmo sem utilizar o recurso de paralelismo (ID 2) gastou um tempo de convergência de 30 segundos a mais (30 segundos) que o ID 1.

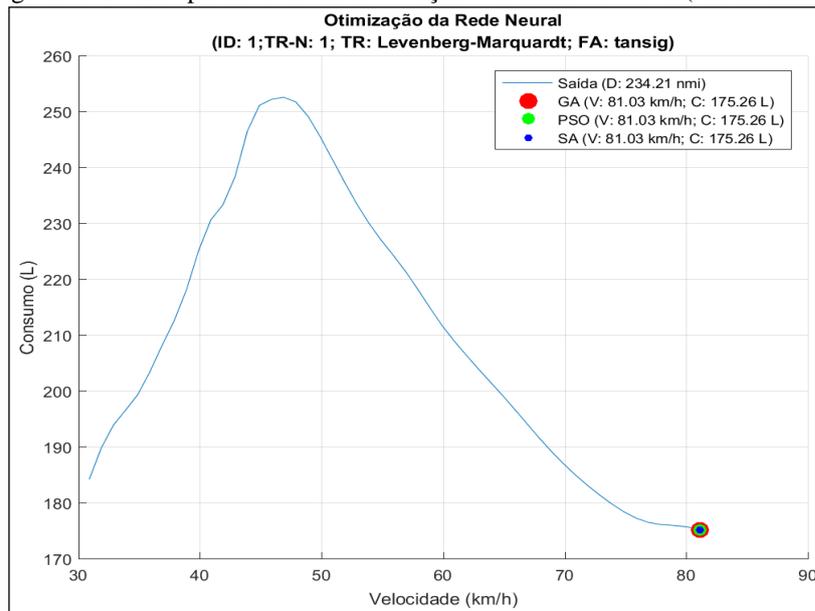
O Algoritmo PSO com paralelismo (ID 3) teve um custo computacional de 7 segundos, enquanto que sem o recurso de computação paralela (ID 4) teve uma convergência que demorou 9 segundos até atingir a velocidade ideal.

Já em relação ao algoritmo SA que não utiliza o recurso da computação paralela (ID 5) o custo computacional foi de 38 segundos.

Analisando a Tabela 4.8 é possível aferir que o algoritmo ID 5 teve o pior resultado, ficando em quinto lugar na classificação de desempenho, o quarto foi o ID 2 que atingiu 30 segundos de execução, em terceiro ficou o ID 1, no qual teve um custo computacional de 20 segundos, em segundo ficou o algoritmo (ID 4) com 9 segundos de convergência e em primeiro o ID 3 com 7 segundos. Logo o algoritmo PSO teve os melhores resultados com ou sem o recurso da computação paralela.

A Figura 4.18 é apresenta o comportamento da relação entre velocidade e consumo e o ponto ótimo entre os dois.

Figura 4.18 - Comportamento da otimização Manaus a Parintins (10000 dados).

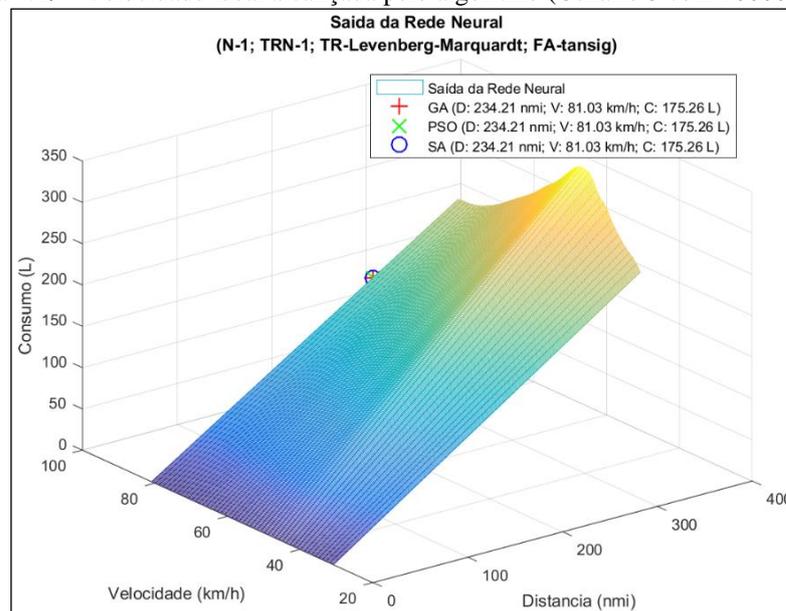


Fonte: AUTOR (2021).

A Figura 4.18 mostra que a partir do ponto em que a velocidade é 32 km/h aproximadamente e o consumo 185 litros, há uma elevação do consumo até a velocidade de 47 km/h aproximadamente, em que o consumo atinge seu potencial máximo, em torno de 253 litros consumidos no trajeto. Deste ponto até atingir o ponto de otimização há uma queda gradual do consumo, atingindo seu ponto mínimo na velocidade de 81,03 km/h, no qual é consumido cerca de 175,26 litros durante o trajeto de Manaus a Parintins.

Na Figura 4.19 é exposto o gráfico de superfície que relaciona a distância, velocidade e consumo.

Figura 4.19 - Velocidade ideal alcançada pelo algoritmo (Cenário 3 com 10000 dados).



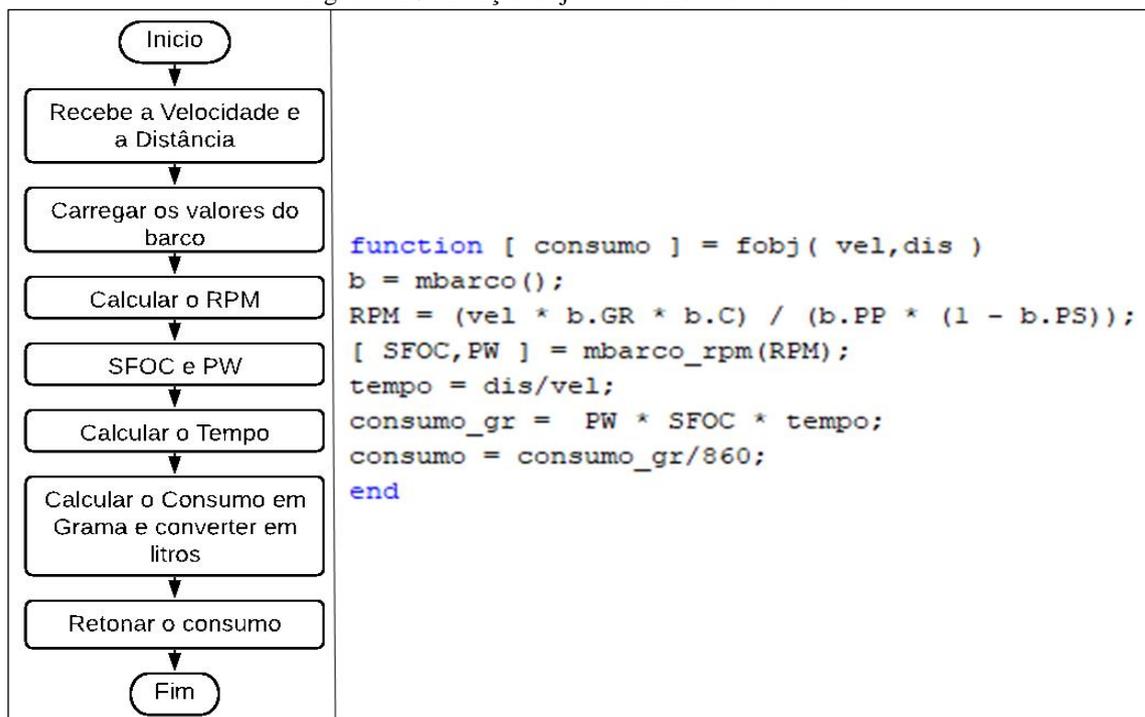
Fonte: AUTOR (2021).

