PONTIFÍCIA UNIVERSIDADE CATÓLICA DO PARANÁ
ESCOLA POLITÉCNICA
MESTRADO EM ENGENHARIA DE PRODUÇÃO E SISTEMAS

ABORDAGENS DA COMPUTAÇÃO NATURAL APLICADAS À PREVISÃO DE SÉRIES TEMPORAIS

CURITIBA
2014

WESLLY PUCHALSKI

ABORDAGENS DA COMPUTAÇÃO NATURAL APLICADAS À PREVISÃO DE SÉRIES TEMPORAIS

Dissertação apresentada ao Programa de Mestrado em Engenharia de Produção e Sistemas, PPGEPS, da Pontifícia Universidade Católica do Paraná, PUCPR, como requisito parcial para obtenção do título de Mestre em Engenharia de Produção e Sistemas.

Orientador: Prof. Dr. Leandro dos Santos Coelho
Coorientador: Prof. Dr. Roberto Zanetti Freire

CURITIBA
2014

WESLLY PUCHALSKI

ABORDAGENS DA COMPUTAÇÃO NATURAL APLICADAS À PREVISÃO DE SÉRIES TEMPORAIS

Dissertação apresentada ao Programa de Pós-Graduação Stricto Senso em Engenharia de Produção e Sistemas da Pontifícia Universidade Católica do Paraná, como requisito à obtenção do título de Mestre em Engenharia de Produção e Sistemas.

COMISSÃO EXAMINADORA

_____
Prof. Dr. Leandro dos Santos Coelho (Orientador)
Pontifícia Universidade Católica do Paraná

_____
Prof. Dr. Roberto Zanetti Freire (Coorientador)
Pontifícia Universidade Católica do Paraná

_____
Prof. Dr. Maria Teresinha Arns Steiner (Membro Interno)
Pontifícia Universidade Católica do Paraná

_____
Prof. Dr. Gideon Villar Leandro (Membro externo)
Universidade Federal do Paraná (UFPR)

Cidade, ____ de _____ de 20__.

RESUMO

Obter um modelo consistente de previsão pode ser a chave para o sucesso em muitas áreas do conhecimento. Neste contexto, esta dissertação tem como objetivo validar a Rede Neural de Wavelets combinada com cincos técnicas de otimização para ajustar os parâmetros necessários da rede, sendo: Evolução Diferencial, Colônia Artificial de Abelhas, Glowworm, Busca Gravitacional e Imperialista Competitivo. O desempenho da rede é avaliado em dois estudos de caso: o preço da saca de soja e a demanda dos grupos de produtos A, B e C de uma empresa do ramo alimentício. Ambas as séries tem um comportamento não linear que dificulta a obtenção de um modelo de previsão apropriado e eficiente. As técnicas são avaliadas em três fases: validação, previsão de curto prazo com uma previsão à frente, e a previsão a longo prazo com previsão de 30 dias à frente para o preço da saca de soja e previsão de 12 meses para a demanda de uma empresa do ramo alimentício. Para a previsão de várias amostras à frente que, devido ao acumulo de erro tona difícil a previsão, a série temporal do preço da saca de soja obteve melhor desempenho com a combinação da Rede Neural de Wavelets com o método Imperialista Competitivo. Para a previsão de demanda dos grupos A e B a melhor combinação foi com método de Evolução Diferencial, e para a previsão do grupo C foi com o método de Colônia Artificial de Abelhas. Contudo, não é possível afirmar se estes métodos são os melhores para essas séries temporais, ainda exige novos testes e comparações empíricas para a escolha da melhor ferramenta para cada situação específica.

Palavras chaves: Previsão, Séries temporais, Redes Neurais, Computação Natural.

ABSTRACT

Get a consistent model of prediction can be the key to success in many areas of knowledge. In this context, this work aims to validate the Wavelets Neural Network combined with five optimization techniques to set required network parameters, being: Differential Evolution, Artificial Bee Colony, Glowworm, Gravitational Search and Imperialist Competitive. Network performance is evaluated in two case studies: the price of soybean sack and demand of a food company. Both series has a non-linear behavior difficult to obtain an appropriate and efficient prediction model. The techniques are evaluated in three stages: validation, short-term forecast and long-term forecast, where, soybean sack it was forecast 30 days ahead and for demand of Food Company it was forecast 12 months ahead. For the forecast of several samples ahead that is the forecast more difficult due to the accumulation of the error, the time series of the soybean sack price performed better with the combination of Wavelet Neural Network with the Imperialist Competitive method. To forecast demand of groups A and B, the best combination was with of Differential Evolution method, and for the forecast of group C was Artificial Bee Colony method. However, it is not possible to say if these methods are best for these time series, still requires further testing and empirical comparisons for choosing the best tool for each specific situation.

Key words: Forecasting, Time Series, Neural Networks, Natural Computing.

“Todos os modelos são errados, mais alguns são úteis”.

GEORGE BOX

AGRADECIMENTOS

Os agradecimentos desse trabalho vão para os meu pais, Élvio e Lisete, para minha irmã Keitry, namorada Marina B. Vida pela ajuda, compreensão e apoio no desenrolar do presente trabalho.

Outras pessoas que devo o agradecimento são aos professores e orientadores Leandro dos Santos Coelho e Roberto Zanetti Freire que estiveram sempre presente com ideias, correções e comentários, também ao Gabriel Fiori Neto que sempre ajudou no que pôde.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE QUADROS

LISTA DE ABREVIATURAS E SIGLAS

ABC: *Artificial Bee Colony Algorithm*

ADALINE: ADAptive LINear Elements

ADLA: Annealing dynamical learning algorithm

ANN*: Artificial Neural Network*

ARMA: Modelo Média Móvel Auto Regressivo Adaptativo

BFO: Bacterial Foraging Optimization

BFT: *Bacterial Foraging Technique*

CSO: *Cat Swarm Optimization*

DEWNN: Differential evolution trained wavelet neural network

EAP: Estrutura Analítica de Projeto

ED: Evolução Diferencial

FBLMS*: Forward Backward Least Mean Square*

FWNN: Fuzzy wavelet neural network

GSA: *Gravitational Search* Algorithm

ICA: *Imperialist Competitive Algorithm*

MADALIN*: Multiple ADALINE*

MAPE: *Mean Absolute Percentage Error*

MSE: *Mean Square Error*

PDP: *Parallel Distributed Processig*

PSO: *Particle Swarm Optimization*

PSOGSA: Hybrid particle swarm optimization e gravitational search algorithm

PUCPR: Pontifícia Universidade Católica do Paraná

RAM : *Random Access Memory*

RN: Rede Neural

SVR*: - Support Vector Regression*

TAWNN: Threshold accepting trained wavelet neural network

TF: Transformada de Fourier

TNEB*: Tamil Nadu Electricity Board*

TW: Transformada de Wavelets

1. INTRODUÇÃO

O entendimento de fenômenos e a busca por sistemas análogos que podem representar esses fenômenos sempre foram desafios para humanidade. Entende-se por sistema análogo um sistema que consiga reproduzir algumas características do fenômeno observado, se esse sistema é um sistema análogo matemático então diz-se modelo matemático do fenômeno (Aguirre *et al*., 1998).

A modelagem de sistemas está agrupada em três conjuntos, sendo: modelagem caixa branca: é necessário ter um conhecimento apropriado do processo, pois a base dessa modelagem é a utilização de leis da física para as equações; modelagem caixa preta: esta modelagem é necessário somente dados de entrada e saída do sistema, a técnica que é responsável por se adaptar e conseguir reproduzir o sistema sem que necessite conhecer os cálculos utilizados; e modelagem caixa cinza: é a mistura de ambas, sendo o conhecimento do sistema combinado com a identificação da caixa preta.

Em diversas áreas a utilização de modelos são imprescindíveis, tais como: analisar e projetar controladores; otimizar sistemas, detecção de falhas, compreender certas dinâmicas, prever o comportamento, entre outras aplicações (Matko *et al*., 1992).

Uma área de interesse deste trabalho é a previsão de séries temporais. A literatura contém exemplos de aplicações em diversas áreas, tais como: Física (Weiegend, 1994), Meteorologia (Ferraz, 1999), Engenharia Elétrica (Martin, 2005), Ciência da Computação (Leland, 1994) e Ciências Econômicas (Maddala, 2001). Conhecer previamente um valor futuro pode significar a diferença entre o sucesso e o fracasso em determinadas ocasiões, isto torna este estudo promissor.

As séries temporais são um conjunto de observações discretas, realizadas em períodos equidistantes e que apresentam uma dependência serial entre essas observações. Com essa definição, dois conjuntos de dados foram selecionados: o preço da saca de soja e a previsão de demanda de uma empresa (Silva et al., 2007).

Estes estudos de caso serão base para os testes de técnicas da computação natural. Deseja-se que técnicas da computação natural tenham a capacidade de aprender e se adaptar, conseguindo tratar diversos fatores que influenciam em um sistema; isto as tornam um paradigma promissor na previsão de séries temporais, sendo uma vantagem em relação aos métodos clássicos da literatura.

1.1. *Motivação*

Nesta dissertação adota-se dois estudos de caso, cada um com sua motivação. O estudo de caso refere-se ao preço da saca de soja; e a previsão de demanda de uma empresa do ramo alimentício. A soja é um das principais culturas brasileira no agronegócio que por sua vez é uma das atividades mais importantes do Brasil.

Empresas do ramo alimentício tens seus produtos com uma pequena vida útil, e um bom modelo adequado de previsão ajudaria, além de aumentar os lucros diminuir o desperdício de alimentos podendo até conseguir uma redução nos preços dos alimentos.

1.2. *Objetivo*

Essa dissertação tem como objetivo avaliar técnicas para realizar a previsão de séries temporais. Para isto, é utilizado a Rede Neural de Wavelets validada com cinco métodos de otimização para a realização do treinamento da rede: Algoritmo de Evolução Diferencial (ED); Algoritmo de Colônia Artificial de Abelhas (ABC); Algoritmo *Glowworm*; Algoritmo de Busca Gravitacional (GSA) e Algoritmo Imperialista Competitivo (ICA).

1.3. *Justificativa*

Obter um modelo consistente de previsão pode ser a chave para o sucesso em qualquer área. Há uma infinidade de áreas que o tema pode ser aplicado, e que uma previsão aceitável mudaria o planejamento e as decisões tomadas.

A previsão de séries temporais é um tema discutido há algum tempo e com diversos métodos aplicados em diversas áreas do conhecimento. No entanto ainda existem diversos testes e métodos a serem testados, requerendo ainda pesquisas nesta área.

Este trabalho tem a combinação da Rede Neural de Wavelets com cinco métodos de otimização para realizar o treinamento, sendo que, além de buscar uma ferramenta que consiga fornecer uma previsão. Além disso, esta dissertação também avaliará os método de otimização, verificando qual será o mais eficiente em relação ao MSE.

1.4. *Organização da Dissertação*

O Capitulo 1 apresentou a introdução, os objetivos e a justificativa. O restante desta dissertação está dividida nos seguintes capítulos.

O capítulo 2 introduz as redes neurais e seus conceitos, também apresenta as duas RNs utilizadas no trabalho que são: Rede Neural de Wavelets e a Rede Neural RBF.

O capítulo 3 apresenta os métodos de otimização que são utilizados para o treinamento das RNs. Cinco métodos foram escolhidos, que são: Algoritmo de Evolução Diferencial, Algoritmo de Colônia Artificial de Abelhas (ABC), Algoritmo *Glowworm*, Algoritmo de Busca Gravitacional e Algoritmo Imperialista Competitivo.

O capítulo 4 contém a explicação das séries temporais escolhidas, também os indicadores de desempenho que serão utilizados para a comparação dos métodos.

O capítulo 5 apresenta os resultados obtidos, primeiramente apresentado separado pelos métodos de treinamento após uma comparação dos melhores resultados para várias amostras à frente.

O capítulo 6 contém as conclusões da dissertação.

2. REDES NEURAIS ARTIFICIAIS

Apesar de ser um tema aparentemente recente, a ideia de utilização das RNs (Redes Neurais) para à resolução de problemas tem origem em 3000 a.C. com os trabalhos de Hipócrates. Em 1890, o pesquisador William James estudou as atividades do cérebro porém o primeiro pesquisador a inspirar-se no cérebro foi o pesquisador Alan Turing em 1936 (Nelson e Illingworth, 1991).

McCullogh e Pitts foram pioneiros nessa área também, sendo os primeiros a proporem um modelo de neurônio em 1943, com um número finito de entradas e uma saída, eles demonstraram que é possível implementar uma função lógica associando neurônios.

A primeira aplicações de RNs em problemas reais foi com os pesquisadores Bernard Widrow e Marcian Hoff, em 1959, com as redes denominadas *ADALINE* (*ADAptive LINear Elements*) e *MADALINE (Multiple ADALINE*), sendo utilizadas como filtros adaptativos para eliminar ruídos das linhas telefônicas (apud Grassi 2004).

Junto com o entusiasmo da época, as RNs sofreram uma crise quando os pesquisadores Minsky e Papert demostraram, em 1969, as deficiências da *perceptron* e provaram que não é possível resolver problemas não linearmente separáveis como a função de *ou exclusivo*. Com a publicação desse trabalho ocorreu o desencorajamento de pesquisadores e o redirecionamento dos fundos.

As RNs só voltaram a ter interesse após a surgimento do algoritmo de treinamento por retropropagação de erro, que foi apresentado por um grupo de pesquisadores denominado PDP (*Parallel Distributed Processing*) em 1986. Também em seus estudos resultaram na expansão da *perceptron* para várias camadas de neurônios, superando assim as dificuldades do método (Rumelhart *et al*., 1986). Após esse marco, houve uma explosão de aplicações com RNs. Esses e demais detalhes sobre as origens das RNs e suas evoluções são encontradas em Wasserman (1993), Haykin (1994) e Gupta e Rao (1994).

2.1. *Conceitos básicos*

As redes neurais são sistemas computacionais que podem ser implementados em *hardware* ou *software*, elas foram inspirada em uma pesquisa biológica do funcionamento do cérebro humano.

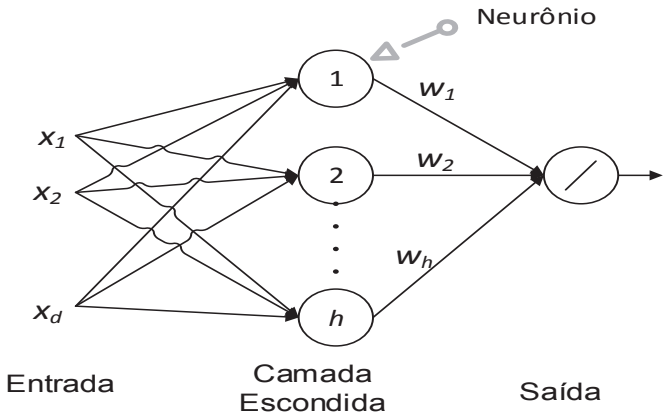
O cérebro humano tem aproximadamente 2,5 bilhões de simples processadores chamados de neurônios que são conectados entre si através de sinapses. (Lekutai,1997). Da mesma forma, as RNs são formadas por vários neurônios artificiais que são interligados sendo que toda a ligação contém um respectivo peso. Esses pesos são o conhecimento da RN e de forma simplificada, pode-se dizer que a RN aprendeu um comportamento após seus pesos serem ajustados (Muller e Reinhardt, 1990). Essa característica de aprendizado que torna as RNs uma ferramenta interessante para diversas áreas. No entanto, as RNs não são capazes de resolver os mesmos problemas que um cérebro humano (Welstead,1994).

Em seu trabalho de 1994, Haykin comenta algumas características que tornam as RNs atraentes, sendo elas:

- habilidade de tratar sistemas não-lineares que é importante para a classificação de padrões e aplicações em identificação de sistemas dinâmicos;
- tolerância à falhas, sendo um conjunto de conexões;
- adaptabilidade que dá à RN a capacidade de se auto ajustar;
- aprendizado, a RN consegue extrair informações e aprender como representar o sistema;
- generalização, ter a capacidade de mapear entradas similares e conseguir reproduzir as saídas, treinamento tem o objetivo de ensinar a rede, fazendo com que ela aprenda o comportamento do sistema, e abstração que faz a RN retirar a essência de um conjunto de entradas.

Em uma forma simples, a rede neural é representada conforme mostrado na Figura 2.1, sendo a entrada submetida a passar por vários neurônios conectados para obter a saída.

Figura 2.1 – Estrutura de uma rede neural.

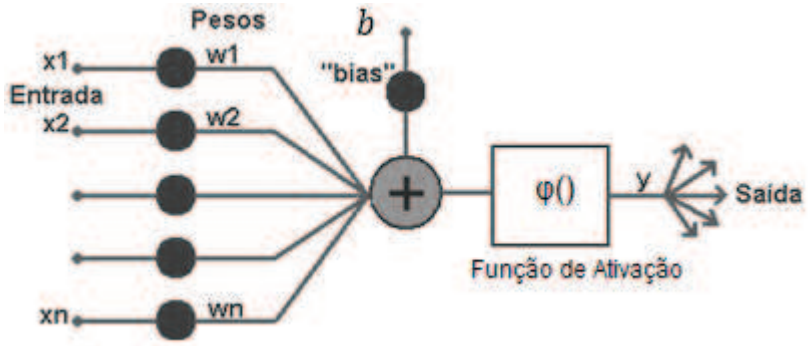


2.2. *Neurônios*

Os neurônios são o coração das RNs, eles que tem a capacidade de modificar as entradas a com os seus pesos e de sua função de ativação para fornecer as saídas.

McCulloch e Pitts(1943) estabeleceram o primeiro modelo de neurônios, um simples modelo onde considera-se várias entradas e uma saída conforme apresentado na Figura 2.2 onde *n* é o número de entradas do neurônio, $x_1, x_2, \dots, x_n$ são as entradas, $w_1, w_2, \dots, w_n$ são os pesos, b são limiares (bias), φ é a função de ativação e y a saída produzida pelo neurônio. A formulação matemática do neurônio é representada pela equação (2.1).

Figura 2.2 – Estrutura de uma rede neural (Adaptado de Grassi 2004).



$$y = \varphi\left(\sum_{n=1}^{i} x_n \cdot w_n + b\right) \tag{2.1}$$

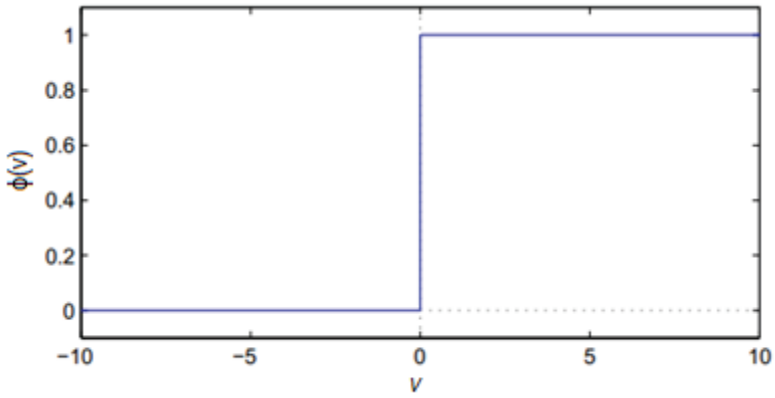
2.3. *Funções de Ativação*

A função de ativação $\varphi(v)$ define a saída do neurônio, sendo os três tipos apresentados abaixo as funções de ativação mais comuns da literatura, são elas:

(a) Função Degrau *(Heaviside Function)*: é normalmente empregada com o modelo de neurônio definido pelo McCulloch e Pitts (Veitch, 2005). Essa função está representada pela Figura 2.3 e é definida da seguinte equação (2.2).

$$\phi(v) = \begin{cases} 1 \text{ se } v \geq 0 \\ 0 \text{ se } v < 0 \end{cases} \tag{2.2}$$

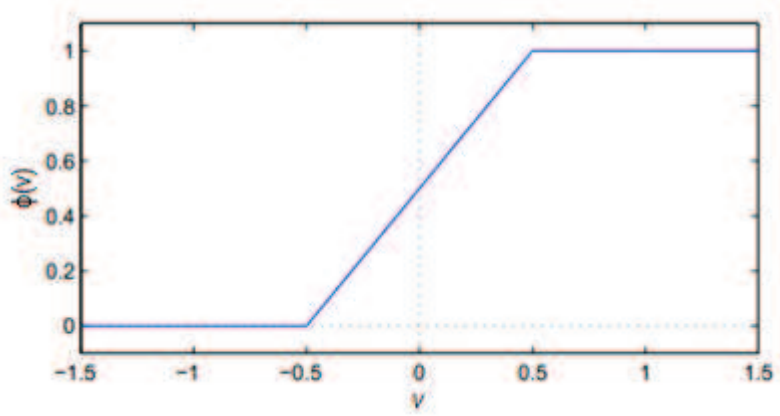
Figura 2.3 – Função degrau



(b) Função Linear por Partes *(Piecewise-Linear Function):* Esta forma de função de ativação é vista como uma aproximação para um amplificador não linear e segundo definição, assume-se que o fator de amplificação é unitário. Este modelo está representada pela Figura 2.4 e pela Equação (2.3) (Veitch, 2005):

$$\phi(v) = \begin{cases} 1 & se\ v \geq \frac{1}{2} \\ v & \text{se} -\frac{1}{2} < v < \frac{1}{2} \\ 0 & se\ v \leq -\frac{1}{2} \end{cases} \tag{2.3}$$

Figura 2.4 - Função linear por partes

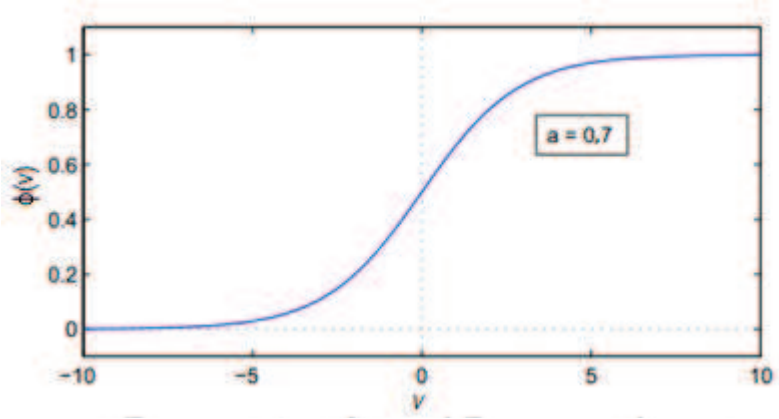


(c) *Função* Sigmóide (*Sigmoid Function*): é a forma mais comum de função de ativação utilizado em redes neurais. Um exemplo é a função logística, definida pela Equação (2.4).

$$\phi(v) = \frac{1}{1 + e^{(-av)}} \tag{2.4}$$

Onde $a > 0$ é o parâmetro *slope*, porém quando $a \to \infty$ acaba-se tornado uma função degrau, no entanto, diferente da função degrau, a função sigmóide é uma função contínua, o que é um fato importante quando a rede começa a realizar o aprendizado (Veitch, 2005), Essa função está representado pela Figura 2.5.

