

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
ENGENHARIA DE CONTROLE E AUTOMAÇÃO

FABIO AUGUSTO GUIDOTTI DOS SANTOS
VICTOR HIDEKI YOSHIZUMI

**EVOLUÇÃO DA ARQUITETURA DE PERCEPTRON MULTICAMADAS BASEADA
EM OTIMIZAÇÃO POR ENXAME DE PARTÍCULAS E COLÔNIA ARTIFICIAL DE
ABELHAS**

TRABALHO DE CONCLUSÃO DE CURSO

CORNÉLIO PROCÓPIO
2018

FABIO AUGUSTO GUIDOTTI DOS SANTOS
VICTOR HIDEKI YOSHIKUMI

**EVOLUÇÃO DA ARQUITETURA DE PERCEPTRON MULTICAMADAS BASEADA
EM OTIMIZAÇÃO POR ENXAME DE PARTÍCULAS E COLÔNIA ARTIFICIAL DE
ABELHAS**

Trabalho de Conclusão de Curso de graduação, apresentado à disciplina Trabalho de Conclusão de Curso 2, do curso de Engenharia de Controle e Automação da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para a obtenção do título de Bacharel.

Orientador: Prof. Dra. Maria Eugenia Dajer
Coorientador: Prof. Dr. Danilo Hernane Spatti

CORNÉLIO PROCÓPIO
2018



Universidade Tecnológica Federal do Paraná
Campus Cornélio Procópio
Departamento Acadêmico de Elétrica
Curso de Engenharia de Controle e Automação



FOLHA DE APROVAÇÃO

Fabio Augusto Guidotti dos Santos

Evolução da arquitetura de Perceptron Multicamadas baseada em Otimização por Exame de Partículas e Colônia Artificial de Abelhas

Trabalho de conclusão de curso apresentado às 16:00hs do dia 23/11/2018 como requisito parcial para a obtenção do título de Engenheiro de Controle e Automação no programa de Graduação em Engenharia de Controle e Automação da Universidade Tecnológica Federal do Paraná. O candidato foi arguido pela Banca Avaliadora composta pelos professores abaixo assinados. Após deliberação, a Banca Avaliadora considerou o trabalho aprovado.

Prof(a). Dr(a). María Eugenia Dajer - Presidente (Orientador)

Prof(a). Dr(a). Danilo Hernane Spatti - (Coorientador)

Prof(a). Dr(a). Danilo Sipoli Sanches - (Membro)

Prof(a). Dr(a). Wagner Fontes Godoy - (Membro)



Universidade Tecnológica Federal do Paraná
Campus Cornélio Procópio
Departamento Acadêmico de Elétrica
Curso de Engenharia de Controle e Automação



FOLHA DE APROVAÇÃO

Victor Hideki Yoshizumi

Evolução da arquitetura de Perceptron Multicamadas baseada em Otimização por Enxame de Partículas e Colônia Artificial de Abelhas

Trabalho de conclusão de curso apresentado às 16:00hs do dia 23/11/2018 como requisito parcial para a obtenção do título de Engenheiro de Controle e Automação no programa de Graduação em Engenharia de Controle e Automação da Universidade Tecnológica Federal do Paraná. O candidato foi arguido pela Banca Avaliadora composta pelos professores abaixo assinados. Após deliberação, a Banca Avaliadora considerou o trabalho aprovado.

Prof(a). Dr(a). María Eugenia Dajer - Presidente (Orientador)

Prof(a). Dr(a). Danilo Hernane Spatti - (Coorientador)

Prof(a). Dr(a). Wagner Fontes Godoy - (Membro)

Prof(a). Dr(a). Danilo Sipoli Sanches - (Membro)

Dedicamos este trabalho primeiramente à Deus, à nossa família e a todos que fizeram parte da nossa jornada até aqui.

AGRADECIMENTOS

Agradecemos primeiramente à Deus por seu infinito amor, nos dando o dom da vida e abençoando os nossos caminhos durante a realização deste trabalho.

Aos nossos pais que sempre estiveram ao nosso lado e enfrentaram tantas dificuldades, fazendo tudo o que fosse possível para que conseguíssemos chegar até este momento e a todos os familiares pelo amor, apoio e incentivo.

Aos nossos orientadores, Maria Eugenia Dajer e Danilo Hernane Spatti, por nos auxiliarem durante esta jornada, fornecendo todo suporte e ajuda necessária. Agradecemos pelos ensinamentos, carinho, paciência compreensão e principalmente pela confiança depositada em nós.

Aos amigos e colegas que sempre estiveram presentes, que não negaram esforços e ficaram na torcida, nosso muito obrigado pelas cachaças e companheirismo. Sem vocês, os momentos que passamos na universidade não teriam sido tão incríveis como foram, principalmente as sextas-feiras insanas na República Tortuga.

Ao Grupo de Estudos em Sinais Biológicos (GE4Bio) pelas constantes colaborações, genialidade e amizade.

A Rami N. Khushaba, PhD, por disponibilizar a base de dados de sinais eletromiográficos utilizada.

Agradecemos a Universidade Tecnológica Federal do Paraná por conceder as ferramentas necessárias para produção deste trabalho.

Enfim, a todos os que contribuíram para a realização deste trabalho, deixamos aqui os nossos mais sinceros agradecimentos.

“Talvez não tenha conseguido fazer o melhor, mas lutei para que o melhor fosse feito. Não sou o que deveria ser, mas graças a Deus, não sou mais o que era antes.”

Martin Luther King

RESUMO

DOS SANTOS, Fabio A. G. e YOSHIZUMI, Victor H. **Evolução da arquitetura de Perceptron Multicamadas baseada em Otimização por Enxame de Partículas e Colônia Artificial de Abelhas**. 2018. 93 f. Trabalho de Conclusão de Curso (Graduação) – Engenharia de Controle e Automação. Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2018.

O presente trabalho se propõe a desenvolver dois sistemas inteligentes híbridos baseados em Otimização por Enxame de Partículas e Colônia Artificial de Abelhas, capazes de realizar a otimização da arquitetura de redes neurais artificiais do tipo *Perceptron* multicamadas. Os parâmetros internos da arquitetura escolhidos para otimização, foram os que possuem maior influência no desempenho da rede, sendo eles: número de camadas escondidas, quantidade de neurônios por camada e a taxa de aprendizado. Desta forma, para realizar o processo de escolha da topologia da rede que melhor se adequa ao conjunto de dados a serem classificados, os algoritmos bioinspirados utilizam como *fitness* o erro quadrático médio gerado pelo *Perceptron* multicamadas na fase de teste. A obtenção e análise dos resultados dos sistemas propostos foram realizadas em duas etapas: validação e teste, onde foram treinadas um total de 192 mil redes. Durante a etapa de validação foram utilizadas as bases de dados de diagnósticos médicos “*Breast Cancer*”, “*Mammographic Mass*” e “*Thyroid*” obtidas a partir do repositório de aprendizado de máquina, UCI. Enquanto, para a fase de teste foram utilizadas as bases de dados de sinais eletromiográficos “*Classes de Movimentos*” e “*Níveis de Contração*”, tratados utilizando a Transformada *Wavelet Packet* da família Haar. A partir dos resultados obtidos foi possível verificar a eficiência de ambos os sistemas, que se apresentam como ótimas ferramentas para auxiliar profissionais responsáveis pelo projeto de redes neurais artificiais, capazes de se adaptar a qualquer classe de problemas envolvendo a otimização da topologia de redes do tipo *Perceptron* multicamadas.

Palavras-chave: Evolução de Arquitetura de *Perceptron* Multicamadas. Otimização por Enxame de Partículas. Colônia Artificial de Abelhas. Transformada *Wavelet Packet*. Sinais Eletromiográficos.

ABSTRACT

DOS SANTOS, Fabio A. G. e YOSHIZUMI, Victor H. **Evolving Multilayer Perceptron architecture based on Particle Swarm Optimization and Artificial Bee Colony.** 2018. 93 f. Trabalho de Conclusão de Curso (Graduação) – Engenharia de Controle e Automação. Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2018.

The present work proposes to develop two hybrid intelligent systems based on Particle Swarm Optimization and Artificial Bee Colony, capable of realizing the optimization of the architecture of artificial neural networks of the multilayer Perceptron type. The internal parameters of the architecture chosen for optimization were those that have the greatest influence on the performance of the network, being: number of hidden layers, number of neurons per layer and learning rate. In this way, to perform the choosing process of the network topology that best fits the set of data to be classified, the bioinspired algorithms use as fitness the average quadratic error generated by the multilayer Perceptron in the test phase. The obtaining and analyze of the proposed systems results were performed in two steps: validation and test, where a total of 192 thousand networks were trained. During the validation stage, the databases of medical diagnoses "Breast Cancer", "Mammographic Mass" and "Thyroid" obtained from the machine learning repository, UCI, were used. While, for the test phase, the databases of electromyographic signals "Classes of Movements" and "Levels of Contraction" were used, treated using the Wavelet Packet Transform of the Haar family. From the obtained results it was possible to verify the efficiency of both systems, which are presented as great tools to assist professionals responsible for the design of artificial neural networks capable of adapting to any class of problems involving the optimization of multilayer Perceptron type network topology.

Keywords: Evolving Multilayer Perceptron Architecture. Particle Swarm Optimization. Artificial Bee Colony. Wavelet Packet Transform. Electromyographic signals.

LISTA DE FIGURAS

Figura 1 - Neurônio Artificial.....	21
Figura 2 - Ilustração de rede <i>Perceptron</i> multicamadas.....	26
Figura 3 - Fases de treinamento da rede MLP.....	27
Figura 4 - Fatores influenciáveis em $x_i + 1_j$	30
Figura 5 - Modelos de topologia PSO.....	31
Figura 6 - Pseudocódigo do algoritmo PSO.....	32
Figura 7 – Função Sigmoide.....	33
Figura 8 - Comportamento das abelhas na busca de néctar.....	36
Figura 9 - Pseudocódigo do algoritmo ABC.....	39
Figura 10 - Três classes de sistemas híbridos.....	42
Figura 11 - Geração do potencial de ação da unidade motora.....	44
Figura 12 - Decomposição da WPT em árvore binária.....	46
Figura 13 - Exemplo de janelamento.....	47
Figura 14 - Configuração dos eletrodos EMG e do acelerômetro.....	51
Figura 15 - Orientações do antebraço.....	52
Figura 16 – Limites máximos para a taxa de aprendizado.....	56
Figura 17 - Pseudocódigo do algoritmo PSO implementado.....	58
Figura 18 - Exemplo do método de escolha da fonte a ser explorada.....	59
Figura 19 – Pseudocódigo do cálculo da dissimilaridade.....	60
Figura 20 - Pseudocódigo do algoritmo ABC implementado.....	61
Figura 21 – Representação binária das variáveis otimizadas.....	62
Figura 22 - Fluxograma de execução do PSO.....	64
Figura 23 - Fluxograma de execução do ABC.....	65
Figura 24 – Fluxograma das etapas de validação e teste.....	67
Figura 25 – Análise comparativa entre os sistemas - <i>Breast Cancer</i>	70
Figura 26 – Análise comparativa entre os sistemas - <i>Mammographic Mass</i>	70
Figura 27 - Análise comparativa entre os sistemas – <i>Thyroid</i>	71
Figura 28 - Análise comparativa entre os sistemas – Tempo.....	72
Figura 29 – Análise comparativa das configurações – ABC-MLP.....	74
Figura 30 – Análise comparativa das configurações – PSO-MLP.....	75
Figura 31 – Análise comparativa de desempenho evolutivo – Melhores configurações.....	76
Figura 32 - Análise comparativa de desempenho evolutivo.....	79

LISTA DE TABELAS

Tabela 1 – Faixas de saturação das saídas da RNA	55
Tabela 2 – Restrições dos parâmetros de otimização.....	63
Tabela 3 - Configurações dos algoritmos para validação dos sistemas	68

LISTA DE QUADROS

Quadro 1 – Trabalhos Relacionados.....	49
Quadro 2 – Características das bases de dados utilizadas no trabalho.....	53
Quadro 3 – Melhor arquitetura encontrada - Fitness.....	73
Quadro 4 – Análise comparativa entre os sistemas – Classes de Movimentos.	77
Quadro 5 – Análise comparativa entre os sistemas – Níveis de Contração.....	77
Quadro 6 – Melhor arquitetura encontrada - Fitness.....	78
Quadro 7 – Matriz confusão.....	80
Quadro 8 – Matriz confusão.....	80
Quadro 9 – Matriz confusão.....	81
Quadro 10 – Matriz confusão.....	81
Quadro 11 – Melhor arquitetura encontrada - Erro.....	82

LISTA DE SIGLAS

ABC	Colônia Artificial de Abelhas
EMG	Eletromiográfico
MLP	Perceptron Multicamadas
PSO	Otimização por Enxame de Partículas
RNA	Rede Neural Artificial
SI	Sistema Inteligente
SIH	Sistema Inteligente Híbrido
WT	Transformada Wavelet
WPT	Transformada Wavelet Packet

SUMÁRIO

1	INTRODUÇÃO E JUSTIFICATIVA	15
1.1	Objetivos	16
1.1.1	Objetivo geral	16
1.1.2	Objetivos específicos	17
1.2	Organização do texto	17
2	FUNDAMENTAÇÃO TEÓRICA	18
2.1	Sistemas inteligentes	18
2.1.1	Redes neurais artificiais	18
2.1.2	Otimização por Enxame de Partículas	28
2.1.3	Colônia Artificial de Abelhas	34
2.2	Sistemas inteligentes híbridos	41
2.3	Eletromiografia	42
2.4	Transformada Wavelet Packet	44
2.5	Trabalhos relacionados	47
3	MATERIAIS E MÉTODOS	50
3.1	Materiais	50
3.1.1	Banco de dados	50
3.1.2	Ferramenta utilizada	53
3.2	Métodos	53
3.2.1	Perceptron multicamadas	54
3.2.2	Otimização por Enxame de Partículas	56
3.2.3	Colônia Artificial de Abelhas	58
3.2.4	Sistema inteligente híbrido	62
3.2.5	Tratamento do banco de dados eletromiográficos	66
3.2.6	Validação e testes	67
4	RESULTADOS	69
4.1	Resultados de validação	69
4.2	Resultados de teste	77
5	ANÁLISE E DISCUSSÃO	83
5.1	Análise dos resultados	83
5.2	Discussão	84
6	CONCLUSÕES E TRABALHOS FUTUROS	86
6.1	Conclusões	86
6.2	Perspectivas futuras	87
7	TRABALHOS REALIZADOS	88
7.1	Publicados	88
7.2	Aceitos para publicação	88
	REFERÊNCIAS	89

1 INTRODUÇÃO E JUSTIFICATIVA

Atualmente, existem inúmeras pesquisas que têm como foco a classificação de padrões, sejam elas sobre dados estatísticos (YI; ZHENG; GUO, 2013), controle populacional (DOLEZEL; SKRABANEK; GAGO, 2016), identificação de movimentos (YIN; GELENBE, 2017) ou focadas em patologias e outras questões relacionadas à saúde (RAJAGURU; PRABHAKAR, 2017; DA SILVA *et al.*, 2017). Deste modo, as Redes Neurais Artificiais (RNAs) emergem como uma poderosa ferramenta para solucionar uma ampla gama de problemas, podendo ser consideradas até mesmo como essenciais em aplicações como o reconhecimento de padrões (RAMCHOUN *et al.*, 2016).

Um dos modelos mais populares e conhecidos de RNAs é o *Perceptron* multicamadas (MLP, do inglês *Multilayer Perceptron*), caracterizado pela presença de uma ou mais camadas escondidas de neurônios (SILVA; SPATTI; FLAUZINO, 2010, p. 91). No entanto, para que uma rede MLP apresente um bom desempenho é necessário definir uma boa topologia para a mesma, esta determinação abrange diversos parâmetros como a quantidade de camadas escondidas a serem utilizadas, o número de neurônios por camada, taxa de aprendizado, função de ativação, dentre outros. Em geral, esta escolha é feita por um especialista através de experimentação, técnica conhecida na literatura como método empírico, sendo caracterizado como um processo tedioso e demorado. Sendo assim, além de consumir tempo e exigir experiência do responsável por esta tarefa, uma rede mal configurada pode resultar em fenômenos conhecidos como *underfitting* e *overfitting*, acarretando em uma baixa taxa de acertos e no aumento da carga computacional exigida durante a operação (BELOV; ZOLOTOV, 2015; SHAHRAKI; ZAHIRI, 2017).

Afim de contornar essa dificuldade, métodos envolvendo rotinas automatizadas e sistemas otimizados para a determinação da arquitetura de RNAs, vem sendo utilizados para melhorar o desempenho das mesmas. Em Binoti, Da Silva Binoti e Leite (2014) utiliza-se um método com rotinas automatizadas para encontrar a melhor configuração para arquitetura de uma RNA. Desta forma, é verificado que para espaços de busca tecnicamente pequenos, a quantidade de combinações possíveis para os parâmetros a serem otimizados resulta em um gigantesco número de redes a serem treinadas, tornando sua aplicação inviável em casos de alta complexidade. Para encontrar bons resultados em problemas mais complexos, os

métodos envolvendo sistemas otimizados permitem aumentar este espaço de busca sem inviabilizar sua aplicação, onde o tempo necessário para evoluir uma arquitetura adaptada a um problema específico é plenamente justificável, quando comparado ao tempo gasto num procedimento empírico de “tentativa e erro” (CARVALHO; RAMOS; CHAVES, 2010). Deste modo, a partir da utilização de técnicas que abordam as RNAs em conjunto com algoritmos bioinspirados é possível obter ganhos significativos em relação a sistemas não otimizados (BARROS *et al.*, 2017; LIMA *et al.* 2017).

Dentre as abordagens comumente utilizadas para realizar esta tarefa, os algoritmos genéticos e suas variações já foram amplamente investigados. No entanto, o algoritmo de Otimização por Enxame de Partículas (PSO, do inglês *Particle Swarm Optimization*) proposto por Kennedy e Eberhart (1995) é uma ferramenta pouco explorada para ajuste de pesos e menos ainda, para definição de arquiteturas de redes MLP (CARVALHO, 2007). Através da literatura, verifica-se que o algoritmo de Colônia Artificial de Abelhas (ABC, do inglês *Artificial Bee Colony*), que é uma entre as várias ferramentas bioinspiradas emergentes na área de otimização apresentada por Karaboga (2005), também é pouco utilizada para solução deste tipo de problema. Desta forma, a seleção dos algoritmos foi realizada com base nos resultados promissores apresentados por ambos, como visto em Carvalho (2007), Çam *et al.* (2015) e Shahraki e Zahiri (2017).

Sendo assim, este trabalho se propõe a desenvolver dois sistemas inteligentes híbridos (SIH) para auxiliar na otimização da arquitetura das redes e treiná-las. Um desses sistemas estará composto por RNA e PSO e o outro combinará RNA e ABC. Ambos os sistemas foram validados utilizando 3 diferentes bases de dados obtidas a partir do repositório de aprendizado de máquina, UCI. Posteriormente, foi selecionado a melhor configuração de cada sistema para ser utilizada na classificação de sinais de EMG de 6 diferentes classes de movimentos da mão e os 3 níveis de contração muscular, tratados utilizando Transformadas *Wavelets Packet* (WPT, do inglês *Wavelet Packet Transform*) com o intuito de verificar seus desempenhos em problemas específicos.

1.1 Objetivos

1.1.1 Objetivo geral

Desenvolver dois sistemas inteligentes híbridos capazes de otimizar os parâmetros da arquitetura de uma rede neural artificial do tipo MLP, tornando-a adaptada a problemas específicos e facilitando a tarefa do projetista no processo de configuração da rede.

1.1.2 Objetivos específicos

1. Implementar o *Perceptron* multicamadas;
2. Implementar os algoritmos bioinspirados de Otimização por Enxame de Partículas e Colônia Artificial de Abelhas;
3. Evoluir a arquitetura da MLP a partir da implementação do sistema inteligente híbrido;
4. Validar os sistemas implementados a partir de bases de dados do repositório de aprendizado de máquina, UCI;
5. Analisar e discutir os resultados obtidos na fase de validação, para definir qual configuração apresenta o melhor desempenho médio em cada sistema para serem utilizadas na aplicação com os sinais EMG;
6. Realizar o processamento das bases de dados de sinais EMG, utilizando Transformada *Wavelet Packet*;
7. Verificar o desempenho dos sistemas na solução de problemas específicos, utilizando as bases de dados de EMG.

1.2 Organização do texto

Este trabalho está organizado em 7 capítulos. Neste capítulo introdutório, foram apresentados a motivação e os objetivos do mesmo. No capítulo 2 são expostos os fundamentos teóricos, divididos em 5 seções que tratam a respeito dos assuntos abordados, para proporcionar um melhor entendimento da teoria que cerca este trabalho. No capítulo 3, encontram-se descritos os materiais e métodos necessários para realizar implementação, bem como a validação e teste dos sistemas. O capítulo 4 apresenta os resultados obtidos no presente trabalho. O capítulo 5 foi dedicado a análise dos resultados e discussão do trabalho desenvolvido. No capítulo 6 são expostas as considerações finais e as perspectivas futuras a respeito do mesmo. Por fim, o capítulo 7 é dedicado a apresentar os trabalhos realizados pelos autores.

2 FUNDAMENTAÇÃO TEÓRICA

Faz-se necessário introduzir conceitos relacionados ao tema proposto, revisando a teoria já consolidada na literatura. Sendo assim, este capítulo dedica-se em apresentar ao leitor a teoria de Sistemas Inteligentes (SIs), detalhando especialmente as redes neurais artificiais, os algoritmos de Otimização por Enxame de Partículas, Colônia Artificial de Abelhas e os Sistemas Inteligentes Híbridos. Também é apresentado os conceitos básicos relacionados a eletromiografia e a área de processamento digital de sinais a partir da Transformada *Wavelet Packet*, permitindo o entendimento necessário para a aplicação das ferramentas utilizadas, finalizando com a apresentação de alguns trabalhos correlatos.

2.1 Sistemas inteligentes

Os Sistemas Inteligentes apresentam pontos-chaves como a habilidade de usar o conhecimento para realizar tarefas, além de trabalhar com problemas complexos que se assemelham a problemas reais através da capacidade de se aproveitar de associações e inferências. Desta forma, para que um sistema computacional seja “inteligente” ele precisa possuir habilidades como a de armazenar e recuperar de maneira eficiente uma grande quantidade de informação, possibilitando usá-las para tomar decisões e resolver problemas (REZENDE, 2003).

Ainda em Rezende (2003), é apresentado as principais técnicas e metodologias usadas pelos SIs como sendo: Aquisição de Conhecimento, Aprendizado de Máquina, Redes Neurais, Lógica *Fuzzy*, Computação Evolutiva, Agentes e Multiagentes, Mineração de Dados e de Textos. Onde, cada uma dessas técnicas é capaz de oferecer uma variedade de graus de habilidade para representar o conhecimento humano. Entretanto, este trabalho se propõe a realizar uma abordagem utilizando RNAs em conjunto com os algoritmos de Otimização por Enxame de Partículas e Colônia Artificial de Abelhas (pertencentes a classe de CE).

2.1.1 Redes neurais artificiais

Como retratado em Silva, Spatti e Flauzino (2010, p. 24), RNAs são modelos computacionais que têm como inspiração o sistema nervoso dos seres vivos.

As mesmas são capazes de realizar a aquisição e manutenção do conhecimento, sendo compostas por um conjunto de unidades de processamento, chamadas de neurônios artificiais, interligados por um grande número de interconexões.

Em Haykin (2007, p. 28), uma rede neural é apresentada, em sua forma mais geral, como uma máquina que é projetada para modelar a maneira como o cérebro realiza uma tarefa ou função de interesse, e então definida da seguinte maneira:

Uma rede neural artificial é um processador maciçamente paralelamente distribuído constituído de unidade de processamento simples, que têm a propensão natural para armazenar conhecimentos experimental e torná-lo disponível para o uso. Ela se assemelha ao cérebro “humano” em dois aspectos:

- 1- O conhecimento é adquirido pela rede a partir de seu ambiente através de um processo de aprendizagem.
- 2- Forças de conexão entre neurônios, conhecidas como pesos sinápticos, são utilizadas para armazenar o conhecimento adquirido.

A aplicação de RNAs envolve características e capacidades bastantes relevantes, dentre essas Haykin (2007, p. 29) e Silva, Spatti e Flauzino (2010, p. 24) destacam:

- **Não-linearidade:** um neurônio artificial pode ser linear ou não-linear, deste modo, uma RNA constituída de neurônios não-lineares, também será não linear. A não-linearidade é uma propriedade muito importante, principalmente se o mecanismo físico responsável pelo sinal de entrada for não linear;
- **Adaptação por experiência:** as RNAs são chamadas adaptativas, pois seus parâmetros internos, tipicamente seus pesos sinápticos, são ajustados a partir da apresentação de exemplos relacionados ao comportamento do processo, ou seja, adquirem conhecimento por experimentações;
- **Capacidade de aprendizado:** através da aplicação de um método de treinamento, a rede é capaz de extrair a relação existente entre as diversas informações que compõem a aplicação;
- **Habilidade de generalização:** após o processo de treinamento da rede, a mesma é capaz de generalizar o conhecimento adquirido, tornando possível a estimação de soluções até então desconhecidas;
- **Organização de dados:** por ser baseada em características intrínsecas que envolve um conjunto de informações a respeito de um processo, a rede é

capaz de se organizar internamente possibilitando o agrupamento de padrões;

- **Tolerância a falhas:** devido ao grande número de interconexões entre os neurônios artificiais, a RNA torna-se tolerante a falhas quando parte de sua estrutura interna é corrompida sensivelmente;
- **Armazenamento distribuído:** o conhecimento sobre o comportamento de determinado processo dentro de uma arquitetura neural é feito de forma distribuída entre as diversas sinapses de seus neurônios, possibilitando um incremento da robustez da arquitetura mesmo que alguns neurônios se tornem inoperantes;
- **Facilidade de prototipagem:** a maioria das arquiteturas de RNAs podem ser facilmente prototipada, dependendo da especificidade da aplicação, visto que após o processo de treinamento, os resultados são normalmente obtidos por algumas operações matemáticas elementares.

Sendo assim, as RNAs se constituem como ferramentas poderosas e bastante versáteis, podendo ser aplicadas nas mais variadas áreas da ciência. Algumas dessas áreas são citadas em Hagan *et al* (2014, p. 1-5), onde podem ser vistas com mais detalhes.