Na Figura 4.19 é mostrado que os algoritmos de otimização (GA, PSO e SA) conseguiram atingir o mesmo ponto de otimização que é referenciado pela velocidade a 81,03 km/h consumindo 175,26 litros de combustível em uma distância de 234,21 milhas náuticas.

4.5 Modelo Calculado para Validação de Resultados

O modelo calculado tem como intuito executar os passos do algoritmo de otimização proposto considerando o modelo matemático para se obter o consumo de combustível. Para isso, foi necessário adaptar a função objetiva conforme ilustrado na Figura 4.20.

Figura 4.20 - Função objetiva do modelo calculado.



Fonte: AUTOR (2021).

Passo 1: a função é chamada pelo Algoritmo de Otimização recebendo 2 argumentos denominados velocidade em km/h e distância em milhas náuticas e retornando a variável consumo em litros;

Passo 2: com os 2 parâmetros recebidos, a função realiza os devidos cálculos matemáticos;

Passo 3: carrega no espaço de trabalho os parâmetros relativos à embarcação sendo eles: ‘Gear Radio’, ‘Propeller Pitch’, ‘Propeller Slip’ e constante para conversão;

Passo 4: a função calcula o RPM;

Passo 5: a função calcula o consumo em gramas;

Passo 6: a função realiza a conversão de gramas para litros;

Passo 7: a função retorna o consumo calculado.

4.6 Resultados do Modelo Calculado

4.6.1 Cenário 1 (Manaus à Barcelos)

Para o cenário de Manaus a Barcelos, alcançou-se os valores contidos na Tabela 4.9, na qual possui as seguintes colunas:

ID: que identifica o item analisado;

Algoritmo: especifica qual otimização foi testada;

Paralelo: parâmetro que indica o uso de computação paralela;

Tempo: indica o tempo gasto para execução da otimização;

Velocidade: variável de entrada em km/h;

Consumo: variável de saída em litros.

Vale ressaltar que devido a velocidade média encontrada ser muito próxima em todas as configurações testadas, o que é levado em consideração para definir o algoritmo mais eficiente é o tempo de execução do algoritmo, dado em segundos.

A velocidade média ideal alcançada para este cenário foi de aproximadamente 81 km/h, que gasta 178,2583 litros de combustível neste percurso, é necessário evidenciar que este gasto de combustível é para o percurso de Manaus a Barcelos somente, e não de Manaus a Barcelos e Barcelos a Manaus.

Os algoritmos utilizados foram GA, PSO e SA, formando um modelo híbrido entre Rede Neural e Algoritmo de Otimização.

Tabela 4.9 - Resultantes da simulação calculada de Manaus à Barcelos.

ID	Algoritmo	Paralelo	Tempo	Velocidade (km/h)	Consumo (L)
1	GA	TRUE	00:00:15	81,030869	178,258316
2	GA	FALSE	00:00:03	81,030869	178,258316
3	PSO	TRUE	00:00:05	81,030869	178,258316
4	PSO	FALSE	00:00:01	81,030869	178,258316
5	SA	FALSE	00:00:31	81,030869	178,258316

Fonte: AUTOR (2021).

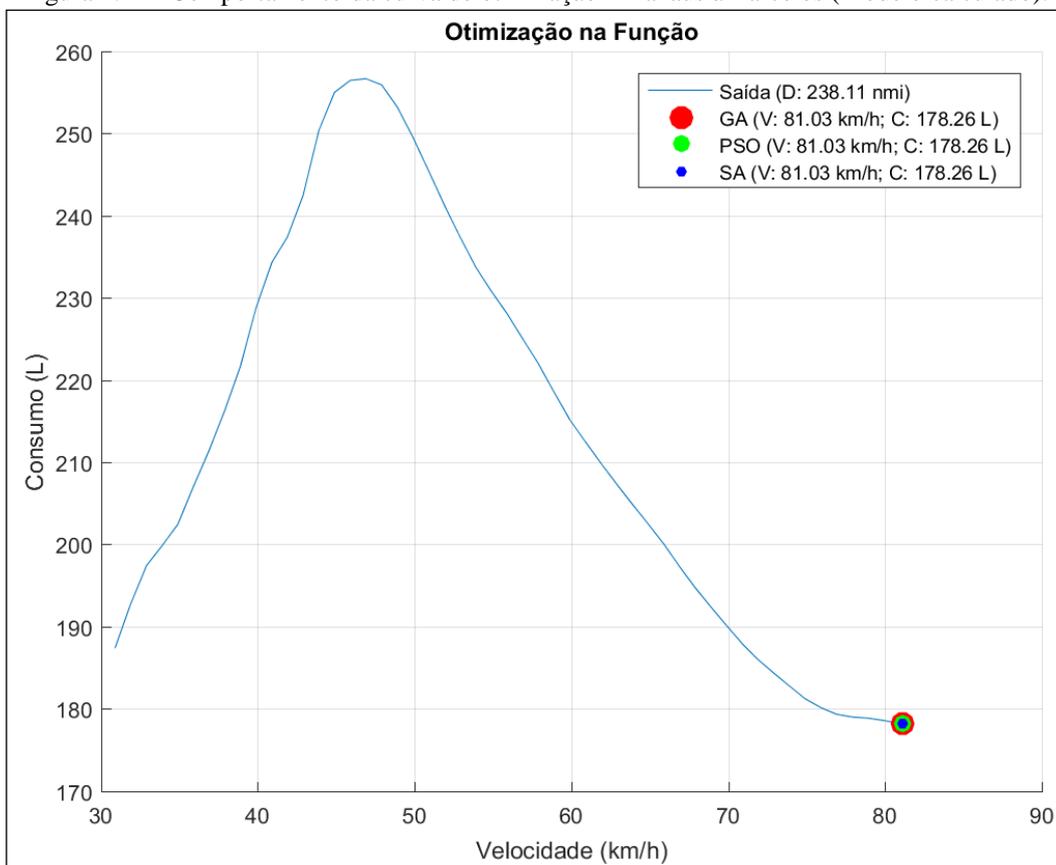
Analisando a Tabela 4.9, é possível afirmar que o algoritmo com a pior eficiência neste cenário foi o SA com 31 segundos gastos para a convergência, em quarto lugar ficou o algoritmo GA com paralelismo (ID 1) com o tempo de 15 segundos, em terceiro está o PSO com

paralelismo (ID 3) que obteve um tempo gasto de 5 segundos, enquanto que o segundo lugar ficou ocupado pelo GA sem paralelismo (ID 2), com 3 segundos de convergência. Logo, o vencedor que obteve o melhor tempo foi o PSO sem paralelismo (ID 1) com um tempo de convergência de 1 segundo.

Na Figura 4.21 é apresentado o comportamento da otimização com GA, PSO e SA, indicando que os valores encontrados da velocidade ideal são muito similares, e que por este motivo adotou-se o tempo de convergência como fator principal para escolher o algoritmo de otimização mais eficiente.

Tendo em vista que o consumo de combustível que é o alvo a ser minimizado foi próximo para todos, no gráfico não é possível mensurar valores tão próximos devido as casas decimais, por serem tão longas foram desconsideradas, adota-se para este cenário o valor de 81,03 km/h para todos os algoritmos testados (GA, PSO e SA).