Figura 2.5 - Função sigmóide

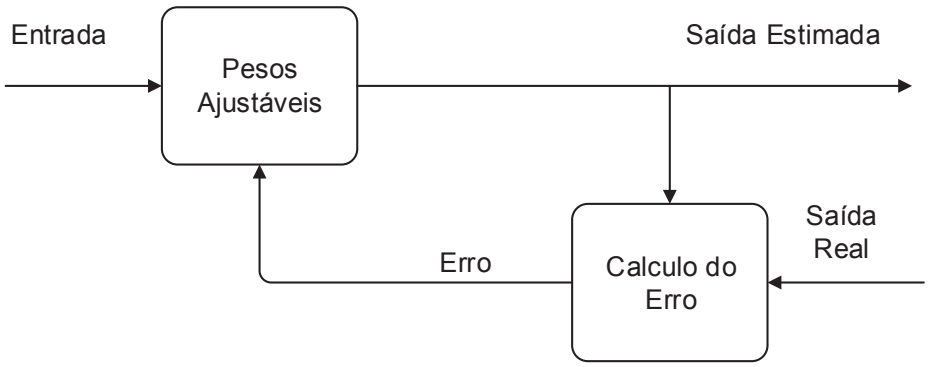


2.4. *Métodos de Treinamento*

O treinamento é uma etapa crucial para as RNs, é nessa etapa que a RN adquire seu conhecimento e só assim consegue responder ao que lhe é estimulado.

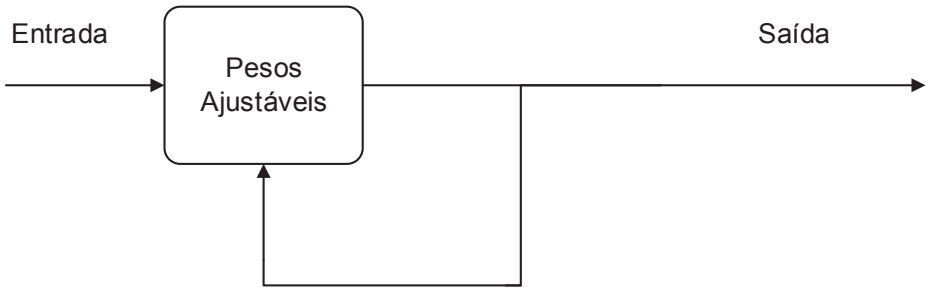
Existem basicamente dois tipos de treinamentos, o treinamento supervisionado e o treinamento não supervisionado. Neste trabalho serão utilizados métodos de otimização para realizar o treinamento supervisionado. Esse procedimento é feito de maneira iterativa buscando reduzir o erro da saída da rede com os dados reais. O treinamento fica no fluxo descrito pela Figura 2.6 até que alguma restrição seja satisfeita. Dentre essas restrições as mais utilizadas é o número de épocas do algoritmo de treinamento, se já atingiu o objetivo da função custo ou se está há algumas iterações sem conseguir melhora.

Figura 2.6 – Treinamento supervisionado



O treinamento não supervisionado é baseado apenas nos estímulos das entradas da rede neural, não há um professor ou crítico inspecionando o processo de aprendizado (Kohonen, 1997). Os seus parâmetros são ajustados a fim de extrair uma tendência estatística dos dados de entrada. O fluxo de treinamento não supervisionado está representado pela Figura 2.7.

Figura 2.7 – Treinamento não-supervisionado



2.5. *Rede Neural de Wavelets*

A rede neural de wavelets surgiu com combinação do conceito de redes neurais com a transformada de wavelets (Lekutai, 1997). Ela vem sendo utilizada com sucesso em uma grande gama de aplicações.

Como exemplo, Ulagammai *et al.*(2007) apresentaram a aplicação do métodos BFT (*bacterial foraging technique*) para realizar o treinamento da wavenet. A rede foi testada em um problema prático da Índia, *Tamil Nadu Electricity Board (TNEB) 69 bus system,* e o métodos BFT conseguiu superar os demais testados.

Outra aplicação, apresentado por Pourtaghi e Lotfollahi-Yaghin (2012,) *tem foco na* minimização do risco de tunelamento, realizando a previsão das deformações do solo induzidas pelo tunelamento.

Fora estes trabalhos, ainda existem várias aplicações em diversas áreas, como em precisão de carga a curto prazo (Bashir e El-Hawary, 2000; Benaouda *et al.,* 2006; Gao e Tsoukalas, 2001), na previsão de séries temporais (Chen, Yang e Dong, 2006), classificação e compreensão de sinais (Kadamble e Srinivasan, 2006) e modelos não linear (Billings e Wai, 2005) entre outros.

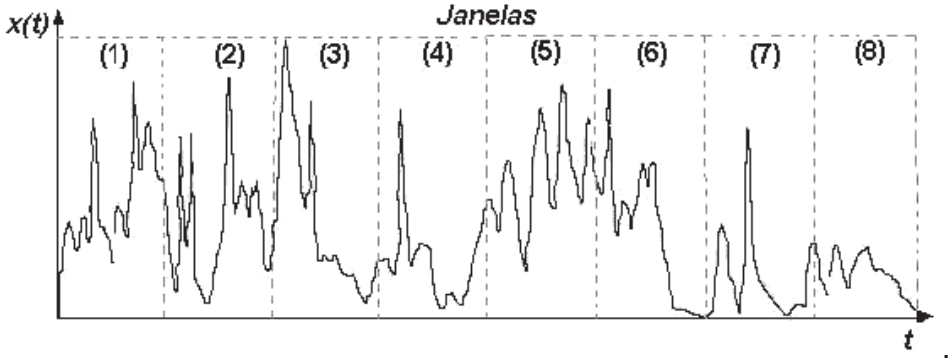
Visto um pouco de suas aplicações, as seções a seguir mostram um pouco sobre a transformada de wavelets e a arquitetura da rede neural de wavelets.

2.5.1. Transformada de Wavelets

Segundo Daubechies (1992) e Meyer (1993) a literatura tem registros de tratamentos semelhantes a transformada de wavelets do começo do século XX, mas ela só passou a ter identidade a partir da década de 70, quando o francês Jean Morlet propôs uma modificação na Transformada de Fourier.

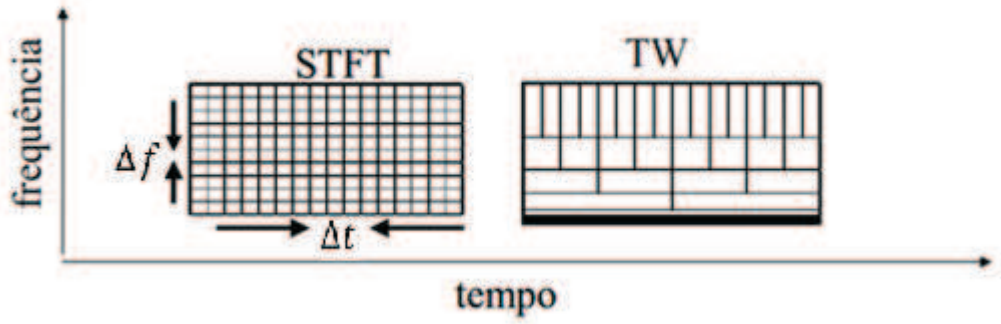
Jean Morlet trabalhava com analise de sinais geofísicos onde os conteúdos de frequência mudavam muito rapidamente com o tempo e a Transformada de Fourier não consegue de adequar a essas mudanças, sendo uma das limitações da TF. Surgiram então, a Transformada de Fourier Janelada ou a Transformada de Fourier de Curta Duração que dividem o sinal em janelas e é aplicado a TF separadamente em cada uma delas conforme a Figura 2.8. Contudo, segundo Gottlieb e Shu (1997) (apud Grassi 2004), esses cortes abruptos causam um problema conhecido como efeito de borda ou fenômeno de Gibbs. Além disso, existe um outro problema que é na escolha da largura das janelas, pois quanto mais janelas melhor fica a análise de cada janela, porém perde-se a resolução espacial, isto é um problema conhecido na área de processamento de sinais como Princípio da Incerteza de Hainsenberg (Cerqueira *et al.,*2000 apud Grassi 2004).

Figura 2.8 – Representação da Transformada de Fourier Janelada (Adaptado de Grassi 2004).



A transformada de wavelets foi introduzida para contornar esses problemas, sendo ela que utiliza-se de janelas sem cortes abruptos e permite uma fácil reconstrução do sinal a partir dos coeficientes da transformada. Também as janelas da TW são de largura variável, conseguindo assim ser bem ajustada para cada pedaço do sinal (Kuçuk e Agiralioglu, 2006). A Figura 2.9 mostra como a Transformada de Fourier de Curta Duração trata o sinal diferente da Transformada de Wavelets no plano tempo-frequência.

Figura 2.9 – Análise no plano tempo-frequência das transformadas (Adaptado de Grassi, 2004).



A wavelet é a função de uma pequena onda que consegue crescer e decair em um período de tempo finito (Veitch, 2005). E é expressa de forma continua pela equação (2.5), sendo x o sinal e $h_{\lambda,t}$ a wavelet, tal que

$$\left(x, h_{\lambda,t}\right) = \int_{-\infty}^{\infty} h_{\lambda,t}(u)x(u)du \qquad (2.5)$$

Existem vários tipos de wavelets, que poder ser escolhidas, O Quadro 2.1 mostra algumas famílias de wavelets.

Quadro 2.1 – Wavelets

NOME	*Wavelets*
MORLET	$cos(\omega_0)exp(-0{,}5u^2)$
MEXICAN HAT	$\frac{2}{\sqrt{3}\sqrt[4]{\pi}}(1-t^2)e^{\frac{-t^2}{2}}$
RASP1	$\frac{u}{(u^2+1)^2}$
RASP2	$\frac{ucos(u)}{u^2+1}$
RASP3	$\frac{sen(\pi u)}{u^2-1}$
POLYWOG1	$u\ exp\left(-\frac{u^2}{2}\right)$
POLYWOG 2	$(u^3-3u)exp\left(-\frac{u^2}{2}\right)$
POLYWOG 3	$(u^4-6u^2+3)exp\left(-\frac{u^2}{2}\right)$
SHANNON	$\frac{sen(2\pi u)-sen(\pi u)}{\pi u}$

2.5.2. Arquitetura da Rede Neural de Wavelets

Explicado o conceito da transformada de wavelets e visto as potencialidades da mesma, ela foi combinada com o conceito de RNs, criou-se a rede neural de wavelets também conhecida como wavenet (Zhang e Benveniste, 1992). A wavenet como as demais redes, tem como objetivo aprender o comportamento de algum problema e reproduzi-lo com o menor erro possível.

A diferença da wavenet para as demais redes é que, os neurônios contém como funções de ativação as funções de wavelets e ficam denominados como neurônios de wavelets ou wavelons. Essa combinação faz com que a wavenet divida o problema em janelas, semelhante ao explicado na TW. Com isso, ela facilita a resolução e necessita menor esforço computacional, mas surgem mais duas variáveis que devem ser encontradas no treinamento além dos pesos w e das bias b que são as dilações a e translações b das wavelets (Alexandridis e Zapranis, 2013).

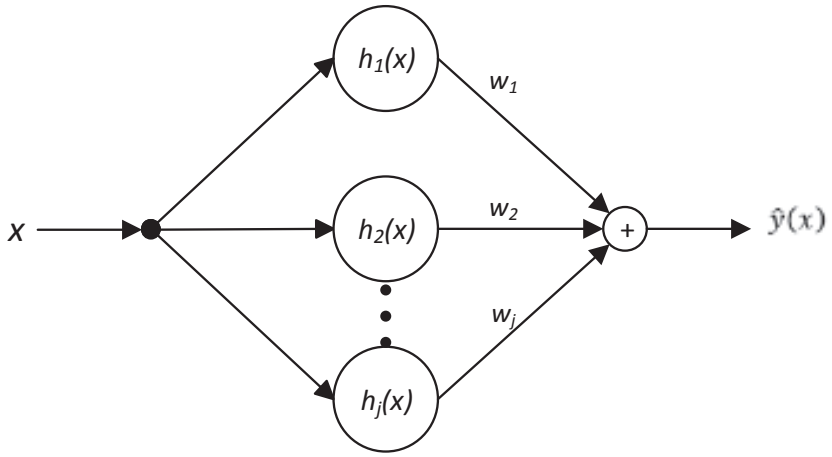
Em Bernard, Mallat e Slotine (1998) são apresentadas diversas razões do porquê utilizar funções de wavelets em vez de outras funções de ativação. Os autores citam

como principal vantagem a habilidade de compressão, também relatam que calcular o valor a um único ponto ou atualizar a estimativa da função de uma nova medida local, envolve apenas um pequeno subconjunto dos coeficientes.

A principal desvantagem da wavenet é que está limitada a aplicações de pequena dimensão de entrada. A razão é que a construção de uma base de wavelet tem um custo computacional acentuado quando a dimensão do vector de entrada é relativamente alto (Zhang, 1997).

A wavenet é uma rede com arquitetura *feed-forward* dividida em três camadas conforme mostrado na Figura 2.10. A primeira camada é a camada de entradas podendo contar uma ou mais entradas. A outra é a camada escondida (*hidden layer*) que contém os wavelons, nessa camada as entradas são transformadas com dilatação e translação e a última etapa é onde aproximação dos valores dos objetivos são estimados (Alexandridis e Zapranis, 2013).

Figura 2.10 – Topologia de Rede Neural de Wavelets



A saída da rede neural pode ser representada pela equação (2.6), onde w_j são os pesos da rede neural, θ são as bias e h_j é definido pela equação (2.7), tal que

$$\hat{y}(x) = \sum_{j=1}^{m} w_j h_j(x) + \theta. \tag{2.6}$$

Na equação (2.7) aplica a dilatação e translação na wavelet mãe ψ, sendo a e b os coeficientes de dilatações e translações respectivamente, ambos são números reais e a deve ser positivo, surgindo assim as wavelets filhas.

$$h_j(x) = a_j^{-\frac{1}{2}} \psi \left(\frac{x - b_j}{a_j} \right) \quad (2.7)$$

A wavelet ψ mãe pode ser qualquer wavelets, porém na bibliografia, as três wavelets mãe mais utilizadas são: a derivada Gaussiana, *Mexican Hat e Morlet*, porém a seleção da wavelet mãe depende da aplicação e não limitadas as citadas.

2.6. *Revisão Bibliográfica*

Segundo Zainunddin, e Pauline (2011), o projeto da rede neural de wavelets é crucial para seu desempenho. Assim eles testaram diferentes tipos de funções de ativação de wavelets e aplicaram na *piecewise function, eles* também aplicaram em um problema real para a previsão da série temporal da poluição. Os resultados mostraram que a integração de diferentes famílias de wavelets tem um melhor desempenho.

Em 2008 como o objetivo de estimar e prever a carga de sedimentos em suspensão em rios os autores, Partal e Cigizoglu(2008), utilizaram a rede neural de wavelets e compararam com métodos convencionais e concluirão que a Wavenet é uma ferramenta apropriada para utilizar com as demais técnicas convencionais.

Mirjalili et al. (2012) realizaram um trabalho com o objetivo de testar o método GSA, PSO e a combinação de ambos PSOGSA para realizar o treinamento de uma rede neural *feedforward*, os resultados mostraram que o método PSOGSA se sobressaiu em relação aos demais tanto em termos de velocidade quanto na capacidade de evitar mínimos locais.

Yand (2012), *et al* utilizaram a rede neural de wavelets para prever o número de taxi. A simulação mostrou que a rede neural de wavelets teve melhor acuracidade na previsão do que a rede neural perceptron multicamadas com treinamento usando algoritmo da retro propagação do erro.

Apresentado no *Journal of Hydrology*, os autores Adamowski e Chan, (2011) relataram a importância de um modelo preciso e confiável na previsão água subterrâneas para um uso sustentável e um adequado abastecimento na área urbana e rural. Dessa forma compararam a Wavenet com a rede neural clássica e com um modelo auto-regressivo integrado de média móvel (ARIMA). Eles concluíram relatando um alto potencial da wavenet para a previsão do nível das águas subterrâneas.

Aquecedor de ar solar é um sistema multi-variável e difícil de modelar com métodos convencionais, assim, Esen, *et al.* (2009) testaram uma rede neural de wavelets e pelo resultados indicaram que a rede pode ser utilizada com sucesso para estimar alguns parâmetros com uma precisão razoável.

Chauhn, Ravi e Karthink (2009) propuseram uma rede neural de wavelets treinada pelo algoritmo de evolução diferencial. Eles testaram para a previsão de falência, também testaram em alguns problemas clássicos como *Iris, Wine* e *Wisconsin Breast*. A rede neural com evolução diferencial que denominaram DEWNN foi comparada com a TAWNN apresentada por Kumar *et al.I* (2008) e com a rede neural de wavelets original. A DEWNN superou ambas em termos de precisão e sensibilidade.

Com a união da rede neural de wavelets com a sistema *fuzzy* Takagi-Sugeno, Yilmaz e Oysal (2010) criaram a FWNN. Realizaram simulações e compararam com outros métodos da literatura e concluíram que a FWNN tem uma capacidade de generalização promissora.

Chia-Nan (2012) testa uma rede neural de wavelets treinada com *annealing dynamical learning algorithm* (ADLA), também *usa support vector regression* (WSVR) para determinar as dilatações e transações e os pesos da wavenet

Ling, *et al.* (2008) propuseram uma rede neural hibrida de enxame de partículas com wavelets. A rede hibrida foi testa na modelagem da distribuição de fluidos para montagem eletrônica (MFD-EP). Nesse estudo de caso o método se mostrou eficiente.

No trabalho de Jin e Liu (2008) propuseram uma rede com a combinação do conceito da rede de base radial com a rede neural de wavelets. A *wavelet basis function neural networks* (WBFNN) se mostrou promissora.

A rede neural de wavelets foi utilizada por Martínez-Morales, Palacios, Valázquez-Carrillo (2013) visando a modelagem de um motor a gasolina. O modelo contém como parâmetros a velocidade do motor, o momento da injeção, o fluxo de massa do combustível injetado e o ângulo da válvula reguladora de pressão de admissão.

3. MÉTODOS DE OTIMIZAÇÃO

Neste capitulo são apresentados os métodos de otimização que serão utilizados para realizar o treinamento das redes neurais.

3.1. *Algoritmo de Evolução Diferencial*

O algoritmo de Evolução Diferencial foi proposto por Storn e Prince em 1995 em um relatório técnico (Storn e Prince, 1995). O algoritmo de otimização é simples porém eficiente e ganhou destaque após apresentar excelentes resultados nas edições de 1996 e 1997 do *International Conteste on Evolutionary Optimization* da IEEE *Internacional Conference on Evolutionary Computation,* ficando em terceiro lugar na edição de 1996 perdendo para dois métodos mais especializados e menos gerais. Na edição de 1997, teve uma atualização das funções de teste e o algoritmo de evolução diferencial foi o que apresentou os melhores resultados (Storn e Price, 1996; Price,1997).