No entanto, apesar de ser uma ferramenta poderosa, o projeto de uma RNA deve ser feito com bastante cautela, pois o aumento indiscriminado de neurônios pode levar a saída da rede para circunstâncias de memorização excessiva, chamada de *overfitting*. Neste tipo de ocorrência, o erro quadrático tende a ser muito baixo durante a fase de aprendizado, no entanto, o mesmo tende a assumir valores elevados frente aos subconjuntos de teste, este fato ocorre devido ao fato de que a rede acaba decorando as suas respostas diante dos estímulos apresentados em suas entradas. Em contrapartida, uma topologia com um número muito reduzido de neurônios pode resultar em uma situação de debilidade neural denominada de *underfitting*, em que a rede se torna incapaz de realizar a extração e armazenamento de características que permitam implementar hipóteses a respeito do comportamento do processo. Nesses casos, o erro quadrático apresenta valores significativos tanto na fase de aprendizado como na fase teste (SILVA; SPATTI; FLAUZINO, 2010 p. 153).

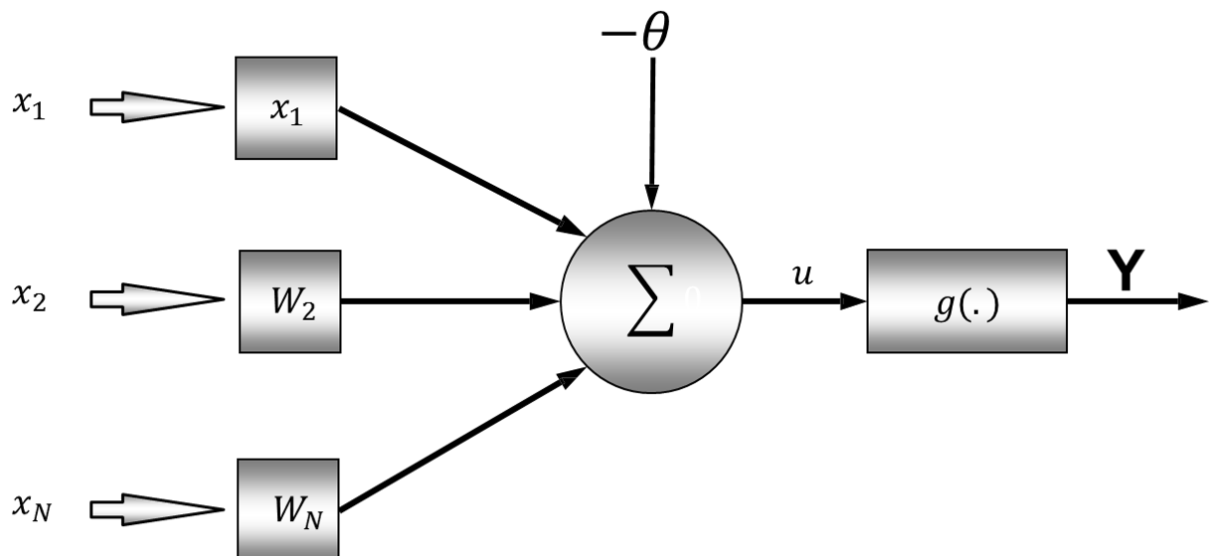
Deste modo, como abordado por Wilamowski (2009), é importante escolher o menor número de neurônios possível para modelagem, visto que uma rede com quantidade reduzida de neurônios e interconexões geralmente apresenta uma melhor

generalização, mesmo que seja necessário sacrificar uma pequena parcela de acertos na etapa de treinamento.

A) Neurônio artificial

A estrutura das RNAs foi desenvolvida tendo como inspiração modelos conhecidos de sistemas nervosos biológicos e o próprio cérebro humano, onde os neurônios artificiais utilizados nas mesmas, são modelos simplificados dos neurônios biológicos (SILVA; SPATTI; FLAUZINO, 2010, p. 33). O modelo de neurônio artificial mais simplificado, mas que ainda é o mais utilizado nas diferentes arquiteturas de RNAs é o proposto por McCulloch e Pitts (1943), pois, engloba as principais características de uma rede neural biológica. A representação do mesmo pode ser vista na Figura 1.

Figura 1 - Neurônio Artificial.



Fonte: Adaptado de Silva, Spatti e Flauzino (2010, p. 34).

Considerando a Figura 1, é possível verificar os sete elementos básicos que constitui o neurônio artificial (SILVA; SPATTI; FLAUZINO, 2010, p. 33):

- a) **Sinais de entrada** $\{x_1, x_2, \dots, x_n\}$: são sinais ou medidas do meio externo que representam os valores assumidos pelas variáveis de uma determinada aplicação. Os sinais de entrada são comumente normalizados para auxiliarem na eficiência computacional dos algoritmos utilizados na aprendizagem;

- b) **Pesos sinápticos** $\{w_1, w_2, \dots, w_n\}$: são valores que serão utilizados para ponderar cada variável de entrada da rede, quantificando as suas relevâncias em relação à funcionalidade do respectivo neurônio;
- c) **Combinador linear** $\{\Sigma\}$: recebe a função de agregar todos os sinais de entrada ponderados pelos respectivos pesos sinápticos, com o intuito de produzir um valor de potencial de ativação;
- d) **Limiar de ativação** $\{\theta\}$: esta variável especifica qual será o patamar apropriado para que o resultado produzido pelo combinador linear possa gerar um valor de disparo em direção à saída do neurônio;
- e) **Potencial de ativação** $\{u\}$: é o resultado produzido pela diferença do valor produzido pelo combinador linear e o limiar de ativação. Se este valor é positivo, ou seja, se $u \geq \theta$ então o neurônio produz um potencial excitatório, caso contrário, o potencial será inibitório;
- f) **Função de ativação** $\{g\}$: tem como objetivo limitar a saída do neurônio dentro de um intervalo de valores razoáveis;
- g) **Sinal de saída** $\{y\}$: valor final produzido pelo neurônio em relação ao conjunto de sinais de entrada, podendo ser utilizado por outros neurônios que estão sequencialmente interligados.

O resultado produzido pelo neurônio artificial proposto por McCulloch e Pitts (1943) é sintetizado, em termos matemáticos, pelas equações (1) e (2):

$$u = \sum_{i=1}^n w_i * x_i - \theta \quad (1)$$

$$y = g(u) \quad (2)$$

Deste modo, o neurônio artificial compõe a estrutura básica de uma rede neural artificial, definindo também as arquiteturas mais simples de RNAs como *Perceptron* simples e *Adaline*, compostas por apenas um neurônio artificial.

B) Funções de ativação

O funcionamento das RNAs mais comuns é definido pelo conceito de *feedforward* (em tradução livre, “alimentação para frente”) representando que o

processamento dentro da estrutura segue um fluxo da entrada para saída, sem laços ou realimentações. Bishop (1995, p. 117) destaca a importância desse tipo de característica, pois, permite representar funções não-lineares de várias variáveis em termos de composições de funções não-lineares de uma única variável, chamada de função de ativação.

As funções de ativação podem ser divididas em dois principais grupos, as funções parcialmente diferenciáveis e as funções totalmente diferenciáveis. Dentre as funções de ativação mais populares, podemos destacar as funções totalmente diferenciáveis de rampa linear, utilizada com o intuito de mapear o comportamento entre as variáveis de entrada e saída de processos e a tangente hiperbólica, uma função antissimétrica e pertencente à família das funções sigmoidais, que permite que redes do tipo MLP aprenda, em geral, mais rapidamente (SILVA; SPATTI; FLAUZINO, 2010, p. 36).

As principais funções de ativação, podem ser observadas em Hagan *et al.* (2014, p. 2-6), para maior compreensão a respeito do comportamento da sua saída em relação a entrada.

C) Processo de treinamento e aprendizado

Neste processo de treinamento, algumas amostras disponíveis que descrevem o comportamento do sistema em questão são apresentadas para rede. Usualmente, este conjunto de amostras é dividido em dois subconjuntos, um para treinamento e outro para teste, concentrando de 60 a 90% dos dados para o treinamento e de 10 a 40% para o teste. Durante este procedimento, cada apresentação completa do conjunto de amostras para treinamento foi denominada época de treinamento (SILVA; SPATTI; FLAUZINO, 2010, p. 51).

Dentre os tipos básicos de treinamento para a RNA, Silva, Spatti e Flauzino (2010, p. 51) abordam, de forma generalizada, as principais formas de aprendizado que estão associadas aos processos de ajuste de pesos, sendo essas:

- **Treinamento Supervisionado:** é similar a um treinamento com um professor e consiste em ter as respostas certas disponíveis para cada amostra dos sinais de entrada, enquanto a rede, por sua vez, não as conhece. Os erros entre as respostas da rede comparadas com as do “professor” são empregados nos ajustes dos parâmetros livres com o

objetivo de diminuir esta defasagem para valores de erro aceitáveis, levando em consideração os propósitos de generalização de soluções;

- **Treinamento Não-supervisionado:** no método não-supervisionado não há respostas desejadas para a camada de saída, sendo assim, a própria rede deve se auto-organizar em relação às particularidades existentes no conjunto total de amostras para que sejam identificados subconjuntos (*clusters*) de dados similares.

Em Kriesel (2007, p. 62), é ressaltado dois tipos diferentes de aprendizado para uma rede neural, que são:

- **Offline:** um conjunto de amostras de treinamento são apresentadas para a rede simultaneamente, para depois calcular os erros e ajustar os parâmetros livres da rede (peso sinápticos e limiar de ativação). Também é conhecido como procedimento de treinamento em lote (*batch training procedures*), pois todos os valores são corrigidos de uma só vez;
- **Online:** diferentemente do aprendizado *offline*, neste caso o ajuste dos pesos é realizado para o erro de cada amostra de treinamento apresentada para rede.

D) Arquiteturas de redes neurais

Os autores Silva, Spatti e Flauzino (2010, p. 46) dividem as redes neurais artificiais em três partes fundamentais, chamadas de camadas, definidas da seguinte forma:

- **Camada de Entrada:** responsável pelo recebimento dos dados ao sistema (medidas, sinais, características). Essas entradas são geralmente normalizadas nas faixas de variações dinâmicas produzidas pelas funções de ativação, com o intuito de obter uma melhor precisão numérica diante das operações matemáticas realizadas pela rede;
- **Camadas Escondidas (ou intermediárias, ocultas):** compostas pelos neurônios encarregados de extrair as características associadas ao problema proposto. A maior parte do processamento interno da rede acontece nessas camadas;

- **Camada de Saída:** responsável pela produção e apresentação dos resultados finais da rede, obtidos a partir dos processamentos efetuados pelos neurônios das camadas anteriores.

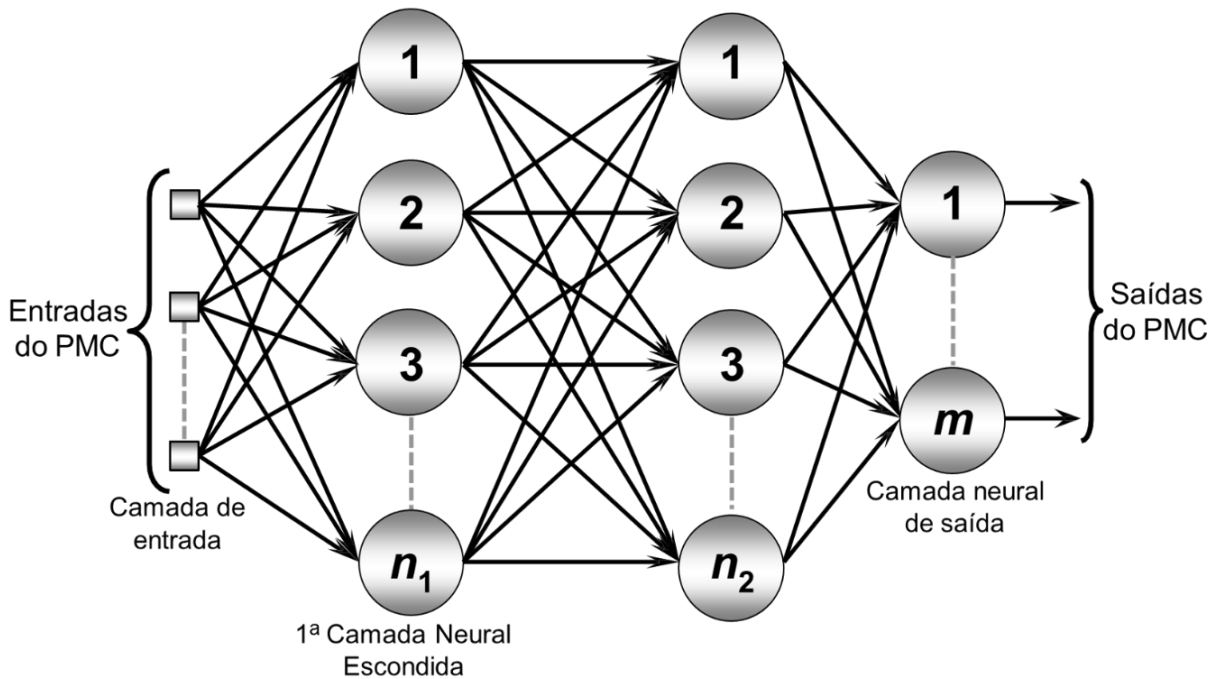
As principais arquiteturas de RNAs, considerando a configuração de seus neurônios e suas formas de interligação, bem como a constituição de suas camadas, podem ser divididas em redes *feedforward* de camada simples, redes *feedforward* de camadas múltiplas, redes recorrentes e redes reticuladas. Neste trabalho foi destacado apenas a *Perceptron* multicamadas, cuja arquitetura é a *feedforward* de camadas múltiplas. Nesta arquitetura o fluxo das informações na rede é unidirecional (da entrada em direção à saída).

E) Perceptron multicamadas

As RNAs do tipo *Perceptron* multicamadas, têm por principal característica a presença de uma ou mais camadas escondidas de neurônios. Estas são utilizadas na solução de problemas relacionados a diversas áreas do conhecimento, tais como aproximação universal de funções, classificação de padrões, identificação e controle de processos, previsão de séries temporais e otimização de sistemas (SILVA; SPATTI; FLAUZINO, 2010, p. 91).

Como observado, a rede MLP pertence à arquitetura *feedforward* de camadas múltiplas, cujo treinamento é realizado de forma supervisionada. A estrutura da rede pode ser observada na Figura 2, juntamente com o fluxo de informações que percorre pela mesma, iniciando na camada de entrada, percorrendo as camadas intermediárias e então finalizando na camada neural de saída (SILVA; SPATTI; FLAUZINO, 2010, p. 92). Pode-se observar que a camada de saída pode possuir mais de um neurônio, tornando possível a solução de problemas onde há não-linearidade.

Figura 2 - Ilustração de rede *Perceptron* multicamadas.

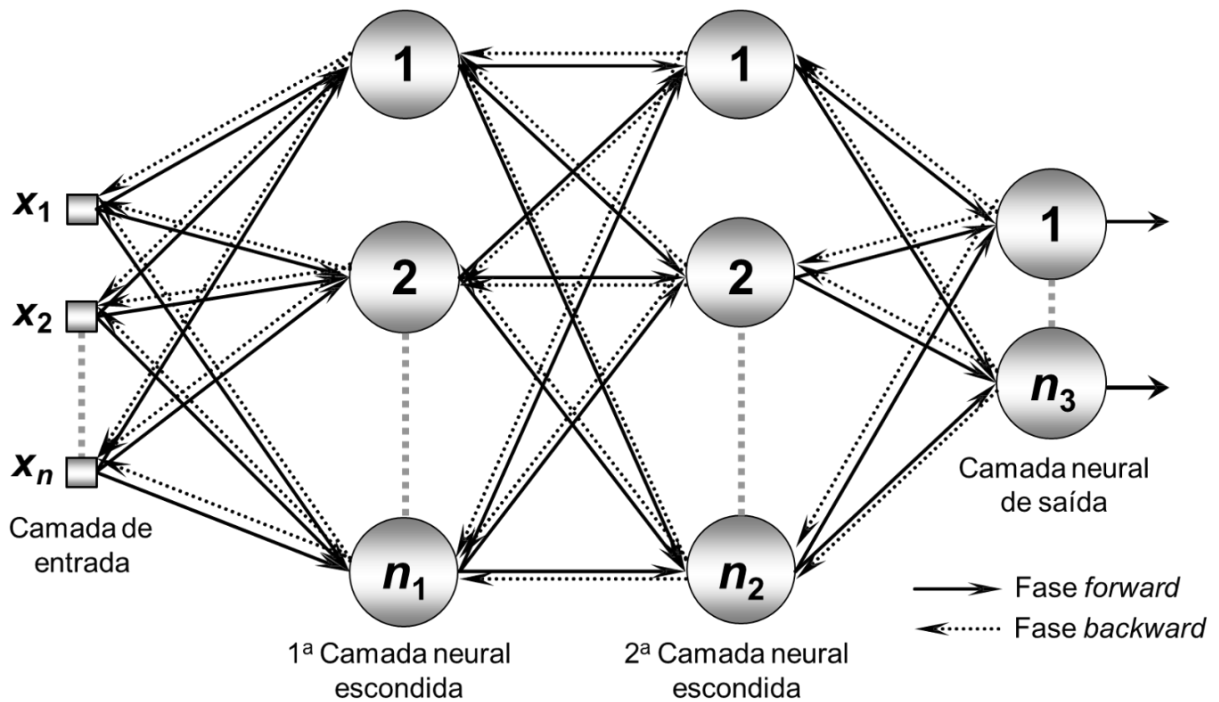


Fonte: Adaptado de Silva, Spatti e Flauzino (2010, p. 92).

Haykin (2007, p. 183), cita que um algoritmo bastante popular para realizar o treinamento supervisionado de MLP é o conhecido como algoritmo *backpropagation* (retropropagação de erro), podendo ser visto como uma generalização do algoritmo de filtragem adaptativa, conhecido como algoritmo do mínimo quadrado médio.

De acordo com Silva, Spatti e Flauzino (2010, p. 94) a aprendizagem por retropropagação de erro é comumente realizado em duas fases bem específicas. A primeira etapa denominada de “propagação adiante” (*forward*), os sinais do conjunto de treinamento são inseridos nas entradas e são propagados camada a camada até a produção da saída em questão. Em seguida é comparado o valor da saída em relação ao valor desejado, visto que esta rede possui treinamento supervisionado, para obtenção dos erros que serão utilizados para realizar o ajuste de pesos e limiares de todos os neurônios, esta fase é chamada de “propagação reversa” (*backward*). A ilustração dessas fases pode ser observada na Figura 3 apresentada a seguir.

Figura 3 - Fases de treinamento da rede MLP.



Fonte: Adaptado de Silva, Spatti e Flauzino (2010, p. 94).

No entanto, a utilização do algoritmo *backpropagation* tende a convergir muito lentamente, pois, o mesmo ajusta os valores das matrizes de pesos da rede MLP na direção oposta do gradiente da função erro, exigindo um elevado esforço computacional. Desta maneira, diversas variações do método *backpropagation* são propostas com o objetivo de tornar o processo de convergência mais eficiente, uma dessas propostas é o método de Levenberg-Marquardt, que é um método gradiente de segunda ordem, baseado no método dos mínimos quadrados para modelos não-lineares, que pode ser incorporado ao algoritmo *backpropagation* com o objetivo de potencializar a eficiência do processo de treinamento (SILVA; SPATTI; FLAUZINO, 2010, p. 116).

Em problemas envolvendo a classificação de padrões, uma rede MLP com m neurônios em sua camada de saída é capaz de classificar, teoricamente, até 2^m classes possíveis. Entretanto, devido à complexidade dos problemas a serem tratados, a adoção desta codificação sequencial de conjuntos pode tornar o treinamento mais difícil, podendo demandar um aumento no número de neurônios nas camadas intermediárias, além da dificuldade para realizar o ajuste topológico. Sendo assim, um dos métodos mais utilizados é o chamado *one of c-class*, que consiste em utilizar a quantidade de neurônios na camada de saída igual ao número de classes do problema (SILVA; SPATTI; FLAUZINO, 2010, p. 130).

2.1.2 Otimização por Enxame de Partículas

Inicialmente apresentado por Kennedy e Eberhart (1995), a otimização por enxame de partículas foi proposta como um método para otimização de funções não-lineares contínuas. Como o nome sugere, o PSO surgiu a partir da análise do comportamento social de enxames, mais especificamente da revoada de pássaros e o comportamento de cardumes de peixes.

A inteligência coletiva, também conhecido como *Swarm Intelligence*, já era objeto de estudo antes mesmo de sua aplicação na área de otimização, sendo utilizada na década de 1980 em sistemas robóticos compostos por uma coleção de agentes simples em um ambiente, interagindo de acordo com regras locais (ZUBEN; ATTUX, 2008). Ainda de acordo com Zuben e Attux (2008), em uma definição formal, pode-se dizer que:

Um sistema de enxame é aquele composto por um conjunto de agentes capazes de interagir entre si e com o meio ambiente. A inteligência de enxame é uma propriedade emergente de um sistema coletivo que resulta de seus princípios de proximidade, qualidade, diversidade, estabilidade e adaptabilidade.

No revoa de pássaros, o objetivo de cada ave é encontrar alimento ou abrigo, para tal objetivo, as mesmas utilizam o aprendizado adquirido por sua própria experiência e também o aprendizado do enxame. Deste modo, os pássaros realizam a busca coletivamente, mas de forma descentralizada, onde cada ave se comporta de maneira independente levando em conta a experiência coletiva e o próprio aprendizado (VESTERSTROM; RIGET, 2002).

Em um contexto computacional, no algoritmo PSO, os pássaros são representados por partículas que corresponde a uma solução no espaço de busca da função a ser otimizada. Sua movimentação dentro deste espaço está diretamente relacionada a sua posição atual e uma determinada velocidade que define a direção do movimento da partícula. A velocidade é modificada levando-se em conta a melhor posição da própria partícula e a melhor posição do grupo (MEDEIROS, 2005).

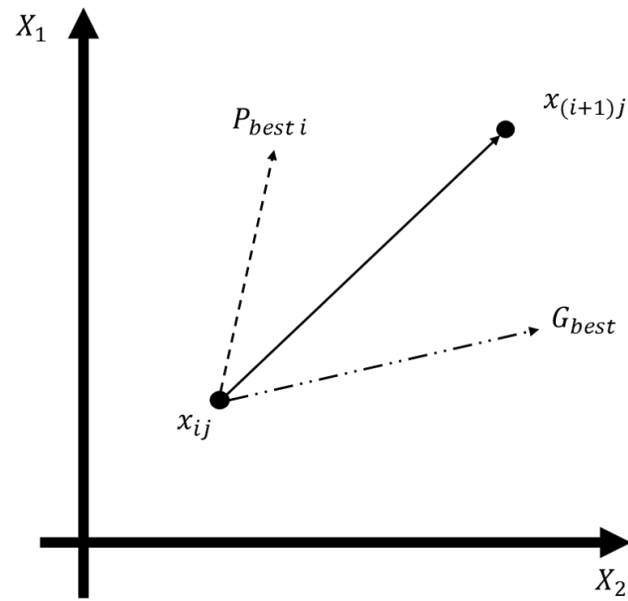
A fim de tornar mais claro o entendimento dos termos contidos no algoritmo PSO, Reyes-Sierra e Coello (2006) definem alguns pontos chave em relação ao mesmo, que são apresentados a seguir:

- **Enxame:** população do algoritmo;

- **Partícula:** indivíduo do enxame. Cada partícula representa uma solução em potencial para o problema tratado dentro do espaço de busca, é representada por um vetor solução;
- **Velocidade:** responsável por realizar a movimentação da partícula no processo de otimização. Calculada individualmente para cada partícula durante o processo iterativo;
- P_{best} : melhor posição encontrada por cada partícula, em individual;
- L_{best} : melhor posição encontrada por uma partícula, considerando uma determinada vizinhança;
- G_{best} : melhor posição encontrada pelas partículas, considerando toda a população;
- **Coeficiente de inercia (ω):** controla a influência dos valores anteriores da velocidade no cálculo de velocidade atual;
- **Fator cognitivo ($c1$):** controla a influência que sua melhor posição tem sobre o cálculo da velocidade;
- **Fator social ($c2$):** controla a influência da melhor posição global sobre o cálculo da velocidade;
- **Topologia de Vizinhança:** estipula as partículas que serão vizinhas durante o processo iterativo, definindo as partículas que irão realizar troca de informação.

A movimentação da partícula dentro do espaço de busca é dependente de três fatores que modificam sua velocidade, sendo eles, o próprio caminho da partícula (v_{ij}), a melhor posição já encontrada por ela mesma (P_{best}) e a melhor posição já encontrada pelo enxame (G_{best}) (TORÁCIO, 2008). Esta relação pode ser vista com maior clareza na Figura 4.

Figura 4 - Fatores influenciáveis em $x_{(i+1)j}$.



Fonte: Adaptado de (Carvalho, 2007, p. 9).

Na qual a partir da velocidade, v_{ij} , uma partícula x_{ij} , se movimenta em um espaço de busca n -dimensional, onde i representa o número de partículas do sistema ($i = \{1, 2, \dots, POP\}$) e j representa o número a serem otimizadas, dimensão do problema ($j = \{1, 2, \dots, D\}$) (TORÁCIO, 2008).

A princípio, para modelar matematicamente a representação da movimentação das partículas Kennedy e Eberhart (1995) testaram vários modelos, até chegar na representação mais eficiente para a atualização da movimentação, que são mostrados nas equações (3) e (4).

$$v_{(i+1)j} = v_{ij} + 2 * rand(0,1) * (P_{best\ i} - x_{ij}) + 2 * rand(0,1) * (G_{best} - x_{ij}) \quad (3)$$

$$x_{(i+1)j} = x_{ij} + v_{(i+1)j} \quad (4)$$

Onde $rand(0,1)$ representa um valor aleatório uniformemente distribuído no intervalo de 0 a 1.

Posteriormente, motivado pela obtenção de melhores controle sobre o processo de busca do algoritmo, Shi e Eberhart (1998) propuseram a inclusão de um parâmetro denominado momento de inercia (ω). Segundo Eberhart e Shi (2000), tal parâmetro atua como controle sobre a velocidade, fazendo com que dependendo do seu valor, as partículas explorem maiores ou menores áreas de busca. Os autores

ainda afirmam que melhores resultados podem ser obtidos utilizando, inicialmente, um valor alto para ω (0,9) e gradativamente reduzir esse valor até um ω (0,4) menor. Desta forma, o algoritmo foi iniciado com maiores áreas de busca e finalizado com uma busca direcionada. A modificação proposta por Shi e Eberhart (1998), na equação (3), referente a velocidade, pode ser vista na equação (5).