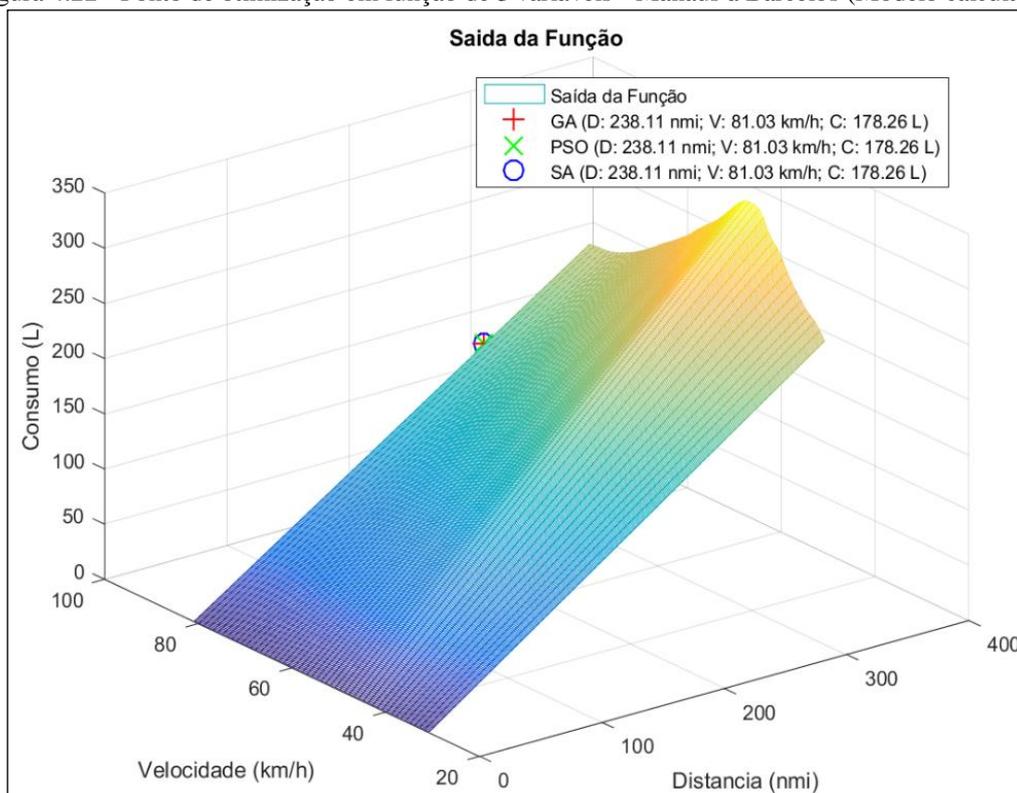
Figura 4.21 - Comportamento da curva de otimização - Manaus à Barcelos (Modelo calculado).



Fonte: AUTOR (2021).

Na Figura 4.22 é mostrado a relação entre distância medida em milhas náuticas e velocidade medida em km/h em função do consumo de combustível em litros, que é o resultado que se obtém através do modelo calculado, ou seja, a partir da função objetivo.

Figura 4.22 - Ponto de otimização em função de 3 variáveis - Manaus à Barcelos (Modelo calculado).



Fonte: AUTOR (2021).

A Figura 4.22 apresenta um gráfico de malha em que é possível identificar que a velocidade ideal alcançada pelo algoritmo híbrido é aproximadamente 81,03 km/h para uma distância de 238,11 milhas náuticas ou 440,98 km, na qual tem um gasto de combustível de 178,26 litros.

4.6.2 Cenário 2 (Manaus à Itacoatiara)

Para o cenário de Manaus à Itacoatiara foi possível alcançar os resultados contidos na Tabela 4.10, a velocidade média ideal alcançada para este cenário foi de aproximadamente 81 km/h, que gasta 78,94 litros de combustível neste percurso, é necessário evidenciar que este gasto de combustível é para o percurso de Manaus a Itacoatiara somente, e não de Manaus a Itacoatiara e Itacoatiara a Manaus.

Tabela 4.10 - Resultantes da simulação calculada de Manaus à Itacoatiara.

ID	Algoritmo	Paralelo	Tempo	Velocidade (km/h)	Consumo (L)
1	GA	TRUE	00:00:12	81,030869	78,946549
2	GA	FALSE	00:00:03	81,030869	78,946549
3	PSO	TRUE	00:00:04	81,030869	78,946549
4	PSO	FALSE	00:00:01	81,030869	78,946549
5	SA	FALSE	00:00:21	81,030869	78,946549

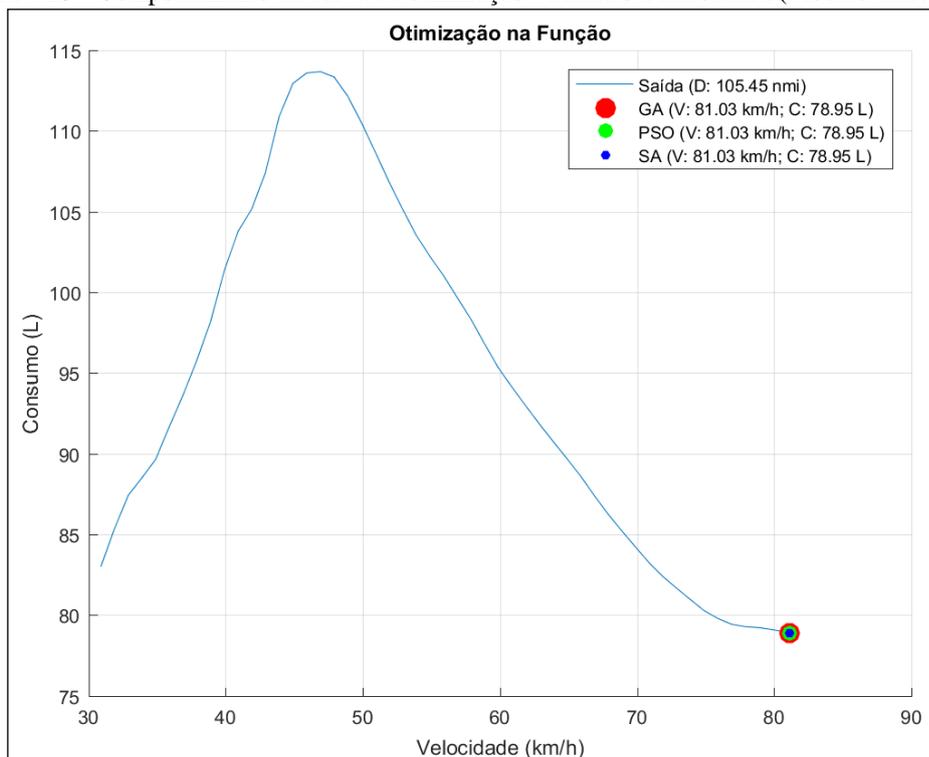
Fonte: AUTOR (2021).

Na Tabela 4.10 é possível afirmar que o algoritmo com a pior eficiência neste cenário foi o SA sem computação paralela (ID 5) com 21 segundos gastos para a convergência, em quarto lugar ficou o algoritmo GA com paralelismo (ID 1) com o tempo de 12 segundos, em terceiro está o PSO com paralelismo (ID 3) que obteve um tempo gasto de 4 segundos, enquanto que o segundo lugar ficou ocupado pelo GA sem paralelismo (ID 2), com 3 segundos de convergência. Logo, o vencedor que obteve o melhor tempo foi o PSO sem paralelismo (ID 1) com um tempo de convergência de 1 segundo.

Na Figura 4.23 é apresentado o comportamento da otimização com GA, PSO e SA, indicando que os valores encontrados da velocidade ideal são muito similares, e que por este motivo adotou-se o tempo de convergência como fator principal para escolher o algoritmo de otimização mais eficiente.

Tendo em vista que o consumo de combustível que é o alvo a ser minimizado foi próximo para todos, no gráfico não é possível mensurar valores tão próximos devido as casas decimais, por serem tão longas foram desconsideradas, adota-se para este cenário o valor de 81,03 km/h para todos os algoritmos testados (GA, PSO e SA).