Várias pesquisas mostram que esse método tem se mostrado eficaz em diversas áreas, desde projetos de filtros (Storn, 1999) até treinamento de RNs (Abbass, 2002; Ilonen *et al*., 2003) que é o objetivo desse trabalho.

Apesar do nome e da sua classificação, o algoritmo não tem inspiração em nenhum processo natural. Todas as operações do método se sustenta em argumentos matemáticos e heurísticos, porém o algoritmo segue uma tendência histórica de testes e evolução da população.

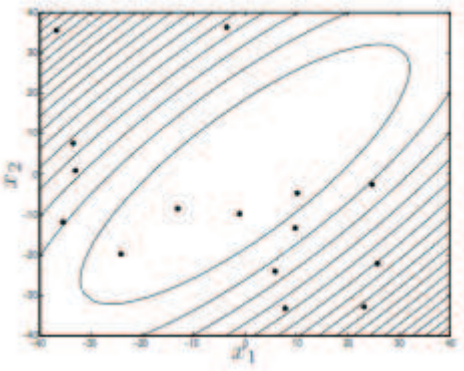
A primeira etapa do método é a criação de uma população de forma aleatória, está população são um conjunto de indivíduos que carregam todos os parâmetros do problema, formando uma matriz conforme apresentado no Quadro 3.1.

Quadro 3.1 – População do algoritmo de evolução diferencial

Indivíduo	P_1	P_2	$\cdots$	P_n
X_1	x_1	x_2	$\cdots$	x_n
X_2	x_1	x_2	$\cdots$	x_n
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
X_k	x_1	x_2	$\cdots$	x_n

Considerando como exemplo um problema de uma função convexa de duas dimensões, a população inicial ficaria distribuída aleatoriamente como na Figura 3.1.

Figura 3.1 – Distribuição da população do ED em um problema de duas dimensões.



Tendo a população, aplica-se o operador de mutação da seguinte forma, é selecionado de forma aleatória dois indivíduos da população e calculada a diferença entre eles, criando um vetor diferencial, e somado a um outro vetor escolhido aleatoriamente tendo assim uma nova solução mutante, conforme descrito matematicamente pela equação (3.1), sendo $v_{t,i}$ a *i*-ésima solução mutante, F é uma constante de mutação e $x_{t,r}$ são os individuos escolhidos de forma aleatória, tal que

$$v_{t,i} = x_{t,r_1} + F\,.\,(x_{t,r_2} - x_{t,r_3}),\ \ r_1, r_2, r_3 \in \{1, \dots, N\}\,. \qquad (3.1)$$

Também é possível utilizar outras estratégias de mutação, conforme mostra o Quadro 3.2.

Quadro 3.2 – Estratégias do algoritmo de evolução diferencial

Estratégia	Mutação	Nomenclatura
1	$U = X_1 + F(X_2 - X_3)$	$ED/rand/1/bin$
2	$U = X_{melhor} + F(X_2 - X_3)$	$ED/melhor/1/bin$
3	$U = X_1 + F(X_2 - X_3 + X_4 - X_5)$	$ED/rand/2/bin$
4	$U = X_{melhor} + F(X_2 - X_3 + X_4 - X_5)$	$ED/melhor/2/bin$
5	$U = X_{antigo} + F\big(X_{melhor} - X_{antigo} + X_1 - X_2\big)$	$ED/rand - melhor/2/bin$
6	$U = X_1 + F(X_2 - X_3)$	$ED/rand/1/exp$
7	$U = X_{melhor} + F(X_2 - X_3)$	$ED/melhor/1/exp$
8	$U = X_1 + F(X_2 - X_3 + X_4 - X_5)$	$ED/rand/2/exp$
9	$U = X_{melhor} + F(X_2 - X_3 + X_4 - X_5)$	$ED/melhor/2/exp$
10	$U = X_{antigo} + F\big(X_{melhor} - X_{antigo} + X_1 - X_2\big)$	$ED/rand - melhor/2/exp$

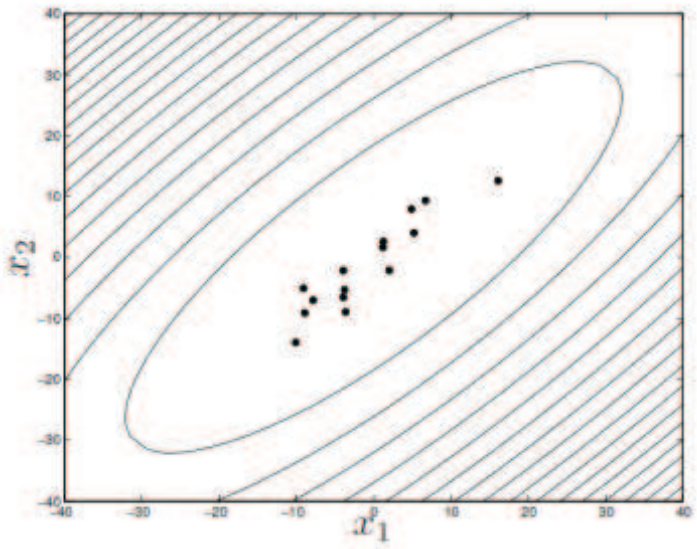
Após a mutação é aplicado o operador de cruzamento, que é a troca de alguns valores de um indivíduo corrente por valores da solução mutante. A formula (3.2) representa essa troca, onde $u_{t,i,j}$ é uma nova solução, $v_{t,i,j}$ a solução mutante e $x_{t,i,j}$ a solução corrente, C é a taxa de crossover e j é um índice escolhido aleatoriamente com distribuição uniforme.

$$u_{t,i,j} = \begin{cases} v_{t,i,j,} & \text{se } rand \leq C \text{ ou } j = \delta_i \\ x_{t,i,j}, & c.c. \end{cases} \tag{3.2}$$

Tendo as novas soluções, elas são testadas e comparadas com as correntes, se as novas soluções forem melhores ela substituem as correntes. Caso contrário, as soluções correntes permanecem por mais uma geração. Esse procedimento é representado pela equação (3.3), onde

$$x_{t+1,i} = \begin{cases} u_{t,i,} & se\ f(u_{t,i}) \leq f(u_{t,i}) \\ x_{t,i,j}, & c.c. \end{cases}. \tag{3.3}$$

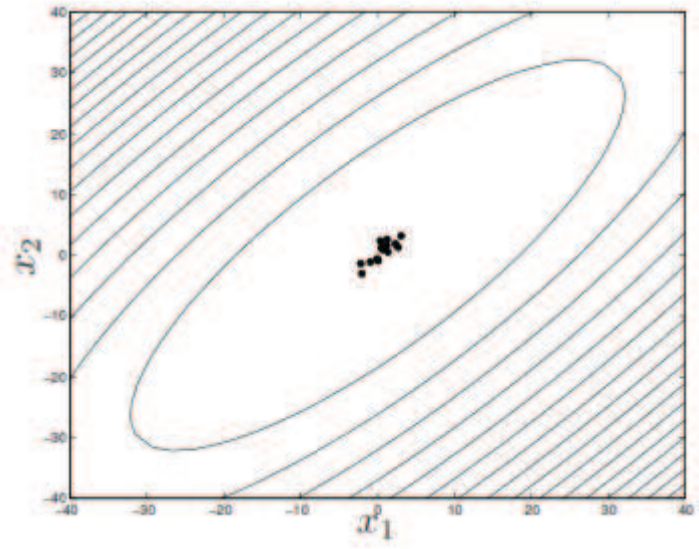
Figura 3.2 – Resultado do ED após 10 gerações.



A Figura 3.2 e a Figura 3.3 mostram como foi a evolução do método a partir da Figura 3.1, sendo a primeira para 10 gerações e a segunda com 20 gerações. Conforme nota-se na Figura 3.2, diferente da Figura 3.1 os indivíduos já estão concentrados perto da solução e mais concentrados ainda na Figura 3.3, o que mostra que com o passar das interações os indivíduos vão se concentrando perto da solução e o passo de

evolução fica menor devido a pequena diferença entre eles conseguindo fazer uma melhor busca local.

Figura 3.3 – Resultado do ED após 20 gerações.



Uma proposta de melhoria foi apresentada por Basu em 2014, trocando a constante de mutação da equação (3.1) por uma variável gaussiana aleatória, formando a equação (3.4),

$$v_{t,i} = x_{t,r_1} + N(0,\sigma)(x_{t,r_2} - x_{t,r_3}) \tag{3.4}$$

onde, $N(0,\sigma)$ representa a variável aleatória gaussiana com média zero e desvio padrão σ. O desvio padrão (σ) é definido pela equação (3.5), onde f_{min} é o menor custo e $f(x_i)$ é o valor custo do indivíduo, tal que

$$\sigma_i = \frac{f(x_i)}{f_{min}}. \tag{3.5}$$

3.2. *Algoritmo de Colônia Artificial de Abelhas*

O algoritmo ABC foi proposto por Karaboga (2005), esse método foi baseado na forma como as abelhas tentam maximizar a quantidade de coleta de néctar usando uma eficiente divisão de trabalho e auto organização. Para realizar esse trabalho existem três tipos de abelhas, abelha batedor, abelha espectador e a abelha empregada.

O processo de coleta inicia-se mandando abelhas batedoras para explorar o local, saindo em diversas direções, enquanto as abelhas espectador ficam na colmeia

esperando as batedoras voltarem. Quando as batedoras encontram uma fonte de alimento elas se tornam abelha empregada e voltam para a colmeia deixar o néctar e escolhem entre três opções: voltar diretamente para a fonte de alimento encontrada, abandonar essa fonte de alimento ou passar informações para as abelhas espectador através da dança sendo que quanto mais intensa a dança maior a fonte de alimento e os movimentos que a abelha reproduz indicam a distância e a posição da fonte. Com a dança agora cabe as abelhas espectador analisar se iram segui-la ou não. Quando a fonte de alimento acaba, as abelhas voltam a se tornar batedoras e buscam novas fontes de alimentos (Karaboga, 2005).

No algoritmo ABC, as fontes de alimentos representam uma possível solução e a quantidade do néctar corresponde a rentabilidade (*fitness*) da solução.

Usando essa analogia das abelhas, torna-se mais fácil o entendimento do método. A primeira etapa é procurar fontes de alimentos aleatoriamente, criando-as conforme a equação (3.6) um vetor de fontes de alimentos que são as soluções iniciais para o problema. Onde $i = 1 \ldots SN$ e $j = 1 \ldots D$, sendo SN o número de fontes de alimento ou soluções, $rand\,(0,1)$ uma função que retorna números aleatórios gerados com distribuição uniforme no intervalo [0 , 1], e D o número de parâmetros do problema (Gao *et al., 2012)*, tal que

$$x_{ij} = x_j^{min} + rand\,(0,1)\left(x_j^{max} - x_j^{min}\right). \tag{3.6}$$

Cada fonte de alimento é associada a somente uma abelha, então o número de fontes de alimentos é igual ao números de abelhas empregadas. Quando a abelha empregada está voltando para a fonte de alimento ela sofre uma pequena modificação do local realizando uma busca na vizinhança. Essa procura na vizinhança pode ser definida matematicamente pela equação (3.7), em que ϕ_{ij} é um número real aleatório uniformemente distribuído no intervalo [-1,1].

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \tag{3.7}$$

Tendo a fonte de alimento x_i, a nova fonte de alimento v_i é encontrada mudando um parâmetro de x_i, essa troca é feita de maneira aleatória devido ao fato de que *j* é uma variável inteira aleatória de tamanho $[1,D]$ e $k \in \{1,2,\ldots SN\}$ também escolhido aleatoriamente.

Após ter gerado novas fontes de alimentos elas podem ser avaliadas pela equação (3.8), onde f_i é o valor custo da solução v_i *tal que*

$$fitness_i = \begin{cases} 1/(1+f_i) \ \ se\ f_i \geq 0 \\ 1 + abs\ (f_i)\ se\ f_i \geq 0 \end{cases}. \tag{3.8}$$

Após as abelhas terem completadas suas buscas, elas compartilham informações sobre a quantidade de néctar e suas posições. As abelhas espectador avaliam todas as informações e escolhem suas fontes de alimentos pela probabilidade da quantidade de néctar. No algoritmo essa probabilidade depende do valor custo do problema, e a seleção pode ser realizada de diversas maneiras, tais como: roleta, baseado em ranking, amostragem universal estocástica, seleção por torneio ou outro tipo de seleção. No ABC clássico, este esquema de seleção é a roleta, em que cada fatia é proporcional em tamanho ao *fitness* representado pela equação (3.9) (Karaboga e Akay, 2009), onde

$$p_i = \frac{fitness_i}{\Sigma_{i=1}^{SN} fitness_i}. \tag{3.9}$$

Essa probabilidade indica que a medida que a quantidade de néctar (valor custo das soluções) aumenta o número de abelhas espectadoras que visita essa fonte também aumenta.

O algoritmo ABC também verifica ao final de cada ciclo se existem fontes de alimento a serem abandonadas, verificando se o contador é maior que o limite estipulado no começo do algoritmo; caso existam elas são abandonadas e criadas outras aleatoriamente no lugar (Akay e Karaboga, 2012).

Uma das propostas recentes é na mudança da criação da população inicial. Gao *et al., (*2012*),* relataram que a população inicial é crucial na evolução do algoritmo, tendo efeito direto na velocidade de convergência e na qualidade da solução final. Desta forma eles propuseram uma nova abordagem empregando a teoria do caos e método de aprendizagem baseado na oposição que possuem ergodicidade, aleatoriedade e irregularidade para gerar a população inicial.

Outra proposta dos autores (Gao *et al., 2012)* é alterar o cálculo da velocidade, trocando a equação (3.7) pelas equações (3.10) e (3.11), nas quais $r_1, r_2, r_3\ e\ r_4$ são inteiros aleatórios escolhidos de $\{1,2,\ldots,SN\}$ e diferente da base i, $\phi_{i,j}$ é um número aleatoriamente uniforme entre [-1,1].

$$v_{\mathrm{i,j}} = x_{best,j} + \varphi_{i,j}(x_{r_1 j} - x_{r_2 j}) \tag{3.10}$$

$$v_{\mathrm{i,j}} = x_{best,j} + \phi_{i,j}\left(x_{r_1 j} - x_{r_2 j}\right) + \phi_{i,j}\left(x_{r_3 j} - x_{r_4 j}\right) \tag{3.11}$$

Estas equações (3.10) e (3.11) foram baseadas nas estratégias do método de evolução diferencial, que podem ser encontradas no

Quadro 3.2.

Outras duas propostas dos autores Akay e Karaboga (2012) são controlar a frequência de perturbação e a magnitude da perturbação.

No algoritmo clássico, a produção da uma nova solução se dá alterando um parâmetro de cada solução, porém isto resulta em uma convergência lenta. Para contornar esse erro, os autores propuseram uma variável MR que controla frequência que essa mutação ocorre. A equação (3.12) representa como esse controle é feito, onde R_{ij} é um valor aleatório e MR uma constante que define a taxa de perturbação.

$$v_{ij} = \begin{cases} x_{ij} + \phi_{ij}\left(x_{ij} - x_{kj}\right), & \text{se } R_{ij} < MR \\ x_{ij}, & c.c. \end{cases} \tag{3.12}$$

A magnitude da perturbação ϕ_{ij}, no ABC clássico, está limitada a uma faixa fixa [-1,1]. Assim a nova proposta é que essa taxa seja variável em uma faixa *[-SF,SF].* Essa variável de controle da magnitude é denominada fator de escala, e se tiver um valor pequeno causa uma convergência lenta, se for grande reduz a capacidade de exploração da perturbação. Por essa razão, o algoritmo proposto pelos autores muda esse fator de escala automaticamente durante a busca de acordo com a regra de *1/5 de Rechenberg's,* o qual afirma que a proporção de mutações bem sucedidas $\varphi(m)$ para todas as mutações devem ser 1/5, representada pela equação (3.13) (Akay e Karaboga, 2012).

$$SF(t+1) = \begin{cases} SF(t) \times 0.85 & \text{se } \varphi(m) < 1/5 \\ \dfrac{SF(t)}{0.85} & \text{se } \varphi(m) > 1/5 \\ SF(t) & \text{se } \varphi(m) = 1/5 \end{cases} \tag{3.13}$$

3.3. *Algoritmo Glowworm*

O algoritmo *glowworm* é baseado na metáfora dos pirilampos e foi desenvolvido por Krishnanand e Ghose (2005.) O algoritmo é uma modificação do método de colônia

de formigas, que utiliza-se da ideia do uso de feromônio para a comunicação. O *glowworm* apoia-se na ideia da troca de informações dos pirilampos, que utilizam a quantidade de luminescência denominada luciferina (*luciferin);* assim, cada pirilampo utiliza a luciferina para passar ao vizinhos a sua localização atual. (Wu *et al*., 2012).

O algoritmo começa distribuindo os pirilampos ou agentes dentro dos limites estipulados, de forma aleatória. Inicialmente todos os pirilampos contém a mesma quantidade de luciferina e em cada iteração do método a luciferina é atualizada (Krishnanand*et al.,* 2006).

Após distribuídos os pirilampos, o algoritmo entra em um *laço* que envolve três principais fases que são as seguintes: atualização da luciferina, movimentação e decisão do alcance (Wu *et al.*, 2012).

Durante a fase de atualização da luciferina, cada pirilampo adiciona à sua quantidade de luciferina uma quantidade proporcional ao seu pressentimento (temperatura, nível de radiação) do seu ponto, sendo esse valor expresso pelo valor custo da função objetivo do problema. Também, ao mesmo tempo é feita uma subtração para simular o decaimento da luciferina com o tempo. Essa fase é representada no algoritmo pela seguinte equação (3.14), tal que

$$l_i(t+1) = (1-p)l_i(t) + \gamma J_i(t+1) \tag{3.14}$$

onde $l_i(t)$ representa o nível de luciferina associado ao pirilampo i no tempo t, p é uma taxa constante de decaimento de luciferina, γ taxa de aprimoramento da luciferina e J_i representa o valor custo da função objetivo do pirilampo i (Zhang *et al.,* 2011).

Na movimentação, os pirilampos são atraídos pelos vizinhos que tem maior brilho. Para representar essa atração, o algoritmo utiliza de um mecanismo de probabilidade para representar o brilho, definido pela equação (3.15), em que $j \in N_i(t)$, $N_i(t) = \{ j : d_{i,j}(t) < r_d^i(t);\ l_i(t) < l_j(t)\}$ é o conjunto da vizinhança do pirilampos i no tempo *t*, $d_{i,j}(t)$ representa a distância Euclidiana entre os pirilampos i e j e r_d^i representa a variável de alcance da vizinhança (Wu *et al.*, 2012),

$$p_{ij} = \frac{l_j(t) - l_i(t)}{\Sigma_{k \in N(t)} l_k(t) - l_i(t)} \tag{3.15}$$

A equação (3.15) é formulada de uma certa forma que as chances de selecionar um pirilampo para mover-se em direção de um vizinho é diretamente proporcional ao nível de luciferina (Krishananand e Ghose, 2008).

Após o pirilampo i ter definido o pirilampo j com base no seu brilho definido pela equação (3.15), a movimentação do pirilampo é realizado conforme a equação (3.16):

$$x_i(t+1) = x_i(t) + s\left(\frac{x_j(t) - x_i(t)}{||x_j(t) - x_i(t)||}\right) \tag{3.16}$$

na qual $x_i(t) \in R^m$ é a localização do pirilampo i, no tempo t, no espeço real m-dimensional R^m, $||.||$ represente o operador Euclidiano e s o tamanho do passo.

Explicado as duas fases anteriores, a última fase da decisão do alcance, que define o alcance de visão na vizinhança é definida pela equação (3.17), com β sendo um parâmetro constante e nt um parâmetro usado para controle do número de vizinhos alcançados (Wu *et al*., 2012),

$$r_d^i(t+1) = \min\{rs, \max\{0, r_d^i(t) + \beta(nt - |N_i(t)|)\}\} \tag{3.17}$$

De acordo com Krishnanand e Ghose (2006), cada pirilampo move uma distância do passo s em direção ao vizinho. No entanto, quando os pirilampos tem distância menor que o passo s, ocorre um efeito chamado *Leapfrogging.* Devido a distância entre eles ser menor que o tamanho do passo, tendo o pirilampo i como líder, quando ocorre a movimentação do pirilampo j, ele passara à frente do pirilampo i tornando-se assim líder. Essa mudança de liderança entre os pirilampos faz com que o algoritmo consiga realizar uma melhor buscar local da solução. (Zhang *et al*., 2011).

Uma proposta de melhoria é alterar o tamanho do passo, definido pela variável s. No algoritmo clássico, s tem um valor constante e é importante na performance e eficiência do método; assim os autores Zhang *et al.,* (2011), propuseram três formas diferentes de tratar essa variável, sendo uma delas um decaimento linear dinâmico e as outras duas um decaimento não linear.