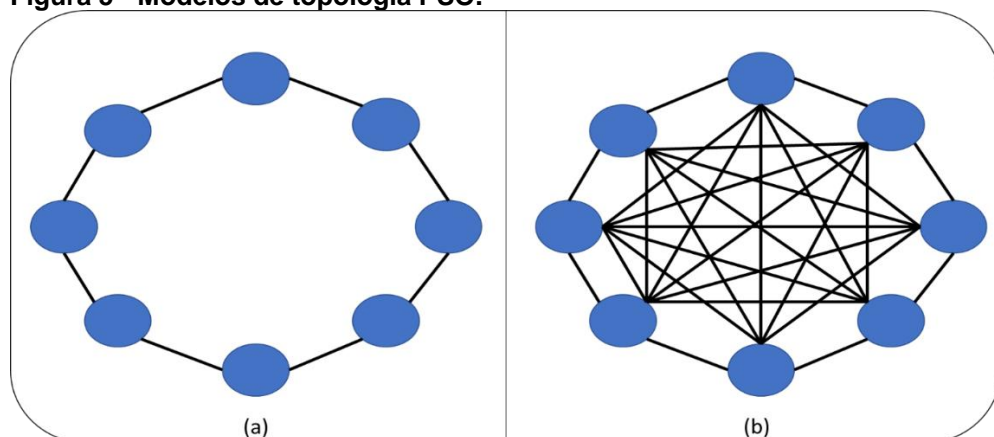
$$v_{(i+1)j} = \omega * v_{ij} + c1 * rand(0,1) * (P_{best} - x_{ij}) + c2 * rand(0,1) * (G_{best} - x_{ij}) \quad (5)$$

Outra modificação existente no algoritmo do PSO, diz respeito a forma como as partículas irão se comunicar durante o processo. Em sua versão clássica o PSO utiliza o modelo G_{best} , onde há a troca de informação entre todas as partículas, este modelo oferece uma rápida taxa de convergência uma vez que o enxame se concentra em um ponto de cada vez, mas deixa o sistema sujeito a uma convergência prematura (VAN DEN BERGH; 2002).

Para contornar a possibilidade de convergência prematura do modelo G_{best} , modelos onde existe comunicação apenas com vizinhos foram propostos, como é o caso do modelo em anel, conhecido como L_{best} . Neste modelo, cada partícula realiza troca de informação apenas com as partículas adjacentes a sua posição, utilizando assim, múltiplos pontos atratores. Diferentemente do G_{best} , a taxa de convergência é menor neste modelo, mas tem menor probabilidade de ficar preso em mínimos locais (CARVALHO, 2007, p. 14).

Os dois modelos citados anteriormente podem ser vistos a seguir, onde são mostradas as formas de comunicação das partículas no modelo L_{best} na Figura 5a e também no modelo G_{best} na Figura 5b.

Figura 5 - Modelos de topologia PSO.



Fonte: Adaptado de Bhargab e Sangita (2015).

O critério de parada definido para o PSO é dado por um determinado número de iterações realizados pelo sistema, assim que é alcançado um determinado valor previamente definido, o algoritmo é interrompido e o melhor resultado encontrado até então é definido com a melhor resposta do sistema. A síntese do funcionamento citado pode ser vista com mais detalhes na Figura 6.

Figura 6 - Pseudocódigo do algoritmo PSO.

```

Atribui parâmetros
PARA cada partícula  $i$  da população FAÇA
    Inicializar  $x_{ij}$  com conjunto  $j$  de soluções aleatória para o problema
    Inicializar  $v_{ij}$  com conjunto  $j$  de velocidades aleatória para o problema
     $fit_i \leftarrow$  Calcula fitness da solução  $x_{ij}$ 
     $P_{best\ i} \leftarrow fit_i$ 
     $G_{best} \leftarrow$  melhor entre  $P_{best\ i}$  e  $G_{best}$ 
FIMPARA
REPETIR
    PARA cada partícula  $i$  da população FAÇA
        Atualizar  $v_{ij}$ , usando (5)
        Atualizar  $x_{ij}$ , usando (4)
         $fit_i \leftarrow$  Calcula fitness da solução  $x_{ij}$ 
         $P_{best\ i} \leftarrow$  melhor entre  $fit_i$  e  $P_{best\ i}$ 
         $G_{best} \leftarrow$  melhor entre  $P_{best\ i}$  e  $G_{best}$ 
        Melhor solução  $\leftarrow x_{ij}$  referente a  $G_{best}$ 
    FIMPARA
ATÉ ciclo máximo
  
```

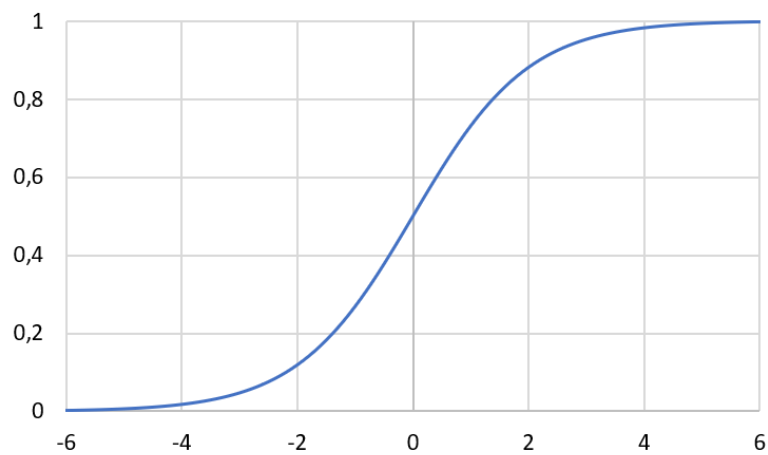
Fonte: Adaptado de Rosendo (2010, p. 20).

A) Otimização por Enxame de Partículas binário

Como uma forma de discretização da aplicação do algoritmo PSO, inicialmente proposto para espaços de busca contínuos, Kennedy e Eberhart (1997) desenvolveram o PSO binário, que através modificações no clássico PSO e a utilização de transformação sigmoide, possibilitou sua aplicação em uma nova classe de problemas.

A versão binária do algoritmo se baseia em simples modificações no clássico PSO, onde o cálculo da velocidade (v_{ij}) é mantido, mas seu valor é convertido no intervalo de 0 a 1 através da função sigmoide, que tem comportamento definido assim como mostrado na Figura 7. Dentro deste contexto, a velocidade convertida ($S(v_{ij})$) representa a probabilidade de um valor se tornar 1 ou 0, por exemplo $S(v_{ij}) = 0,2$ tem 20% de chance de que x_{ij} seja igual a 1 e 80% de chance para 0 (KENNEDY; EBERHART, 1997).

Figura 7 – Função Sigmoide.



Fonte: Autoria própria.

Matematicamente, tal representação pode ser feita através das equações (6) e (7), apresentadas a seguir.

$$S(v_{ij}) = \frac{1}{1 + e^{-v_{ij}}} \quad (6)$$

$$x_{ij} = \begin{cases} 0 \rightarrow \text{caso } rand(0,1) \geq S(v_{ij}) \\ 1 \rightarrow \text{caso } rand(0,1) < S(v_{ij}) \end{cases} \quad (7)$$

Onde $rand(0,1)$ é um valor aleatório distribuído uniformemente entre 0 e 1, sendo considerado a escolha probabilística. É importante notar que pelo fato de a velocidade utilizar a função sigmoide, seus limitantes são definidos de acordo com o intervalo probabilístico desejado, por exemplo, caso a velocidade varie entre (-6 e 6) a probabilidade estará entre (0,9975 e 0,0025), sendo tal parâmetro diretamente relacionado com a diversidade das partículas (KENNEDY; EBERHART, 1997).

Desta forma, destaca-se que apesar de ser uma técnica apresentada em 1997, a versão binária do algoritmo PSO é utilizada em diversos trabalhos recentes (JIN; RAHMAT-SAMII, 2010; WIBOWO *et al*, 2016; KUMARI *et al*, 2017).

2.1.3 Colônia Artificial de Abelhas

O algoritmo de Colônia Artificial de Abelhas, é uma metaheurística bioinspirada de inteligência coletiva baseada no comportamento social das abelhas em busca de alimento, desenvolvida por Karaboga (2005), afim de solucionar problemas de otimização multimodal e multidimensional (GAO; LIU, 2011).

Segundo Bao e Zeng (2009) o que caracteriza a inteligência coletiva das abelhas pode ser dividida em três componentes essenciais: fonte de alimento, abelhas campeiras e abelhas não campeiras. Além de dois comportamentos básicos: recrutar abelhas não campeiras e abandonar uma fonte de alimento. Cada um dos componentes e comportamentos são descritos a seguir:

- **Fonte de alimento (*Food Source*):** representada para as abelhas como a fonte de néctar e caracterizada através de sua posição em relação a colmeia e o potencial de fornecimento. Analogamente, do ponto de vista computacional, a fonte de alimento corresponde a uma possível solução de um problema e o potencial de fornecimento corresponde a aptidão solução;
- **Abelhas campeiras (*Employed foragers*):** as abelhas campeiras, também chamada por outros autores de trabalhadoras (ANDRADE, 2012), são as responsáveis por encontrar as fontes de alimentos e voltar a colmeia com as informações referente a fonte encontrada, essas informações são transmitidas através da dança das abelhas;
- **Abelhas não campeiras (*Unemployed foragers*):** são divididas em dois subgrupos, as abelhas exploradoras (*Scouts bee*) e as oportunistas (*Onlooker bee*). As abelhas exploradoras são responsáveis por sempre estarem em busca de novas fontes de alimentos, representam cerca de 5% a 10% da colmeia (SEELEY, 1995), já as oportunistas observam a dança (*waggle dance*) das abelhas campeiras e ajudam a minerar determinada fonte de alimento. Computacionalmente, as abelhas exploradoras realizam buscas aleatórias na região delimitada (busca global) e as oportunistas

realizam buscas locais ao redor da fonte de alimento indicado pela abelha campeira;

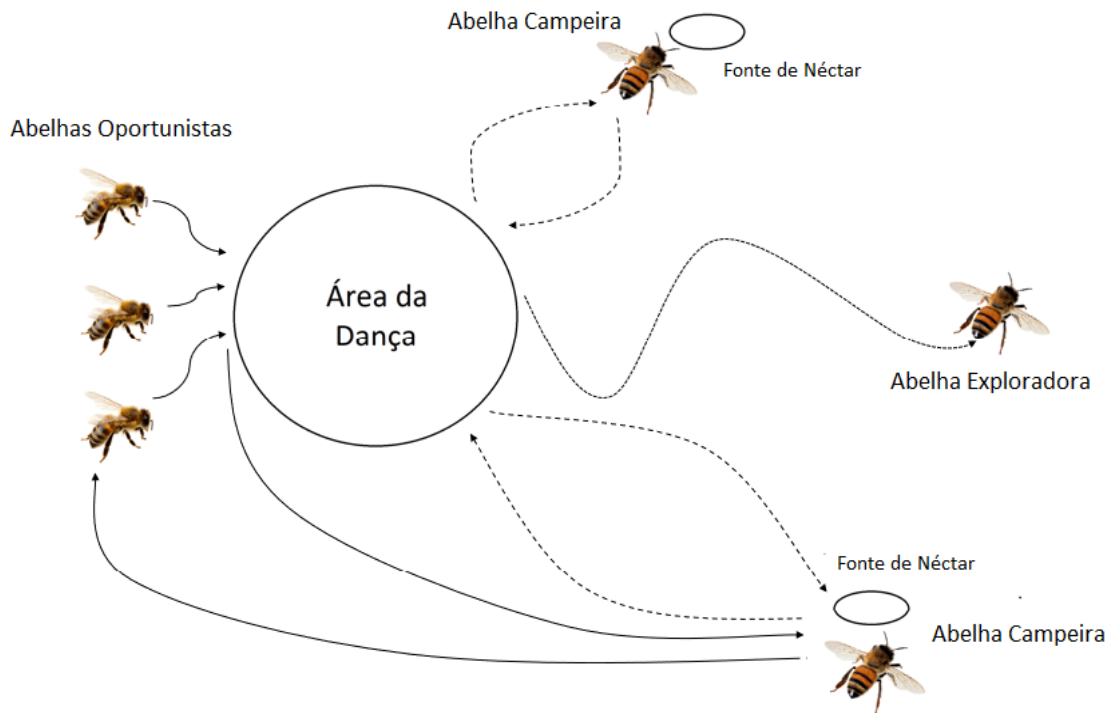
- **Recrutamento:** o recrutamento, como já mencionado acima, é realizado pelas abelhas campeiras, onde as mesmas realizam a dança e as abelhas oportunistas observam a dança e escolhem a fonte onde iram trabalhar;
- **Abandono da fonte:** ocorre quando a fonte de néctar é esgotada, obrigando a abelha campeira a procurar outra fonte de alimento. A implementação deste comportamento é feita através de uma variável limite, que indica a quantidade de iterações que uma fonte permanece sem evoluir, podendo então abandoná-la.

Como destacado por Talbi (2009), uma das características mais interessante na metaheurística, é o fato das simples partículas que compõem o sistema cooperarem de alguma maneira. No caso do ABC essa cooperação ocorre através da troca de informação na área de dança, onde as abelhas campeiras, através da dança, informam a qualidade e a localização da fonte de alimento. As abelhas oportunistas, por sua vez, podem assistir a várias danças e assim escolher a fonte que consideram mais rentável (KARABOGA, 2005).

Um fato peculiar relacionado as abelhas espectadoras, é que raramente elas assistem a dança até o final, portanto, elas não podem tomar a decisão levando em conta a rentabilidade da fonte (SEELEY, 1995). Mesmo assim, as fontes mais rentáveis apresentam a predominância das abelhas oportunista, isso se deve ao fato de que as mesmas apresentam mais abelhas que realizam a dança. Assim, as fontes com maior rentabilidade apresentam, estaticamente, probabilidade maiores de serem escolhidas (TERESHKO; LOENGAROV, 2005).

Além da importância das abelhas campeiras e oportunista, as abelhas exploradoras desempenham um papel essencial na colmeia, garantindo uma boa diversidade de locais para a busca de alimento (ANDRADE, 2012). O modo como as abelhas se interagem e executam suas tarefas pode ser visto com mais clareza na Figura 8.

Figura 8 - Comportamento das abelhas na busca de néctar.



Fonte: Adaptado de Andrade (2012, p. 70).

A modelagem matemática do comportamento das abelhas na busca por néctar, parte inicialmente da definição da quantidade de abelhas trabalhando na colmeia, o que no algoritmo é chamado de população de abelhas ($P = \{1, 2, 3, \dots, SN\}$), sendo que cada abelha representa uma solução ao problema e cada solução representa uma fonte de néctar (KARABOGA, 2005).

Por definição, na modelagem do algoritmo, cada abelha j é representada por um vetor $X_j = \{x_{1j}, x_{2j}, \dots, x_{nj}\}$, onde cada componente do vetor representa uma das variáveis ou parâmetros do problema a ser otimizado. Do mesmo modo que em outras metaheurísticas populacionais, a eficiência do indivíduo da população é apresentada pela resposta da função objetivo, denominado *fitness* (aptidão) (ANDRADE, 2012).

Na metaheurística, as abelhas exploradoras são as responsáveis por inicializar a busca da solução e garantir a diversidade da mesma durante o processo de iteração, evitando que o algoritmo estacione em um mínimo local. Para tal, as exploradoras realizam buscas randômicas pela região limitada das variáveis, conforme é mostrado na equação (8).

$$x_{ij} = x_{i,min} + rand(0,1) \times (x_{i,max} - x_{i,min}) \quad (8)$$

Onde x_{ij} representa a variável i de uma abelha j , $x_{i,min}$ o limite inferior da variável i , $x_{i,max}$ o seu limite superior e $rand(0,1)$ um valor aleatório entre 0 e 1 (KARABOGA; BASTURK, 2007). Sendo essa expressão válida apenas para espaços de busca contínuos (ANDRADE, 2012).

Outra modelagem matemática presente no algoritmo clássico ABC é a representação do comportamento das abelhas oportunistas, que extraem néctar de uma fonte já conhecida. Na otimização da função objetivo, tal comportamento é representado pela busca local, onde as abelhas oportunistas buscam solução em torno de uma solução apresentada pela abelha campeira. A equação matemática que caracteriza a busca local é mostrada na equação (9).

$$v_{ij} = x_{ij} + rand(-1,1) \times (x_{ij} - x_{kj}) \quad (9)$$

Onde v_{ij} é uma nova solução local gerada a partir de x_{ij} , melhor solução da busca local da iteração anterior, $k \in \{1, 2, \dots, N\}$ (tal que $k \neq i$) e $j \in \{1, 2, \dots, D\}$ são índices gerados aleatoriamente. Já a variável $rand(-1,1)$ representa um valor aleatório escolhido entre -1 e 1 . Desta forma, as equações (8) e (9) compõem o algoritmo ABC clássico.

Para representar a escolha das fontes pelas abelhas oportunistas ao assistir a dança das abelhas campeiras, foi definido que a chance de uma oportunista escolher uma fonte é diretamente proporcional a sua qualidade, para isto, é utilizado a equação (10), onde para cada fonte é calculado sua probabilidade, em porcentagem, de ser escolhida.

$$p_i = \frac{fit_i}{\sum_{n=1}^N fit_n} \quad (10)$$

Onde p_i é a probabilidade de seleção da fonte x_i e fit_i é o *fitness* da função objetivo a ser otimizada. Vale frisar que a equação (10), que simboliza o comportamento das abelhas na área da dança é o que garante a intensificação de buscas em fonte mais promissoras, visto que, a mesma tem maior probabilidade de ser escolhida por uma quantidade maior de abelhas oportunistas.

Por fim, para que as abelhas artificiais não fiquem presas em uma fonte sem qualidade, foi definido um parâmetro de controle para o abandono da fonte pouco produtiva. Se a mesma não apresentar evolução durante um número definido de gerações a fonte é abandonada e a abelha campeira se torna uma exploradora em busca de uma nova solução. O parâmetro de controle que determina o máximo de gerações para o abandono é chamado de LIMIT (KARABOGA, 2005).

O critério de parada definido para a Colônia Artificial de Abelhas é dado pelo número de iterações realizados pelo sistema. Assim que é alcançado um determinado valor previamente definido, o algoritmo é interrompido e o melhor resultado encontrado até então é definido como a melhor resposta do sistema (KARABOGA, D.; BASTURK, B.; 2007).

Figura 9 - Pseudocódigo do algoritmo ABC.

```

Atribui parâmetros
PARA cada abelha campeira  $i$  da população FAÇA
    Inicializar  $x_{ij}$  com conjunto  $j$  de soluções aleatória para o problema, usando (8)
     $fit_i \leftarrow$  Calcula fitness da solução  $x_{ij}$ 
     $Best_i \leftarrow fit_i$ 
FIMPARA
REPETIR
    PARA cada abelha campeira  $i$  da população FAÇA
        Gerar  $x_{ij}$ , com base na própria abelha campeira, usando a equação (9)
         $fit_i \leftarrow$  Calcula fitness da solução  $x_{ij}$ 
    FIMPARA
     $\overrightarrow{Best} \leftarrow$  recebe as melhores soluções entre  $\overrightarrow{fit}$  e  $\overrightarrow{Best}$ 
    Calcular probabilidade  $p_i$  para as soluções  $x_{ij}$ , a partir da equação (10)
    Determinar abelha campeira escolhida pela da abelha oportunista
    PARA cada abelha oportunista  $i$  da população FAÇA
        Gerar  $v_{ij}$ , com base na abelha oportunista escolhida, usando a equação (9)
         $fit_i \leftarrow$  Calcula fitness da solução  $v_{ij}$ 
    FIMPARA
     $\overrightarrow{Best} \leftarrow$  recebe as melhores soluções entre  $\overrightarrow{fit}$  e  $\overrightarrow{Best}$ 
    PARA cada abelha campeira  $i$  da população FAÇA
        Determinar o abandono da fonte  $i$ 
        SE fonte  $i$  abandonada
            Gerar  $x_{ij}$ , chamando abelha exploradora, equação (8)
             $fit_i \leftarrow$  Calcula fitness da solução  $x_{ij}$ 
        FIMSE
    FIMPARA
    Melhor solução  $\leftarrow x_{ij}$  referente ao melhor fitness de  $\overrightarrow{Best}$ 
ATÉ ciclo máximo

```

Fonte: Adaptado de KARABOGA, D.; BASTURK, B.; 2007).

A) Colônia Artificial de Abelhas para otimização binária

Como o modelo clássico do algoritmo ABC proposto por Karaboga (2005) foi, a princípio, implementado com objetivo de trabalhar com espaços de busca contínuos, otimizações envolvendo variáveis discretas ou binárias necessitam de

modificações para a viabilização de sua aplicação (OZTURK; HANCE; KARABOGA, 2015; PAMPARÁ; ENGELBRECHT, 2011; LI; MENG, 2017). Dentre as diversas formas de discretização existentes na literatura, destaca-se o método proposto por Kashan, Nahavandi e Husseinzadeh Kashan (2012), onde é desenvolvido uma versão binária do ABC que não leva em consideração apenas os bits individualmente, mas a relação de um vetor binário, através da dissimilaridade.

Para o cálculo da dissimilaridade entre dois vetores, os bits de cada vetor são, primeiramente, comparados um a um de acordo com sua posição e a quantidade de vezes que cada combinação de bits ocorre é quantificada. As quatro combinações possíveis são mostradas a seguir (KASHAN, NAHAVANDI E HUSSEINZADEH KASHAN, 2012):

- M_{11} : número de bits em que $x_i = y_i = 1$;
- M_{01} : número de bits em que $x_i = 0$ e $y_i = 1$;
- M_{10} : número de bits em que $x_i = 1$ e $y_i = 0$;
- M_{00} : número de bits em que $x_i = y_i = 0$.

Com base nos valores de M_{11} , M_{01} e M_{10} a dissimilaridade é calculada conforme é mostrado na equação (11).

$$Dissim(x_{ij}, x_{kj}) = 1 - \frac{M_{11}}{M_{01} + M_{10} + M_{11}} \quad (11)$$

Neste método a equação (10), do modelo clássico, é modificada levando como base a dissimilaridade entre os vetores x_{ij} e x_{kj} , onde j e k passam a representar um conjunto de bits que correspondem a uma variável escolhida aleatoriamente. A modificação citada pode ser vista na equação (12).

$$Dissim(v_{ij}, x_{ij}) \approx rand(-1,1) \times Dissim(x_{ij}, x_{kj}) \quad (12)$$

Onde $Dissim(x_{ij}, x_{kj})$ representa a dissimilaridade entre dois vetores e $Dissim(v_{ij}, x_{ij})$ a dissimilaridade que deve ser calculada a partir da determinação de M_{11} , M_{01} e M_{10} , que pode ser encontrada computacionalmente através do sistema de minimização a seguir:

$$\min \left| 1 - \frac{M_{11}}{M_{01} + M_{10} + M_{11}} - A \right| \quad (13)$$

Sujeito a:

$$M_{11} + M_{01} = n_1 \quad (14)$$

$$M_{10} \leq n_0 \quad (15)$$

$$M_{11}, M_{10}, M_{01} \geq 0 \quad (16)$$

Sendo A igual a $Dissim(x_{ij}, x_{kj})$, n_1 o número de bits igual a 1 e n_0 o número de bits igual 0 de x_{ij} .

A partir dos valores M_{11} , M_{01} e M_{10} calculados, o vetor v_{ij} pode ser criado seguindo alguns passos:

- Inicializar o vetor v_{ij} com zeros, sendo seu tamanho igual ao de x_{ij} ;
- Criar o conjunto P, contendo as posições onde x_{ij} vale um;
- Selecionar aleatoriamente dentro do conjunto P, M_{11} posições;
- Tornar as posições selecionadas dentro do conjunto P igual a um;
- Criar o conjunto Q, contendo todas as posições onde x_{ij} vale zero;
- Selecionar aleatoriamente dentro do conjunto Q, M_{10} posições;
- Tornar as posições selecionadas dentro do conjunto Q igual a um.

2.2 Sistemas inteligentes híbridos

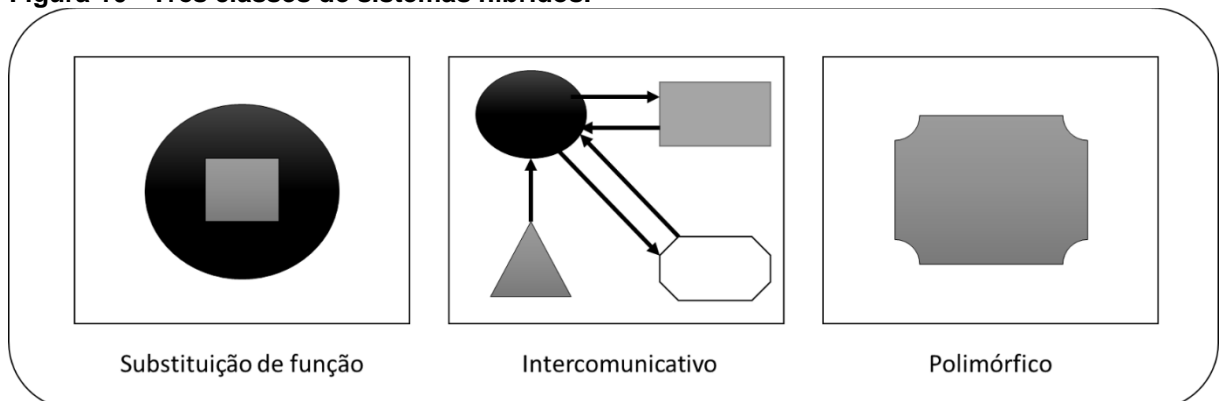
Segundo Goldschmidt, Passos e Bezerra (2015), pode ser definido como um sistema híbrido aqueles utilizam a combinação de duas ou mais técnicas de forma sinérgica para a solução de um problema. Ainda de acordo com Goldschmidt, Passos e Bezerra (2015), “Este sinergismo se reflete na obtenção de um sistema mais

poderoso (em termos de poder de interpretação, de aprendizado, de estimativa de parâmetros, de generalização, dentre outros) e com menos deficiências”.

Como é mostrado na Figura 10, a associação destas técnicas pode ser relacionada em três formas básicas para a construção de sistemas inteligentes híbridos, sendo elas (REZENDE, 2003):

- **Substituição de função:** utiliza as técnicas de forma que a função de uma delas complemente a da outra, não acrescentando nenhuma funcionalidade. O objetivo desta união é superar alguma limitação da técnica principal ou otimizar sua execução.
- **Intercomunicativo:** utilizado para resolver problemas complexos, que possam ser divididos em várias subtarefas independentes, onde cada técnica é representada por um módulo responsável por resolver determinada sub tarefa.
- **Polimórfico:** nesta hibridização uma única técnica é adaptada para executar uma tarefa encubada a outra técnica. Tem como objetivo descobrir novas funcionalidades de uma técnica e entender como diferentes técnicas podem se relacionar.

Figura 10 - Três classes de sistemas híbridos.



Fonte: Adaptado de Rezende (2003).