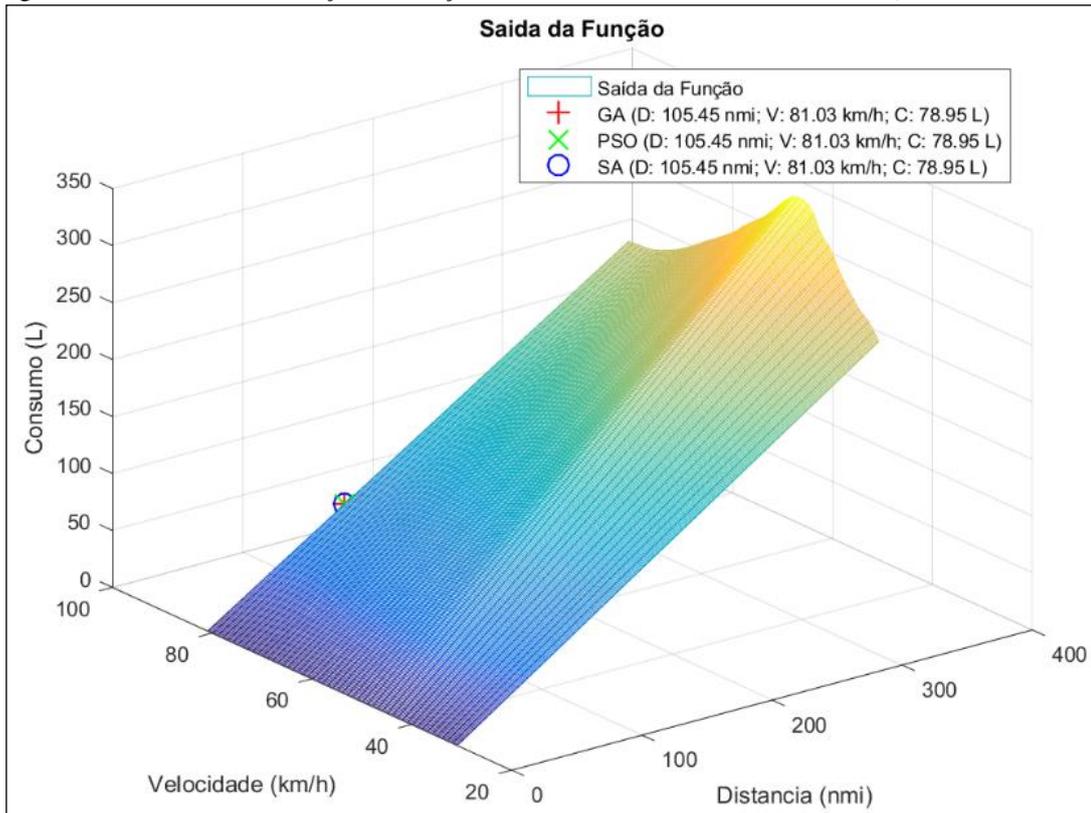
Figura 4.23 - Comportamento da curva de otimização - Manaus à Itacoatiara (Modelo calculado).



Fonte: AUTOR (2021).

Na Figura 4.24 é mostrado a relação entre distância medida em milhas náuticas e velocidade medida em km/h em função do consumo de combustível em litros, que é o resultado que se obtém através do modelo calculado, ou seja, a partir da função objetivo.

Figura 4.24 - Ponto de otimização em função de 3 variáveis - Manaus à Itacoatiara (Modelo calculado).



Fonte: AUTOR (2021).

A Figura 4.24 apresenta um gráfico de malha em que é possível identificar que a velocidade ideal alcançada pelo algoritmo híbrido é aproximadamente 81,03 km/h para uma distância de 105,45 milhas náuticas ou 195,30 km, na qual tem um gasto de combustível de 78,95 litros.

4.6.3 Cenário 3 (Manaus à Parintins)

Para o cenário de Manaus à Parintins foi possível alcançar os resultados contidos na Tabela 4.11, a velocidade média ideal alcançada para este cenário foi de aproximadamente 81,03 km/h, que gasta 172,33 litros de combustível neste percurso, é necessário evidenciar que este gasto de combustível é para o percurso de Manaus a Parintins somente, e não de Manaus a Parintins e Parintins a Manaus.

Tabela 4.11 - Resultantes da simulação calculada de Manaus à Parintins.

ID	Algoritmo	Paralelo	Tempo	Velocidade (km/h)	Consumo (L)
1	GA	TRUE	00:00:15	81,030869	175,335718
2	GA	FALSE	00:00:02	81,030869	175,335718
3	PSO	TRUE	00:00:04	81,030869	175,335718
4	PSO	FALSE	00:00:00	81,030869	175,335718
5	SA	FALSE	00:00:16	81,030869	175,335718

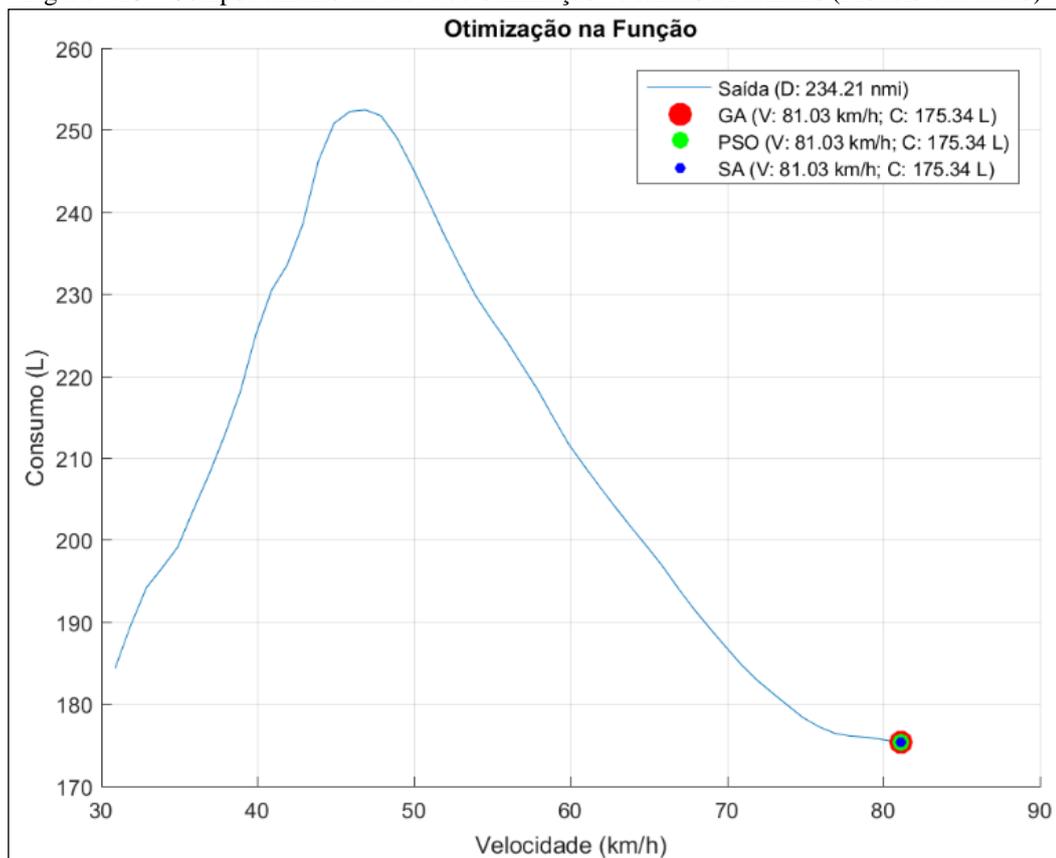
Fonte: AUTOR (2021).

Na Tabela 4.11 é possível afirmar que o algoritmo com a pior eficiência neste cenário foi o SA sem computação paralela (ID 5) com 16 segundos gastos para a convergência, em quarto lugar ficou o algoritmo GA com paralelismo (ID 1) com o tempo de 15 segundos, em terceiro está o PSO com paralelismo (ID 3) que obteve um tempo gasto de 4 segundos, enquanto que o segundo lugar ficou ocupado pelo GA sem paralelismo (ID 2), com 2 segundos de convergência. Logo, o vencedor que obteve o melhor tempo foi o PSO sem paralelismo (ID 1) com um tempo de convergência de 0 segundos.

Na Figura 4.25 é apresentado o comportamento da otimização com GA, PSO e SA, indicando que os valores encontrados da velocidade ideal são muito similares, e que por este motivo adotou-se o tempo de convergência como fator principal para escolher o algoritmo de otimização mais eficiente.

Tendo em vista que o consumo de combustível que é o alvo a ser minimizado foi próximo para todos, no gráfico não é possível mensurar valores tão próximos devido as casas decimais, por serem tão longas foram desconsideradas, adota-se para este cenário o valor de 81,03 km/h para todos os algoritmos testados (GA, PSO e SA).