Para um decaimento linear utilizaram a equação (3.18) e os decaimentos não lineares pelas equações (3.19) e (3.20). Dessa forma o passo é alterado pelo número de iterações do método, em que s_0 é o valor inicial do passo s_{min} é o valor mínimo do passo, T_{max} é o número máximo de iterações e *t* é o número da iteração em que o método se encontra.

$$s(t) = \frac{(s_0 - s_{min})(T_{max} - t)}{T_{max}} + s_{min} \quad (3.18)$$

$$s(t) = (s_{min} - s_0)(t/T_{max})^2 + s_0 \quad (3.19)$$

$$s(t) = (s_0 - s_{min})(t/T_{max})^2 + (s_{min} - s_0)(2\,t/T_{max}) + s_0 \quad (3.20)$$

Os autores Zhang *et al.,* (2011) avaliaram esta mudança pelo número de iterações que o método leva para chegar na solução ótima, tendo uma melhora.

Uma outra melhoria proposta pelos autores Zhang *et al.,* (2011) é um comportamento de auto exploração, que é utilizada quando o pirilampo tem um valor custo abaixo de um limite e não tem nenhum vizinho que tenha esse valor acima do limite. A auto exploração começa alterando a localização do pirilampo na forma de uma espiral aleatória, se durante um número de voltas pré-definido da espiral o pirilampo não tem um valor custo acima do limite, o pirilampo volta à posição inicial e recomeça a espiral ou muda para um comportamento de busca aleatória em forma de Z.

Uma outa proposta de melhoria é em utilizar duas novas estratégias na fase da movimentação, trocando assim a equação (3.16) pelas novas equações, que tem como inspiração o algoritmo ABC e o PSO (*Particle Swarm Optimization*). Essa proposta foi apresentada por Wu *et al.*, 2012.

A estratégia baseada no método ABC seria a adaptação da equação (3.7) desenvolvendo assim a equação (3.21), na qual de forma similar ao método ABC, os pirilampos selecionam um vizinho aleatoriamente e mudam seus parâmetros um a um até encontrar um valor custo melhor atualizando assim sua posição.

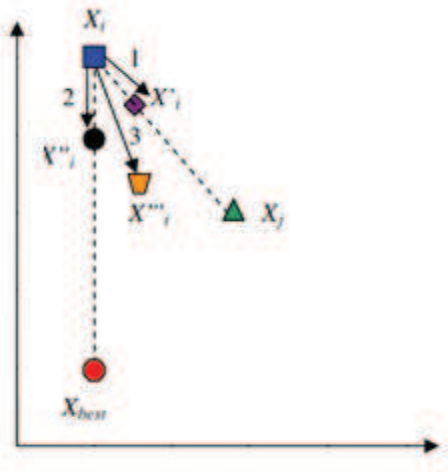
$$x_i(t+1) = x_i(t) + r(x_j - x_i) \quad (3.21)$$

Do PSO monta-se a estratégia definida pela equação (3.22), em que ω é chamado de valor de inércia que controla os impactos das soluções anteriores, c_1 e c_2 são constantes positivas que controlam a influência da melhor solução e da solução do vizinho, respectivamente, onde

$$x_i(t+1) = \omega x_i(t) + rc_1[x_{best}(t) - x_i(t)] + rc_2[x_j(t) - x_i(t)]. \quad (3.22)$$

A Figura 3.4 representa um exemplo das três estratégias, sendo que x_i' representa a posição calculada pela equação (3.16), x_i'' representa a posição calculada pela equação (3.21) e x_i''' a posição calculada pela equação (3.22).

Figura 3.4 – Movimentação das três estratégias (Adaptado de Wu *et al.*, 2012).



3.4. *Algoritmo de Busca Gravitacional*

O algoritmo *Gravitational Search* ou GSA foi proposto por Rashedi *et al.* (2009) e trata-se de um algoritmo estocástico baseado na lei da gravidade e na iteração entre as massas.

Este algoritmo é um método iterativo que simula a atração das massas e se move em múltiplas dimensões do espaço influenciado pela força da gravidade, a Figura 3.5 expressa essa atração na qual os agentes (soluções) são considerados objetos e suas massas são definidas pelos seus desempenhos.

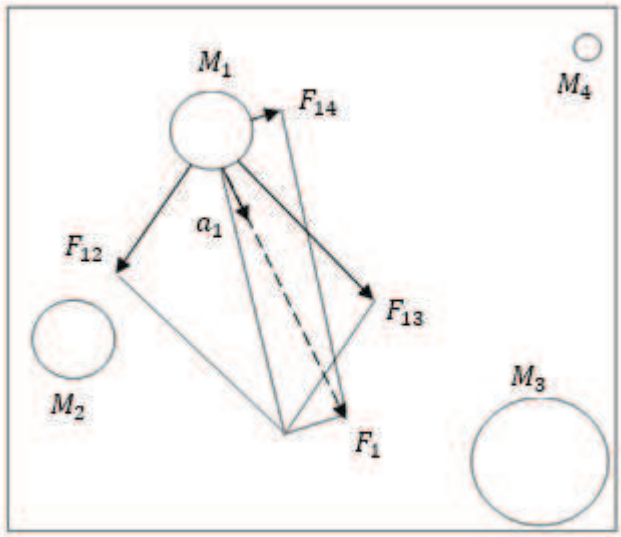
O algoritmo é definido por cinco passos:

1 - Gerar os objetos de forma aleatória montando um vetor de soluções representado pela equação (3.23),

$$X_i = (x_i^1, \dots, x_i^d, \dots, x_i^n), \quad i=1,2,\dots,k \tag{3.23}$$

na qual X_i^d representa a posição do objeto i na dimensão d e n é a dimensão do espaço (Swain *et al.,* 2012).

Figura 3.5 – Atração dos corpos (Adaptado de Rashedi *et al.*, 2009)



2 - Avaliar os objetos na função objetivo e atualizar as variáveis *best(t), worst(t) e* $M_i(t)$ *segundo as equações* (3.24) *até* (3.28)*, (Swain et al.,* 2012):

$$M_i = M_{i,j}, i = 1,2, \dots, k. \tag{3.24}$$

$$m_i(t) = \frac{fit_i(t) - worst(t)}{best - worst(t)} \tag{3.25}$$

$$M_i(t) = \frac{m_i(t)}{\sum_{j=1}^{k} m_j(t)} \tag{3.26}$$

$$best(t) = min_{j \in \{1,\dots,k\}} fit_j(t) \tag{3.27}$$

$$worst(t) = max_{j \in \{1,\dots,k\}} fit_j(t) \tag{3.28}$$

em que $fit_i(t)$ é o valor custo da função objetivo do objeto i no tempo *t* e $M_i(t)$ a massa gravitacional do objeto *i* e, por fim *k* representa o tamanho da população.

3 - Computar a força, definida pela equação (3.29), em que $M_j(t)$ é a massa gravitacional do agente j, $M_i(t)$ é a massa gravitacional do agente i, $G(t)$ é uma constante gravitacional definida pela equação (3.30), ε uma pequena constante e $R_{i,j}(t)$ é a distancia Euclidiana entre os agente i e j (Rashedi *et al.*, 2009).

$$F_{i,j}^d(t) = G(t) \frac{M_{pi}(t) \times M_{aj}(t)}{R_{ij}(t) + \varepsilon} (x_j^d(t) - x_i^d(t)) \tag{3.29}$$

$$G(t) = G_0 e^{-\propto \frac{t}{T}} \tag{3.30}$$

Para dar uma característica estocástica ao algoritmo, Rashedi *et al* (2009) supuseram que a força atual que atua no agente i na dimensão d deve ser um somatório ponderando aleatoriamente as forças exercidas de outros agentes, representado matematicamente pela equação abaixo (Equação (3.31)), onde $\mathrm{rand_j}$ é um numero aleatório dentro do intervalo *[0,1],*

$$F_i^d(t) = \sum_{j=1, j\neq i}^{N} rand_j F_{i,j}^d(t) \tag{3.31}$$

4 - Calcular a aceleração e a velocidade. Baseado na lei do movimento, a aceleração do agente i no tempo t na direção d, $a_i^d(t)$ *é dado pela equação* (3.32).

$$a_i^d(t) = \frac{F_i^d(t)}{M_{ii}(t)} \tag{3.32}$$

Calculada a aceleração, a próxima variável a ser considerado é a velocidade, representado pela equação (3.33).

$$v_i^d(t+1) = rand_i \times v_i^d(t) + a_i^d(t) \tag{3.33}$$

5 - Atualizar a posição dos objetos, somando a velocidade à sua posição atual, conforme a equação (3.34), finalizando assim a iteração do algoritmo, que volta para o passo 2 até que algum critério de parada seja satisfeito, terminando assim o método.

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1). \tag{3.34}$$

Sarafrazi, *et al. (2011),* comentaram que ter duas soluções muito próximas é inútil, então propuseram a inclusão de uma taxa de proximidade que é calculada pela equação (3.35),

$$\frac{R_{i,j}}{R_{i,best}} < C \tag{3.35}$$

em quem $R_{i,j}$ e $R_{i,best}$ são distâncias euclidianas entre as massas i e j, e entre as massas i e a massa com a melhor solução, respectivamente, e C é a taxa limite de proximidade

que deve ser grande enquanto as massas não convergem, para proporcionar maior exploração, e menor quando as massas estão mais próximas.

A posição de cada massa que satisfazer a equação (3.35), será definida pela Equação (3.36), com $U(-0{,}5{,}0{,}5)$ sendo um número aleatório uniformemente distribuído no intervalo $[-0{,}5{,}0{,}5]$, D definido pela equação (3.37) e ρ é uma constante.

$$x_i(t+1) = x_i(t) \times D \tag{3.36}$$

$$D = \begin{cases} R_{i,j} \times U(-0{,}5{,}0{,}5)\ se\ R_{i,best} \geq 1 \\ (1 + \rho \times U(-0{,}5{,}0{,}5) \qquad c.c. \end{cases} \tag{3.37}$$

Essa mudança obteve melhoras no algoritmo conforme Sarafrazi, *et al. (2011),* apresentaram em seu trabalho.

Outra modificação foi apresentado por Khajehzadeh *et al.,* 2012, que relatam que o ajuste de distâncias que os corpos se movimentos(velocidade) definido pela equação (3.33) no algoritmo padrão, é uma variável estocástica e sujeita a criar trajetórias não controladas. Assim, eles introduziram uma restrição a esta variável de acordo com a equação (3.41), na qual v_{max} define a velocidade máxima permitida.

$$-v_{max} \leq v_i^d \leq v_{max} \tag{3.38}$$

A máxima velocidade define o quão rápido um solução pode mudar as suas posições em cada iteração. Se esse parâmetro for alto, as massas podem passar pela solução sem percebe-las, contudo, se for pequeno pode ser que não consiga explorar bem o espaço de busca. Para achar esse ponto de equilíbrio, os autores propuseram a equação (3.42),

$$V_{max} = \left[1 - (\frac{t}{t_{max}})^h\right] \times V_{max0} \tag{3.39}$$

onde h é uma constante positiva*, e* V_{max0} é a velocidade inicial e considerada uma fração da distância entre os limites, representada pela equação (3.40), que tem $\propto$ uma constante no intervalo $[0{,}1]$, $\mathrm{tal\ que}$

$$V_{max0} = \propto \times (x_{max} - x_{min}). \tag{3.40}$$

Dessa forma o algoritmo vai diminuindo a velocidade conforme se vai realizando as interações do métodos, deixando o método melhor para realizar uma busca local. Os autores mostraram em seu trabalho uma melhora nos resultados.

3.5. *Algoritmo Imperialista Competitivo*

O algoritmo imperialista competitivo ou ICA (*Imperialist Competitive Algorithm)* foi introduzido por Atashpaz e Lucas (2007). Como o próprio nome diz, o algoritmo é inspirado em uma competição imperialista, onde os imperadores tentam controlar mais países usando seu poder.

O objetivo da otimização é achar a solução ótima em termos das variáveis do problema. Assim, da mesma forma como é feita em outros métodos, é formada uma estrutura inicial com tamanho do número de variáveis do problema, essa estrutura é chamado de país neste método e pode ser representado pela Equação (3.41):

$$country = [p_1, p_2, p_3, \dots, p_{N_{var}}] \tag{3.41}$$

em que $\mathrm{p_i}$ $\mathrm{sendo}\, i = 1, 2, 3, \dots, N_{var}$ são as variáveis a serem otimizadas. O custo é calculado de acordo com a Equação (3.42):

$$cost = f(country) = \left[p_1, p_2, p_3, \dots, p_{N_{var}}\right]. \tag{3.42}$$

Para iniciar a otimização $\mathrm{N_{país}}$ são criados e os mais poderosos são selecionados *N-imperadores,* os restantes dos países se tornaram *N-colônias* dividido entre os imperadores criando assim um império.

O número de colônias que cada imperador recebe é proporcional ao seu poder. Para a divisão é necessário o custo normalizado de cada imperador definido pela equação (3.43), na qual c_n é o custo do imperador n e C_n é o custo normalizado, tal que

$$C_n = c_n - max_i\{c_i\}. \tag{3.43}$$

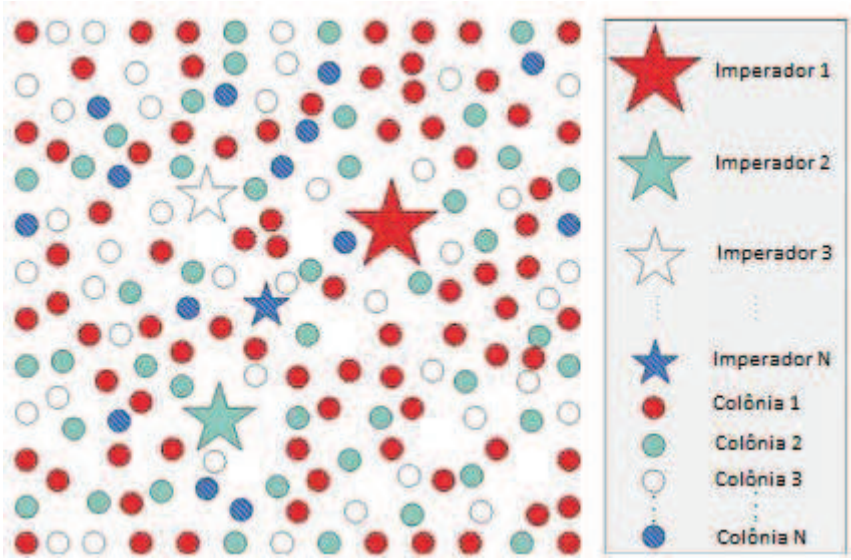
Tendo o custo normalizado de cada imperador calcula-se o poder normalizado de cada um pela Equação (3.44). Com isso o número de colônias de cada imperador é definido pela Equação (3.45), na qual N_{col} é o número total de colônias e $N.C._n$ é o número de colônias que cada imperador receberá.

$$p_n = \left|\frac{C_n}{\sum_{i=1}^{N_{imp}} C_i}\right| \tag{3.44}$$

$$N.C._n = round\{p_n \times N_{col}\} \tag{3.45}$$

A distribuição das colônias acontece de forma uniformemente aleatória entre os imperadores. A Figura 3.6 representa essa divisão mostrando que o imperador mais poderoso tem um número maior de colônias do que os imperadores mais fracos.

Figura 3.6 – Distribuição dos Imperadores e suas colônias (Adaptado de Yousefia *et al.,* 2012*)*.



Com as colônias distribuídas, elas começam a andar em direção ao imperador, conforme a equação (3.46),

$$\{x\}_{new} = \{x\}_{old} + U(0, \beta \times d) \times \{V_1\} \quad (3.46)$$

$$\theta \sim U(-\gamma, \gamma) \quad (3.47)$$

onde β é um parâmetro de controle, d é a distância euclidiana entre a colônia e o imperador, $U(0, \beta \times d)$ é um número aleatório uniformemente distribuído entre 0 e $\beta \times d$, e $\{V_1\}$ tem o ponto anterior e sua direção para o imperador porém não necessariamente seguem em linha reta ao imperador, a equação (3.47) insere uma perturbação para aumentar o campo de pesquisa onde γ é um parâmetro de ajuste de desvio. A

Figura 3.7 representa esta mudança de posição (Talatahari *et al.,* 2012).

Durante a movimentação, caso a colônia tenha um custo menor que o imperador, ocorre a troca entre eles com o imperador ficando no lugar da colônia e a colônia no lugar do imperador. Isto é ilustrado na Figura 3.8.

Figura 3.7 – Movimento da colônia em direção ao imperador. (Adaptado de Talatahari *et al.,* 2012)

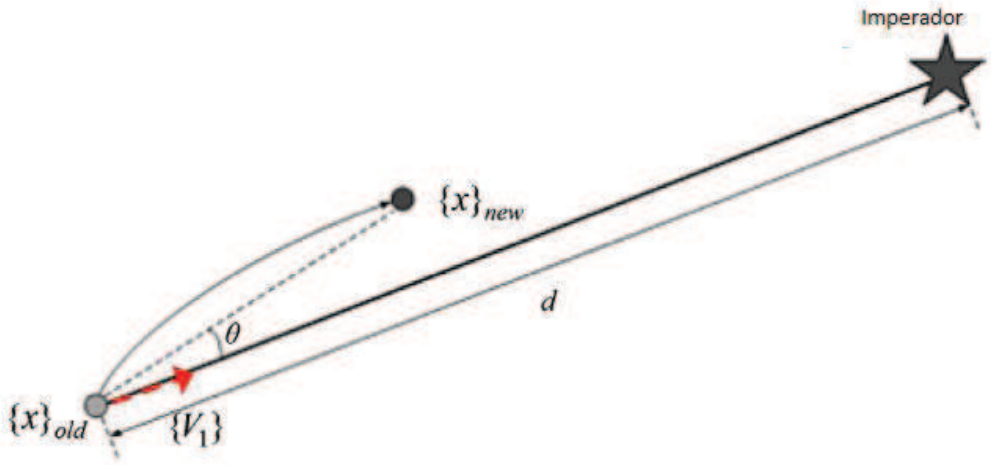
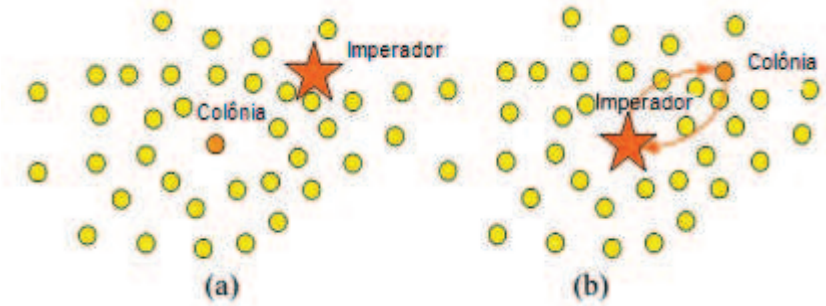


Figura 3.8 – Troca de posição (a) A colônia tem um custo menor que o Imperador (b) Ocorreu a troca de posição da colônia com o imperador (Adaptado de Atashpaz-Gargari e Lucas, 2007).



Uma das partes importantes do ICA é a competição dos imperadores, onde eles tentam tomar posse de colônias de outros imperadores. Isto ocorre de acordo com uma probabilidade representada pela equação (3.48),

$$P_{p_n} = \left| \frac{N.T.C._n}{\sum_{i=1}^{N_{imo}} N.T.C._i} \right| \tag{3.48}$$

em que N_{imp} é o número de imperadores e $N.T.C._n$ é o custo total normalizado do império *n* que é definido pela equação (3.49), e o custo total $T.C._n$ é calculado pela equação (3.50),

$$N.T.C._n = T.C._n - max_i\{T.C._i\} \quad (3.49)$$

$$T.C._n = cost(Imp_n) + \varepsilon.mean\{cost(colônias_n)\} \quad (3.50)$$

sendo que ξ é uma constante positiva.

Calculada a probabilidade de cada imperador tem-se um vetor P (3.51). Esse vetor é subtraído de um vetor de mesmo tamanho R (3.52), que é criado com uma distribuição uniforme aleatória, resultando em D (3.53), sendo que o império que tiver o maior valor em D conquista a colônia.

$$P = [p_{p_1}, p_{p_2}, p_{p_3}, \dots, p_{p_{N_{imp}}}] \quad (3.51)$$

$$R = [r_1, r_2, r_3, \dots, r_{N_{imp}}] \quad (3.52)$$

$$D = P - R = \left[D_1, D_2, D_3, \dots, D_{N_{imp}}\right] = [p_{p_1} - r_1, p_{p_2} - r_2, p_{p_3 - r_3}, \dots, p_{P_{N_{imp}}} - r_{N_{imp}}] \quad (3.53)$$

Esse processo de seleção é similar a uma roleta utilizado em outro métodos, porém bem mais rápido.

Após a decisão da colônia, o método chega ao fim de uma iteração e volta para a movimentação das colônias e fica nesse ciclo até que um critério de para seja satisfeito.