2.3 Eletromiografia

Segundo Enoka (2000), a eletromiografia é definida como uma técnica de monitoramento elétrico biológico, que fornece as medidas dos potenciais de ação, sendo essas representadas pela tensão em função do tempo. Este sinal pode ser utilizado para diversos fins, como por exemplo: identificação de doenças (MERLETTI;

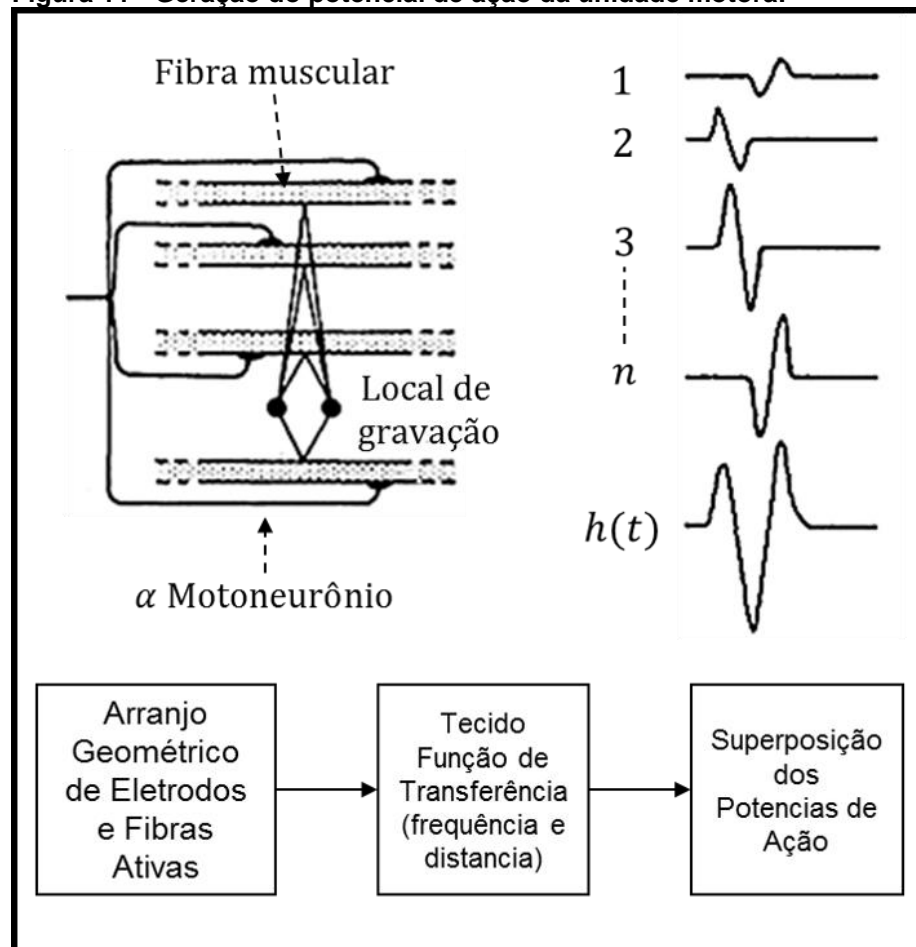
FARINA; GRANATA, 1999) e reconhecimento de padrão de movimentos (AL-TIMEMY *et al.*, 2015).

A aquisição dos dados eletromiográficos podem ser feitas através de dois métodos, o invasivo e o de superfície. No método invasivo detectam-se maiores amplitudes e frequências, no entanto, sua técnica gera mais inconvenientes, visto que são utilizadas agulhas em contato direto com o músculo. No método superficial, eletrodos são colocados sobre a pele, trazendo vantagens como a facilidade de uso e a possibilidade do estudo de movimentos, porém, também conta com pontos negativos como as interferências causada pelo tecido existente entre as fibras musculares e o local do eletrodo, criando um efeito de filtro passa-baixa (KONRAD, 2005; TAKAHASHI, 2006).

Desta forma, um sinal EMG obtido por sensores denominados de eletromiógrafos, fornece dados instantâneos para um computador ou um sistema de aquisição de dados. A leitura eletromiográfica é analógica, sendo assim, é necessário que esse sinal seja convertido para um sistema discreto para que possa ser estudado e manipulado. Um requisito que deve ser respeitado durante o processo de discretização é a frequência de amostragem, que deve estar de acordo com o teorema de Nyquist, para evitar o efeito de *aliasing*. Em geral, as frequências eletromiográficas variam entre 400 e 500Hz (MARCHETTI; DUARTE, 2006).

A leitura de tensão captada pelo eletrodo é chamada de potencial de ação da unidade motora, que é a soma dos potenciais de ação individuais de todas as fibras musculares da unidade motora ($h(t)$), controlado pelos motoneurônios. A Figura 11 representa uma unidade motora, onde é possível observar os potenciais de cada fibra e a soma do potencial ($h(t)$), vista pelo eletrodo (DE LUCA, 1979).

Figura 11 - Geração do potencial de ação da unidade motora.



Fonte: Adaptado de De Luca (1979).

Na análise do sinal EMG é bastante recorrente a utilização de métodos de processamento para realizar o tratamento do mesmo. A frequência e a amplitude do sinal EMG são fatores importantes no seu processamento, tais informações podem ser obtidas através das análises no domínio temporal e no domínio da frequência. Desta forma, a análise temporal descreve quando algo ocorre e qual a amplitude de sua ocorrência, enquanto a análise de frequência permite identificar a forma dos potenciais de ação (ROBERTSON *et al.*, 2004).

2.4 Transformada *Wavelet Packet*

A análise de sinais representa uma das mais importantes ferramentas no processamento digital de sinais. Essa análise pode ser realizada por meio das Transformadas de Fourier, porém, uma das grandes deficiências da análise de Fourier surge do fato de que ela não apresenta um caráter local, mas sim um “comportamento

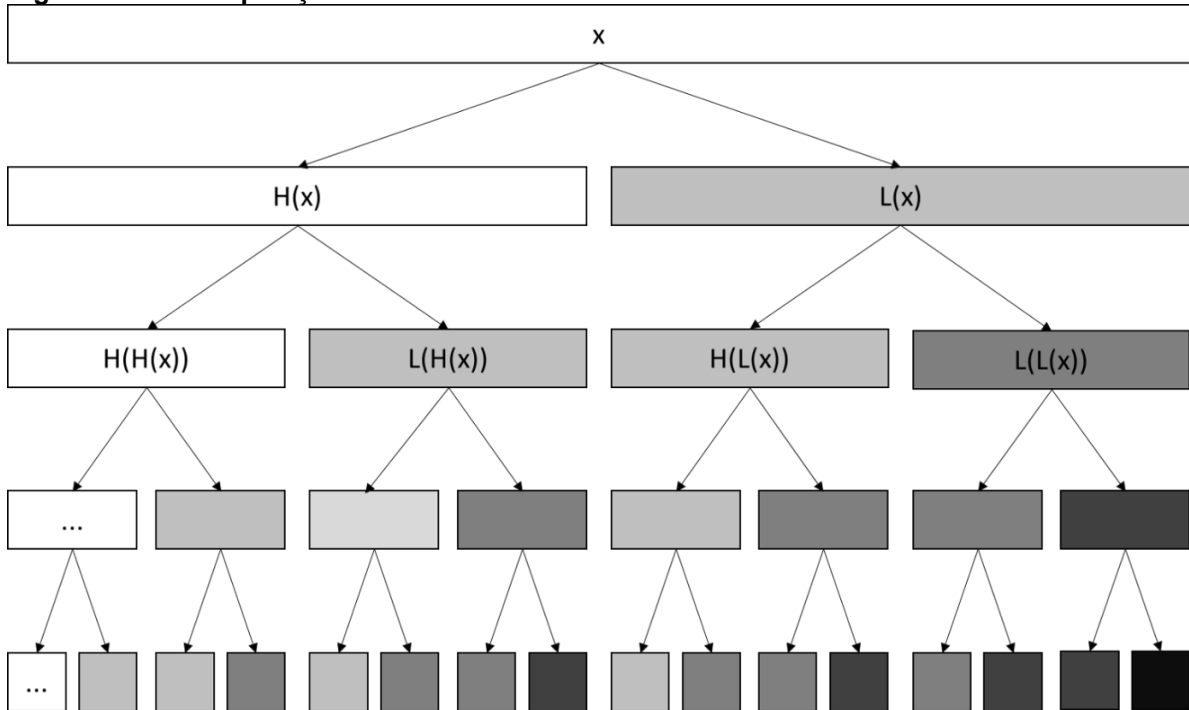
global médio”, além de não fornecer uma análise temporal, apenas frequencial (OLIVEIRA, 2007).

A Transformada *Wavelet* surgiu como uma alternativa a Transformada de Fourier, permitindo uma análise tempo-frequência. Deste modo, a WT compõe uma ferramenta com uma larga gama de aplicações que permite unificar um vasto número de técnicas de processamento e análise, como a análise de sinais médicos, análise de imagens, codificação em sub-bandas, análise multirresolução para visão artificial em computadores e modelagem de sistemas variantes no espaço-tempo (OLIVEIRA, 2007)

Uma forma bastante eficiente de aplicar a Transformada *Wavelet* Discreta é através de filtros, técnica desenvolvida por Mallat (1989), onde tem-se a decomposição da *wavelet* implementando uma análise multiresolução. A Transformada *Wavelet Packet*, proposta por Coifman e Wickerhauser (1992), é uma generalização do conceito da Transformada *Wavelet* Discreta.

Na Transformada *Wavelet* Discreta, o sinal é dividido em coeficientes de aproximação e coeficientes de detalhes, resultantes da filtragem passa-baixas e da filtragem passa-altas, respectivamente. No entanto, na decomposição, apenas os coeficientes de aproximação são novamente subdivididos em coeficientes de aproximação e coeficientes de detalhes. Na análise *wavelet packet* tanto os coeficientes de aproximação quanto os de detalhes são subdivididos novamente, deste modo, para uma resolução j , 2^j sub-bandas são obtidas, cada uma delas correspondendo a uma faixa de frequência específica (MALLAT, 2008). Desta forma, os coeficientes são decompostos em forma de árvore binária, chamada árvore *wavelet packet*, este conceito pode ser visualizado de maneira mais clara por meio da Figura 12.

Figura 12 - Decomposição da WPT em árvore binária.



Fonte: Adaptado de Prarraga (2002, p. 59).

Dentro desse contexto, no processamento digital de sinais, a análise baseada na energia para os coeficientes da Transformada *Wavelet Packet* é uma das mais comuns, portanto, este é um dos parâmetros relevantes a serem extraídos do sinal após realizada a decomposição por meio da Transformada *Wavelet Packet*.

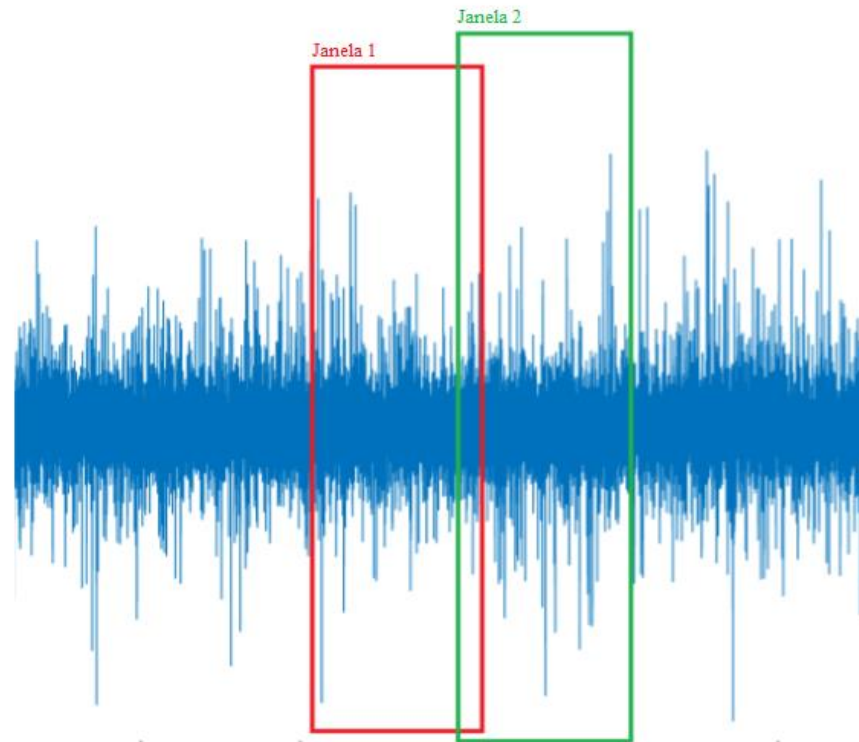
De acordo com Da Silva *et al.* (2017), a energia dos coeficientes *wavelet packet*, para cada sub-banda i da resolução j , é definida por meio da equação (17) como:

$$E_i = \frac{1}{n^2} \sum_{k=1}^n |C_{j,k}^i|^2 \quad (17)$$

onde n é a quantidade de coeficientes da sub-banda, k é o fator de translação e $C_{j,k}^i$ são os coeficientes *wavelet packet*.

A WPT é dividida em famílias, cada uma com um perfil diferente para extração. Esta é aplicada em níveis de decomposição, produzindo coeficientes ou nós para um determinado conjunto de dados, selecionados em intervalos, chamados de janelamento, a partir dos dados no tempo como mostra a Figura 13, podendo haver sobreposição das janelas de dados. Os nós no último nível de decomposição, são chamados de folhas de árvores ou terminal nodes (JIAO; SHI; LIU, 2017).

Figura 13 - Exemplo de janelamento.



Fonte: Autoria própria.

2.5 Trabalhos relacionados

No decorrer do processo de revisão bibliográfica, foi possível observar a existência de propostas relacionadas na literatura, que fizeram a aplicação de métodos utilizados neste trabalho como o uso de redes neurais artificiais para classificação de padrões e também de algoritmos bioinspirados para otimização de seus parâmetros internos com o intuito de melhorar seu desempenho.

Em Zhang e Shao (2000), é proposto pelos autores um SIH que utiliza o PSO para realizar a otimização dos pesos sinápticos e da arquitetura de RNAs do tipo MLP. Uma parte do sistema envolve a otimização do número de neurônios da camada escondida da rede através de um enxame de partículas pertencentes ao domínio dos naturais, enquanto a outra parte utiliza o PSO padrão (domínio dos reais) para ajustar os pesos sinápticos relativos a cada arquitetura (partícula) do primeiro enxame. Deste modo, esses dois processos são repetidos alternadamente por um número de iterações. Já em Carvalho (2007, p. 40), é proposto um algoritmo baseado na metodologia sugerida por Zhang e Shao (2000), no entanto, utiliza variações do modelo original do PSO no subprocesso de ajuste de pesos sinápticos. Em ambos os

trabalhos, a otimização é realizada para os pesos e a quantidade de neurônios de apenas uma camada escondida, enquanto para o presente trabalho, esta otimização foi realizada utilizando dois algoritmos distintos, sendo um PSO e um ABC, onde as variáveis a serem otimizadas foram a quantidade de camadas, número de neurônios em cada camada e a taxa de aprendizado, não abordando a otimização de pesos sinápticos a partir dos algoritmos bioinspirados.

Em Carvalho, Ramos e Chaves (2010), os autores apresentam quatro metaheurísticas para otimização de arquitetura de redes neurais artificiais. Os parâmetros otimizados neste trabalho são o número de camadas escondidas, a quantidade de neurônios em cada camada, taxa de aprendizado, momentum e função de ativação. Com base nos resultados obtidos neste trabalho, Carvalho, Ramos e Chaves (2010), destacam que é totalmente justificável evoluir a arquitetura de uma RNA se o objetivo é obter uma solução dedicada e adaptada ao contexto do problema. Este trabalho visa abordar a aplicação de metaheurísticas diferentes das utilizadas por Carvalho, Ramos e Chaves (2010), para otimizar o número de camadas escondidas, o número de neurônios em cada uma delas e a taxa de aprendizado de uma MLP.

Já em Çam *et al.* (2015), os autores desenvolvem um sistema para a otimização da topologia do *Perceptron* multicamadas baseado em três técnicas de otimização, sendo elas: Algoritmo Genético, Colônia Artificial de Abelhas e Enxame de Partículas, fazendo a comparação de seus resultados. No caso de Çam *et al.* (2015) os parâmetros escolhidos para a otimização são a taxa de aprendizado, o número de neurônios da camada oculta e o coeficiente momentum, onde apenas uma camada neural escondida foi utilizada. A partir dos resultados obtidos, os autores concluem que dependendo das bases de testes utilizadas, uma técnica comportou-se melhor do que a outra. As principais diferenças entre o presente trabalho e Çam *et al.* (2015), é o fato de não estar limitado a apenas uma camada neural escondida, já que isto também foi incluído como um valor a ser otimizado e a utilização de apenas dois tipos de técnicas para a realização da otimização.

Por fim, em Shahraki e Zahiri (2017), é implementado a otimização da arquitetura de MLP utilizando três diferentes metaheurísticas. As variáveis otimizadas neste trabalho são as camadas escondidas, a quantidade de neurônios em cada uma delas e as funções de ativação. O que difere o presente trabalho do apresentado por Shahraki e Zahiri (2017), é o fato de que neste, o número de camadas escondidas

otimizadas é maior devido à complexidade do problema abordado, além disso, apesar de não ser otimizada a função de ativação, foi otimizado a taxa de aprendizado que é outro parâmetro bastante importante para o desempenho da rede.

O Quadro 1 apresenta uma síntese dos trabalhos mencionados, onde algumas ressalvas são destacadas.

Quadro 1 – Trabalhos Relacionados.

Autores	Trabalho desenvolvido	Ressalvas
Zhang e Shao (2000)	Otimização de arquitetura e pesos sinápticos de RNA utilizando PSO	Utiliza dois sistemas que trabalham alternadamente para otimização de arquitetura e pesos sinápticos. Não otimiza o número de camadas escondidas e a taxa de aprendizado.
Carvalho (2007)	Otimização de arquitetura e pesos sinápticos de redes MLP utilizando PSO	Se baseia no mesmo método proposto por Zhang e Shao (2000), apenas utilizando variações do PSO para otimização dos pesos sinápticos. O método acarreta em grandes tempos médios de execução do algoritmo.
Carvalho, Ramos e Chaves (2010)	Otimização da arquitetura de RNA, utilizando VNS, SA, GEOvar e GA	Demonstra que o erro de generalização, como <i>fitness</i> , apresenta uma maior influência na qualidade da otimização. Destaca que é totalmente justificável evoluir a arquitetura de uma RNA quando se tem o interesse de obter uma solução dedicada e adaptada ao problema.
Çam <i>et al.</i> (2015)	Comparação do AG, ABC e PSO na otimização da arquitetura de MLP	Mostra que o desempenho dos algoritmos pode variar de acordo com a base de dados utilizada. Não realiza a otimização do número de camadas escondidas.
Shahraki e Zahiri (2017)	Utilização de IPO, PSO e GSA na otimização da arquitetura de MLP	Utiliza o EQM como função objetivo para garantir uma rede com baixa complexidade e boa generalização. Desconsidera a otimização da taxa de aprendizado.

Fonte: Autoria própria.

3 MATERIAIS E MÉTODOS

Neste capítulo são apresentados os materiais e a metodologia utilizada no desenvolvimento deste trabalho, onde é explicitado detalhes sobre a implementação e validação dos sistemas.

3.1 Materiais

Na presente seção é elencado os materiais utilizados no decorrer deste trabalho, desta maneira, as subseções a seguir estão organizadas de forma a apresentar o banco de dados e a ferramenta empregada na implementação dos sistemas.

3.1.1 Banco de dados

Para realização deste trabalho foram utilizadas cinco bases de dados, onde três delas são bases de dados públicas, disponível no consolidado repositório de aprendizado de máquina, UCI, e são destinadas para validação dos sistemas. As outras duas bases de dados são compostas por sinais eletromiográficos e, por se tratar de uma base de dados composta por sinais biológicos, foi destinada para verificar o desempenho na solução de problemas específicos, para a configuração que apresentar os melhores de cada um dos sistemas na etapa de validação.

A seguir, é apresentada uma breve descrição da origem e características das bases de dados de diagnósticos médicos, provenientes do repositório de aprendizado de máquina:

- **Breast Cancer data:** Este banco de dados obtido do Hospital da Universidade de Wisconsin, EUA, apresenta um conjunto de dados relacionados ao diagnóstico médico de câncer de mama e possui 683 amostras de massa mamária em um espaço de nove dimensões de características e dividido em duas classes de tumores, malignos e benignos;
- **Mammographic Mass data:** Coletada no Instituto de Radiologia do Universidade Erlangen-Nuremberg entre 2003 e 2006, esta base de dados apresenta um conjunto de dados composto por cinco atributos e 830

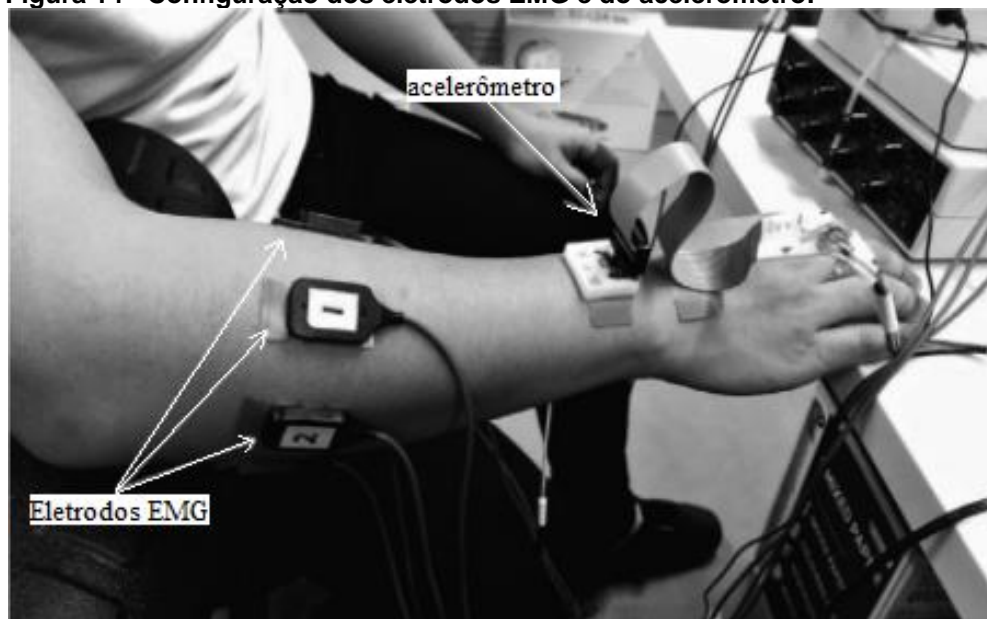
amostras, utilizando no diagnóstico para prever a gravidade de uma lesão de massa mamográfica (malignos ou benignos);

- **Thyroid data:** Este banco de dados, coletado pela Universidade James Cook, Townsville – Austrália, é composto por 215 amostras com cinco atributos, onde o mesmo é usado para tentar prever a tireoide de um paciente entre as classes: eutireoidismo, hipotireoidismo ou hipertireoidismo.

Os dados de EMG que foram utilizados para verificar o desempenho dos sistemas, fazem parte de uma base coletada por Khushaba *et al* (2016), na *University of Technology* em Sidney, Austrália, para classificação de movimentos da mão a partir de sinais eletromiográficos do antebraço. O banco de dados completo é composto por amostras coletadas a partir de 10 participantes voluntários com idades entre 20 e 33 anos e com circunferência média do antebraço de $26,59 \pm 2,41$ cm.

Em Khushaba *et al* (2016) é apresentado que os dados de 6 eletrodos EMG foram capturados com um sistema Bagnoli desktop EMG system, da empresa americana Delsys Inc, com ganho de 1000. Os eletrodos foram dispostos ao longo da circunferência do antebraço, espaçados igualmente, além disso, um eletrodo de referência foi disposto no punho dos voluntários, configuração pode ser observada na Figura 14. Um conversor de 12 bits Analógico/Digital da National Instruments, BNC-2090, foi utilizado para amostrar o sinal a uma frequência de 4000Hz.

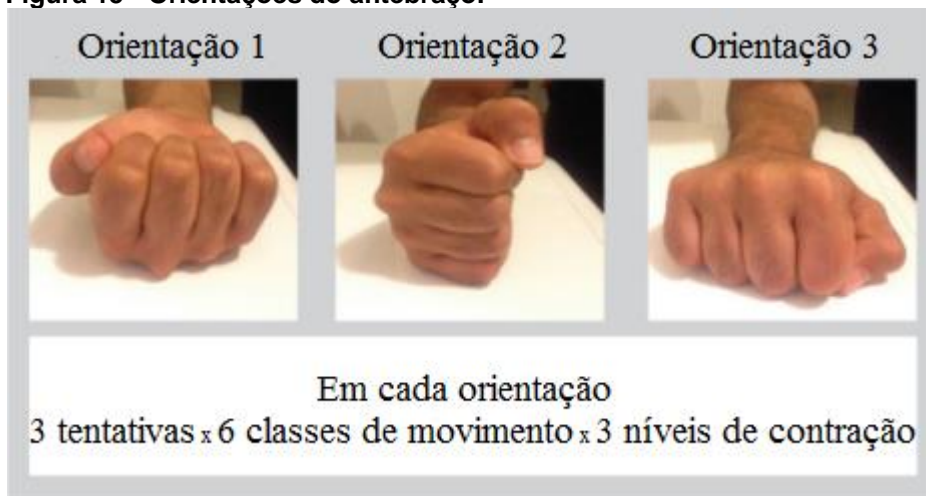
Figura 14 - Configuração dos eletrodos EMG e do acelerômetro.



Fonte: Adaptado de Khushaba *et al.* (2016).

Com os instrumentos devidamente dispostos, os voluntários realizaram uma série de movimentos divididos em seis classes que variam de C1 a C6, sendo esses movimentos: Mão Fechada (C1), Mão Aberta (C2), Extensão do Punho (C3), Flexão do Punho (C4), Desvio Ulnar do punho (C5) e Desvio Radial do Punho (C6). Para execução dos movimentos também foram consideradas três diferentes orientações do antebraço: punho totalmente supinado, em repouso e totalmente pronado, que são representados pelas orientações de 1 a 3 da Figura 15. Todos os seis movimentos foram repetidos pelos participantes, para cada orientação do antebraço, em três níveis distintos de contração muscular: baixa, média e alta. Para cada um destes movimentos foram registrados 3 ensaios, iniciando pelo nível mais baixo de contração muscular e aumentando sua intensidade gradualmente, totalizando assim, 162 tentativas por voluntário. Com o intuito de evitar a fadiga, cada tentativa teve duração de 5 segundos e os participantes tiveram até 10 segundos de descanso entre cada uma delas.

Figura 15 - Orientações do antebraço.



Fonte: Adaptado de Khushaba et al. (2016).