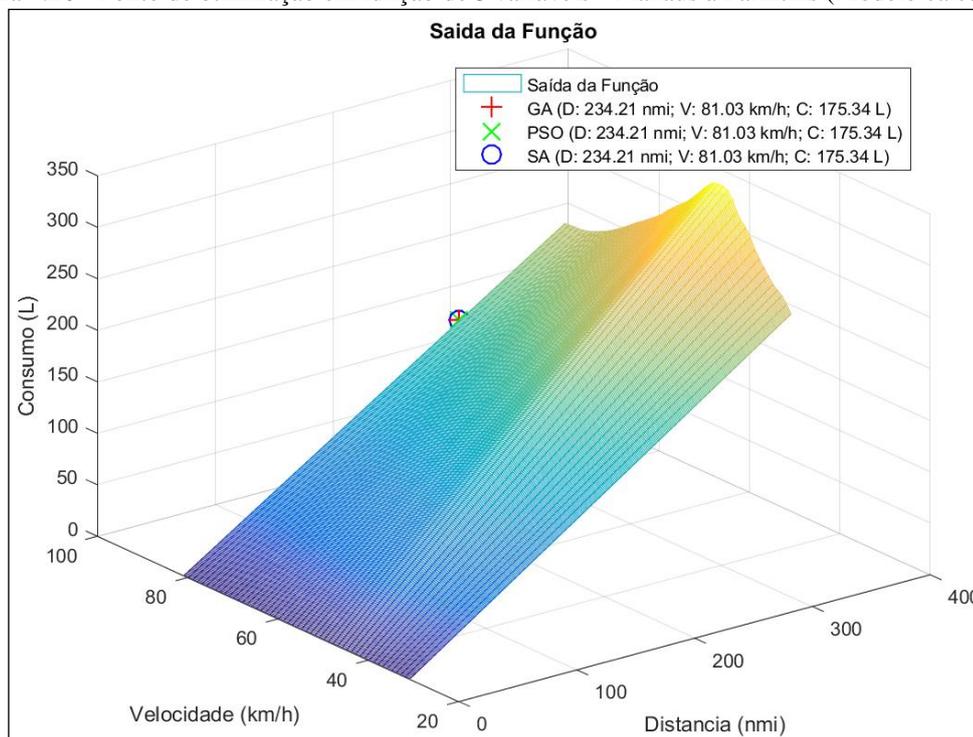
Figura 4.25 - Comportamento da curva de otimização - Manaus à Parintins (Modelo calculado).



Fonte: AUTOR (2021).

Na Figura 4.26 é mostrado a relação entre distância medida em milhas náuticas e velocidade medida em km/h em função do consumo de combustível em litros, que é o resultado que se obtém através do modelo calculado, ou seja, a partir da função objetivo.

Figura 4.26 - Ponto de otimização em função de 3 variáveis - Manaus à Parintins (Modelo calculado).



Fonte: AUTOR (2021).

A Figura 4.26 apresenta um gráfico de malha em que é possível identificar que a velocidade ideal alcançada pelo algoritmo híbrido é aproximadamente 81,03 km/h para uma distância de 234,21 milhas náuticas ou 433,75 km, na qual tem um gasto de combustível de 175,34 litros.

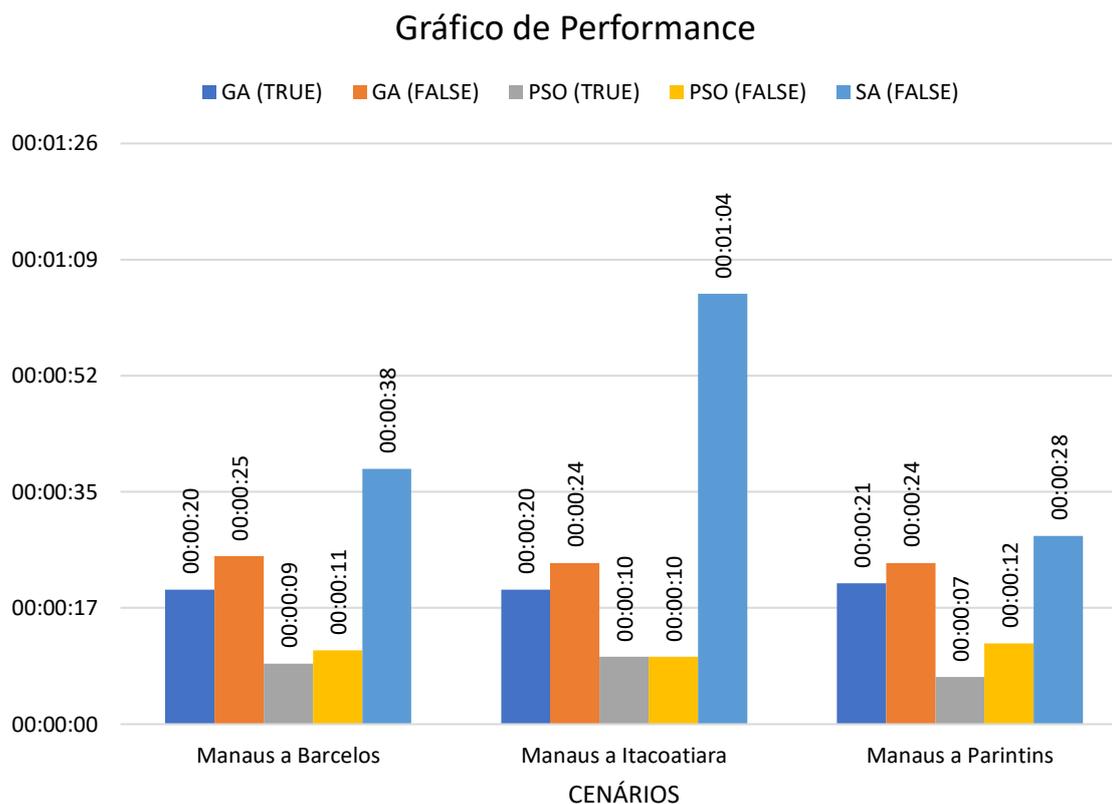
4.7 Resultados Finais

4.7.1 Resultados Finais das Simulações

Nesta seção é mostrado a comparação dos testes realizados de forma geral, no qual é possível verificar qual o algoritmo de otimização foi o melhor e conseqüentemente é o escolhido para o algoritmo híbrido.

No Gráfico 4.6 é apresentado a comparação dos cenários testados (Manaus a Barcelos, Manaus a Itacoatiara e Manaus a Parintins) com 500 dados.

Gráfico 4.6 - Comparação de performance – 500 dados.



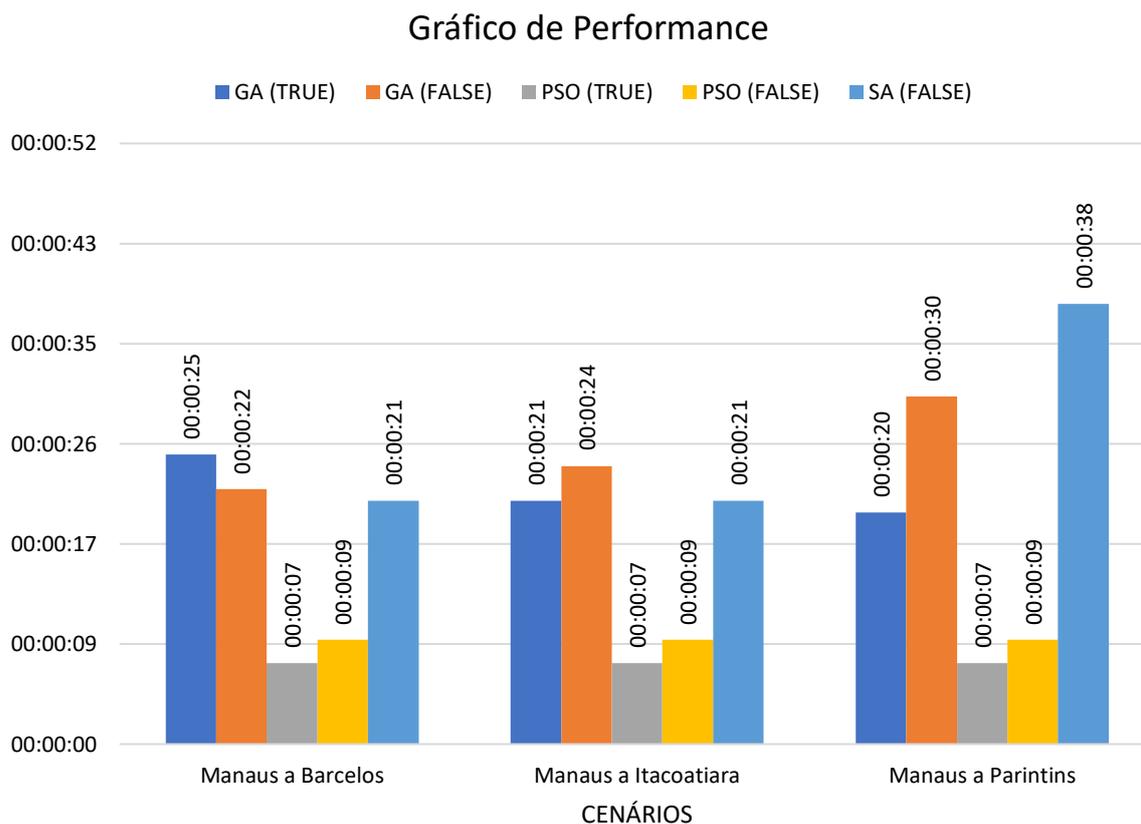
Fonte: AUTOR (2021).