Recentemente, Kaveh e Talatahari, 2010, propuseram a troca da equação (3.46) pela equação (3.54), na qual $\{V_2\}$ é perpendicular a $\{V_1\}$; outra mudança foi $U(-1,1)$ que ficou como um número aleatório distribuindo uniformemente no intervalo [-1, 1]. Após essa mudança o algoritmo pode ser denominado de Algoritmo Imperialista Competitivo Ortogonal.

$$\{x\}_{new} = \{x\}_{old} + \beta \times d \times \{rand\} \otimes \{V_1\} + U(-1,1) \times \tan(\theta) \times d \times \{V_2\}, \quad \{V_1\} \cdot \{V_2\} = 0, \quad \|\{V_2\}\| = 1 \quad (3.54)$$

De forma similar ao que Kaveh e Talatahari (2010), os autores Talatahari *et al.* (2012), propuseram três variações no ICA baseado em mapas caóticos. Na primeira proposta, eles modificaram a equação (3.54) retirando o vetor $\{rand\}$ por $\{cm\}$ que é um vetor caótico, como mostra a equação (3.55).

$$\{x\}_{new} = \{x\}_{old} + \beta \times d \times \{cm\} \otimes \{V_1\} + U(-1,1) \times \tan(\theta) \times d \times \{V_2\}, \quad \{V_1\} \cdot \{V_2\} = 0, \quad \|\{V_2\}\| = 1 \quad (3.55)$$

A segunda modificação foi retirar o parâmetro $U(-1,1)$ e colocar um vector caótico $\{cm\}$ no lugar conforme a equação (3.56).

$$\{x\}_{new} = \{x\}_{old} + \beta \times d \times \{rand\} \otimes \{V_1\} + \{cm\} \times \tan(\theta) \times d \times \{V_2\}, \quad \{V_1\} \cdot \{V_2\} = 0, \quad \|\{V_2\}\| = 1 \tag{3.56}$$

A última proposta é a combinação das duas primeiras, realizando a troca de $\{rand\}\ e\ U(-1,1)$ por vetores caóticos, montando a equação (3.57) a seguir.

$$\{x\}_{new} = \{x\}_{old} + \beta \times d \times \{cm\} \otimes \{V_1\} + cm \times \tan(\theta) \times d \times \{V_2\}, \quad \{V_1\} \cdot \{V_2\} = 0, \quad \|\{V_2\}\| = 1 \tag{3.57}$$

4. CASOS AVALIADOS PARA PREVISÃO

Neste capítulo são apresentados as séries temporais escolhidas e também os indicadores de desempenho que serão avaliados.

4.1. *Previsão do Preço da Saca de Soja*

A previsão de séries temporais do agronegócio já vem sendo analisadas por vários especialistas em diferentes situações, variáveis e diferentes métodos.

Yonenaga e Figueiredo (1999) realizaram a previsão da saca de soja utilizando uma rede neural com o método de retropropagação do erro. O trabalho foi realizado utilizando o preço coletado mensalmente de janeiro de 1975 a julho de 1995 totalizando 247 observações que foi dividido entre padrões de treinamento e de teste. A previsão foi realizada para os dez meses seguintes e os autores concluíram que as redes neurais podem ser uma boa alternativa aos métodos estatísticos.

Balanagammal *et al.* (2000) estudaram duas culturas: o arroz e a cana de açúcar. Nelas eles utilizaram séries temporais do tipo ARIMA para realizar a previsão da área plantada, da produção e da produtividade da cultura.

Ramirez e Fadiga (2003) em seu trabalho de previsão do preço de commodities agrícolas utilizando o modelo assimétrico GARCH, trataram tipos de culturas como a soja, o sorgo, e o trigo e concluíram que o modelo assimétrico GARCH é uma alternativa interessante para realizar previsões de séries temporais principalmente quando as variáveis dependentes são assimétricas.

Lima *et al.* (2007) realizaram a previsão de quatro séries temporais: preço do açúcar, preço do café, preço do soja, e o preço do boi gordo utilizando modelos de séries temporais do tipo ARMA/ARIMA e comparou com as séries dos tipo ARFIMA e chegou à conclusão que o modelo ARFIMA apresentou um menor erro quadrado médio, exceto na série da soja.

Kumar *et al.* (2010) estudaram as séries temporais com lógica fuzzy para realizar a previsão da produção de trigo, e relataram da importância para o administrador da fazenda e para as indústrias locais realizarem as suas gestões.

A fim de testar novas ferramentas esse presente trabalho utiliza os dados retirados do CEPEA(Centro de Estudos Avançados em Economia Aplicada – ESALQ/USP) do período de 29 de julho de 1997 a 25 de setembro de 2014. A Figura 4.1 representa esta série de dados.

Figura 4.1 – Serie de dados do preço da saca de soja de julho de 1997 a setembro de 2014 (CEPEA).

4.2. *Previsão de Demanda*

O dados para a previsão de demanda são de uma empresa de alimentos, a qual tem representatividade no mercado brasileiro, tem um variado mix de produtos dentro das divisões de produto de linha seca, congelados e laticínios. Este dados estão em propriedade e somente é utilizado a fim de comparações.

Tendo alta rentabilidade para a empresa e seus produtos com um curto ciclo de vida, a divisão de laticínios necessita de uma boa ferramenta de previsão de demanda para um melhor planejamento estratégico, sendo esta divisão escolhida para o teste de novas técnicas de previsão.

Com a ideia que a previsão de demanda de um nível agregado é mais interessante do que dos itens individuais, os produtos da divisão foram agrupados em grupos de acordo com algumas características como: composição, linha de produção, prazo de validade e embalagens, resultando em cinco grupos conforme descrito a seguir.

a) Grupo A: este grupo é composto por produtos que representam 70% do total de vendas da empresa.
b) Grupo B: este grupo é composto por produtos que representam 10,5% do total de vendas da empresa.
c) Grupo C: este grupo é composto por produtos que representam 8,5% do total de vendas da empresa.

d) Grupo D: este grupo é composto por produtos que representam 10% do total de vendas da empresa.
e) Grupo E: este grupo é composto por produtos que representam 1% do total de vendas da empresa.

Os dados utilizados foram coletado no período janeiro de 2005 a dezembro de 2013, sendo utilizados para a avaliação das técnicas somente os grupos A, B e C devido sua maior representatividade de demanda da empresa.

Figura 4.2 representa os dados do grupo A e a Figura 4.3 representa os demais grupos.

Figura 4.2 – Dados de demanda dos produtos do grupo A no período de 2005 a 2012.

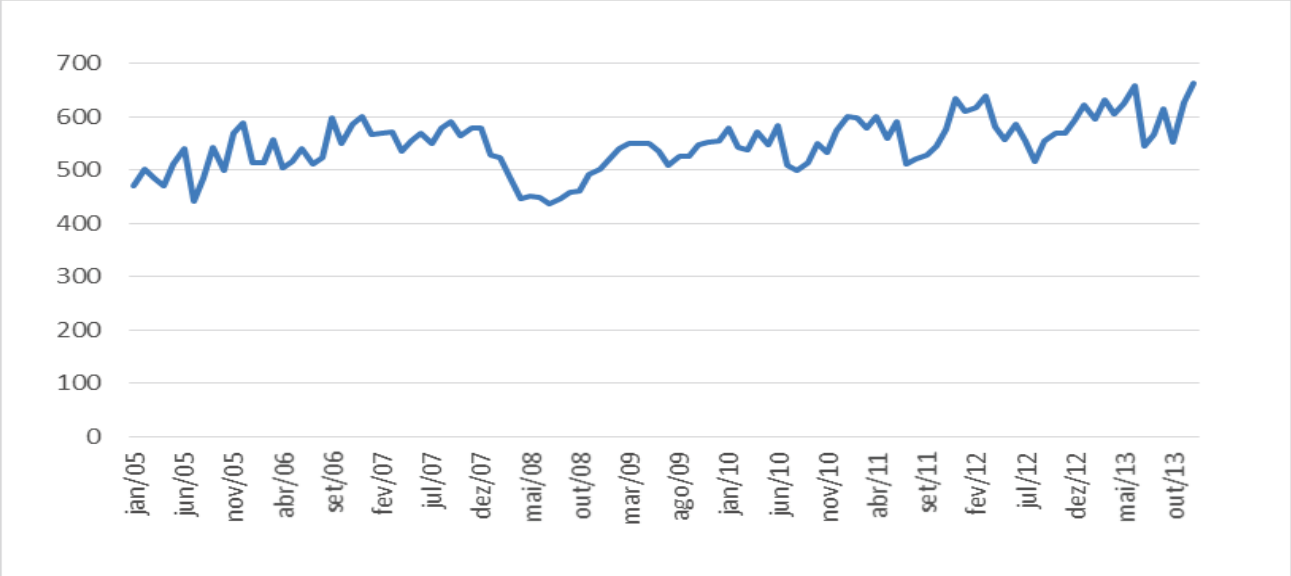
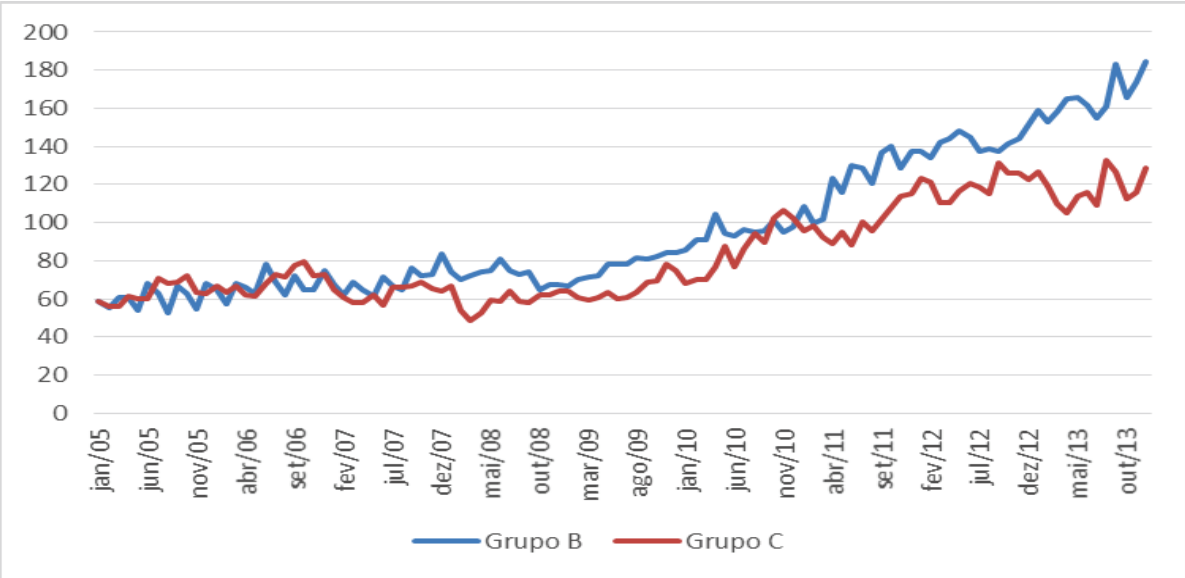


Figura 4.3 – Dados de demanda dos produtos dos grupos B-C no período de 2005 a 2013.

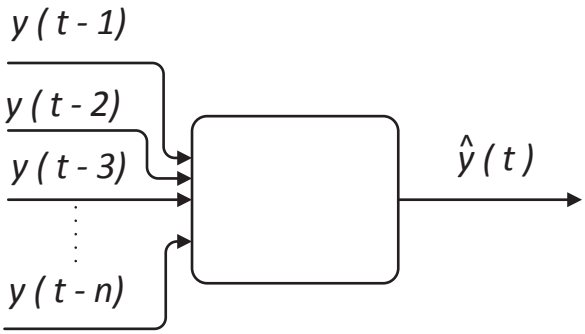


4.3. *Entradas*

Os dados para a previsão serão divididos em duas parcelas. A primeira parcela, contém a maior parte dos dados é utilizada para realizar o treinamento das RNs, a segunda parcela é utilizada para verificar a qualidade da previsão das RNs através dos indicadores de desempenho explicados no tópico a seguir.

Para a previsão de ambos os estudos de caso, são utilizados somente os dados históricos. Com isto, as entradas serão os mesmos dados de saída da rede, porém com uma defasagem no tempo. Uma possibilidade para tentar melhorar a previsão é aumentar o número de entradas atrasadas no tempo, conforme mostra a Figura 4.4, sendo $y(t-n)$ os dados dos estudos de caso atrasados e $\hat{y}(t)$ o valor previsto no instante t.

Figura 4.4 – Entradas e saída das RNs.



4.4. *Indicadores de Desempenho*

A comparação dos métodos será realizada utilizando dois indicadores de desempenho: MSE (*Mean Square Error*), MAPE (*Mean Absolute Percentage Error).*

O MSE é uma medida estatística que compara os estimadores levando em consideração a média do erro que o modelo apresentou relacionando a saída real e a saída estimada. A equação (4.1) representa a forma de calcular o MSE, na qual y_i é a entrada real, e $\hat{y}_i$ *é* a saída estimada pelo modelo de previsão*.*

$$MSE = \sum_{i=1}^{n} \frac{(y_i - \hat{y}_{i)}{}^{2}}{n} \tag{4.1}$$

O MAPE, que é uma das medidas de erro mais usadas para avaliar métodos de previsão, ele representa o valor absoluto médio que permite avaliar a margem de acerto do método. O MAPE é definido pela equação (4.2):

$$MAPE_{\mathrm{n}} = \frac{\sum_{i=1}^{h} \left| \frac{\mathrm{y}_i - \hat{y}_i}{\mathrm{y}_i} \right|}{\mathrm{n}} \times 100 \tag{4.2}$$

5. RESULTADOS

Neste capitulo é exposto os resultados das previsões dos dois estudos de casos e uma comparação das previsões no final desse capítulo.

Para a previsão de demanda foram utilizados somente os grupos A, B e C por representar 89% da demanda da empresa.

Os testes foram realizados utilizando o *software* Matlab R2013a em um computador com processador Athlon Phenom II X6 1090T e 16GB de memória RAM.

Os resultados são apresentados contendo a etapa aprendizagem ou validação da rede e dois tipos de previsões sendo: uma previsão a frente, 30 previsões à frente para a previsão do preço da saca de soja e 12 previsões a frente para a previsão de demanda.

Os testes foram realizados utilizando a rede neural de wavelets com função de ativação de *mexican hat* apresentada no Quadro 2.1*,* alterando somente o número de neurônios e o número de entradas dos dados conforme explicado na seção 4.3. A rede foi testada com 5, 10, 20, 40 e 80 neurônios e com 1, 2, 3, 4, 5 entradas e os métodos de otimização foram utilizados em sua forma clássica com parâmetros encontrados na literatura.

5.1. *Parâmetros*

Devido aos métodos de treinamento se tratarem todos de métodos evolucionários, contém alguns parâmetros em comum que estão apresentados no Quadro 5.1.

Quadro 5.1 – Parâmetros em comum dos métodos

Parâmetro	Valor
Limite Superior	10
Limite Inferior	-10
Tamanho da população	300
Máximo de iterações	1000

O algoritmo de Evolução diferencial tem três parâmetros para ser ajustados, mutação, crossover e a estratégia de mutação utilizada. A constante de mutação *F* tem que estar no intervalo entre [0 2], Storn e Price (1997) recomendam que o valor de F esteja entre 0.4 e 1, sendo 0,5 uma boa escolha inicial.

A taxa de *crossover* deve estar ente 0 e 1, sendo quanto mais perto de zero mais lento a evolução e quanto mais perto de perto de 1 mais arriscado. Com isso Mayer *et al.*(2005) comenta que aparentemente 0,5 é uma boa escolha.

A estratégia utilizada é a número 1 do
Quadro 3.2, devido ser a clássica do método. Assim, os parâmetros do algoritmo de evolução computacional fica conforme o Quadro 5.2 abaixo.

Quadro 5.2 – Parâmetros utilizados no algoritmo de evolução diferencial.

Parâmetro	Valor
Mutação	0,5
Crossover	0,5
Estratégia	1

No algoritmo de colônia artificial de abelhas não tem parâmetros além dos mostrados no Quadro 5.1. Abaixo segue os resultados para cada estudo de caso.

No algoritmo glowwom, tem cinco parâmetros para ajustar. Estes parâmetros foram ajustados igual em Zhang *et al* (2011) onde o autor citou extenso testes em Torn, A., Zilinskas, A. e estão definido pelo Quadro 5.3.

Quadro 5.3 – Parâmetros utilizados no algoritmo glowwom.

Parâmetro	Valor
Taxa de decaimento da luciferina (p)	0,4
Taxa de aprimoramento da luciferina (γ)	0,6
Parâmetro (β)	2
Alcance (nt)	3
Tamanho do passo (s)	0.01

O algoritmo de busca gravitacional tem dois parâmetros a ser ajustado. Os parâmetros são os mesmo utilizados em Rashedi *et al.*, (2009) sendo $G_0 = 100$ e $\propto = 20$.

Além dos parâmetros em comum dos métodos, o ICA tem mais 3 parâmetros utilizando os mesmos parâmetros utilizados no trabalho de Khabbazi, A. *et al.(2009)* e estão apresentados no Quadro 5.4

Quadro 5.4 - Parâmetros utilizados no algoritmo imperialista competitivo.

Parâmetro	Valor
Número de Imperadores	8
Coeficiente (β)	2
Angulo (γ)	0,5

5.2. *Previsão do Preço da Saca de Soja*

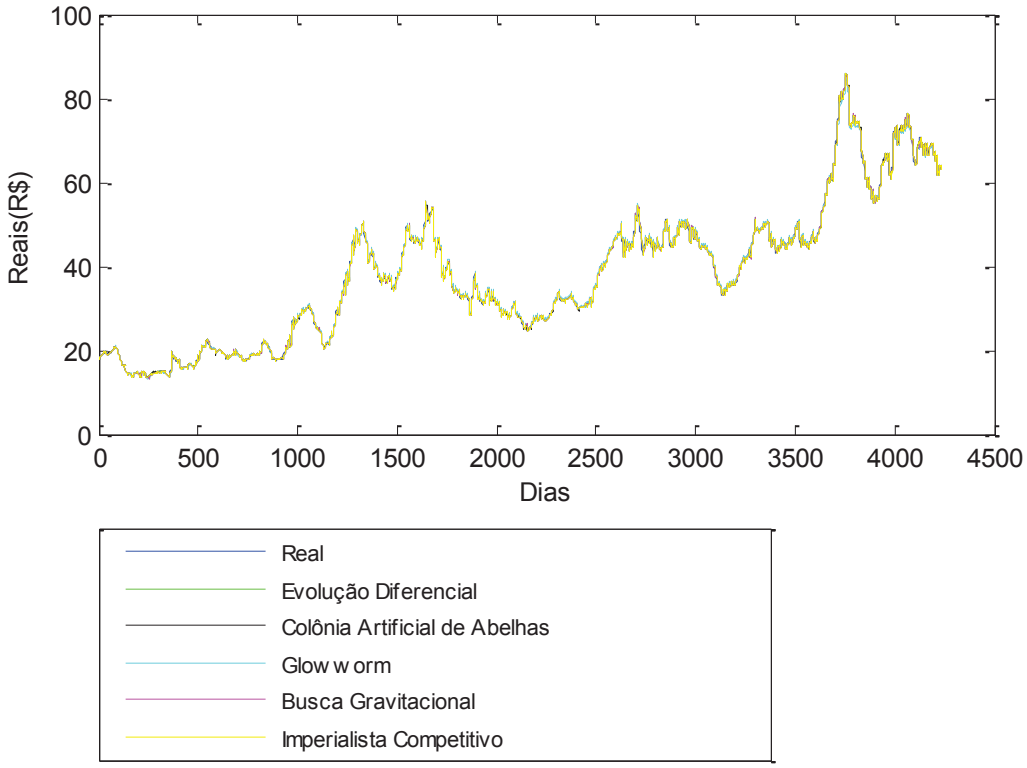
No estudo de caso da previsão da saca de soja, os algoritmos de Evolução Diferencial, Colônia Artificial de Abelhas e *Glowworm* tiveram os melhores resultados com 1 entrada e 5 neurônios, os algoritmos de Busca Gravitacional e Imperialista Competitivo tiveram os melhores resultados com 1 entrada e 20 neurônios, conforme o Quadro 5.5 onde eles foram avaliados pelo MSE.

Quadro 5.5 - Resultados da validação dos métodos para o preço da saca de soja.