No entanto, neste presente trabalho, foram utilizados os dados referentes a um único paciente e organizados em duas bases distintas, onde uma é direcionada a classificação dos movimentos da mão e a outra para classificação dos níveis de contração muscular.

As informações referentes às características das bases de dados utilizadas neste trabalho, podem ser visualizadas com mais clareza no Quadro 2.

Quadro 2 – Características das bases de dados utilizadas no trabalho.

Características	Breast Cancer	Mammographic Mass	Thyroid	Classes de Movimentos	Níveis de Contração
Número de amostras	683	830	215	3708	3708
Quantidade de atributos	9	5	5	48	48
Quantidade de classes	2	2	3	6	3

Fonte: Autoria própria.

3.1.2 Ferramenta utilizada

O *software* MATLAB®, da MathWorks, foi a ferramenta utilizada para elaboração deste trabalho. A escolha desta ferramenta para realização de todas as etapas práticas foi feita devido a familiaridade dos autores com a mesma, além de ser um *software* bastante popular, com a presença de *toolboxes* amplamente utilizadas e consolidadas no desenvolvimento de RNAs e estar disponível para elaboração de estudos e pesquisas nos laboratórios da UTFPR-CP.

3.2 Métodos

A proposta deste trabalho se baseia no desenvolvimento de dois SIH Auxiliares que utilizam PSO e ABC para realizar a otimização dos valores de parâmetros internos de uma RNA do tipo MLP.

Os parâmetros que foram otimizados a partir da implementação desses sistemas são: número de camadas neurais escondidas, quantidade de neurônios em cada uma dessas camadas e o valor da taxa de aprendizado. Com isto, pretende-se comprovar que a utilização de algoritmos bioinspirados em conjunto com RNAs pode auxiliar na melhoria dos resultados da classificação de padrões, comparado com sistemas que são baseados somente em RNAs.

Desta forma, as subseções seguintes detalham os métodos e procedimentos que foram realizados para o desenvolvimento desses sistemas de otimização e cada uma das partes pertencentes aos mesmos, tais como: classificador de padrões (MLP), PSO e ABC (algoritmos de otimização), SIH (união das técnicas) e as informações sobre o tratamento do banco de dados EMG que foi utilizado na fase de testes com o algoritmo selecionado. Por fim, é descrito a metodologia de validação e comparação do desempenho dos sistemas otimizados.

3.2.1 *Perceptron* multicamadas

Como apresentado anteriormente, a aplicação do *Perceptron* multicamadas utilizada foi a de classificador de padrões que utiliza-se da técnica de reconhecimento de padrões como meio de classificar dados de entrada em determinadas classes ou objetos, fundamentando-se em características-chave (MATHWORKS, 2017).

Para estratégia empregada neste presente trabalho, o foco foi direcionado ao método supervisionado. O mesmo possui este nome por utilizar algoritmos de aprendizado onde as saídas desejadas são conhecidas e as alterações dos pesos sinápticos são calculadas de forma que a defasagem entre as saídas da rede passe a diminuir para valores de erro aceitáveis em relação a desejada, levando em consideração os propósitos de generalização de soluções.

Outra decisão importante está ligada a estrutura do sistema classificador, que é composto por uma única RNA responsável por saídas que agrupam os resultados entre todas as classes determinadas. Diferente das redes especialistas, que são um subconjunto de RNAs, onde cada uma é responsável por identificar se uma entrada é ou não pertencente a uma classe em particular.

Além disso, para a implementação deste classificador de padrões, procedimentos de preparação dos dados devem ser adotados. Sendo assim, é importante salientar que os bancos de dados de sinais eletromiográficos que foram utilizados neste trabalho passaram por uma etapa de pré-processamento antes da sua utilização nos experimentos. Além disso, antes da realização da validação dos sistemas e dos testes, cada uma das bases de dados foi embaralhada a partir de um método baseado em carteadado, para modificar a ordem original das amostras de dados da mesma, de modo que a cada experimento tenha-se uma ordem diferente dos dados de entrada.

O algoritmo empregado para treinamento da MLP se trata de uma versão aperfeiçoada do *Backpropagation*, o método de Levenberg-Marquardt, implementado através da função de treinamento de rede '*trainlm*' do MATLAB. Sua seleção foi feita pois, este é frequentemente recomendado como algoritmo supervisionado de primeira escolha, além de ser apresentado em Silva, Spatti e Flauzino (2010, p. 117), como um método que reduz o tempo de convergência e o esforço computacional requerido pelo algoritmo *Backpropagation*.

Segundo Haykin (2007, p. 206), uma MLP pode, em geral, aprender mais rápido se a função de ativação for antissimétrica do que quando for não-simétrica. Um exemplo popular de função antissimétrica é a tangente hiperbólica. Deste modo, dentre as funções de ativação mais populares, as escolhidas para elaboração deste trabalho foram: tangente hiperbólica para as camadas escondidas, que pertence à família das funções sigmoidais e são frequentemente utilizadas para reconhecimento de padrões, e rampa linear para a camada de saída, permitindo a identificação de grau de confiança dos resultados obtidos. Ambas foram aplicadas a partir das funções de transferência neural '*tansig*' e '*purelin*' do MATLAB.

Os parâmetros de treinamento da rede como o número de épocas e o valor do *goal*, ou erro, foram fixados mediante a testes realizados com a RNA isolada dos algoritmos bioinspirados, onde os valores obtidos para esses parâmetros foram 200 e 1×10^{-10} , respectivamente.

Por fim, foi utilizado o método *one of c-class*, que associa a saída de cada neurônio diretamente à classe correspondente, portanto, neste caso a quantidade de neurônios na camada de saída é igual ao número de classes do problema. Na saída da RNA, foi efetuado um pós-processamento dos dados, realizando-se a saturação das saídas obtidas para -1 e 1, ou enviar o valor para 2 tornando possível a identificação dos acertos, erros e incertezas, ou seja, determinar a confiabilidade das respostas fornecidas pela rede. Essa saturação foi, a priori, determinada conforme a Tabela 1.

Tabela 1 – Faixas de saturação das saídas da RNA.

Saídas da RNA (y)	Resposta
$y \geq 0.98$	1
$-0.98 < y < 0.98$	2
$y \leq -0.98$	-1

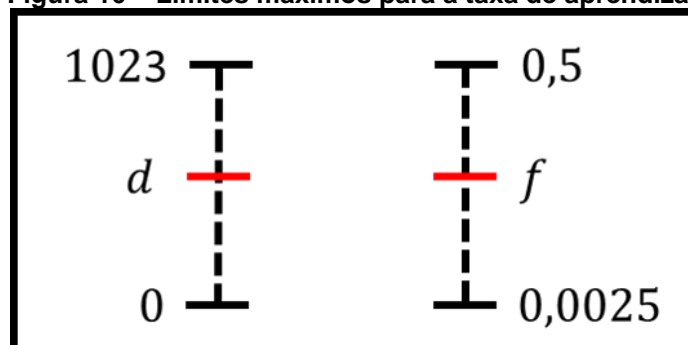
Fonte: Autoria própria.

3.2.2 Otimização por Enxame de Partículas

A implementação do sistema de otimização por enxame de partículas, em seu funcionamento básico, se baseia em fazer com que o seu *fitness* caminhe em direção a uma resposta que faça a função objetivo ter o melhor resultado possível a partir de uma dada condição inicial, seja maximizando ou minimizando um valor. O *fitness* utilizado neste trabalho foi o erro quadrático médio (EQM) gerado pela RNA, de forma análoga ao proposto por Carvalho (2007) e por Shahraki e Zahiri (2017). Para tal, foi necessário levar em consideração alguns pontos cruciais de funcionamento lógico do algoritmo, tais como: variáveis a serem otimizadas, inicialização da população, velocidade das partículas e topologia do sistema.

Como os parâmetros que mais afetam a resposta da RNA são referentes as camadas escondidas e a taxa de aprendizado da RNA, os mesmos foram escolhidos para serem otimizados. Tendo em vista que as variáveis otimizadas pertencem a dois domínios diferentes, foi verificado em Yoshizumi *et al.* (2018) a necessidade de se trabalhar com a versão binária do algoritmo, afim de evitar soluções aproximadas e garantir a busca local. A quantidade de neurônios em cada camada escondida teve sua resolução dada em 6 bits, correspondente ao intervalo de 0 a 63. Enquanto a resolução da taxa de aprendizado é dada por 10 bits, que corresponde proporcionalmente ao intervalo de 0,0025 a 0,5, como apresentado na Figura 16.

Figura 16 – Limites máximos para a taxa de aprendizado.



Fonte: Autoria própria.

Da teoria dos segmentos proporcionais, pode-se derivar uma expressão matemática para o cálculo do valor de d (SILVA, 2017):

$$\frac{0,5 - 0,0025}{f - 0,0025} = \frac{1023 - 0}{d - 0} \quad (18)$$

desta forma, tem-se:

$$d = 1023 \times \frac{f - 0,0025}{0,5 - 0,0025} \quad (19)$$

onde, d é a representação decimal da taxa de aprendizado e f é o número em ponto flutuante a ser convertido.

Para realizar a inicialização da população e a atualização das soluções da mesma, deve-se considerar a importância do método randômico utilizado nos cálculos, como esta implementação é realizada no software MATLAB®, foi utilizado a função “*rand()*”, que retorna um único número aleatório uniformemente distribuído no intervalo de 0 a 1.

Outro ponto relevante no algoritmo PSO, diz respeito ao comportamento da velocidade da partícula durante o processo de busca na região factível. A fim de obter um comportamento mais eficiente, foi adicionado o parâmetro inercial (w), responsável por amortecer a velocidade com o decorrer das iterações, valor que inicia-se em 0,9 e termina em 0,4, e os parâmetros cognitivo ($c_1 = 2$) e social ($c_2 = 2$), responsável por ponderar o auto aprendizado e o aprendizado social das partículas.

A forma como as partículas se comunicam durante o processo iterativo no PSO é conhecido como topologia de otimização. Pode-se citar como topologias amplamente utilizadas neste algoritmo a “Melhor global”, onde todas as partículas se comunicam e “Melhor local”, onde as partículas se comunicam apenas com seus vizinhos. Na implementação da otimização para a RNA foi empregado a topologia mais popular e consolidada, a “Melhor global”.

Considerando todas as configurações e procedimentos citados para a implementação da otimização por enxame de partículas, pode-se resumir suas funções básicas em inicializar soluções, calcular *fitness*, realizar troca de informação e calcular nova solução. A síntese do funcionamento do algoritmo de Otimização por Enxame de Partículas implementado, pode ser vista com mais detalhes na Figura 17.

Figura 17 - Pseudocódigo do algoritmo PSO implementado.

```

Atribui parâmetros
PARA cada partícula  $i$  da população FAÇA
    Inicializar  $x_{ij}$  com conjunto  $j$  de soluções aleatória para o problema
    Inicializar  $v_{ij}$  com conjunto  $j$  de velocidades aleatória para o problema
     $fit_i \leftarrow$  Calcula fitness da solução  $x_{ij}$ 
     $P_{best\ i} \leftarrow fit_i$ 
     $G_{best} \leftarrow$  melhor entre  $P_{best\ i}$  e  $G_{best}$ 
FIMPARA
REPETIR
    PARA cada partícula  $i$  da população FAÇA
        Atualizar  $v_{ij}$ , usando (5)
        Converter  $v_{ij}$  em probabilidade ( $S(v_{ij})$ ), usando (6)
        Atualizar  $x_{ij}$ , com base em  $S(v_{ij})$ , usando (7)
         $fit_i \leftarrow$  Calcula fitness da solução  $x_{ij}$ 
         $P_{best\ i} \leftarrow$  melhor entre  $fit_i$  e  $P_{best\ i}$ 
         $G_{best} \leftarrow$  melhor entre  $P_{best\ i}$  e  $G_{best}$ 
        Melhor solução  $\leftarrow x_{ij}$  referente a  $G_{best}$ 
    FIMPARA
ATÉ ciclo máximo
  
```

Fonte: Autoria própria.

3.2.3 Colônia Artificial de Abelhas

A metaheurística ABC, como citado, se baseia no comportamento das abelhas na busca por alimento, sendo assim, este método de otimização tenta por meio deste comportamento, buscar dentro de um espaço limitado a melhor solução possível. O *fitness* utilizado para o ABC foi o mesmo utilizado para o PSO, o erro quadrático médio (EQM) gerado pela RNA. Para a implementação deste algoritmo foi necessário levar em conta alguns pontos essenciais para seu funcionamento adequado, como por exemplo, configuração dos parâmetros, método de busca aleatório, diversidade das soluções e busca local.

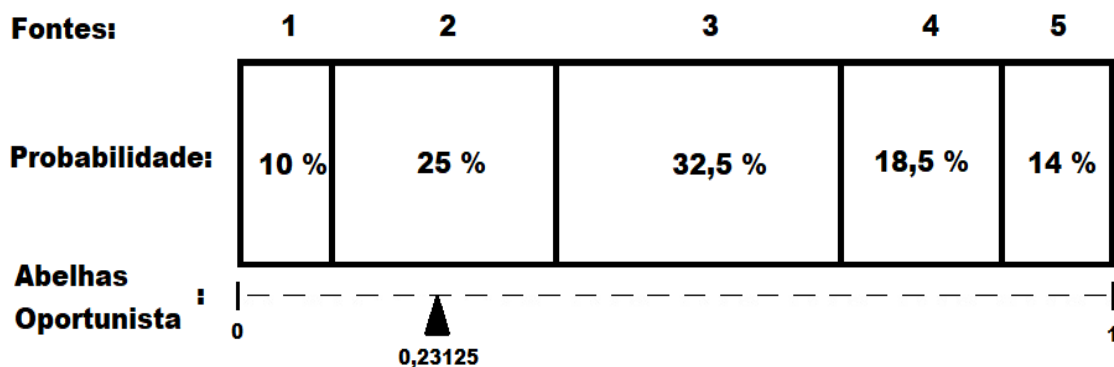
Assim como realizado para o Algoritmo de Otimização por Enxame de Partículas, tendo em vista que a quantidade de neurônios em cada camada escondida pertence ao conjunto dos números inteiros e a taxa de aprendizado ao conjunto dos

reais, foi implementado uma versão binária do algoritmo ABC. Desta forma, a quantidade de neurônios por camada tem sua resolução dada em 6 bits, enquanto a resolução da taxa de aprendizado é dada por 10 bits, que correspondem aos intervalos de 0 a 63 e 0,0025 a 0,5, respectivamente. Esta conversão também é realizada utilizando a Equação (19), apresentada na seção anterior.

Para realizar a inicialização da população e a atualização das soluções da mesma, foi preciso considerar a importância do método randômico utilizado nos cálculos. Como esta implementação é realizada no software MATLAB®, foi utilizado a função “*rand()*”, que retorna um único número aleatório uniformemente distribuído no intervalo de -1 a 1 para as atualizações e de 0 a 1 para as inicializações.

O método utilizado para representar o comportamento da escolha da fonte a ser explorada pelas abelhas oportunistas se baseia na probabilidade, calculada pela equação (10) e um valor aleatório entre 0 e 1. A fim de facilitar o entendimento da relação entre as fontes, a probabilidade de escolha de uma das fontes e as abelhas oportunista, um exemplo é mostrado na Figura 18, onde observa-se que a área de ocupação referente a cada fonte é proporcional a sua probabilidade de escolha. Assim, as fontes que apresentarem maiores probabilidades, possuem maior chance de serem escolhidas. A seta apresentada na parte de baixo da imagem, representa a escolha da abelha, sendo um valor aleatório entre 0 e 1, no exemplo da Figura 18, o valor escolhido aleatoriamente foi de (0,23125), logo, nesta exemplificação a fonte escolhida foi a número 2.

Figura 18 - Exemplo do método de escolha da fonte a ser explorada.



Fonte: Autoria própria.

A busca local realizada pelas abelhas oportunistas foi implementada seguindo a equação (9), onde as novas soluções são variações das soluções encontradas pelas abelhas campeiras.

Um dos parâmetros mais importantes na configuração da otimização ABC é o LIMIT que representa quando uma fonte encontrada por uma abelha campeira se torna improdutivo. No algoritmo implementado, esse parâmetro foi configurado para abandonar uma determinada fonte após 30 iterações sem evolução.

Como é desejado realizar comparações entre os sistemas PSO e ABC, é interessante que ambos processem o mesmo número de redes durante a fase de otimização, para tal, a otimização por colônia artificial de abelhas foi modificada a fim de fixar a quantidade de redes por iteração. A modificação foi realizada em relação ao abandono de uma fonte, onde ao invés de adicionar uma nova abelha exploradora, para buscar uma nova fonte, uma das abelhas oportunistas foi transformada em exploradora, desta forma, manteve-se um número fixo de abelhas por iteração.

Considerando todas as configurações e procedimentos citados para a implementação da otimização por Colônia Artificial de Abelhas, pode-se resumir suas funções básicas em inicializar soluções, calcular *fitness*, calcular probabilidade, decidir qual fonte será explorada pelas abelhas oportunistas, realizar busca local e verificar abandono da fonte. A síntese do funcionamento do algoritmo de Colônia Artificial de Abelhas implementado, pode ser vista com mais detalhes nas Figuras 19 e 20.

Figura 19 – Pseudocódigo do cálculo da dissimilaridade.

FUNÇÃO dissimilaridade (x_{ij})

Determinar M_{11}, M_{01} e M_{10} , usando x_{ij}

Calcular $Dissim(x_{ij}, x_{kj})$, usando (11)

Gerar $Dissim(v_{ij}, x_{ij})$, usando (12)

Calcular novos valores de M_{11}, M_{01} e M_{10} , minimizando (13) sujeito a (14), (15) e (16)

Determinar v_{ij} a partir dos novos valores de M_{11}, M_{01} e M_{10}

RETORNAR (v_{ij})

Fonte: Autoria própria.

Figura 20 - Pseudocódigo do algoritmo ABC implementado.

```

Atribui parâmetros
PARA cada abelha campeira  $i$  da população FAÇA
    Inicializar  $x_{ij}$  com conjunto  $j$  de soluções aleatória para o problema, usando (8)
     $fit_i \leftarrow$  Calcula fitness da solução  $x_{ij}$ 
     $Best_i \leftarrow fit_i$ 
FIMPARA
REPETIR
    PARA cada abelha campeira  $i$  da população FAÇA
        Verificar se a fonte da abelha campeira  $i$  será abandonada
        SE fonte  $i$  abandonada
            Gerar  $x_{ij}$ , chamando abelha exploradora, equação (8)
             $fit_i \leftarrow$  Calcula fitness da solução  $x_{ij}$ 
        SENÃO
             $x_{ij} \leftarrow$  dissimilaridade ( $x_{ij}$ )
             $fit_i \leftarrow$  Calcular fitness da solução  $x_{ij}$ 
        FIMSE
    FIMPARA
     $\overrightarrow{Best} \leftarrow$  recebe as melhores soluções entre  $\overrightarrow{fit}$  e  $\overrightarrow{Best}$ 
    Calcular probabilidade  $p_i$  para as soluções  $x_{ij}$ , a partir da equação (10)
    Determinar abelha campeira escolhida pelas da abelha oportunista
    PARA cada abelha oportunista  $i$  da população FAÇA
         $v_{ij} \leftarrow$  dissimilaridade (abelha campeira escolhida pela abelha oportunista  $i$ )
         $fit_i \leftarrow$  Calcular fitness da solução  $v_{ij}$ 
    FIMPARA
     $\overrightarrow{Best} \leftarrow$  recebe as melhores soluções entre  $\overrightarrow{fit}$  e  $\overrightarrow{Best}$ 
    PARA cada abelha campeira  $i$  da população FAÇA
        Determinar o abandono da fonte  $i$ 
        SE fonte  $i$  abandonada
            Incrementar  $\rightarrow$  Quantidade_fontes_abandonadas
            Chamar abelha exploradora, busca aleatória, equação (8)
        FIMSE
    FIMPARA
    Melhor solução  $\leftarrow x_{ij}$  referente ao melhor fitness de  $\overrightarrow{Best}$ 
ATÉ ciclo máximo

```

Fonte: Autoria própria.

3.2.4 Sistema inteligente híbrido

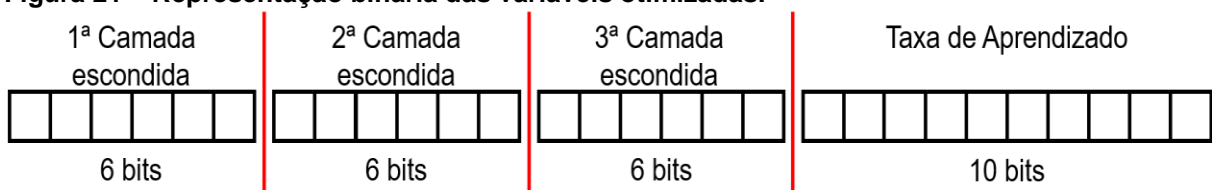
Neste trabalho foram implementados dois sistemas híbridos inteligentes, sendo ambos classificados como SIH de substituição de função, onde os algoritmos bioinspirados PSO e ABC, foram utilizados de forma que auxiliem na classificação de padrões realizada pela RNA. Como pode ser observado, do ponto de vista dos sistemas de otimização, a RNA é responsável por calcular o *fitness* da solução que, como apresentado anteriormente, foi escolhido como sendo o erro quadrático médio (EQM). Este erro de classificação é calculado na etapa de validação da rede, ou seja, é calculado com base nos dados que ainda não foram vistos pela rede.

Os dois sistemas de otimização auxiliarão a RNA no processo de escolha da topologia que melhor se adequa ao conjunto de dados a serem classificados. Os parâmetros internos da topologia da RNA que possuem maior influência na classificação dos dados e, que por esse motivo foram escolhidos com parâmetros a serem otimizados, são:

- **Número de camadas escondidas:** quantidade de camadas entre a camada de entrada e a camada de saída, diretamente relacionada a complexidade de classificação dos dados;
- **Neurônios em cada camada escondida:** quantidade de neurônios existente em cada camada escondida, também relacionado a complexidade de classificação dos dados. Mal configurado pode causar *overfitting* ou *underfitting* (SILVA; SPATTI; FLAUZINO, 2010 p. 153);
- **Taxa de aprendizado:** velocidade de aprendizado dos pesos sinápticos, quando mal configurado prende a rede em mínimos locais, seu valor assim como os outros, varia de acordo com base de dados.

A representação das variáveis otimizadas na forma de bits pode ser observada na Figura 21.

Figura 21 – Representação binária das variáveis otimizadas.



Fonte: Autoria própria.

Como os dois sistemas de otimização necessitam de uma região de busca definida, os parâmetros citados foram limitados por regiões que sejam condizentes com as variáveis e ao mesmo tempo possa trazer bons resultados. Os intervalos de limitação que foram utilizados para os parâmetros otimizados são apresentados na Tabela 2.

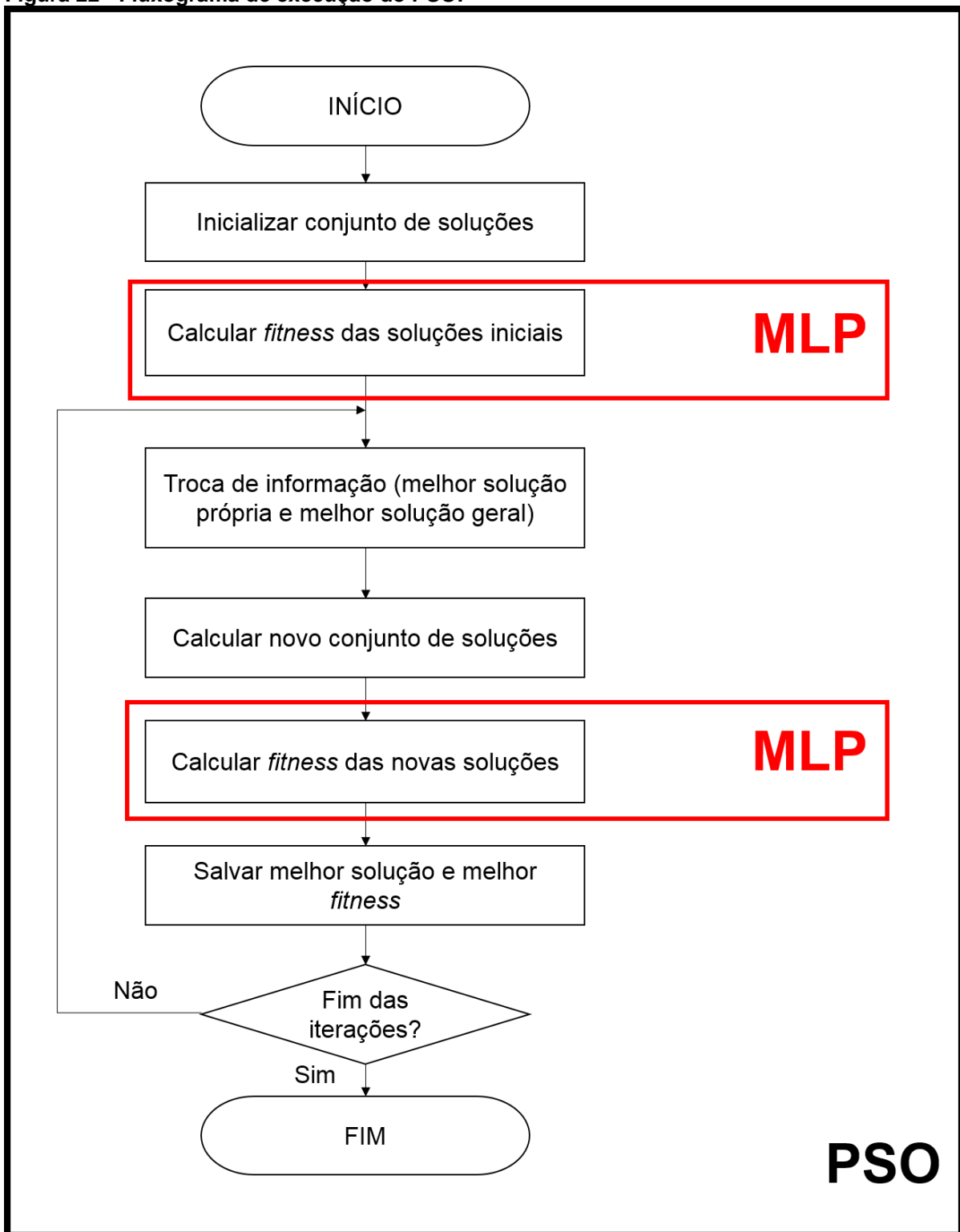
Tabela 2 – Restrições dos parâmetros de otimização.

Parâmetros	Restrições
Quantidade de neurônios na 1º camada	$0 \leq qnt_n1 \leq 63$
Quantidade de neurônios na 2º camada	$0 \leq qnt_n2 \leq 63$
Quantidade de neurônios na 3º camada	$0 \leq qnt_n3 \leq 63$
Taxa de aprendizado (δ)	$0,0025 \leq \delta \leq 0,5$

Fonte: Autoria própria.