No Gráfico 4.6 pode-se observar que o pior algoritmo foi o SA que não possui o recurso da computação paralela, atingindo nos três cenários testados um valor acima de todos os outros testados. Em quarto lugar ficou o algoritmo GA sem o paralelismo, ficou em quarto lugar para todos os cenários analisados com 500 dados utilizados para treinamento da rede neural. Já o mesmo algoritmo com o recurso de paralelismo obteve o terceiro lugar, e isso ocorreu também para todos os cenários testados. O segundo lugar ficou com o PSO sem a utilização de paralelismo, que no cenário de Manaus a Itacoatiara ficou empatado em primeiro com o PSO com paralelismo.

O melhor algoritmo de otimização testado foi o PSO utilizando o recurso de computação paralela, sendo o melhor em todos os cenários testados, conforme pode ser observado no Gráfico 4.6.

No Gráfico 4.7 é mostrado a comparação dos testes realizados nos cenários de Manaus a Barcelos, Manaus a Itacoatiara e Manaus a Parintins, no qual é comparado a performance dos algoritmos de otimização por meio do tempo de convergência.

Gráfico 4.7 - Comparação de performance – 10000 dados.



Fonte: AUTOR (2021).

No Gráfico 4.7 é possível observar que o pior algoritmo de otimização utilizado nos testes foi o SA, pois obteve resultados de convergência muito altos em comparação aos demais algoritmos. Em quarto lugar ficou o GA sem paralelismo, em terceiro o GA com paralelismo e em segundo o PSO sem o recurso de computação paralela.

O melhor algoritmo utilizado para os três cenários testados com a entrada de 10000 dados na rede neural, foi o PSO com a computação paralela.

Logo pode-se afirmar que o melhor algoritmo foi o PSO, tanto sem paralelismo quanto com paralelismo, sendo que quando o algoritmo utilizou a computação paralela, o custo computacional de convergência foi menor que os demais, sendo esta, a melhor configuração do algoritmo híbrido para encontrar a velocidade média que ao longo de um trajeto específico tende a fazer com que a embarcação consuma menos combustível.

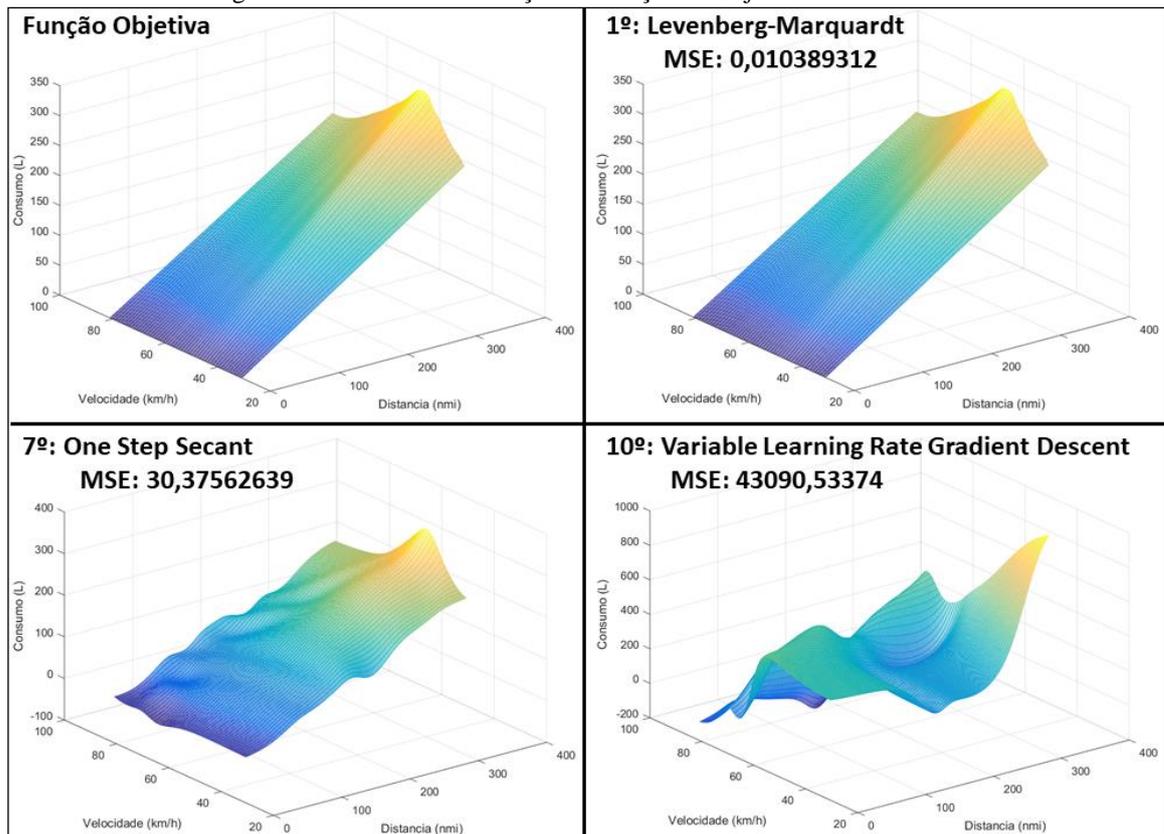
4.7.2 Resultados Finais do Modelo Calculado

Mediante os resultados apresentados neste capítulo, foi possível entender os procedimentos executados para se chegar nos resultados de otimização, contudo, a importância

do modelo de previsão com RNA possui grande importância no modelo de otimização, de tal forma que, o grau de distorção da curva objetiva aumenta em função do erro projetado na RNA.

Em virtude disso, a Figura 4.27 mostra uma comparação da curva de distorção resultantes de 3 algoritmos de treinamento mediante o resultado da função objetiva calculada. Nesta figura é possível identificar o quão importante é o ajuste de uma rede, de tal forma que quanto mais próximo de 0 melhor é a curva objetiva, os gráficos que apresentam a distorção em funções do algoritmo de treinamento variam as taxas de erros, fato que compromete o valor resultante da otimização.

Figura 4.27 - Curva de distorção em função do objetivo e taxa de erro.



Fonte: AUTOR (2021).

CAPÍTULO 5

5 CONCLUSÕES E SUGESTÕES

5.1 CONCLUSÕES

Apresente dissertação teve os objetivos estabelecidos inicialmente alcançados, no qual foi realizado o mapeamento das variáveis relativas ao motor, de forma a otimizar o consumo de combustível, depois deste mapeamento foi realizado o desenvolvimento do algoritmo híbrido que utilizou rede neural e algoritmos de otimização (GA, PSO e SA) para ajudar a encontrar a velocidade ótima a partir da distância e os dados de consumo do combustível, minimizando assim o consumo de combustível da embarcação pela velocidade média que menos consome combustível.