		Validação				
		Resultados do Algoritmo de Evolução Diferencial				
MSE		Neurônios				
		5	10	20	40	80
Entradas	1	0,24481	0,62753	0,45785	1,73905	14,76558
	2	0,37681	1,13513	2,82994	6,19377	17,91665
	3	0,53901	1,34267	2,54531	57,48078	33,43872
	4	0,68385	3,54953	1,36325	7,41959	163,48779
	5	0,68543	1,43402	11,81485	7,69641	311,56609
		Resultados do Algoritmo de Colônia Artificial de Abelhas				
Entradas	1	0,203519	0,215522	0,214986	0,500807	0,565308
	2	0,380834	0,513281	0,405885	0,394512	2,482299
	3	0,613493	0,466082	1,054721	3,264717	4,229186
	4	0,724404	0,784983	1,809568	2,470075	27,505662
	5	1,202066	1,286075	2,022455	2,535765	52,956636
		Resultados do Algoritmo Glowworm				
Entradas	1	0,33776	0,50434	9,28444	30,85866	97,68326
	2	1,35017	4,72561	24,13694	47,86830	1478,67275
	3	43,63027	0,82958	32,17561	420,16388	2715,88435
	4	13,16071	62,24671	260,07326	1211,11736	10548,82379
	5	2,65273	317,30718	57,71223	3941,78323	11603,80924
		Resultados do Algoritmo de Busca Gravitacional				
Entradas	1	3,54123	0,28613	0,20531	0,20847	0,20543
	2	5,44104	1,30398	0,33223	0,61591	1,15607
	3	6,07093	0,95016	0,68895	0,67655	1,53876
	4	6,37072	8,37624	0,93246	3,40530	0,93434
	5	3,44182	6,31666	1,58628	2,89334	3,60166
		Resultados do Algoritmo Imperialista Competitivo				
Entradas	1	0,20513	0,21181	0,20322	0,21164	0,20323
	2	0,31272	0,31785	0,31737	0,31504	1,93186
	3	0,43049	0,43462	0,43743	0,49314	0,70287
	4	0,55101	0,64869	0,73921	1,27390	2,28813
	5	0,67804	0,67590	0,68008	0,68173	1,61805

As configurações que tiveram os melhores resultados na validação estão expressas graficamente pela Figura 5.1. Com essas configurações o método de Evolução Diferencial teve um MAPE igual a 0,9724, a última época que o algoritmo evoluiu foi a 964 e demorou aproximadamente 8,8 minutos para concluir o treinamento, o algoritmo de Colônia Artificial de Abelhas teve um MAPE igual a 0,7990, a última época que o algoritmo evoluiu foi a 644 e demorou aproximadamente 9,2 minutos para concluir o treinamento, *Glowworm* teve um MAPE igual a 1,1198, a última época que o algoritmo evoluiu foi a 76 e demorou aproximadamente 26,8 minutos para concluir o treinamento, na Busca Gravitacional teve um MAPE igual a 0,8073, o algoritmo evoluiu até a última época e demorou aproximadamente 54,5 minutos para concluir o treinamento e por fim o algoritmo Imperialista Competitivo teve um MAPE igual a 0,7995, a última época que o algoritmo evoluiu foi a 994 e demorou aproximadamente 14,4 minutos para concluir o treinamento.

Figura 5.1 – Melhores configurações dos métodos na validação do preço da saca de soja.



Passado a validação, os métodos foram submetidos para dois tipos de previsão, sendo uma com somente uma previsão a frente e com 30 previsões a frente. O Quadro 5.6 apresenta os resultados com as mesmas configurações utilizadas na validação,

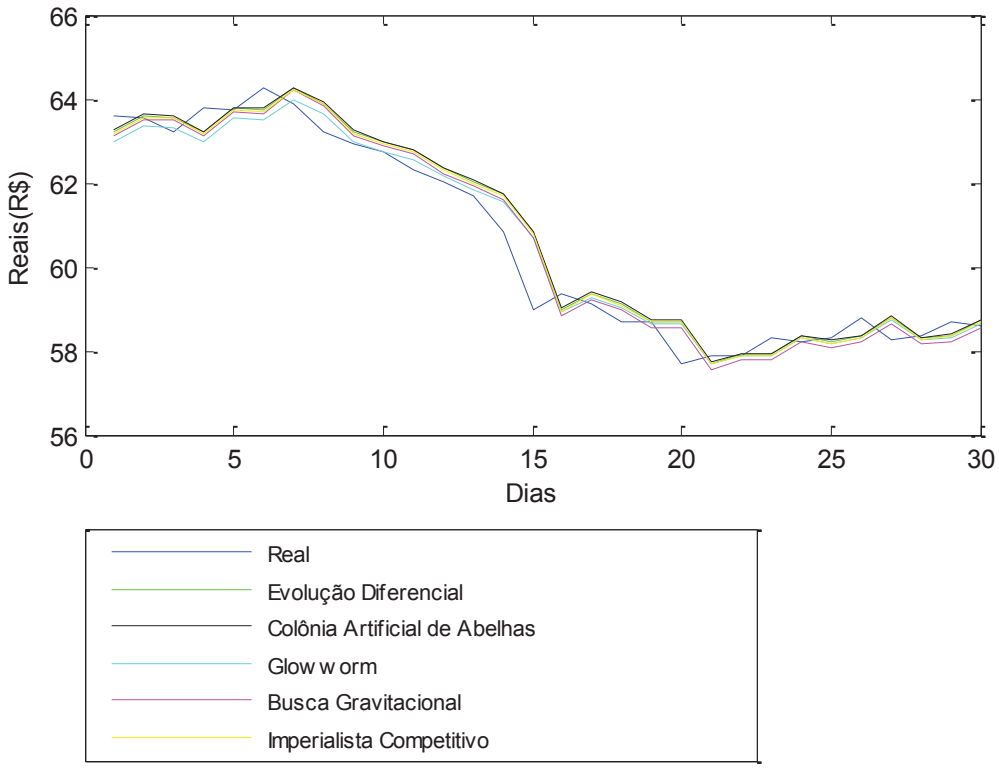
porém avaliando a previsão dos métodos a um passo à frente. Os métodos de Evolução Diferencial, Colônia Artificial de Abelhas e o Imperialista Competitivo mantiveram como a melhor configuração as mesmas da etapa de treinamento ou validação. O método *Glowworm* teve como melhor configuração 1 entrada e 10 neurônios e o método de Busca Gravitacional com 1 entrada e 80 neurônios.

Quadro 5.6 - Resultados de 1 previsão à frente dos métodos para o preço da saca de soja.

1 Previsão à frente						
Resultados do Algoritmo de Evolução Diferencial						
MSE		Neurônios				
		5	10	20	40	80
Entradas	1	0,31684	0,34336	0,42686	0,50043	26,67893
	2	0,38198	1,87719	8,24931	7,66998	10,63125
	3	0,79824	2,13710	0,58293	11,66890	16,77049
	4	0,98948	4,20754	1,93645	4,91986	310,81405
	5	0,76045	5,73024	1,23000	8,44869	62,66701
Resultados do Algoritmo de Colônia Artificial de Abelhas						
Entradas	1	0,294758	0,311845	0,300174	0,370722	0,297575
	2	0,530188	0,471811	0,504971	0,589500	4,288820
	3	0,749754	0,674861	1,008253	10,100379	1,410056
	4	0,672201	0,886260	2,799301	1,648473	21,608703
	5	0,659228	2,660894	3,023513	5,598475	4,045865
Resultados do Algoritmo Glowworm						
Entradas	1	0,30595	0,25460	1,90639	1,50047	35,79838
	2	0,54259	3,28320	102,41196	116,78658	909,04108
	3	87,13009	0,65918	20,05656	49,97890	1862,10664
	4	7,21358	156,37629	421,41448	2452,88068	39280,56896
	5	0,91154	767,13531	47,41797	4042,09847	20833,76082
Resultados do Algoritmo de Busca Gravitacional						
Entradas	1	5,09077	0,55229	0,29746	0,36964	0,25925
	2	1,75753	2,56058	0,33118	0,51289	1,27546
	3	2,61274	2,15583	1,46171	0,52202	6,49557
	4	2,90828	2,30864	2,44640	10,13115	2,23679
	5	1,89053	3,97940	4,27646	10,15834	9,46785
Resultados do Algoritmo Imperialista Competitivo						
Entradas	1	0,28411	0,27988	0,27473	0,28930	0,28051
	2	0,41314	0,46819	0,46985	0,36777	0,94381
	3	0,52640	0,60745	0,63059	0,43379	2,31004
	4	0,74077	1,05628	2,15011	1,12926	4,11022
	5	1,09387	0,95969	0,83034	0,85778	6,34460

As melhores configurações dessa etapa estão expressas graficamente pela Figura 5.2. O métodos de Evolução Diferencial teve um MAPE igual 0,7004, a Colonia Artificial de Abelhas teve um MAPE igual a 0,6618, *Glowworm* teve um MAPE igual a 0,5871, a Busca Gravitacional teve um MAPE igual a 0,6288 e o Imperialista Competitivo teve um MAPE igual a 0,6318.

Figura 5.2 - Melhores configurações dos métodos em 1 previsão à frente do preço da saca de soja.



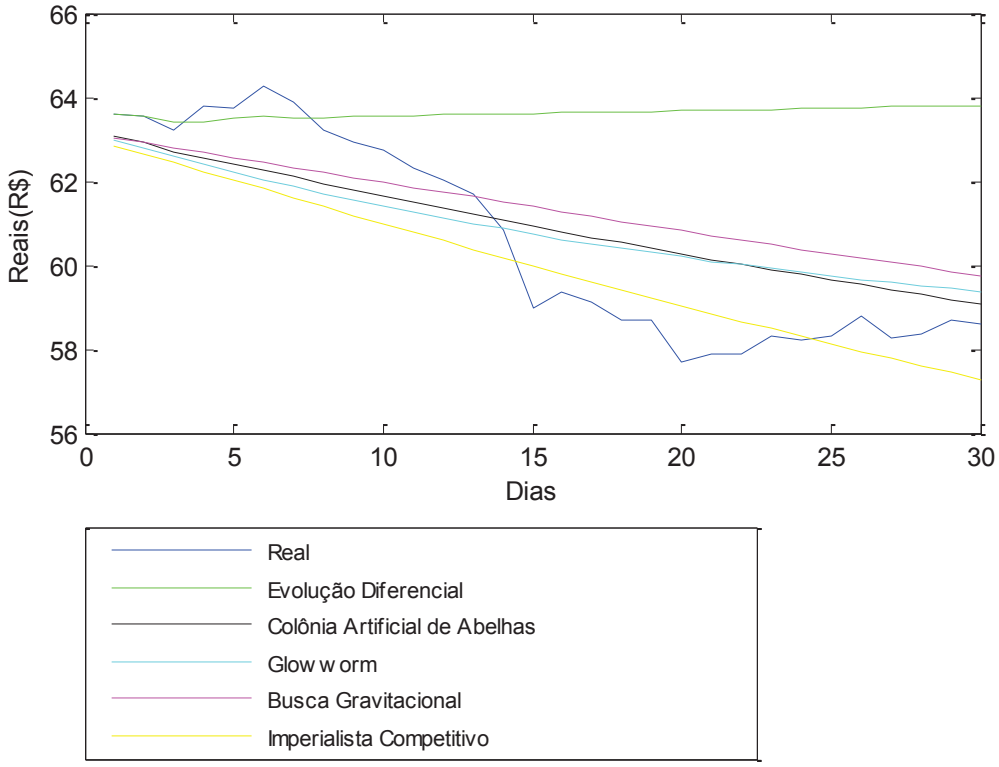
Para *n* previsões a frente, o métodos de Evolução diferencial e o Colônia Artificial de Abelhas tiveram os melhores resultados com 5 entradas e 5 neurônios, *Glowworm* com 1 entrada e 10 neurônios, Busca Gravitacional com 2 entradas e 20 neurônios e o Imperialista Competitivo com 3 entradas e 40 neurônios, conforme mostra o Quadro 5.7.

A Figura 5.3 mostra os resultados graficamente, onde Evolução Diferencial teve um MAPE igual a 3,3620, Colônia Artificial de Abelhas teve um MAPE igual a 2,0709, *Glowworm* teve um MAPE igual a 2.2236, Busca Gravitacional teve um MAPE igual a 7,3301 e Imperialista Competitivo teve um MAPE igual a 8,0552.

Quadro 5.7 - Resultados de *n* previsão à frente dos métodos para o preço da saca de soja.

30 Previsões à frente						
Resultados do Algoritmo de Evolução Diferencial						
MSE		Neurônios				
		5	10	20	40	80
Entradas	1	38,29566	7,01892	35,46257	59,21867	194,58260
	2	8,23137	96,92196	329,20834	165,25011	761,66762
	3	45,47680	143,55220	7,56541	12,76178	312,30200
	4	40,60386	317,82194	204,54164	1948,99225	89,88513
	5	5,786342714	236,5572622	84,32837995	202,3364527	7568,64457
Resultados do Algoritmo de Colônia Artificial de Abelhas						
Entradas	1	16,984351	29,934390	15,978351	1462,078274	4,945982
	2	33,249103	14,495527	50,899066	45,370656	770,243805
	3	47,621287	28,403899	52,738409	235,446256	561,450622
	4	13,469301	16,125110	59,699244	26513,384975	26,375711
	5	1,910382	42,199369	260,751760	116,081595	15,824195
Resultados do Algoritmo Glowworm						
Entradas	1	10,55101	2,10079	832,44650	269323,31985	1203769,07760
	2	30,84283	167,22878	523,29912	301,83512	680,78428
	3	234,54837	51,13632	274719,47589	271,10539	1283,20806
	4	16495,33857	48374,45073	11087,72212	392,03867	410227,30169
	5	6,095540474	39867,51935	1250,103604	1290,499032	628349,6346
Resultados do Algoritmo de Busca Gravitacional						
Entradas	1	428,50792	68,07863	12,95553	25,72654	3,72185
	2	13,46025	79,92062	2,82745	14,34149	19,04238
	3	18,18214	76,58932	100,69533	4,50302	100,71867
	4	17,82610	10,92989	65,12520	70,64723	105,00357
	5	15,05042	22,83851	65,60683	78,79966	53,29459
Resultados do Algoritmo Imperialista Competitivo						
Entradas	1	14,77499	9,98333	9,92365	12,87262	11,96763
	2	16,25993	25,80303	24,58904	9,43711	20,75300
	3	11,97176	19,28395	20,63908	1,50817	112,25545
	4	14,40652	42,67763	79,73773	25,96428	70,62621
	5	21,13996	14,85868	9,08165	11,33482	96,59517

Figura 5.3 - Melhores configurações dos métodos em *n* previsões à frente do preço da saca de soja.



5.3. *Previsão de Demanda*

Para a previsão de demanda, são realizados os mesmo teste para a previsão da saca do soja, aplicados a demanda três grupos de uma empresa do ramo alimentício, os resultados são apresentados a seguir.

5.3.1. Grupo A

Na etapa de validação dos métodos, os resultados estão apresentados no Quadro 5.8, onde o método de Evolução Diferencial teve melhor resultado com 1 entrada e 10 neurônios, Colônia Artificial de Abelhas teve com 1 entrada e 40 neurônios, *Glowworm* teve os melhores resultados com 1 entrada e 10 neurônios, Busca gravitacional com 1 entrada e 40 neurônios e o Imperialista Competitivo com 1 entrada e 5 neurônios.

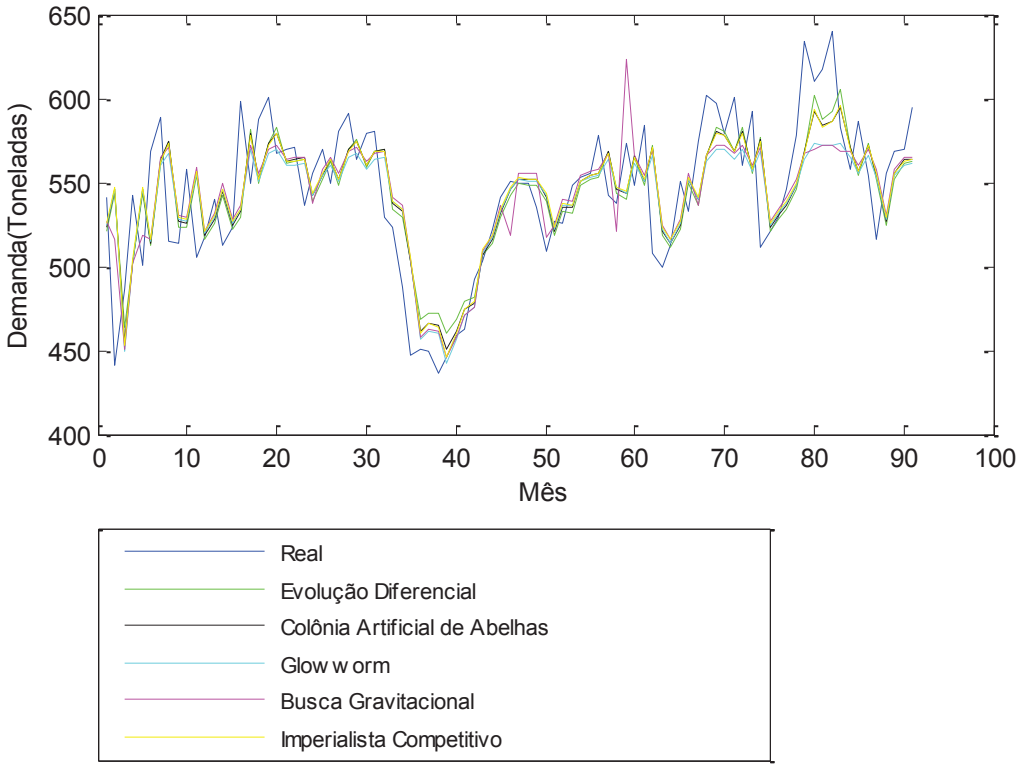
As melhores configurações estão apresentadas pela Figura 5.4, na qual Evolução Diferencial teve um MAPE de 4,4814, evoluiu até a época 252 e demorou aproximadamente 2,1 minutos o concluir o treinamento, Colônia Artificial de Abelhas teve

Quadro 5.8 - Resultados de validação dos métodos para grupo A.

Validação						
Resultados do Algoritmo de Evolução Diferencial						
MSE		Neurônios				
		5	10	20	40	80
Entradas	1	917,22311	907,42378	910,89330	1122,19147	1031,89477
	2	958,59464	957,27167	1017,89580	1092,54730	1222,08443
	3	975,24160	992,92014	1017,75002	1198,51148	3725,86521
	4	1154,05994	1149,24885	1206,84107	1376,27867	1289,58856
	5	1330,04934	1375,43128	1380,03221	1730,67539	1392,66653
Resultados do Algoritmo de Colônia Artificial de Abelhas						
Entradas	1	891,63840	893,16259	892,66965	891,55262	903,98988
	2	948,65109	950,41764	949,25409	953,51662	950,56804
	3	966,04928	959,64399	965,14311	965,12539	978,22370
	4	1143,29522	1141,90242	1139,33088	1152,35140	1184,70051
	5	1312,27577	1309,54308	1311,36481	1354,39740	1393,35797
Resultados do Algoritmo Glowworm						
Entradas	1	929,85064	914,35661	919,07508	945,49624	1156,71994
	2	1046,96559	987,12676	1007,83670	1273,23277	3242,99876
	3	995,32108	999,80953	1129,15986	1034,26771	2512,60329
	4	1283,48463	1293,90554	1672,77888	1521,22363	1827,19858
	5	1388,86012	1511,74530	1842,88604	8462,80260	3762,22179
Resultados do Algoritmo de Busca Gravitacional						
Entradas	1	902,20630	897,37831	885,55576	853,77800	883,34041
	2	952,92316	958,28591	948,23077	953,66642	946,20034
	3	991,35283	979,33694	977,44784	969,19073	967,23849
	4	1144,64164	1142,94077	1143,14817	1144,87999	1144,61662
	5	1314,22609	1312,39441	1317,73186	1310,02235	1309,17174
Resultados do Algoritmo Imperialista Competitivo						
Entradas	1	887,94642	896,68865	891,97767	890,83850	889,77540
	2	952,08109	951,76414	951,75680	947,54940	946,37456
	3	965,57325	965,69747	967,96559	963,43245	965,53561
	4	1143,40545	1143,86748	1140,63191	1139,56849	1138,21780
	5	1309,85341	1309,08454	1308,99446	1311,35014	1314,95479

um MAPE de 4,3713 e a última época que o algoritmo evoluiu foi a 172 demorando 2,7 minutos o treinamento, *Glowworm* teve um MAPE de 4,3803, evoluiu até a época 804, e demorou aproximadamente 17,2 minutos o concluir o treinamento, Busca Gravitacional teve um MAPE de 4,3076, evoluiu até a época 212 e demorou aproximadamente 22,4 minutos o concluir o treinamento e o Imperialista Competitivo teve um MAPE de 4,3563, evoluiu até a última época, e demorou aproximadamente 1,1 minutos o concluir o treinamento.

Figura 5.4 - Melhores configurações dos métodos na validação do grupo A.