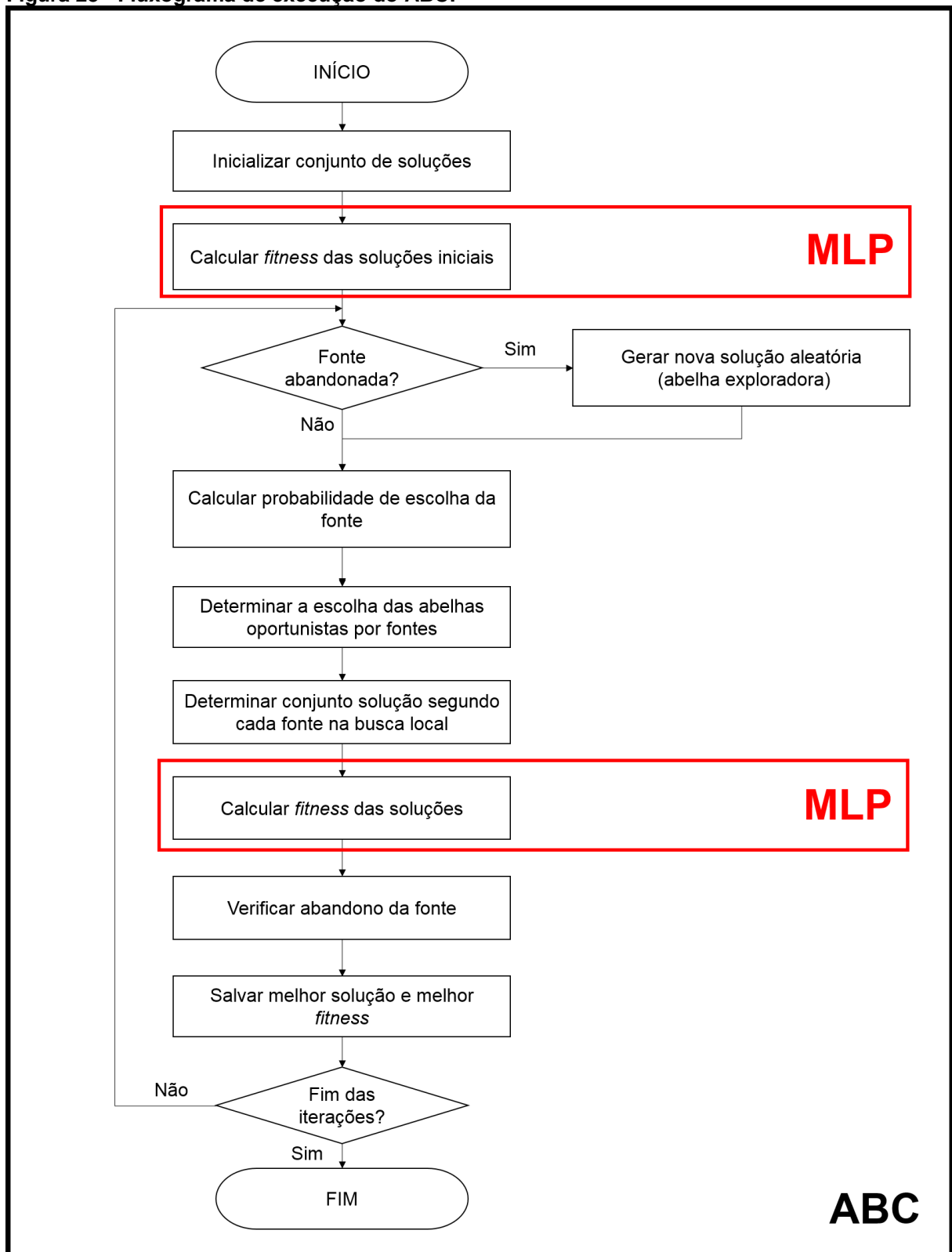
A fim de facilitar o entendimento da relação entre os sistemas inteligentes, é apresentado nas Figuras 22 e 23 o fluxograma de funcionamento de cada sistema de otimização e os momentos onde a RNA é requisitada.

Figura 22 - Fluxograma de execução do PSO.



Fonte: Autoria própria.

Figura 23 - Fluxograma de execução do ABC.



Fonte: Autoria própria.

Por fim, com o intuito de tornar o ABC e o PSO mais adaptados a RNA, foi implementado um critério de desempate para o caso de duas soluções diferentes possuírem o mesmo *fitness*. O critério escolhido como desempate foi a complexidade

computacional da solução, desta forma, caso duas soluções distintas tenham o mesmo valor de *fitness*, a melhor solução é aquela com menor complexidade computacional, considerando que as RNAs possuem complexidade algorítmica da ordem de $O(n^2)$. Desta forma, a complexidade é calculada segundo a equação (20).

$$C = \sum_{i=0}^3 n_i^2 \quad (20)$$

onde, C é a complexidade e n_i é o número de neurônios na camada escondida i .

3.2.5 Tratamento do banco de dados eletromiográficos

Há algum tempo técnicas de extração de características de sinais vêm sendo empregadas. Tais técnicas facilitam a análise dos sinais, nos apresentando informações mais claras, além de tornar possível a aplicação de tratamentos para eliminação de ruídos. Contudo, as Transformadas *Wavelets*, extraem informações tanto no tempo quanto em frequência, diferente de outras técnicas que extraem características apenas no domínio da frequência (HAFIZ; ABERCROMBIE; EATON, 2017). Por este motivo, optou-se pela utilização das Transformadas *Wavelets* no tratamento dos dados EMG utilizados neste trabalho.

Os dados referentes a cada paciente foram unificados e organizados em um único banco de dados, para que o mesmo possa ser tratado utilizando a Transformada *Wavelet Packet* com a finalidade de realizar a extração das características dos sinais eletromiográficos, bem como diminuir a quantidade de dados que foram apresentados a RNA.

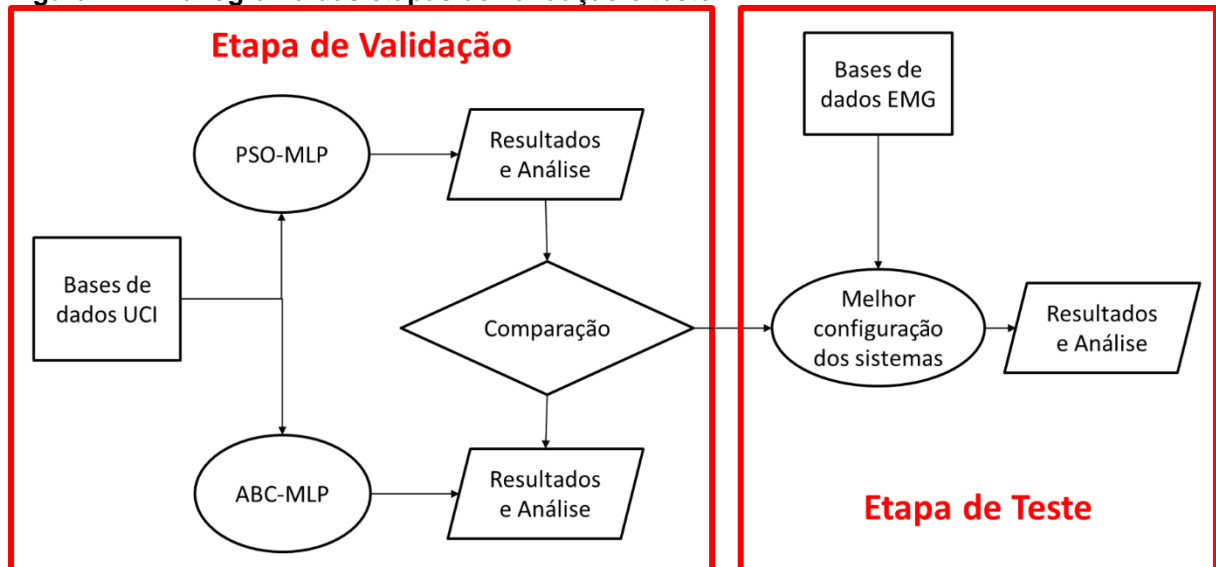
A escolha da Transformada *Wavelet Packet* utilizada para o tratamento dos dados EMG foi realizada com base em resultados de estudos prévios realizados no Grupo de Estudos em Sinais Biológicos (GE4Bio) da UTFPR-CP. Essa escolha é apresentada com mais detalhes em Dos Santos *et al* (2018), onde, a partir dos resultados obtidos verificou-se que a aplicação de *wavelets* da família Haar aos dados EMG, apresentou resultados superiores quando comparada com as famílias Daubechies (2-5), Coiflets (1-5) e Symlets (2-5). Todas com nível de decomposição 3 e janelamento de 2048 amostras com sobreposição de informações de 50%.

Vale ressaltar que antes da sua utilização nos experimentos e testes que foram realizados, o banco de dados tratado utilizando as Transformada *Wavelet Packet* teve seus valores normalizados entre -1 e 1, através da função ‘*mapminmax*’ do MATLAB.

3.2.6 Validação e testes

Os resultados obtidos a partir da aplicação dos sistemas ABC-MLP e PSO-MLP foram analisados em duas etapas sendo elas, validação e teste, como apresenta a Figura 24.

Figura 24 – Fluxograma das etapas de validação e teste.



Fonte: Autoria própria.

Durante a etapa de validação de desempenho dos sistemas propostos neste trabalho, foi realizada uma análise comparativa entre os sistemas e verificado a configuração que apresentou os melhores resultados utilizando as bases de dados de diagnósticos médicos “*Breast Cancer*”, “*Mammographic Mass*” e “*Thyroid*” obtidas a partir do repositório de aprendizado de máquina, UCI.

Os sistemas de otimização tiveram sua validação realizada a partir de 10 configurações diferentes, com cada configuração sendo executada 10 vezes, acarretando em um total de 100 execuções. Cada combinação resultante da quantidade de iterações e número da população irá gerar um total de 300 redes por execução, deste modo, 30.000 redes foram geradas para cada base de dados,

totalizando um valor de 90.000 redes para cada um dos sistemas. Esta relação pode ser observada na Tabela 3, a seguir.

Tabela 3 - Configurações dos algoritmos para validação dos sistemas.

Configuração	Nº de execuções	Nº de redes	Iterações	População
1	10	3000	60	5
2	10	3000	50	6
3	10	3000	30	10
4	10	3000	25	12
5	10	3000	20	15
6	10	3000	15	20
7	10	3000	12	25
8	10	3000	10	30
9	10	3000	6	50
10	10	3000	5	60
Total	100	30.000		

Fonte: Autoria própria.

Para definir qual configuração apresentou o melhor desempenho médio em cada um dos sistemas para ser utilizada com as bases de dados EMG, foi realizada uma análise a partir dos resultados obtidos na fase de validação. Sendo assim, foi averiguado o melhor *fitness* encontrado em cada configuração, a média dos *fitness* das melhores redes encontradas em cada conjunto de configuração, o tempo médio de execução de cada sistema, bem como os gráficos de evolução dos sistemas.

Na etapa de teste, a aquisição dos resultados foi realizada utilizando as bases de dados EMG, “Classes de Movimentos” e “Níveis de Contração”. Os sistemas foram executados com as configurações que apresentaram o melhor desempenho médio, escolhidas na etapa de validação. Sendo assim, os mesmos foram executados 10 vezes, gerando um total de 3000 redes testadas por sistema, para cada uma das bases EMG.

Tendo em mãos os resultados alcançados, foi verificado o desempenho dos sistemas propostos para averiguar o seu comportamento em relação aos problemas abordados. Para isto, foi analisado o melhor *fitness* encontrado, a média dos *fitness* das melhores redes encontradas em cada execução, o tempo médio de execução de cada sistema, os gráficos de evolução dos mesmos, bem como a eficiência de classificação em cada um dos problemas, fazendo o uso das matrizes de confusão.

4 RESULTADOS

Neste capítulo, inicia-se a apresentação e discussão dos resultados obtidos por meio dos experimentos executados segundo a metodologia descrita no capítulo anterior.

A primeira seção é destinada para a apresentação dos resultados obtidos durante a etapa de validação do sistema, empregando-se a classificação de padrões com ambos os sistemas otimizados, fazendo o uso das bases de dados de diagnósticos médicos “*Breast Cancer*”, “*Mammographic Mass*” e “*Thyroid*” obtidas a partir do repositório de aprendizado de máquina, UCI. Ainda nesta seção é realizado a comparação entre os sistemas bem como uma análise de desempenho para verificar qual configuração apresenta o melhor desempenho médio em cada um dos sistemas.

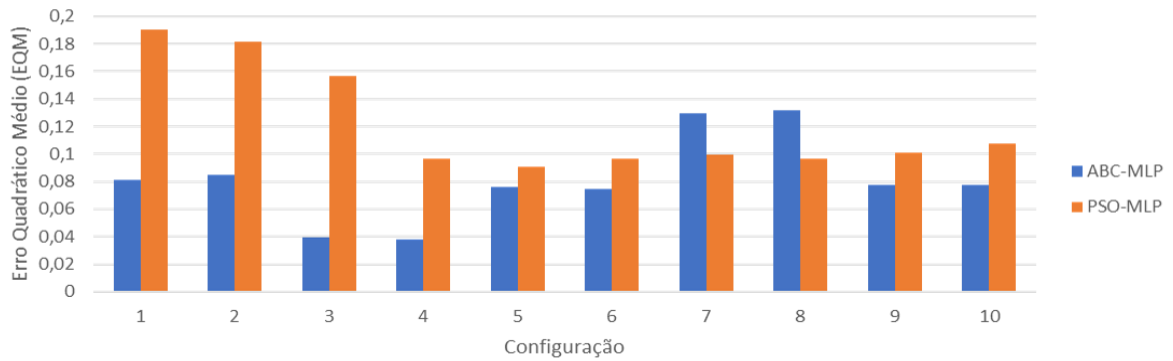
A segunda seção tem como objetivo verificar o desempenho dos sistemas na solução de problemas específicos, onde, para este presente trabalho foi representado por duas bases de dados de sinais EMG, referentes a classificação de 6 diferentes classes de movimentos da mão e dos 3 níveis de contração muscular em que esses movimentos foram realizados, respectivamente. Proporcionando uma perspectiva mais ampla a respeito do comportamento dos sistemas ao lidar com diferentes classes de problemas.

4.1 Resultados de validação

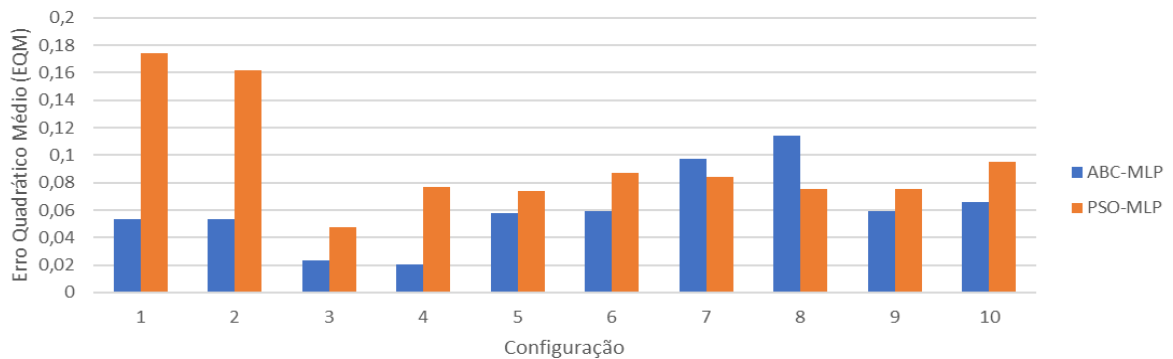
Durante todo o processo da etapa de validação foi realizado o treinamento de um total de 180 mil redes. A partir da otimização dos parâmetros das RNAs foi possível obter a resposta de cada execução para cada uma das configurações, o que nos forneceu um total de 10 redes por configuração, para cada um dos sistemas. A partir desses resultados, foi realizada uma análise comparativa para verificar o desempenho dos sistemas em minimizar o *fitness*. Desta forma, as Figuras 25, 26 e 27 apresentam os resultados referentes ao valor médio do *fitness* para as 10 melhores redes de cada configuração, bem como o menor valor encontrado, para as bases de dados “*Breast Cancer*”, “*Mammographic Mass*” e “*Thyroid*”, respectivamente.

Figura 25 – Análise comparativa entre os sistemas - *Breast Cancer*.

(a) Valor Médio



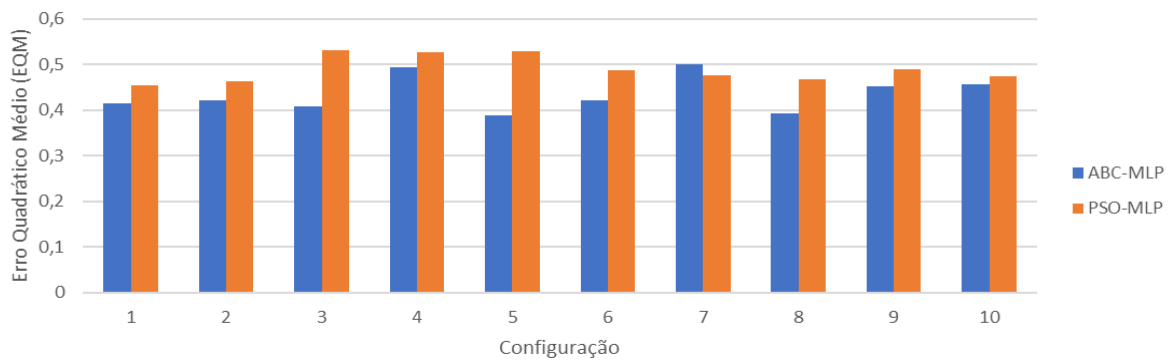
(b) Valor Mínimo



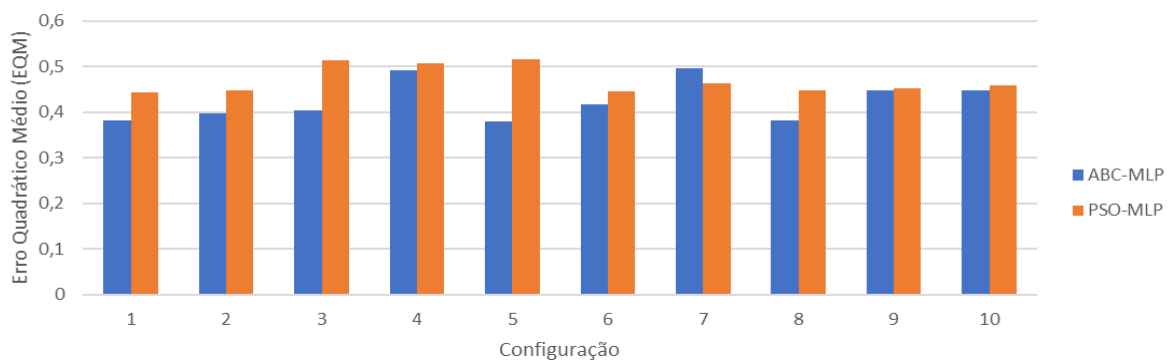
Fonte: Autoria própria.

Figura 26 – Análise comparativa entre os sistemas - *Mammographic Mass*.

(a) Valor Médio

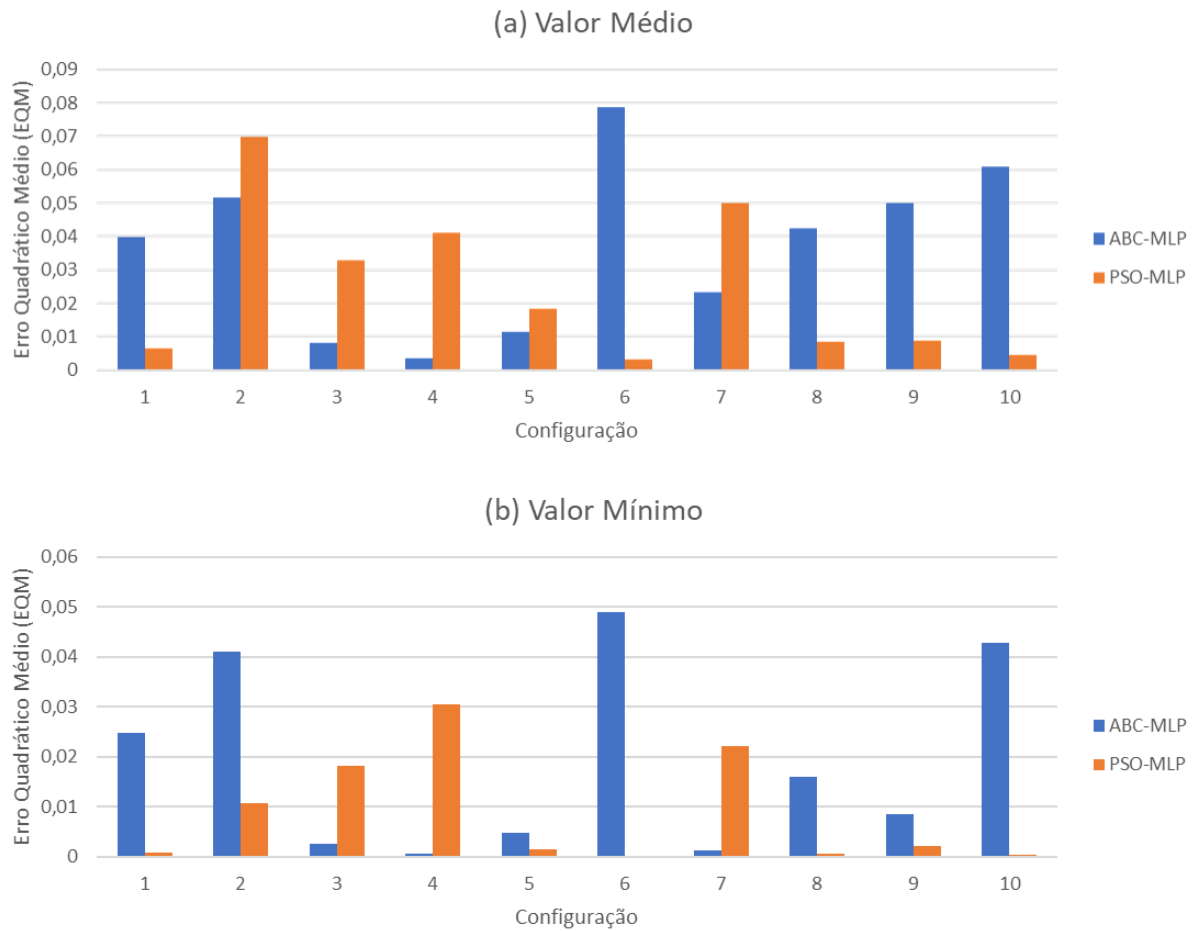


(b) Valor Mínimo



Fonte: Autoria própria.

Figura 27 - Análise comparativa entre os sistemas – *Thyroid*.

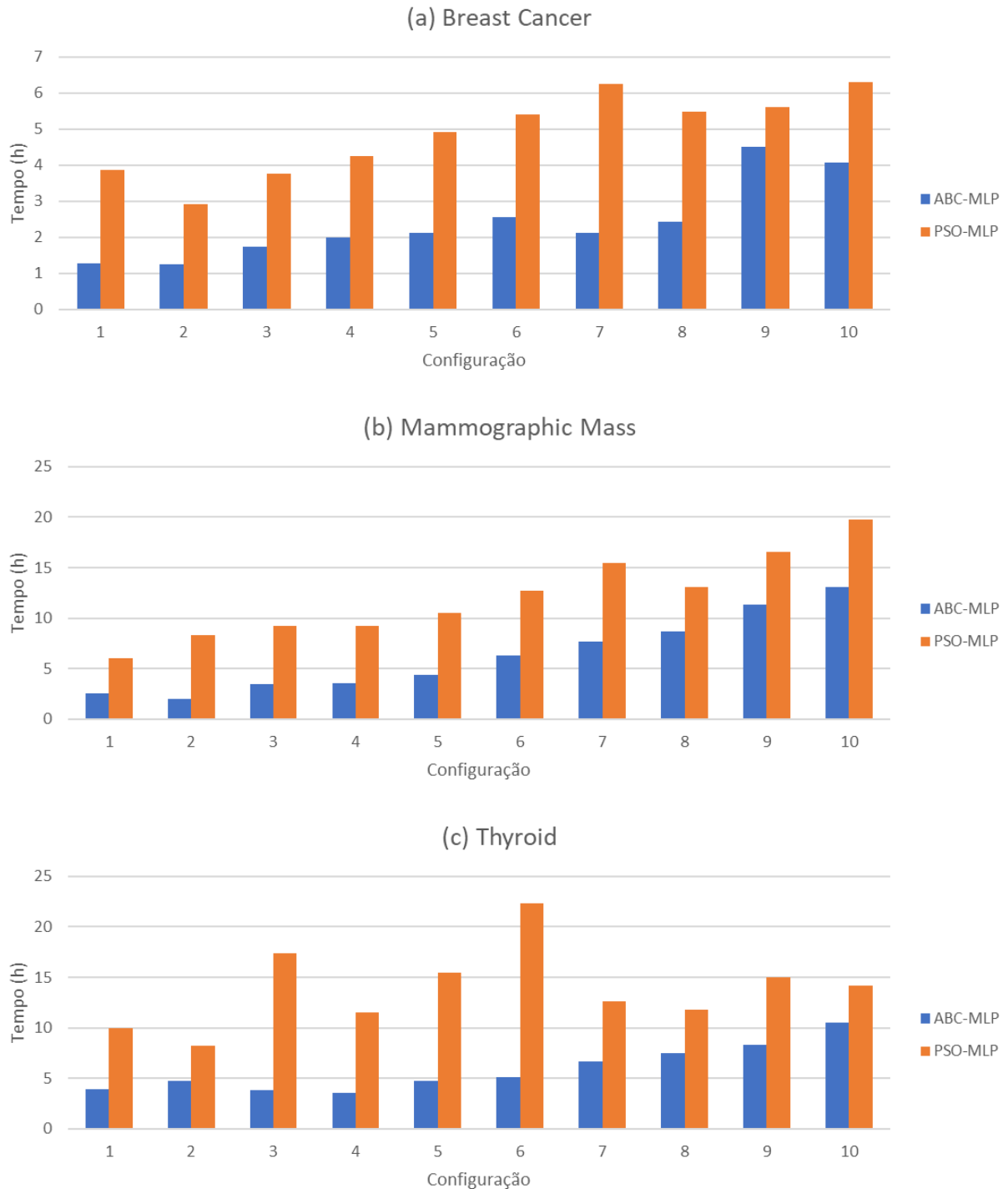


Fonte: Autoria própria.