Devido ao algoritmo utilizar rede neural e algoritmos de otimização a rede neural foi testada e selecionado o melhor modelo para que fosse substituído na função objetivo dos algoritmos de otimização.

O modelo híbrido foi testado e os algoritmos de otimização GA PSO e SA foram tiveram seus desempenhos analisados de forma a verificar qual dos algoritmos teve o melhor resultado, esses testes foram analisados em três cenários distintos, sendo eles o cenário composto pelo percurso de Manaus a Barcelos, o segundo cenário de Manaus a Itacoatiara e o terceiro de Manaus a Parintins, no qual o melhor algoritmo de otimização do modelo híbrido foi o PSO com a utilização do recurso de computação paralela.

Conclui-se que o algoritmo teve bons resultados de otimização do consumo de combustível e que para cenários de longa distância que é o caso dos cenários testados, a metodologia empregada de utilizar IA para otimizar o consumo de combustível se torna uma alternativa aplicável no cenário do porto de Manaus, considerando embarcações de pequeno e médio porte, já que quanto maior é a distância maior é o consumo de combustível.

5.2 SUGESTÕES PARA TRABALHOS FUTUROS

As sugestões para trabalhos futuros estão pautadas principalmente nas limitações do estudo e que podem ser listados da seguinte forma:

- Aplicar o modelo híbrido para cenários com distancias maiores do que as testadas.
- Levar em consideração peso, resistência da água e tempo de travessia de um ponto A para um ponto B.

- Considerar características de motores de combustão interna das categorias marinizado e terrestres para encontrar a melhor eficiência em cada categoria.
- Considerar as variáveis de emissão de gases, velocidade média e custo de combustível como variável de saída e aplicar algoritmos multiobjectivos para encontrar o melhor ponto de otimização entre eles.

REFERÊNCIAS

- ABD ELAZIZ, Mohamed; EWEES, Ahmed A.; OLIVA, Diego. Hyper-heuristic method for multilevel thresholding image segmentation. **Expert Systems with Applications**, v. 146, p. 113201, 2020.
- ARORA, J. S. **EDITION, THIRD. INTRODUCTION TO OPTIMUM DESIGN**. Elsevier, 2012.
- ARRUDA JÚNIOR, Jurandi da Silva. Desenvolvimento de um método para redução do consumo de combustível no transporte rodoviário de cargas: capacitação, aplicações de torque e telemetria para veículos pesados. 2014.
- BASKOV, Vladimir; IGNATOV, Anton; POLOTNYANSCHIKOV, Vladislav. Assessing the influence of operating factors on the properties of engine oil and the environmental safety of internal combustion engine. **Transportation Research Procedia**, v. 50, p. 37-43, 2020.
- BERTONI JUNIOR, Ivan Luiz. Análise de desempenho e emissões de um motor ciclo diesel operando com fumigação de água e etanol super-hidratado. 2020.
- BOOB, Digvijay; DEY, Santanu S.; LAN, Guanghui. Complexity of training relu neural network. **Discrete Optimization**, p. 100620, 2020.
- BOUZON, M. F. Estudo de algoritmos de otimização inspirados na natureza aplicados ao treinamento de redes neurais artificiais. 2021.
- BRUNETTI, Franco. Introdução ao estudo dos motores de combustão interna. In: **MOTORES DE COMBUSTÃO INTERNA VOLUME 1**, São Paulo, Edgard Blücher Ltda. 2018.
- BUENO, Fabrício. Métodos Heurísticos. **Teoria e implementações. Araranguá: IFSC**, 2009.
- CHEN, Ming et al. Heuristic algorithms based on deep reinforcement learning for quadratic unconstrained binary optimization. **Knowledge-Based Systems**, v. 207, p. 106366, 2020.
- CHEN, Wei-Neng et al. Particle swarm optimization with an aging leader and challengers. **IEEE transactions on evolutionary computation**, v. 17, n. 2, p. 241-258, 2012.
- DADA, Emmanuel Gbenga et al. Predicting protein secondary structure based on ensemble Neural Network. **ITEGAM-JETIA**, v. 7, n. 27, p. 49-56, 2021.

DALPRÁ, Agostinho J. et al. Experimentação planejada para análise dos fatores que influenciam os pontos ótimos de funcionamento de um motor de combustão interna. 2020.

DEL VALLE, Yamille et al. Particle swarm optimization: basic concepts, variants and applications in power systems. **IEEE Transactions on evolutionary computation**, v. 12, n. 2, p. 171-195, 2008.

DELGADO-HIDALGO, Liliana; RAINWATER, Chase; NACHTMANN, Heather. A computational comparison of cargo prioritization and terminal allocation problem models. **Computers & Industrial Engineering**, v. 144, p. 106450, 2020.

DUFOUR, Jean-Marie; NEVES, Julien. Finite-sample inference and nonstandard asymptotics with Monte Carlo tests and R. In: **Handbook of Statistics**. Elsevier, 2019. p. 3-31.

FAGUNDEZ, J. L. S. et al. Joint use of artificial neural networks and particle swarm optimization to determine optimal performance of an ethanol SI engine operating with negative valve overlap strategy. **Energy**, v. 204, p. 117892, 2020.

GAO, K. Z. et al. A survey on meta-heuristics for solving disassembly line balancing, planning and scheduling problems in remanufacturing. **Swarm and Evolutionary Computation**, v. 57, p. 100719, 2020.

GONG, Yue-Jiao et al. Genetic learning particle swarm optimization. **IEEE transactions on cybernetics**, v. 46, n. 10, p. 2277-2290, 2015.

GONZALES, Gill Velleda; DOS SANTOS, Elizaldo Domingues; NETO, Antônio José Silva. Uma comparação entre os algoritmos de evolução diferencial e recozimento simulado associados ao design construtal para a otimização geométrica de uma cavidade em forma de duplo-t. **Revista Mundi Engenharia, Tecnologia e Gestão (ISSN: 2525-4782)**, v. 3, n. 2, 2018.

GUAZZELLI, Alex. Fundamentação de modelos de redes neurais e seus metodos de aplicação no reconhecimento de caracteres. **Salão de Iniciação Científica (3.: 1991: Porto Alegre).[Resumos]. Porto Alegre: UFRGS, 1991.**, 1991.

HATAMI, Mohammad; HASANPOUR, Maryam; JING, Dengwei. Recent developments of nanoparticles additives to the consumables liquids in internal combustion engines: Part I: Nano-fuels. **Journal of Molecular Liquids**, p. 114250, 2020.

HOSSEINIOUN, Pejman et al. A new energy-aware tasks scheduling approach in fog computing using hybrid meta-heuristic algorithm. **Journal of Parallel and Distributed Computing**, v. 143, p. 88-96, 2020.

HUANG, Renfang et al. Energy performance prediction of the centrifugal pumps by using a hybrid neural network. **Energy**, v. 213, p. 119005, 2020.

HUI, Tianyu; ZENG, Wenjie; YU, Tao. Core power control of the ADS based on genetic algorithm tuning PID controller. **Nuclear Engineering and Design**, v. 370, p. 110835, 2020.

JÚNIOR, JORGE DE ALMEIDA BRITO. **SOLUÇÃO PARA O DESPACHO ECONÔMICO AMBIENTAL DE UM SISTEMA DE GERAÇÃO TÉRMICA POR RECOZIMENTO SIMULADO**. 2018. Tese de Doutorado.

KIRKPATRICK, Scott; GELATT, C. Daniel; VECCHI, Mario P. Optimization by simulated annealing. **science**, v. 220, n. 4598, p. 671-680, 1983.

KOÇAK, Y., & ÜSTÜNDAĞ ŞIRAY, G. New activation functions for single layer feedforward neural network. **Expert Systems with Applications**, 164, 113977. doi:10.1016/j.eswa.2020.113977. 2021.

KUMAR, A. et al. On solving complex reliability optimization problem using multi-objective particle swarm optimization. In: **Mathematics Applied to Engineering**. Academic Press, 2017. p. 115-131.