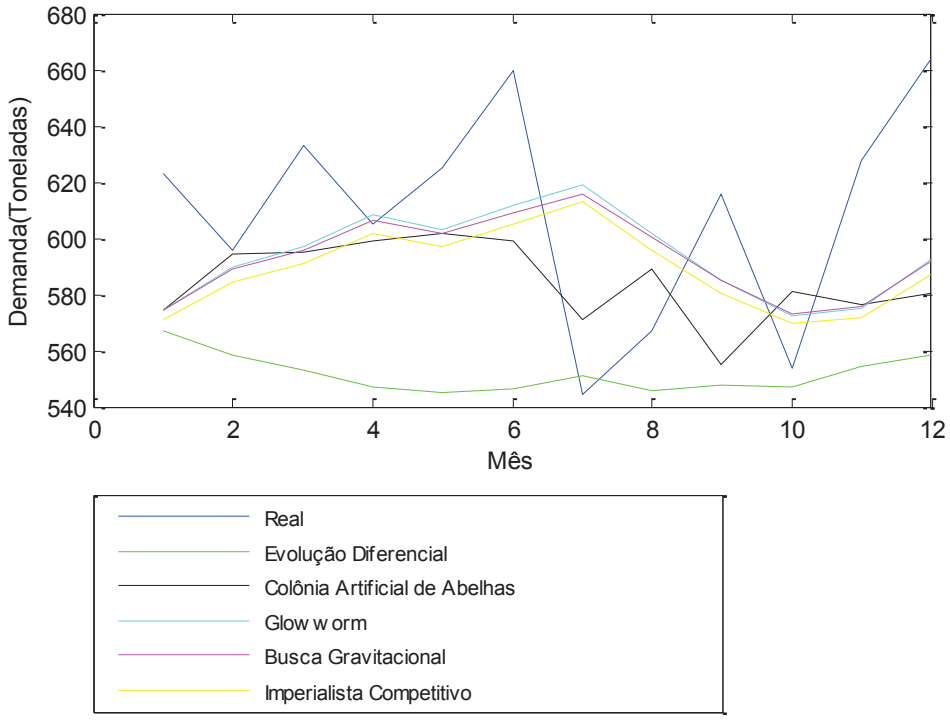
Para uma previsão a frente, todos os métodos tiveram melhores resultados com configurações diferentes da parte da validação conforme o Quadro 5.9. Evolução Diferencial teve melhor resultado com 5 entradas e 40 neurônios e um MAPE de 5,5798, Colônia Artificial de Abelhas com 2 entradas e 80 neurônios e um MAPE de 6,0386, *Glowworm* com 3 entradas e 5 neurônios e um MAPE de 6,1022, Busca Gravitacional com 3 entradas e 20 neurônios e um MAPE de 6,0832 e Imperialista Competitivo com 3 entradas e 80 neurônios com um MAPE de 6,3980. Estes resultados podem ser visualizados graficamente pela Figura 5.5.

Quadro 5.9 - Resultados de 1 previsão à frente dos métodos para grupo A.

1 Previsão à frente Resultados do Algoritmo de Evolução Diferencial						
MSE		Neurônios				
		5	10	20	40	80
Entradas	1	2710,34467	2517,15087	2928,06019	5093,93859	3269,15008
	2	2354,73831	2341,92362	2328,05784	3557,30795	2297,24206
	3	2268,89952	2092,82651	2867,90997	3357,20993	15013,08386
	4	2377,17646	2979,29901	3839,04184	3085,83176	4316,70779
	5	3604,51967	4549,71878	3810,16312	1625,33316	3468,09686

	Resultados do Algoritmo de Colônia Artificial de Abelhas					
Entradas	1	2692,77849	2545,02760	2555,62232	2537,79409	2810,57232
	2	2440,15567	2475,24816	2448,56696	2316,05382	1943,85538
	3	2088,17854	2132,79087	2100,51756	2096,46986	2058,71757
	4	2784,91907	2824,29178	2642,02142	2384,05208	2262,07800
	5	2892,42294	3021,43384	3078,84000	3906,61169	3117,83727
	Resultados do Algoritmo Glowworm					
Entradas	1	2654,27672	3109,37105	2545,73624	3135,80390	3564,97438
	2	2282,37919	2274,86464	2515,75962	5322,17445	6106,49980
	3	1865,05561	2243,52624	3154,49800	2697,01371	4992,60544
	4	4356,98419	4146,29279	5753,43660	6593,13687	7852,19613
	5	3255,15593	2905,62082	6216,64000	41431,90859	4502,88285
	Resultados do Algoritmo de Busca Gravitacional					
Entradas	1	2607,85473	2829,17694	2656,68038	2996,44952	2588,79053
	2	2339,87829	2280,27650	2400,56946	2446,15365	2390,18690
	3	1912,38668	1929,33965	1853,29690	2222,55141	2200,44286
	4	2616,28857	2682,95814	2574,81211	2784,18147	2776,23762
	5	3224,65846	2895,48127	3347,63222	3109,60349	3066,45529
	Resultados do Algoritmo Imperialista Competitivo					
Entradas	1	2570,11899	2499,38307	2509,85309	2534,90039	2542,53146
	2	2397,21238	2410,60673	2393,10879	2489,52255	2677,14635
	3	2140,03379	2179,26609	2031,88470	2111,45796	2023,88767
	4	2637,09753	2626,64143	2714,56166	2631,73634	2708,89363
	5	2937,13088	3008,65764	3032,13601	3063,55587	3009,72432

Figura 5.5 - Melhores configurações dos métodos em 1 previsão à frente do grupo A.

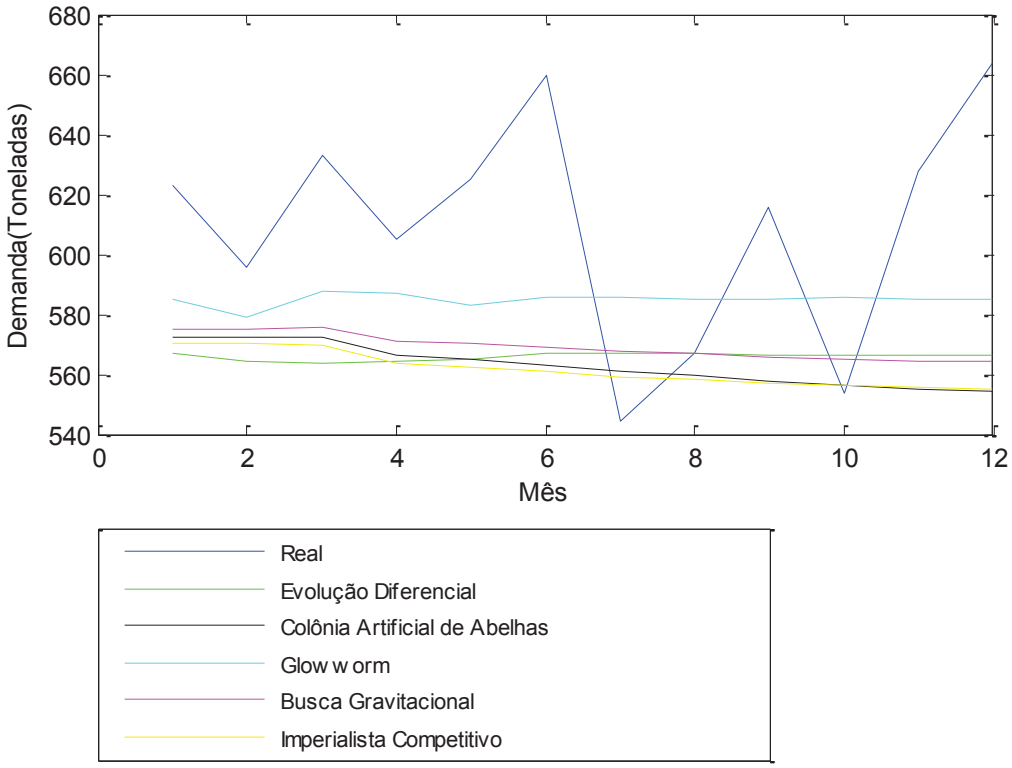


O Quadro 5.10 mostra os resultados para *n* previsões à frente. Evolução Diferencial teve melhor resultado com 5 entradas e 40 neurônios e um MAPE de 5,4643, Colônia Artificial de Abelhas com 2 entradas e 80 neurônios com um MAPE de 7,8779, *Glowworm* com 3 entradas e 5 neurônios e um MAPE de 6,4278, Busca Gravitacional com 3 entradas e 20 neurônios teve um MAPE de 7,3301 e Imperialista Competitivo com 3 entradas e 80 neurônios teve um MAPE de 8.055289. A Figura 5.6 contém esses resultados expressos graficamente.

Quadro 5.10 - Resultados de *n* previsões à frente dos métodos para grupo A.

12 Previsões à frente						
Resultados do Algoritmo de Evolução Diferencial						
MSE		Neurônios				
		5	10	20	40	80
Entradas	1	5234,58976	4695,27948	4931,31354	3862,64523	3606,38413
	2	4186,14729	4095,68941	3292,19372	4660,29960	1837,38297
	3	4288,10833	3964,00324	4413,64548	5173,97467	2532,36650
	4	3243,00003	4073,48766	3739,35130	4088,19386	6377,75038
	5	4070,24388	4005,58287	3588,55946	1570,08521	4755,80613
Resultados do Algoritmo de Colônia Artificial de Abelhas						
Entradas	1	4342,94970	4271,21871	4428,29728	4322,02728	4093,61324
	2	3966,86259	4047,32691	4160,94025	3578,60389	4059,40898
	3	3971,19639	4073,55292	3734,05253	3795,13742	3506,21771
	4	3808,45959	4140,57965	4288,05571	3976,44749	3853,75433
	5	4165,06977	4227,64369	3989,18428	4109,61404	5271,76370
Resultados do Algoritmo Glowworm						
Entradas	1	6029,59738	4626,88248	5130,10767	4195,93008	242344,31394
	2	2503,36396	3554,47132	5074,10980	4630,73537	1943,94316
	3	3446,75689	4052,90966	2897,25378	3437,65270	6211,42245
	4	5751,64721	4593,73115	3952,23489	3663,78113	4840,04790
	5	5110,95262	5620,17782	3663,57895	4029,16014	7334,67761
Resultados do Algoritmo de Busca Gravitacional						
Entradas	1	4932,45317	5058,16503	4204,06969	3782,04617	4253,09786
	2	3786,04983	3523,44004	4587,79587	3904,72810	4281,42859
	3	3112,60520	2954,78121	3228,18002	3574,86453	3606,60898
	4	3837,71956	3849,64648	3754,94250	3834,26039	3836,28282
	5	4331,79058	4118,89117	4106,05351	4061,67789	4106,73145
Resultados do Algoritmo Imperialista Competitivo						
Entradas	1	4246,87009	4450,47322	4190,75493	4115,49991	4166,50624
	2	4017,99338	3982,25458	4057,08029	4338,83745	4476,59005
	3	3626,49721	3626,48804	3664,97338	4147,68301	3744,12782
	4	3844,13683	3830,97921	4317,62901	4264,76326	4028,64661
	5	4235,77993	4164,33920	4131,21013	4011,48857	4447,62534

Figura 5.6 - Melhores configurações dos métodos em *n* previsões à frente do grupo A.



5.3.2. Grupo B

Na etapa de validação dos métodos para o grupo B, os resultados estão expressos no Quadro 5.11. O método de Evolução Diferencial teve melhor resultado com 3 entradas e 20 neurônios, Colônia Artificial de Abelhas e *Glowworm* tiveram com 3 entradas e 10 neurônios, Busca gravitacional com 3 entradas e 80 neurônios e o Imperialista Competitivo com 3 entradas e 40 neurônios.

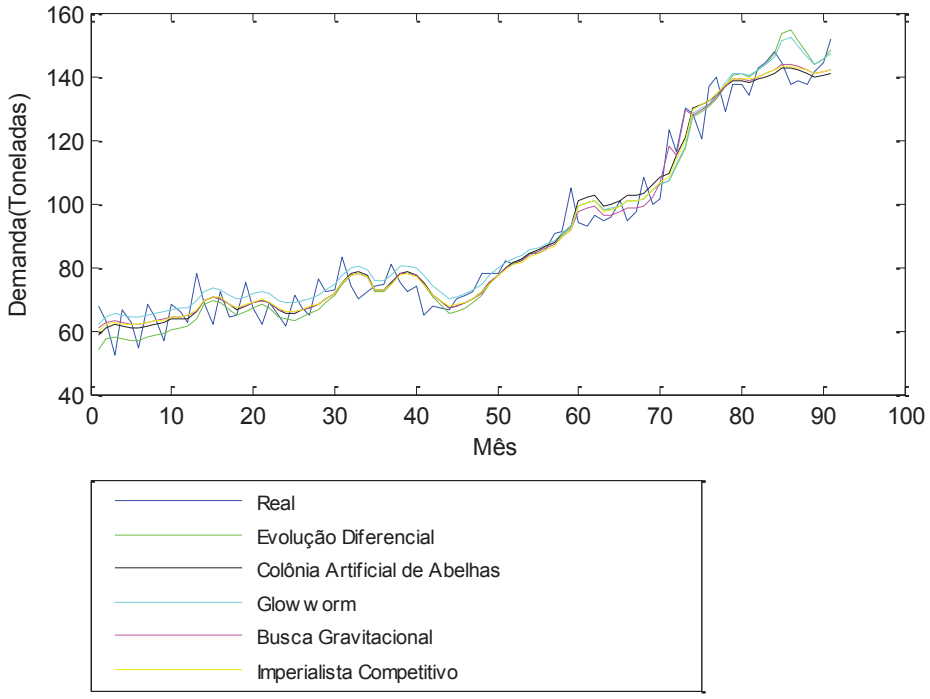
A Figura 5.7 apresenta os resultados dessas configurações graficamente. O algoritmo de Evolução Diferencial teve um MAPE de 5,1077 e a última época que o algoritmo evoluiu foi a 733 demorando aproximadamente 2,8 minutos o treinamento, Colonia Artificial de Abelhas teve um MAPE de 5,1604 e a última época que o algoritmo evoluiu foi a 705 demorando aproximadamente 1,6 minutos para concluir o treinamento, *Glowworm* teve um MAPE de 5,7628 e a última época que o algoritmo evoluiu foi a 50 demorando aproximadamente 17,5 minutos o treinamento, Busca Gravitacional teve um MAPE de 4,7097, o algoritmo evoluiu até a última época e demorou aproximadamente 39,1 minutos o treinamento e Imperialista Competitivo teve um MAPE de 5,0390, a última

época que o algoritmo evoluiu foi a 991 demorando aproximadamente 2,3 minutos no treinamento.

Quadro 5.11 - Resultados de validação dos métodos para grupo B.

Validação						
Resultados do Algoritmo de Evolução Diferencial						
MSE		Neurônios				
		5	10	20	40	80
Entradas	1	48,78677	48,50391	49,45956	58,15074	71,91326
	2	41,34015	41,19822	54,22651	44,03236	66,63813
	3	30,06838	31,59634	29,48280	55,33894	209,05459
	4	37,15425	38,34629	51,47415	200,44930	300,37119
	5	39,79303	40,64174	46,48591	77,18906	2024,02099
Resultados do Algoritmo de Colônia Artificial de Abelhas						
Entradas	1	47,64004	46,03038	43,98269	47,05768	44,05049
	2	41,01207	39,00876	37,97196	40,06270	40,29130
	3	29,01256	28,65911	29,40372	29,80774	33,48501
	4	33,85717	35,04993	34,81798	35,80955	35,67919
	5	35,01839	34,41668	36,04447	40,60255	96,07638
Resultados do Algoritmo Glowworm						
Entradas	1	49,50955	49,06802	57,58982	108,28136	237,04971
	2	43,22444	39,36665	72,24895	52,02530	1531,77249
	3	83,15367	35,66241	61,05317	324,89918	578,82128
	4	48,74228	194,08781	137,86455	142,49390	20421,68306
	5	195,30290	670,50331	125,45225	285,09355	6558,20178
Resultados do Algoritmo de Busca Gravitacional						
Entradas	1	43,86054	44,48712	42,76233	36,34926	38,38251
	2	38,57676	36,50112	36,15300	35,21234	34,71469
	3	29,75230	28,63215	28,45866	27,76568	23,77951
	4	33,41795	33,39978	33,19119	33,18076	32,89036
	5	35,50755	33,84991	34,14614	33,93720	33,51478
Resultados do Algoritmo Imperialista Competitivo						
Entradas	1	42,09909	47,26512	44,29780	46,61889	43,57398
	2	37,20799	38,33256	36,89226	39,79178	40,18353
	3	28,24751	30,10939	27,91899	27,80595	28,49821
	4	34,42885	36,68530	33,98983	38,08397	35,68268
	5	33,89372	33,75362	36,71699	36,33569	38,84223

Figura 5.7 - Melhores configurações dos métodos na validação do grupo B.



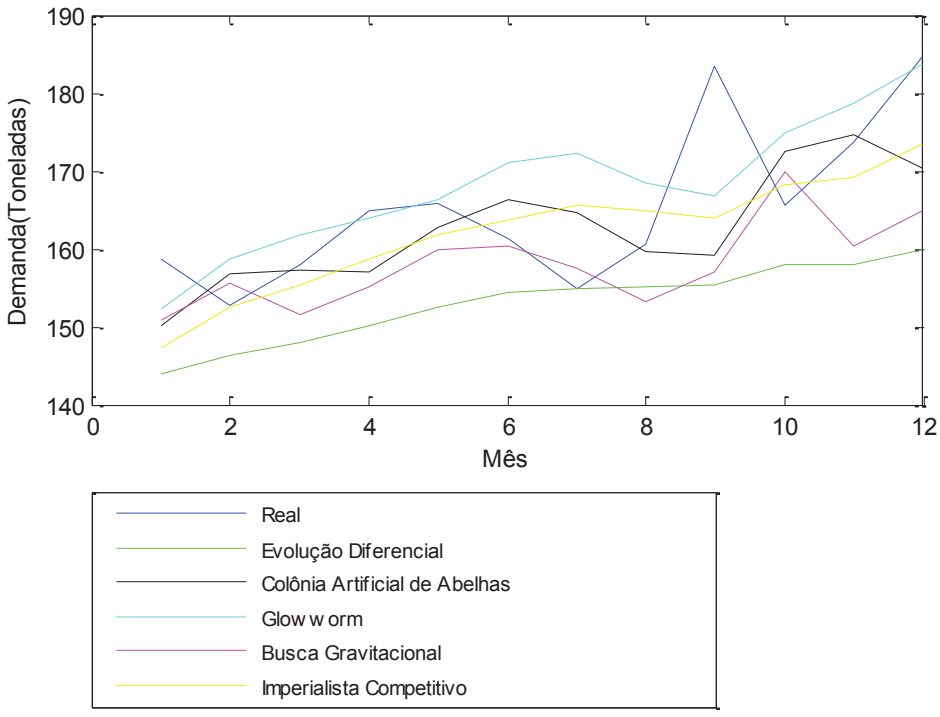
Para uma previsão a frente do grupo B, o único método que teve o melhor resultado na validação e com 1 previsão com a mesma configuração foi o algoritmo de *Glowworm, os demais* métodos tiveram melhores resultados com configurações diferentes da parte da validação conforme o Quadro 5.9. Evolução Diferencial teve melhor resultado com 5 entradas e 20 neurônios e um MAPE de 3,8340, Colônia Artificial de Abelhas com 2 entradas e 5 neurônios e um MAPE de 4,2261, *Glowworm* com 3 entradas e 10 neurônios e um MAPE de 4,2666, Busca Gravitacional com 1 entrada e 10 neurônios e um MAPE de 5,2151 e Imperialista Competitivo com 4 entradas e 40 neurônios com um MAPE de 3,9018. Estes resultados podem ser visualizados graficamente pela Figura 5.8.

Quadro 5.12 - Resultados de 1 previsão à frente dos métodos para grupo B.

1 Previsão à frente Resultados do Algoritmo de Evolução Diferencial						
MSE		Neurônios				
		5	10	20	40	80
Entradas	1	128,93060	110,24753	110,57391	418,10374	421,66053
	2	96,67942	93,45317	101,75033	100,98362	819,47689
	3	178,90454	187,57552	405,98742	781,47462	1425,23265
	4	168,12563	235,18575	61,79256	31200,03114	2810,00717
	5	76,56270	180,05404	60,29010	19537,15731	40249,28271

	Resultados do Algoritmo de Colônia Artificial de Abelhas					
Entradas	1	98,12747	257,62024	204,72835	97,03795	141,53756
	2	93,15020	4886,92355	206,86497	2409,90826	2907,52212
	3	1328,56298	769,27954	407,30242	136,94643	173,18996
	4	957,20862	2495,86236	3852,90978	276,45050	2219,95336
	5	183,38124	1259,51420	916,01951	1175,86412	12731,83099
	Resultados do Algoritmo Glowworm					
Entradas	1	103,92141	104,09297	305,70281	493,58943	2036,53288
	2	124,74512	245,08256	133,73529	737,83887	17692,84758
	3	830,50940	77,46719	1143,16958	7642,85257	247914,44895
	4	176,90474	4503,45346	3738,13003	4763,77489	107711,28075
	5	93559,92278	12874,40351	1089,66819	245,55037	231585,80589
	Resultados do Algoritmo de Busca Gravitacional					
Entradas	1	144,41312	131,82948	557,99385	1160,69444	1191,99613
	2	141,46845	861,69589	447,09282	1268,45350	346,12881
	3	470,37419	380,05502	370,41776	541,59003	666,12942
	4	324,75800	335,49475	580,39150	540,64293	542,34288
	5	569,68774	430,25982	453,01566	481,38480	565,49899
	Resultados do Algoritmo Imperialista Competitivo					
Entradas	1	98,57231	99,92384	632,69406	115,99833	277,88118
	2	114,90375	263,27359	700,68721	3224,58018	204,49771
	3	230,31817	629,68578	532,04447	1045,78901	832,47465
	4	302,16931	89,81983	473,86753	71,42753	4404,41443
	5	504,41785	949,89855	206,22001	712,36336	409,85421

Figura 5.8 - Melhores configurações dos métodos em 1 previsão à frente do grupo B.