Através dos gráficos apresentados, é possível observar que o sistema ABC-MLP apresentou resultado superior ao PSO-MLP, para as bases de dados “*Breast Cancer*” e “*Mammographic Mass*”. Já para a base de dados “*Thyroid*”, que se trata de uma base relativamente simples e com um pequeno número de amostras, ambos os sistemas apresentaram um ótimo resultado, obtendo valores extremamente baixos para o fitness. Entretanto, em uma análise geral, tanto para o valor médio, quando para o valor mínimo encontrado para o fitness, é possível observar um melhor desempenho por parte do sistema PSO-MLP.

Outra análise realizada foi em relação ao custo computacional, onde o tempo médio, em horas, para executar cada configuração em cada um dos sistemas é apresentado a seguir na Figura 28.

Figura 28 - Análise comparativa entre os sistemas – Tempo.



Fonte: Autoria própria.

Com base na Figura 28, é possível observar que o sistema ABC-MLP obteve uma melhor resposta, em relação ao custo computacional, para todas as configurações e com todas as bases de dados utilizadas na etapa de validação. Tendo em vista que todas as respostas foram obtidas utilizando computadores com as mesmas configurações (processador Intel Core i5 – 3330, 4 GB de memória RAM e

SSD de 256 GB), tal resultado tem como causa o fato de que o sistema ABC-MLP tende a convergir de forma mais eficaz para regiões que apresentam como solução RNAs que atingem os critérios de parada mais rapidamente, proporcionando então, um melhor custo computacional.

As melhores arquiteturas encontradas em cada um dos sistemas para as bases de dados utilizadas durante a validação, são apresentadas no Quadro 3.

Quadro 3 – Melhor arquitetura encontrada - *Fitness*.

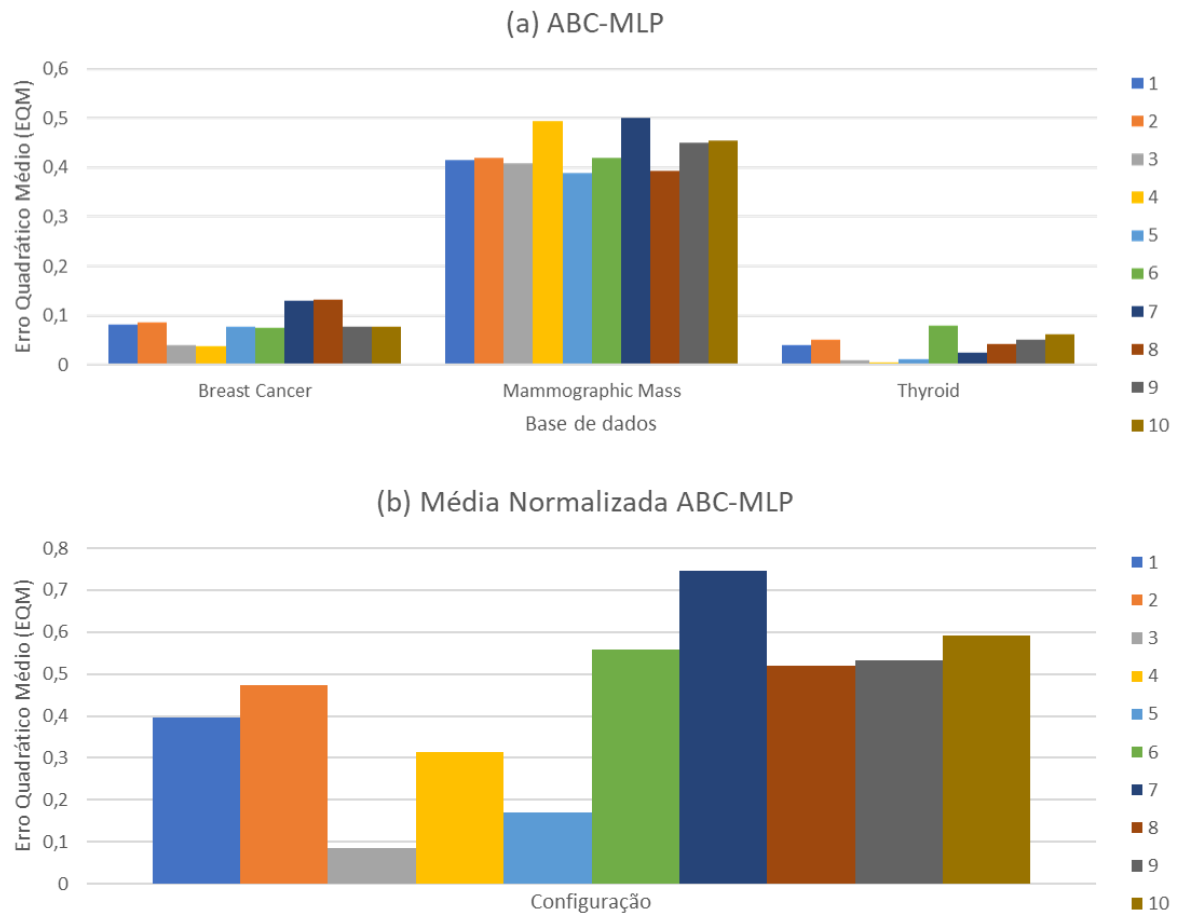
Bases de dados	ABC-MLP	PSO-MLP
Breast Cancer	[2 1 0 0,0336]	[20 21 4 0,3152]
Mammographic Mass	[2 2 2 0,0030]	[3 1 1 0,0113]
Thyroid	[15 16 41 0,0044]	[41 5 8 0,4232]

Fonte: Autoria própria.

Considerando as restrições dos parâmetros definidas na Tabela 2, foi determinado um espaço de busca relativamente grande para os sistemas, no entanto, verificou-se que apesar deste detalhe, os mesmos apresentaram soluções proporcionais a complexidade dos problemas abordados em cada base de dados. Para as bases de dados “*Mammographic Mass*” e “*Thyroid*”, ambos os sistemas apresentaram resultados aproximados em relação a complexidade das redes, onde o sistema PSO-MLP demonstrou uma leve vantagem. Porém, analisando a base de dados “*Breast Cancer*” é possível identificar uma larga vantagem para a solução apresentada pelo sistema ABC-MLP.

Terminadas as análises comparativas entre os sistemas, foi realizado uma análise para determinar qual das 10 configurações apresentou um melhor desempenho médio em cada sistema. Os resultados das Figuras 29(b) e 30(b) foram obtidos realizando o cálculo das médias normalizadas apresentadas nas Figuras 29(a) e 30(a), visto que os valores médios dos *fitness* em cada um dos problemas, apresentam intervalos de variação distintos.

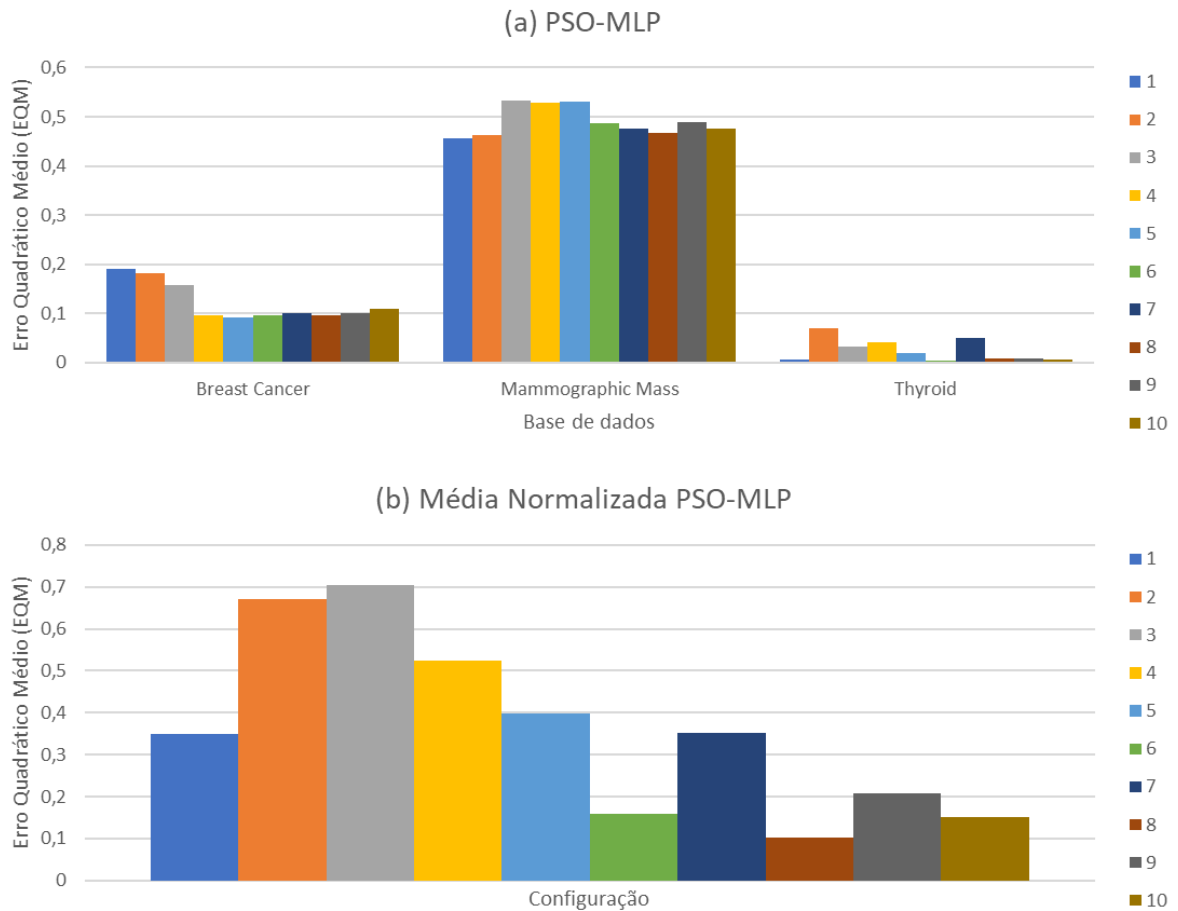
Figura 29 – Análise comparativa das configurações – ABC-MLP.



Fonte: Autoria própria.

Ao observar a Figura 29(a), verifica-se que para a base de dados “*Mammographic Mass*” a configuração que apresentou o melhor desempenho foi a de número 5, já para as bases “*Thyroid*” e “*Breast Cancer*” o melhor resultado encontrado foi utilizando a configuração 4. No entanto, é possível observar através da Figura 29(b) que a configuração que apresentou o melhor desempenho médio para todas as bases de dados utilizadas, foi a de número 3.

Figura 30 – Análise comparativa das configurações – PSO-MLP.

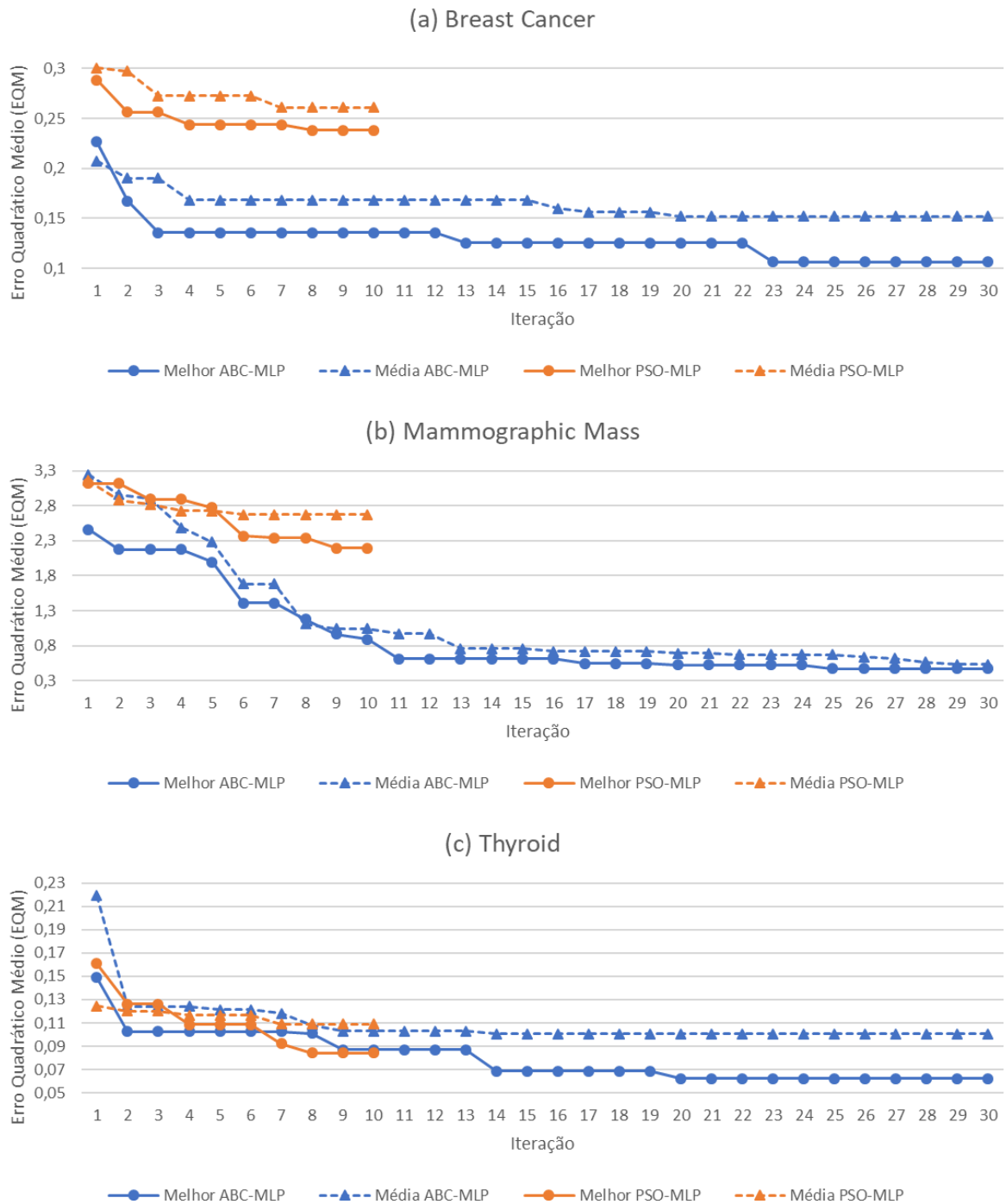


Fonte: Autoria própria.

Ao observar a Figura 30(a), verifica-se que para a base de dados “*Mammographic Mass*” a configuração que apresentou o melhor desempenho foi a de número 1, para a base “*Thyroid*” o melhor resultado encontrado foi utilizando a configuração 6, já para a base de dados “*Breast Cancer*”, a configuração 5 foi a que apresentou o melhor desempenho. No entanto, é possível observar através da Figura 30(b) que a configuração que apresentou o melhor desempenho médio para todas as bases de dados utilizadas, foi a de número 8.

Com base na análise anterior, foi selecionada as configurações 3 e 8 para os sistemas ABC-MLP e PSO-MLP, respectivamente. Por fim, tendo em mãos as configurações com os melhores desempenhos médios, foi realizada a análise comparativa de desempenho entre os sistemas, onde é possível verificar na Figura 31 a evolução média das 10 execuções, bem como a resposta evolutiva da melhor execução.

Figura 31 – Análise comparativa de desempenho evolutivo – Melhores configurações.



Fonte: Autoria própria.

Como é possível verificar em todos os gráficos comparativos, devido a inicialização aleatória das variáveis a serem otimizadas, o *fitness* se inicializa em pontos diferentes para todos os casos. Entretanto, constata-se que para as configurações analisadas, mesmo o sistema ABC-MLP sendo inicializado em uma posição favorável para a base de dados “*Breast Cancer*” apresentada na Figura 31(a), o sistema ainda demonstra um melhor desempenho evolutivo, mesmo quando

inicializado em posições espaciais desfavoráveis como apresentado na Figura 31(b) e (c).

4.2 Resultados de teste

Tendo em vista os resultados adquiridos na etapa de validação, tornou-se possível realizar a etapa de testes. Nesta etapa, foi utilizada a configuração que apresentou o melhor desempenho médio em cada um dos sistemas, como apresentado na seção anterior, gerando um total de 12 mil redes treinadas.

Desta forma, a partir da otimização dos parâmetros das RNAs durante a etapa de teste, foi possível obter a resposta de cada execução para cada um dos sistemas. A partir desses resultados, foi realizada uma análise comparativa para verificar o desempenho dos sistemas em minimizar o *fitness*. Desta forma, os Quadros 4 e 5 apresentam os resultados referentes ao valor médio do *fitness*, erro e tempo para as 10 melhores redes de cada sistema, bem como o menor valor encontrado para o *fitness* e o erro, para as bases de dados “Classes de Movimentos” e “Níveis de Contração”, respectivamente.

Quadro 4 – Análise comparativa entre os sistemas – Classes de Movimentos.

Classes de Movimentos	Fitness (EQM)		Erro (%)		Tempo (horas)
	Média	Mínimo	Média	Mínimo	Média
ABC-MLP	$4,72 \times 10^{-6}$	$3,57 \times 10^{-9}$	0,07	0,00	27,42
PSO-MLP	$1,08 \times 10^{-6}$	$1,39 \times 10^{-8}$	0,08	0,00	109,48

Fonte: Autoria própria.

Quadro 5 – Análise comparativa entre os sistemas – Níveis de Contração.

Níveis de Contração	Fitness (EQM)		Erro (%)		Tempo (horas)
	Média	Mínimo	Média	Mínimo	Média
ABC-MLP	$2,25 \times 10^{-1}$	$1,97 \times 10^{-1}$	15,35	14,4	22,59
PSO-MLP	$2,05 \times 10^{-1}$	$1,93 \times 10^{-1}$	14,68	13,48	84,17

Fonte: Autoria própria.

Ao observar o Quadro 4, verifica-se que o sistema PSO-MLP apresentou um melhor resultado em relação média dos *fitness*. Entretanto, nos quesitos de melhor *fitness*, média dos erros e custo computacional, o sistema ABC-MLP teve um desempenho superior. Vale destacar que a base “Classes de Movimentos” possui uma complexidade de classificação relativamente simples, onde ambos sistemas atingiram uma taxa de acerto de 100%.

Analisando o Quadro 5, pode-se constatar que o sistema PSO-MLP apresentou um desempenho levemente superior em todos os quesitos, com a exceção do custo computacional que para ambas as bases se apresentou em torno de 4 vezes maior do que para o sistema ABC-MLP.

As arquiteturas encontradas que apresentaram o melhor *fitness* em cada um dos sistemas para as bases de dados utilizadas durante o teste, são apresentadas no Quadro 6.

Quadro 6 – Melhor arquitetura encontrada - *Fitness*.

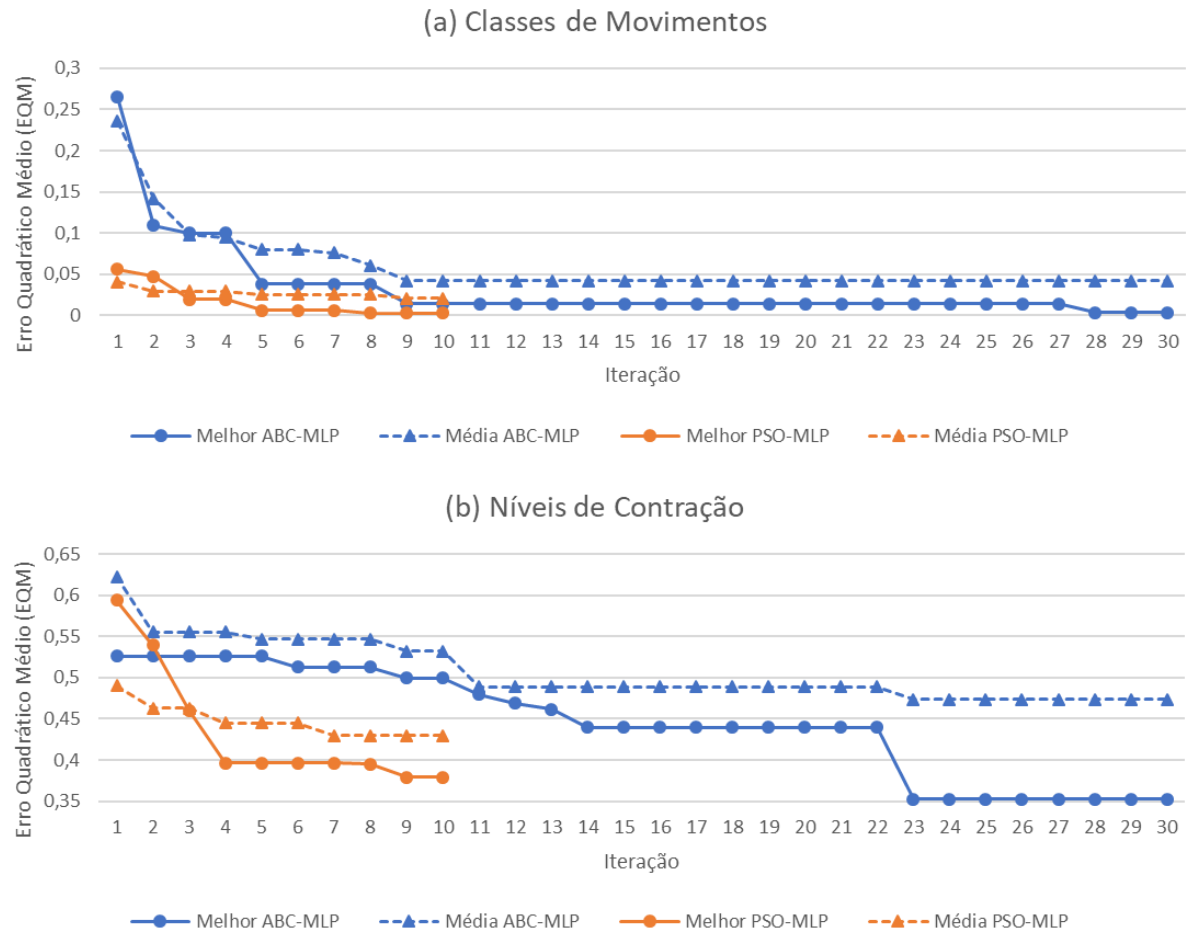
Bases de dados	ABC-MLP	PSO-MLP
Classes de Movimentos	[33 8 13 0,4271]	[50 5 6 0,1518]
Níveis de Contração	[58 51 4 0,0064]	[49 59 3 0,3745]

Fonte: Autoria própria.

Para as bases de dados “Classes de Movimentos” e “Níveis de Contração”, ambos os sistemas apresentaram resultados aproximados em relação a complexidade das redes, entretanto, o sistema ABC-MLP demonstrou um melhor resultado para a primeira base, enquanto o sistema PSO-MLP teve vantagem com a segunda.

Com o intuito de verificar o desempenho evolutivo dos sistemas diante dos problemas abordados, foi realizada uma análise comparativa de desempenho entre os mesmos, onde é possível verificar na Figura 32 a evolução média das 10 execuções, bem como a resposta evolutiva da melhor execução para cada uma das bases de dados.

Figura 32 - Análise comparativa de desempenho evolutivo.



Fonte: Autoria própria.

Como é possível verificar em todos os gráficos comparativos, devido a inicialização aleatória das variáveis a serem otimizadas, o *fitness* se inicializa em pontos diferentes para todos os casos. Entretanto, o sistema PSO-MLP teve uma inicialização favorável em relação ao ABC-MLP em ambos os casos, onde, na Figura 32(a) nota-se que o valor inicial médio do sistema PSO-MLP encontra-se inferior ao valor médio final por iteração do sistema ABC-MLP.

Desta forma, analisando os gráficos apresentados na Figura 32, observa-se que o sistema PSO-MLP demonstra um melhor desempenho evolutivo médio em ambos os casos, no entanto, ao analisar o desempenho da melhor execução dos sistemas, pode-se averiguar que o sistema ABC-MLP consegue atingir resultados que são até mesmo superiores que o do sistema PSO-MLP, mesmo inicializando em pontos desfavoráveis.

Com o intuito de realizar uma análise mais aprofundada a respeito do desempenho dos sistemas, foi verificado a eficiência de classificação das respostas

obtidas para os problemas em cada um dos sistemas, a partir das matrizes de confusão, que nos permitem averiguar tanto os verdadeiros positivos quanto os verdadeiros negativos, além dos valores classificados como incertos, como apresentadas nos Quadros 7, 8, 9 e 10.

Quadro 7 – Matriz de confusão ABC-MLP – Classes de Movimentos.

Classes de Movimentos	ABC-MLP - Classificação (%)							
	Classes	1	2	3	4	5	6	Incerteza
Desejado	1	100	0	0	0	0	0	0
	2	0	100	0	0	0	0	0
	3	0	0	100	0	0	0	0
	4	0	0	0	100	0	0	0
	5	0	0	0	0	100	0	0
	6	0	0	0	0	0	100	0
Total de acertos		100						

Fonte: Autoria própria.

Quadro 8 – Matriz de confusão PSO-MLP – Classes de Movimentos.

Classes de Movimentos	PSO-MLP - Classificação (%)							
	Classes	1	2	3	4	5	6	Incerteza
Desejado	1	100	0	0	0	0	0	0
	2	0	100	0	0	0	0	0
	3	0	0	100	0	0	0	0
	4	0	0	0	100	0	0	0
	5	0	0	0	0	100	0	0
	6	0	0	0	0	0	100	0
Total de acertos		100						

Fonte: Autoria própria.

Para a base de dados “Classes de Movimentos”, ambos os sistemas apresentaram como resposta soluções com taxa de acerto de 100%, fato que pode ser confirmado através da diagonal principal das matrizes de confusão apresentadas nos Quadros 7 e 8.

Quadro 9 – Matriz de confusão ABC-MLP – Níveis de Contração.

Níveis de Contração	ABC-MLP - Classificação (%)				
	Classes	1	2	3	Incerteza
Desejado	1	90,74	3,09	0,31	5,86
	2	0,93	82,41	6,17	10,49
	3	0,62	6,17	83,64	9,57
Total de acertos		85,60			

Fonte: Autoria própria.

Quadro 10 – Matriz de confusão PSO-MLP – Níveis de Contração.

Níveis de Contração	PSO-MLP - Classificação (%)				
	Classes	1	2	3	Incerteza
Desejado	1	92,59	1,54	0,62	5,25
	2	1,54	79,63	5,56	13,27
	3	0,62	2,78	87,35	9,26
Total de acertos		86,52			

Fonte: Autoria própria.