LEITE, ALLAN RODRIGO. Um esquema para redução do consumo de combustível em sistemas de condução férrea baseado em otimização distribuída de restrição. **Curitiba: Pontifícia Universidade Católica do Paraná**, v. 85, 2009.

LOBATO, Fran Sérgio et al. Design and identification problems of rotor bearing systems using the simulated annealing algorithm. **Simulated annealing-single and multiple objective problems**, p. 197-16, 2012.

MARINI, Federico; WALCZAK, Beata. Particle swarm optimization (PSO). A tutorial. **Chemometrics and Intelligent Laboratory Systems**, v. 149, p. 153-165, 2015.

MARIO, Mauricio Conceição; DA SILVA FILHO, João Inácio; ABE, Jair Minoru. REDE NEURAL ARTIFICIAL HÍBRIDA-REDE PARA-NEURAL: IMPLEMENTAÇÃO DA

LÓGICA PARACONSISTENTE EM NEURÔNIOS ARTIFICIAIS. **ANAIS ASPECTOS DE SISTEMAS INTELIGENTES BASEADOS EM LÓGICAS ANOTADAS**, p. 192, 2021.

MARTINS, Murilo Battistuzzi et al. Otimização energética de um trator agrícola utilizando normas técnicas em operações de gradagem. **Revista Engenharia na Agricultura**, v. 26, n. 1, p. 52-57, 2018.

MCCULLOCH, Warren S.; PITTS, Walter. A logical calculus of the ideas immanent in nervous activity. **The bulletin of mathematical biophysics**, v. 5, n. 4, p. 115-133, 1943.

MENZEL, Germano et al. Multi-objective optimization of the volumetric and thermal efficiencies applied to a multi-cylinder internal combustion engine. **Energy Conversion and Management**, v. 216, p. 112930, 2020.

MOHAMMADI, Farzaneh et al. Modeling and sensitivity analysis of the alkylphenols removal via moving bed biofilm reactor using artificial neural networks: Comparison of levenberg marquardt and particle swarm optimization training algorithms. **Biochemical Engineering Journal**, v. 161, p. 107685, 2020.

NAKA, Shigenori et al. A hybrid particle swarm optimization for distribution state estimation. **IEEE Transactions on Power systems**, v. 18, n. 1, p. 60-68, 2003.

NASCIMENTO, Manoel Henrique Reis et al. New solution for resolution of the economic load dispatch by different mathematical optimization methods, turning off the less efficient generators. **Journal of Engineering and Tecnology for Industrial Applications**, v. 3, p. 10, 2017.

OSABA, Eneko et al. Community detection in networks using bio-inspired optimization: Latest developments, new results and perspectives with a selection of recent meta-heuristics. **Applied Soft Computing**, v. 87, p. 106010, 2020.

PIOTROWSKI, Adam P.; NAPIORKOWSKI, Jaroslaw J.; PIOTROWSKA, Agnieszka E. Population size in particle swarm optimization. **Swarm and Evolutionary Computation**, v. 58, p. 100718, 2020.

QIN, Quande et al. Particle swarm optimization with interswarm interactive learning strategy. **IEEE transactions on cybernetics**, v. 46, n. 10, p. 2238-2251, 2015.

RAMUJANE, Nelson Maneira. **Desenvolvimento de um algoritmo em matlab para determinar coeficientes hidrodinâmicos pelo método de mapeamento conforme multiparâmetro**. 2019. Tese de Doutorado.

RAUBER, Thomas Walter. Redes neurais artificiais. **Universidade Federal do Espírito Santo**, p. 29, 2005.

ROME2RIO. <https://www.rome2rio.com/pt/>. Consultado em: 08/04/2021.

ROSENBLATT, Frank. The perceptron: a probabilistic model for information storage and organization in the brain. **Psychological review**, v. 65, n. 6, p. 386, 1958.

RUFINO, Caio Henrique et al. Conceptual study of an internal combustion engine with adjustable cubic capacity and compression ratio: Estudo conceitual de um motor com cilindrada e taxa de compressão ajustáveis. 2020.

RUMELHART, David E.; HINTON, Geoffrey E.; WILLIAMS, Ronald J. Learning representations by back-propagating errors. **nature**, v. 323, n. 6088, p. 533-536, 1986.

SCANIA. DC09 074A. 232 kW (315 hp) EU Stage II, China Phase II and Russia Stage I. 2020.

SCHELLENBERG, Christoph; LOHAN, John; DIMACHE, Laurentiu. Comparison of metaheuristic optimisation methods for grid-edge technology that leverages heat pumps and thermal energy storage. **Renewable and Sustainable Energy Reviews**, v. 131, p. 109966, 2020.

SHEYKHI, Mohammad et al. Investigation of the effects of operating parameters of an internal combustion engine on the performance and fuel consumption of a CCHP system. **Energy**, v. 211, p. 119041, 2020.

SHI, Yuhui; EBERHART, Russell. A modified particle swarm optimizer. In: **1998 IEEE international conference on evolutionary computation proceedings. IEEE world congress on computational intelligence (Cat. No. 98TH8360)**. IEEE, 1998. p. 69-73.

SILVA, Jean da Silva de Abreu et al. PROPOSTA DE IMPLANTAÇÃO DE SISTEMA DE PROTEÇÃO CONTRA POTENCIAL DE FALHA DO MOTOR À DIESEL (DISPARO DO MOTOR). **ITEGAM-JETIA**, v. 5, n. 19, p. 06-11, 2019.

SINGH, Akhilendra Pratap; KUMAR, Vikram; AGARWAL, Avinash Kumar. Evaluation of comparative engine combustion, performance and emission characteristics of low temperature combustion (PCCI and RCCI) modes. **Applied Energy**, v. 278, p. 115644, 2020.

SOARES, Carlos Alberto Santana. Elementos de Cálculo I-Notas de Aulas I Conjuntos, Retas e Circunferências. 2019.

TAN, Roy; DURU, Okan; THEPSITHAR, Prapisala. Assessment of relative fuel cost for dual fuel marine engines along major Asian container shipping routes. **Transportation Research Part E: Logistics and Transportation Review**, v. 140, p. 102004, 2020.

TIMANÁ-PEÑA, Jimena Adriana; COBOS-LOZADA, Carlos Alberto; TORRES-JIMENEZ, Jose. Metaheuristic algorithms for building Covering Arrays: A review. **Revista Facultad de Ingeniería**, v. 25, n. 43, p. 31-45, 2016.

WANG, Dongshu; TAN, Dapei; LIU, Lei. Particle swarm optimization algorithm: an overview. **Soft Computing**, v. 22, n. 2, p. 387-408, 2018.

WANG, Xiaoming et al. Meta-heuristics for unrelated parallel machines scheduling with random rework to minimize expected total weighted tardiness. **Computers & Industrial Engineering**, v. 145, p. 106505, 2020.

XAVIER, Reinaldo S. et al. Heurística para modelagem e minimização do consumo de combustível para rotas de coleta de lixo. **Bento Gonçalves**, v. 12, 2010.

XIAO, Yiyong et al. Development of a fuel consumption optimization model for the capacitated vehicle routing problem. **Computers & operations research**, v. 39, n. 7, p. 1419-1431, 2012.

YANG, Xin-She. **Introduction to algorithms for data mining and machine learning**. Academic press, 2019.

ZF. Reversor Marítimo Marine Gearbox ZF W220. 2020.

APÊNDICES

APÊNDICE A – ARTIGO PUBLICADO

Artigo publicado na revista International Journal for Innovation Education and Research (IJIER) – ISSN: 2411-2933, com Qualis Capes A2.

“Computational meta-heuristics based on Machine Learning to optimize fuel consumption of vessels using diesel engines”

DOI: <https://doi.org/10.31686/ijier.vol9.iss5.3128>

Paulo Oliveira Siqueira Junior, Manoel Henrique Reis Nascimento, Ítalo Rodrigo Soares Silva, Ricardo Silva Parente, Milton Fonseca Júnior e Jandecy Cabral Leite

Publicado em: 01/05/2021.