Levando em conta a base de dados “Níveis de Contração” nota-se que ambos os sistemas obtiveram taxa de acerto total superior a 85%, onde o PSO-MLP teve um desempenho levemente superior. Um importante detalhe que pôde ser validado a partir das matrizes de confusão foi que a segunda classe maior dificuldade de classificação por se tratar de um níveis intermediário, porém apesar da mesma apresentar maior dificuldade de classificação, a robustez das redes encontradas pode ser confirmada tendo em vista as respostas apontam com maior frequência para a taxa de incerteza e não como um verdadeiro negativo. Analisando os extremos é constatado que por se tratar de níveis com diferenças discrepantes, as redes possuem menor tendência de apontar como verdadeiro negativo a extremidade oposta.

As arquiteturas encontradas que apresentaram o menor Erro em cada um dos sistemas para as bases de dados utilizadas durante o teste, são apresentadas no Quadro 11.

Quadro 11 – Melhor arquitetura encontrada - Erro.

Bases de dados	ABC-MLP	PSO-MLP
Classes de Movimentos	[33 8 13 0,4271]	[50 5 6 0,1518]
Níveis de Contração	[58 33 4 0,2068]	[61 53 9 0,3371]

Fonte: Autoria própria.

Analisando a complexidades das soluções encontradas para as arquiteturas que apresentaram os menores erros com as bases de dados “Classes de Movimentos” e “Níveis de Contração”, é possível identificar uma larga vantagem para as soluções apresentadas pelo sistema ABC-MLP.

5 ANÁLISE E DISCUSSÃO

Este capítulo é dedicado à apresentação das análises dos resultados obtidos a partir da otimização da arquitetura do MLP e das discussões que cercam este trabalho, tendo como base a literatura correlata.

5.1 Análise dos resultados

A análise dos sistemas foi realizada em duas etapas, sendo a primeira responsável pela validação dos sistemas utilizando as bases de dados de diagnósticos médicos “*Breast Cancer*”, “*Mammographic Mass*” e “*Thyroid*”, enquanto a segunda é voltada para a verificação do desempenho dos sistemas na solução de problemas específicos, utilizando as bases de dados de sinais EMG “*Classes de Movimentos*” e “*Níveis de Contração*”. Desta forma, foi possível realizar o treinamento de 192 mil redes para verificar o desempenho dos sistemas.

Como as bases de dados utilizadas na etapa de validação apresentam características distintas de número de amostras, quantidade de atributos e níveis de complexidade, era esperado observar comportamentos diferentes para ambos os sistemas em cada configuração. Ao observar os resultados da análise comparativa dos sistemas, é possível corroborar resultados diferentes para cada configuração, sendo que em duas bases de dados o sistema ABC-MLP se mostrou superior, tanto para a análise do valor médio quanto para a análise do valor mínimo. Com esses resultados também foi possível destacar a importância da escolha de uma configuração adequada afim de maximizar a eficiência dos sistemas.

Tendo em vista o espaço de busca aplicado nas problemáticas e as melhores arquiteturas encontradas para cada base de dados em cada sistema, fica visível a efetividade da otimização de ambos sistemas, onde para bases simples como “*Mammographic Mass*” as arquiteturas encontradas apresentam poucos neurônios, entre 1 e 3.

Considerando a comparação entre os sistemas realizada na etapa de teste verificou-se uma competição mais acirrada entre eles. Nas análises de *fitness* e erro, ambos obtiveram valores relativamente próximos para as duas bases de dados analisadas, sendo o resultado mais distante o *fitness* mínimo, obtido pela base de dados “*Classes de Movimentos*”.

Levando em consideração as bases de dados da etapa de teste e seu objetivo, classificar seis classes de movimentos da mão e os três diferentes níveis de contração em que eles foram realizados, é verificado através da análise das matrizes de confusão que tal objetivo foi atingido com taxa de 100% de acerto para a classificação dos movimentos e de 86,52% para os níveis de contração. Vale destacar que apesar dos níveis de contração apresentarem uma taxa de erro de 13,48%, desse total apenas 4,22% são verdadeiros negativos e 9,26% é classificado como incerteza.

Outra análise importante presente na etapa de validação e teste, é a análise comparativa de desempenho evolutivo, onde é possível verificar o comportamento dos sistemas de otimização no processo iterativo nas diversas bases de dados. Neste processo é possível identificar que mesmo com a inicialização aleatória favorecendo um sistema ou outro nas bases de dados, foi possível analisar o desempenho dos sistemas devido as diversas execuções.

Realizando uma análise específica em relação ao custo computacional dos sistemas, através do tempo de execução de ambos sistemas rodando uma quantidade fixa de redes é visível a superioridade do ABC-MLP, onde suas execuções se demonstraram cerca de 4 vezes mais rápidas que o sistema PSO-MLP.

5.2 Discussão

Tendo em vista que este trabalho realiza o emprego de algumas abordagens similares as utilizadas nos trabalhos relacionados, fazendo o uso de bases de dados utilizadas nos mesmos para validar os sistemas, pode-se comparar os resultados obtidos tendo como base os resultados apresentados na literatura correlata.

Utilizando diferentes versões do algoritmo bioinspirado PSO para realizar a otimização da arquitetura e dos pesos sinápticos de RNAs do tipo MLP, Carvalho (2007) obteve como melhor média do percentual de erro de classificação da base de dados "*Breast Cancer*" um valor igual a 4,14%, utilizando o sistema PSO-PSO:WD. Para os sistemas propostos ABC-MLP e PSO-MLP, a melhor média do percentual de erro de classificação obtida com a mesma base de dados, entre as 10 configurações testadas foi de 1,51% e 3,66%, respectivamente. Tal resultado pode estar relacionado ao fato de que, apesar de também realizar a otimização dos pesos sinápticos, Carvalho (2007) otimiza apenas a quantidade de neurônios de uma única camada

escondida, enquanto neste presente trabalho são otimizadas 3 camadas escondidas e a taxa de aprendizado, o que confirma a influência das variáveis selecionadas no desempenho da rede.

Em Çam *et al.* (2015), onde são desenvolvidos três diferentes sistemas de otimização utilizando ABC, Algoritmo Genético e PSO, a melhor taxa de acerto encontrada para as bases de dados “*Thyroid*” e “*Mammographic Mass*” foi de 96,26% e 88,43%, respectivamente. Confrontando os resultados apresentados por Çam *et al.* (2015) com os obtidos neste trabalho, verifica-se que para a base de dados “*Thyroid*” ambos os sistemas desenvolvidos, ABC-MLP e PSO-MLP, apresentam uma taxa de acerto superior, 98,46% e 100%, respectivamente. Em contrapartida, para a base de dados “*Mammographic Mass*”, as taxas de acertos obtidas foram de 74,30% e 73,90% para os sistemas ABC-MLP e PSO-MLP. Entretanto, os sistemas propostos neste trabalho utilizam um rigoroso critério de confiabilidade, como apresentado na Tabela 1, o que pode resultar em uma taxa de classificação significativamente menor.

Entre os trabalhos relacionados, Shahraki e Zahiri (2017) apresentam um desenho metodológico muito próximo com o proposto nesta pesquisa. O sistema de otimização utilizando algoritmo *Inclined Planes Optimization* apresentou os melhores resultados entre os propostos por Shahraki e Zahiri (2017), onde, para a base de dados “*Breast Cancer*” a taxa de acerto foi de 99,04% com EQM de 0,0127. Para o presente trabalho, a melhor taxa de acerto encontrada para o sistema ABC-MLP foi de 99,51% com EQM de 0,0201, enquanto para o PSO-MLP foi de 97,56% com o valor de 0,0473 para o EQM. Desta forma, verifica-se a importância da otimização de novas variáveis que influenciam no desempenho da rede, como por exemplo, as funções de ativação.

Outro fator apresentado em Shahraki e Zahiri (2017) e constatado no desenvolvimento deste trabalho, é a eficiência do uso do EQM como *fitness* para garantir uma rede com baixa complexidade e boa generalização. Tal fato pode ser observado a partir dos resultados utilizando a base de dados “*Classes de Movimentos*”, onde a rede com menor complexidade, dentre todas as que tiveram taxa de acerto de 100%, também é a que apresenta o menor EQM.

6 CONCLUSÕES E TRABALHOS FUTUROS

Este capítulo tem como objetivo apresentar uma síntese dos principais resultados obtidos, bem como algumas considerações a respeito das perspectivas futuras de pesquisas relacionadas a este trabalho.

6.1 Conclusões

Como pode-se observar, as RNAs têm se apresentado como poderosas ferramentas para solução de uma ampla gama de problemas. Contudo, a determinação de valores adequados para os parâmetros relacionados a topologia de sua arquitetura, influencia diretamente no seu desempenho. Esta tarefa é comumente realizada através do método empírico, sendo um processo tedioso e exaustivo. Desta maneira, constata-se a necessidade de auxiliar os projetistas responsáveis pelo desenvolvimento de RNAs no ajuste desses valores.

Sendo assim, neste trabalho foi realizado a implementação do MLP juntamente com os algoritmos bioinspirados PSO e ABC. A evolução da arquitetura do MLP foi realizada empregando a combinação dos sistemas inteligentes, obtendo-se assim os sistemas inteligentes híbridos PSO-MLP e ABC-MLP. Os algoritmos bioinspirados PSO e ABC utilizam como *fitness* o EQM gerado pela rede durante a fase de teste, com o objetivo de otimizar a arquitetura e melhorar o desempenho das RNAs na tarefa de classificação de padrões.

Ambos os sistemas foram validados utilizando três diferentes bases de dados de diagnóstico médico do repositório de aprendizado de máquina, UCI, onde foi possível comparar o desempenho dos sistemas e encontrar qual configuração apresentou o melhor desempenho médio em cada um deles.

Para verificar o comportamento dos sistemas na solução de problemas específicos, foi utilizado duas bases de dados de sinais EMG tratados utilizando a TWP da família Haar, onde constatou-se a eficiência dos sistemas desenvolvidos na classificação de movimentos da mão e dos níveis de contração muscular.

Portanto, ao comparar os resultados obtidos para os dois sistemas, ABC-MLP e PSO-MLP, verificou-se que neste contexto, não é possível afirmar a preeminência de um sistema diante do outro, visto que ambos sistemas apresentaram superioridade em bases de dados distintas. Assim, pode-se concluir que o

desempenho dos mesmos está diretamente relacionado às características do problema abordado.

Assim sendo, as principais contribuições deste trabalho podem ser sintetizadas tais como:

1. Ferramenta capaz de auxiliar e facilitar a tarefa de profissionais responsáveis pelo projeto de RNAs;
2. Aumento da robustez no processo de determinação dos parâmetros relacionados a arquiteturas das RNAs, visto que o mesmo torna-se otimizado e não mais empírico;
3. Melhoria no desempenho das RNAs através da otimização de sua topologia utilizando os algoritmos bioinspirados;
4. Sistemas capazes de se adaptar a qualquer classe de problemas envolvendo a otimização da topologia de RNAs do tipo MLP.

6.2 Perspectivas futuras

Por fim, a partir dos resultados obtidos neste trabalho e das análises realizadas, pode-se apresentar algumas perspectivas de trabalhos futuros:

1. Inserção de novos parâmetros da arquitetura que apresentam grande influência no desempenho das redes MLP a serem otimizados;
2. Utilização de versões aprimoradas dos algoritmos bioinspirados de Otimização por Enxame de Partículas e Colônia Artificial de Abelhas;
3. Análise de novas abordagens para a função *fitness*, utilizando combinações de variáveis e até mesmo funções multi-objetivo, com o intuito de obter uma representação mais fiel para os resultados obtidos pela RNA.

7 TRABALHOS REALIZADOS

7.1 Publicados

1. Treinamento Otimizado de Redes Neurais Artificiais Para a Classificação de Movimentos Musculares. Encontro Nacional de Inteligência Artificial e Computacional, Uberlândia, MG, 2017.
2. Reconhecimento de padrões de movimentos da mão a partir de sinais mioelétricos do antebraço utilizando redes neurais artificiais e algoritmos genéticos. V Congresso Brasileiro de Eletromiografia e Cinesiologia e X Simpósio de Engenharia Biomédica. Uberlândia, MG, 2018.
3. Classification of Hand Movements from EMG Signals using Optimized MLP. International Joint Conference in Neural Network- IJCNN. Rio de Janeiro, RJ, 2018.
4. Reconhecimento de padrões de movimentos da mão a partir de sinais mioelétricos do antebraço utilizando redes neurais artificiais e algoritmos genéticos. Em: Bases da Saúde e Engenharia Biomédica v. 2. cap. 15, p. 92-99. Editora Atena. Belo Horizonte, MG, 2018.
5. Identificação de disfonias Utilizando Redes Neurais Artificiais. Seminário de Iniciação Científica e Tecnológica da UTFPR. Apucarana, PR, 2018.
6. Classificação de Movimentos da Mão Utilizando Redes Neurais Artificiais. Seminário de Iniciação Científica e Tecnológica da UTFPR. Apucarana, PR, 2018.
7. Processamento Digital de Sinais Biológicos Mediante Transformadas Wavelets para Aplicação em Redes Neurais Artificiais. Seminário de Iniciação Científica e Tecnológica da UTFPR. Apucarana, PR, 2018.
8. Sistema de Otimização de Parâmetros Internos de Redes Neurais Artificiais. Seminário de Iniciação Científica e Tecnológica da UTFPR. Apucarana, PR, 2018.

7.2 Aceitos para publicação

1. Optimized Artificial Neural Network for EMG Classification Using Genetic Algorithm. Journal of Control, Automation and Electrical Systems, 2018.

REFERÊNCIAS

- AL-TIMEMY, A. H. *et al.* **Improving the performance against force variation of emg controlled multifunctional upper-limb prostheses for transradial amputees.** IEEE Transactions on Neural Systems and Rehabilitation Engineering, 2015.
- ANDRADE, L. A. C. G. **Heurística baseada em colônia artificial de abelhas para o problema de localização de bases, alocação e realocação de ambulâncias.** Tese de Doutorado. Universidade de São Paulo, 2012.
- BARROS F. K. H. *et al.* **Treinamento Otimizado de Redes Neurais Artificiais Para a Classificação de Movimentos Musculares.** Brazilian Conference on Intelligent Systems, Uberlandia, Brazil, 2017.
- BELOV, M. P.; ZOLOTOV, O. I. **Optimization of parameters of neural networks by genetic algorithm in the control systems of electromechanical objects.** XVIII International Conference on Soft Computing and Measurements (SCM), 2015.
- BHARGAB, C.; SANGITA N. **Particle Swarm Optimization Algorithm for Integer Factorization Problem (IFP).** International Journal of Computer Applications 117(13):14-17, 2015.
- BINOTI, D. H. B.; DA SILVA BINOTI, M. L. M.; LEITE, H. G. **Configuração de redes neurais artificiais para estimação do volume de árvores.** Revista Ciência da Madeira (Brazilian Journal of Wood Science), v. 5, n. 1, p. 10-12953, 2014.
- BISHOP, C. M. **Neural networks for pattern recognition.** Oxford, UK: Oxford university press, 1995.
- ÇAM, Z. G.; ÇIMEN, S.; YILDIRIM, T. **Learning parameter optimization of Multi-Layer Perceptron using Artificial Bee Colony, Genetic Algorithm and Particle Swarm Optimization.** In: Applied Machine Intelligence and Informatics (SAMII), IEEE 13th International Symposium on. IEEE, p. 329-332, 2015.
- CARVALHO, A. R.; RAMOS, F. M.; CHAVES, A. A. **Metaheuristics for the feedforward artificial neural network (ANN) architecture optimization problem.** Neural Computing and Applications, v. 20, n. 8, p. 1273-1284, 2010.
- CARVALHO, M. R. **Uma análise de otimização de redes neurais MLP por enxames de partículas.** Dissertação (Mestrado). Programa de Pós-Graduação em Ciência da Computação, Universidade Federal de Pernambuco, Recife, 2007.
- COIFMAN, R. R.; WICKERHAUSER, M. V. **Entropy-based Algorithms for Best Basis Selection.** IEE Transaction on Information Theory, v. 38, n. 2, 1992.
- DA SILVA, A. F. T. *et al.* **Classificação de sinais de voz através da aplicação da transformada Wavelet Packet e redes neurais artificiais.** Revista Principia - Divulgação Científica e Tecnológica do IFPB. 1. p. 34-41, 2017.
- DE LUCA, C. J. **Physiology and Mathematics of Myoelectric on Biomedical Engineering,** v. BME-26, no. 6, p. 313-325. Signals. IEEE Transactions, 1979.

DOLEZEL, P.; SKRABANEK, P.; GAGO, L. **Pattern recognition neural network as a tool for pest birds detection**. IEEE Symposium Series on Computational Intelligence (SSCI), Athens, p. 1-6, 2016.

DOS SANTOS, F. A. G. *et al.* **Processamento Digital de Sinais Biológicos Mediante Transformadas *Wavelets* para Aplicação em Redes Neurais Artificiais**. Seminário de Iniciação Científica e Tecnológica da UTFPR. Apucarana-PR, 2018.

EBERHART, R. C.; SHI, Y. **Comparing inertia weights and constriction factors in particle swarm optimization**. Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No.00TH8512), La Jolla, CA, p. 84-88 v.1, 2000.

ENOKA, R. M. **Bases neuromecânicas da cinesiologia**. São Paulo: Manole. 450 p, 2000.

GAO, W.; LIU, S. **Improved artificial bee colony algorithm for global optimization**. Information Processing Letters, v.111, p. 871-882, 2011.

GOLDSCHMIDT, R.; PASSOS, E.; BEZERRA, E. **Data Mining: Conceitos, técnicas, algoritmos, orientações e aplicações**. [S.I.]: Elsevier, 2015.

HAFIZ, F.; ABERCROMBIE, S.; EATON, A. **Power quality event identification using wavelet packet transform: A comprehensive investigation**. Region 10 Conference, TENCON 2017 - IEEE, p. 1-6, 2017.

HAGAN, M. T. *et al.* **Neural Network Design. Second Edition**. Boston, MA: Martin Hagan, 2014.

HAYKIN, S. **Redes neurais: princípios e prática**. Bookman Editora, 2007.

JIAO, S.; SHI, W.; LIU, Q. **Self-adaptative partial discharge denoising based on variation mode decomposition and wavelet packet transform**. Chinese Automation Congress (CAC), p. 7301-7306, 2017.

KARABOGA, D.; BASTURK, B. **Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems**. In: International fuzzy systems association world congress. Springer, Berlin, Heidelberg, p. 789-798, 2007.

KARABOGA, D. **An idea based on honey bee swarm for numerical optimization**. Technical report-tr06, Erciyes university, engineering faculty, computer engineering department, 2005.

KASHAN, M. H.; NAHAVANDI, N.; HUSSEINZADEH KASHAN, A. **DisABC: A new artificial bee colony algorithm for binary optimization**. Applied Soft Computing. v. 12, 2012.

KENNEDY, J.; EBERHART, R. **A discrete binary version of the particle swarm algorithm**. IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation, Orlando, FL, v. 5, p. 4104-4108, 1997.

KENNEDY, J.; EBERHART, R. **Particle swarm Optimization**. Neural Networks. Proceedings., IEEE International Conference on, Perth, WA, v. 4, p. 1942-1948, 1995.

KHALIL, T. M.; GORPINICH, A. V. **Optimal conductor selection and capacitor placement for loss reduction of radial distribution systems by selective particle swarm Optimization**. Seventh International Conference on Computer Engineering & Systems (ICCES), Cairo, p. 215-220., 2012.

KHUSHABA, R. N. *et al.* **Combined influence of forearm orientation and muscular contraction on emg pattern recognition**. Expert Systems with Applications, 2016.

KONRAD, P. **The ABC of EMG**. version 1.0. Noraxon INC. USA, 2005.

KRIESEL, D. **A Brief Introduction to Neural Networks**. 2007. Disponível em <http://www.dkriesel.com/_media/science/neuronalenetze-en-zeta2-2col-dkrieselcom.pdf> Acesso em: 19 mar. 2018.

LI, S.; MENG, Z. **Optimal PMU placement based on improved binary artificial bee colony algorithm**. In: Transportation Electrification Asia-Pacific (ITEC Asia-Pacific), IEEE Conference and Exp. IEEE, p. 1-6, 2017.

LIMA, A. A. M. *et al.* **Reconhecimento de Padrões de Movimentos da Mão a Partir de Sinais Mielétricos do Antebraço Utilizando Redes Neurais Artificiais e Algoritmos Genéticos**. V Congresso Brasileiro de Eletromiografia e Cinesiologia /X Simpósio de Engenharia Biomédica. Uberlândia, Brasil, 2017.

MALLAT, S. G. “**A theory of multiresolution signal decomposition: The wavelet representation**”. IEEE Trans. Pat. Anal Mach. Intell., 11:674--693, 1989.

MALLAT, S. **A Wavelet Tour of Signal Processing**. Academic Press, 3 edition, 2008.

MATHWORKS. **Pattern Recognition**. 2017. Disponível em: <<https://www.mathworks.com/discovery/pattern-recognition.html>>. Acesso em: 28 mar. 2018.

MARCHETTI, P. H.; DUARTE, M. **Instrumentação em Eletromiografia**. Lab de Biofís. EEFÉ-USP, 2006.

MCCULLOCH, W. S.; PITTS, W. **A logical calculus of neural analysis in nervous activity**. **Bulletin of Mathematical Biophysics**. v. 5, p. 115-133, 1943.

MEDEIROS, J. A. C. C. **Enxame de partículas como ferramenta de otimização em problemas complexos de engenharia nuclear**. Universidade Federal do Rio de Janeiro, 2005.

MERLETTI, R.; FARINA, D.; GRANATA, A. **Non-invasive assesement of motor unit properties with linear electrode arrays**. **Electroenceph Clinical Neurophysiology**, suppl 50, p. 293-300, 1999.

OLIVEIRA, H. M. **Análise de Fourier e Wavelets: Sinais Estacionários e não Estacionários**. Recife: EDUFPE, 2007.

OZTURK, C.; HANCER, E.; KARABOGA, D. **A novel binary artificial bee colony algorithm based on genetic operators**. Information Sciences, v. 297, 2015.

PAMPARÁ, G.; ENGELBRECHT, A. P. **Binary artificial bee colony optimization**. In: Swarm Intelligence (SIS), IEEE Symposium on. IEEE, p. 1-8, 2011.

PRARRAGA, A. **Aplicação da transformada Wavelet Packet na Análise e Classificação de Sinais de Vozes Patológicas**. Dissertação. Universidade Federal do Rio Grande do Sul, Porto Alegre, 163p, 2002.

RAJAGURU, H.; PRABHAKAR, S. K. **Hilbert transform with Elman backpropagation and multilayer perceptrons for epilepsy classification**. International conference of Electronics, Communication and Aerospace Technology (ICECA), COIMBATORE, India, p. 571-576, 2017.

RAMCHOUN, H. *et al.* **Multilayer perceptron: Architecture optimization and training**. International Journal of Interactive Multimedia and Artificial Intelligence, 2016.

REYES-SIERRA, M.; COELLO, C. A. **Multi-objective particle swarm optimizers: A survey of the state-of-the-art**. International Journal of Computational Intelligence Research, 2006.

REZENDE, S. O. **Sistemas inteligentes: fundamentos e aplicações**. Editora Manole Ltda, 2003.

ROBERTSON, D. G. E. *et al.* **Research methods in biomechanics**. United States: Human Kinetics. 2004.

ROSENDO, M. **Um Algoritmo de Otimização por Nuvem de Partículas para Resolução de Problemas Combinatórios**. Dissertação (Mestrado) – Universidade Federal do Paraná, Curitiba, 2010.

SEELEY, T. D. **The wisdom of the hive: the social physiology of honey bee colonies**. Harvard University Press, 2009.

SHAHRAKI, N. S.; ZAHIRI, S. H. **Inclined planes optimization algorithm in optimal architecture of MLP neural networks**. In: Pattern Recognition and Image Analysis (IPRIA), 3rd International Conference on. IEEE, p. 189-194, 2017.

SHI, Y.; EBERHART, R. C. **A modified particle swarm optimizer**. IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360), Anchorage, AK, p. 69-73, 1998.

SILVA, I. N.; SPATTI, D. H.; FLAUZINO, R. A. **Redes Neurais Artificiais: para engenharia e ciências aplicadas**. São Paulo: Artliber, 399p, 2010.

SILVA, L. P. M. **Segmentos proporcionais**. 2017. Disponível em: <<http://brasilecola.uol.com.br/matematica/segmentos-proporcionais.htm>>. Acesso em: 26 ago. 2018.

TAKAHASHI, L. S. O. **Análise de relação entre eletromiografia e força do músculo quadríceps em exercicios resistidos**. Dissertação (Mestrado) – Programa de Pós-graduação Interrunidades em Bioengenharia (EESC/FMRP/IQSC), Univerdade de São Paulo, São Carlos, 2006.

TALBI, E. G. **Metaheuristics: from design to implementation**. John Wiley & Sons, 2009.

TERESHKO, V.; LOENGAROV, A. **Collective Decision-Making in Honey Bee Foraging Dynamics**. Computing and Information Systems Journal, ISSN 1352-9404, v. 9, no. 3, October 2005.

TORACIO, A. A. P. G. **Aprendizado de regras de classificação com otimização por nuvem de particulas multiobjectivo**. Dissertação (mestrado) - Universidade Federal do Paraná, Setor de Ciencias Exatas, Programa de Pós-Graduação em Informática. Defesa: Curitiba, 2008.

VAN DEN BERGH, F. **An Analysis of Particle Swarm Optimizers**. PhD Computer thesis, University of Pretoria, 2002.

VESTERSTRØM, J. S.; RIGET, J. **Particle Swarms: Extensions for improved local, multi-modal, dynamic search in numerical optimization**. Dissertação (Mestrado) — Faculty of Science, Aarhus Universitet, p. 203, 2002.

WILAMOWSKI, B. M. **Neural network architectures and learning algorithms**. Industrial Electronics Magazine, IEEE, v. 3, issue 4, p. 56-63, 2009.

YI, Y.; ZHENG, W. X.; GUO, L. **Improved Results on Statistic Information Control with a Dynamic Neural Network Identifier**. IEEE Transactions on Circuits and Systems II: Express Briefs, v. 60, no. 11, p. 816-820, Nov. 2013.

YIN, Y.; GELENBE, E. **Single-cell based random neural network for deep learning**. International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, p. 86-93, 2017

YOSHIZUMI, V. H. *et al.* **Sistema de Otimização de Parâmetros Internos de Redes Neurais Artificiais**. Seminário de Iniciação Científica e Tecnológica da UTFPR. Apucarana-PR, 2018.

ZHANG, C.; SHAO, H. **An ANN's evolved by a new evolutionary system and its application**. In: **Decision and Control**. Proceedings of the 39th IEEE Conference on. IEEE, p. 3562-3563, 2000.

ZUBEN, F. J. V.; ATTUX, R. R. F. **Inteligência de Enxame**. DCA/FEEC/Unicamp e DECOM/FEEC/Unicamp, 2008.