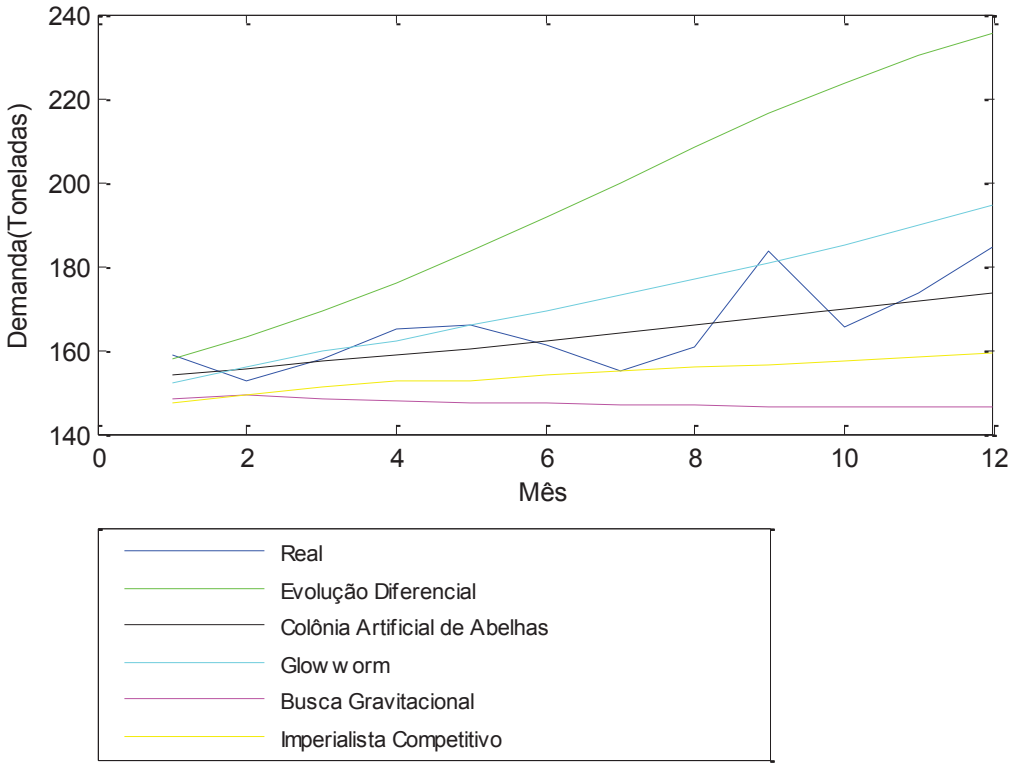


O Quadro 5.13 mostra os resultados para *n* previsões à frente. Evolução Diferencial teve melhor resultado com 1 entrada e 10 neurônios e um MAPE de 3,1308, Colônia Artificial de Abelhas com 1 entrada e 5 neurônios com um MAPE de 3,3104, *Glowworm* com 3 entradas e 10 neurônios e um MAPE de 5,2581, Busca Gravitacional com 2 entradas e 5 neurônios teve um MAPE de 10,5518 e Imperialista Competitivo com 4 entradas e 40 neurônios teve um MAPE de 6,5665. A Figura 5.9 contém esses resultados expressos graficamente.

Quadro 5.13 - Resultados de *n* previsões à frente dos métodos para grupo B.

12 Previsões à frente						
Resultados do Algoritmo de Evolução Diferencial						
MSE		Neurônios				
		5	10	20	40	80
Entradas	1	1747,99805	42,88686	760,60020	649,41606	6535967,85553
	2	324,99123	233,14871	118,13923	236,84952	1378550,56478
	3	543,70663	703,01043	397,93818	242233,47238	45187,58577
	4	507,83724	565,47370	55,93136	2545395,08964	298,85380
	5	175,05646	521,98294	55,05270	512,02717	3466356,30297
Resultados do Algoritmo de Colônia Artificial de Abelhas						
Entradas	1	49,54925	557,49873	475,89076	306,73439	840,47190
	2	216,20635	686,74302	594,06185	684,45547	687,50174
	3	737,55910	690,62913	317,28486	726,73638	830,32066
	4	676,29924	803,54832	654,23852	528,23379	637,20792
	5	450,47385	695,66823	726,14636	395,78640	2743116,55496
Resultados do Algoritmo Glowworm						
Entradas	1	453,95545	475,87249	567785,27138	5407,77506	6476456,03084
	2	882,21176	552,50730	197,40546	817,85433	1521830,32078
	3	169450,27092	120,59700	880527,15068	2446,31313	961094,53537
	4	2851,07234	636315,28926	2801942,74024	384,56423	5363314,25517
	5	33831,62666	2001,88281	1367,78819	3703,85120	9767,01824
Resultados do Algoritmo de Busca Gravitacional						
Entradas	1	520,98619	517,68314	726,70140	833,89892	667,14044
	2	434,65249	650,35049	654,35382	626,27555	680,45747
	3	719,25061	651,65053	644,23917	630,42338	611,62043
	4	562,99630	571,62960	635,40375	630,49837	662,49020
	5	711,08462	600,64937	627,70136	638,39668	624,96414
Resultados do Algoritmo Imperialista Competitivo						
Entradas	1	200,88144	372,94148	788,31928	527,98616	688,70828
	2	393,81260	634,85779	694,82413	798,04682	720,53800
	3	525,93975	703,95294	587,07982	624,56920	611,25736
	4	450,43254	272,84430	548,59510	187,06567	729,93756
	5	591,21534	663,96732	410,46114	555,96779	845,61002

Figura 5.9 - Melhores configurações dos métodos em *n* previsões à frente do grupo B.



5.3.3. Grupo C

O Quadro 5.14 apresenta os resultados da etapa de validação dos métodos para o grupo C. O método de Evolução Diferencial e *Glowworm* tiveram os melhores resultados com 1 entrada e 5 neurônios, Colônia Artificial de Abelhas com 1 entrada e 10 neurônios, Busca gravitacional com 1 entrada e 80 neurônios e o Imperialista Competitivo com 1 entrada e 10 neurônios.

A Figura 5.10 apresenta os resultados dessas configurações graficamente. O algoritmo de Evolução Diferencial teve um MAPE de 5,7444 e a última época que o algoritmo evoluiu foi a 902 demorando aproximadamente 1,8 minutos o treinamento, Colonia Artificial de Abelhas teve um MAPE de 5,6086 e a última época que o algoritmo evoluiu foi a 918 demorando aproximadamente 1,7 minutos o treinamento, *Glowworm* teve um MAPE de 5,7108 e a última época que o algoritmo evoluiu foi a 642 demorando aproximadamente 17,8 minutos o treinamento, Busca Gravitacional teve um MAPE de 5,2093, o algoritmo evoluiu até a última época e demorou aproximadamente 38,6 minutos no treinamento e Imperialista Competitivo teve um MAPE 5,6300 e a última época que o algoritmo evoluiu foi a 988 demorando aproximadamente 1,2 minutos no treinamento.

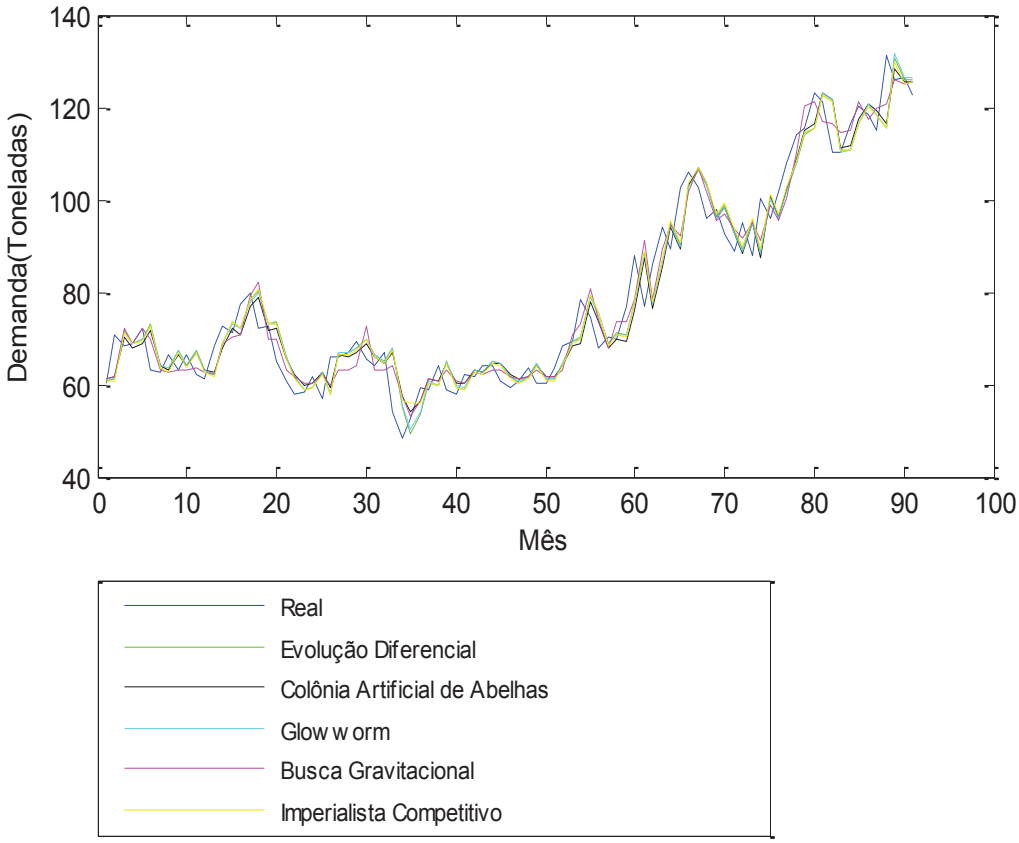
Quadro 5.14 - Resultados de validação dos métodos para grupo C.

Validação						
Resultados do Algoritmo de Evolução Diferencial						
MSE	Neurônios					
		5	10	20	40	80
Entradas	1	30,37900	30,76826	30,47504	38,70934	33,58815
	2	33,56725	33,71093	36,18982	34,55090	58,64065
	3	36,80693	37,46304	36,46238	102,21017	129,25437
	4	43,17572	44,47339	47,36638	67,70405	212,80058
	5	49,22489	54,06734	64,86415	123,59926	175,26701
Resultados do Algoritmo de Colônia Artificial de Abelhas						
Entradas	1	30,29841	29,55620	29,90982	30,01481	29,80813
	2	31,72036	31,59708	32,39334	31,47121	31,13442
	3	35,22105	35,19378	34,85442	35,08523	35,83209
	4	41,83748	42,62929	42,58251	42,92319	58,15323
	5	48,67398	48,02885	49,44661	49,46371	69,41720
Resultados do Algoritmo Glowworm						
Entradas	1	30,37720	30,80300	45,57060	39,30808	175,06275
	2	40,79844	33,57816	51,42121	108,08512	121,08190
	3	39,34163	39,16995	155,97141	1246,59379	4642,91453
	4	46,71906	149,16626	97,04485	127,91942	2362,63215
	5	102,90804	54,79990	336,54970	398,53287	4037,38170
Resultados do Algoritmo de Busca Gravitacional						
Entradas	1	31,14332	29,75086	29,01055	26,53011	23,54619
	2	35,06330	30,74859	30,51934	30,04120	30,44145
	3	40,86355	35,72041	34,30922	34,51165	34,12974
	4	47,40053	41,68726	42,20518	42,51502	41,91076
	5	47,41296	48,20947	48,92361	49,06700	47,74815
Resultados do Algoritmo Imperialista Competitivo						
Entradas	1	30,05823	29,70281	30,23990	30,69779	29,93257
	2	31,67038	32,41803	32,06932	32,47153	30,66653
	3	35,43327	35,64956	35,87409	34,67599	36,21050
	4	42,88168	42,77391	41,78248	42,97208	42,59519
	5	47,57727	46,72990	47,23087	47,69227	47,55640

Para uma previsão a frente do grupo C, todos métodos tiveram melhores resultados com configurações diferentes da parte da validação conforme o Quadro 5.15. Evolução Diferencial teve melhor resultado com 5 entradas e 20 neurônios e um MAPE de 6,4121, Colônia Artificial de Abelhas com 3 entradas e 20 neurônios e um MAPE de 6,0599, *Glowworm* com 5 entradas e 10 neurônios e um MAPE de 6,2706, Busca Gravitacional com 3 entradas e 10 neurônios e um MAPE de 6,0546 e Imperialista

Competitivo com 3 entradas e 40 neurônios com um MAPE de 6,3265. Estes resultados podem ser visualizados graficamente pela Figura 5.11.

Figura 5.10 - Melhores configurações dos métodos na validação do grupo C.

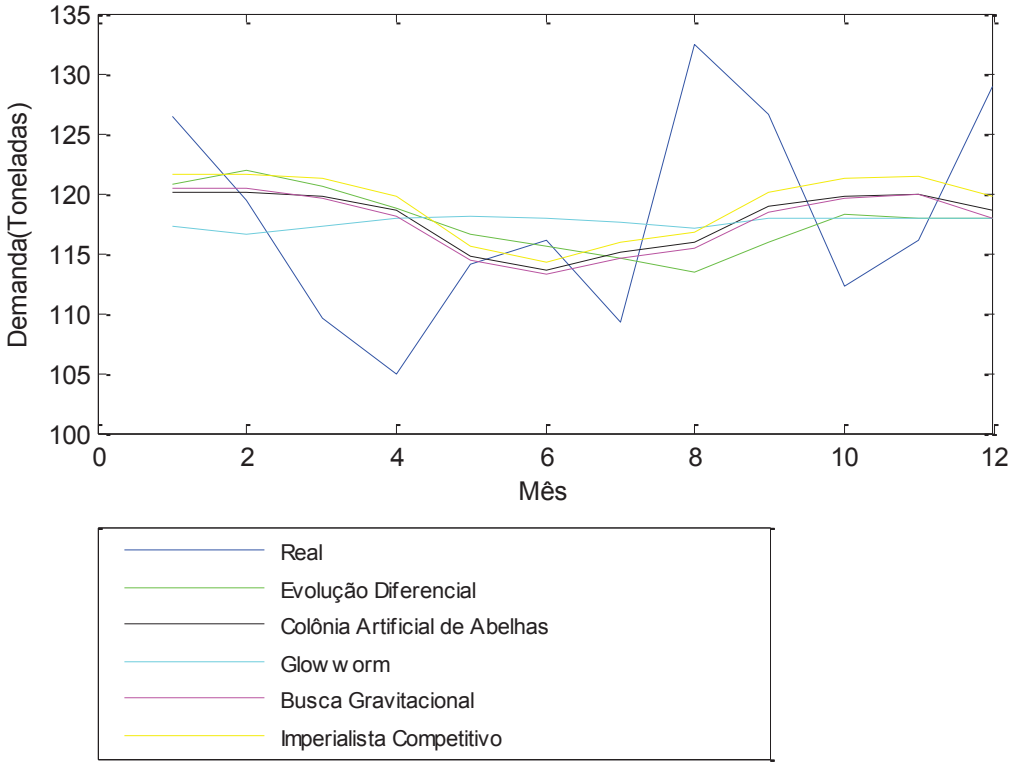


Quadro 5.15 - Resultados de 1 previsão à frente dos métodos para grupo C.

1 Previsão à frente						
Resultados do Algoritmo de Evolução Diferencial						
MSE		Neurônios				
		5	10	20	40	80
Entradas	1	101,64341	109,43081	103,90713	137,44321	115,27224
	2	129,74321	128,54945	122,92268	120,43924	362,80902
	3	99,56352	95,66693	86,63263	135,09389	426,95227
	4	95,55736	89,57722	84,45621	102,70634	576,86441
	5	119,74994	99,31858	73,97001	359,32326	349,14951
Resultados do Algoritmo de Colônia Artificial de Abelhas						
Entradas	1	101,88256	95,34822	94,39111	99,73400	85,97544
	2	91,28969	105,12701	105,51169	98,06901	84,33323
	3	82,16252	89,80795	73,37022	85,27974	74,01554
	4	86,23000	88,85826	97,38100	79,37385	110,43098
	5	106,78520	107,97366	127,20774	82,36941	178,77716

	Resultados do Algoritmo Glowworm					
Entradas	1	106,64343	114,26873	90,80070	137,50873	411,04706
	2	95,80331	97,38381	161,82056	178,17214	290,68929
	3	90,95013	131,63816	893,37977	3149,10550	3866,47650
	4	151,05298	479,71687	476,80313	594,76294	297,42367
	5	274,68901	72,44415	435,47294	237,70489	4804,28412
	Resultados do Algoritmo de Busca Gravitacional					
Entradas	1	92,79837	91,74881	89,99201	79,79580	103,45273
	2	102,88295	83,44935	88,33556	76,58329	82,96880
	3	84,61001	74,32563	81,72824	78,97983	85,96180
	4	147,62545	83,57195	77,32439	77,97197	78,86489
	5	93,04901	82,07043	79,38306	79,08255	86,70014
	Resultados do Algoritmo Imperialista Competitivo					
Entradas	1	97,54126	98,07392	103,53533	99,00351	95,09778
	2	113,36739	100,98173	110,86678	110,73645	90,10367
	3	88,11168	96,58351	75,76325	75,65256	85,48322
	4	102,15269	96,51735	83,74778	95,62886	104,38086
	5	96,79423	114,24776	142,69230	142,57346	141,94698

Figura 5.11 - Melhores configurações dos métodos em 1 previsão à frente do grupo C.

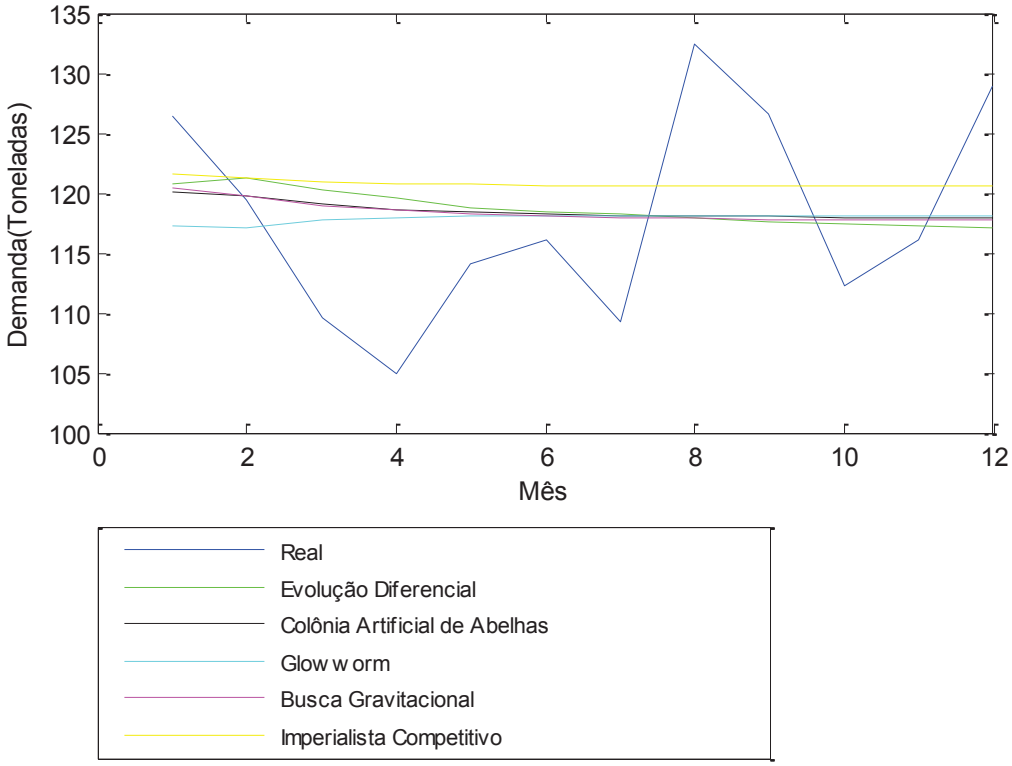


O Quadro 5.17 mostra os resultados para *n* previsões à frente do grupo C. Evolução Diferencial teve melhor resultado com 5 entradas e 20 neurônios e um MAPE de 6,4922, Colônia Artificial de Abelhas com 3 entradas e 80 neurônios com um MAPE de 6,1132, *Glowworm* com 5 entradas e 10 neurônios e um MAPE de 6,2610, Busca Gravitacional com 3 entradas e 10 neurônios teve um MAPE de 6,0729 e Imperialista Competitivo com 3 entradas e 80 neurônios teve um MAPE de 6,8699. A Figura 5.12 contém esses resultados expressos graficamente

Quadro 5.16 - Resultados de *n* previsões à frente dos métodos para grupo C.

12 Previsões à frente						
Resultados do Algoritmo de Evolução Diferencial						
MSE		Neurônios				
		5	10	20	40	80
Entradas	1	80,57055	102,64223	80,68323	4655,65900	412,70177
	2	876,25303	625,35876	210,97095	100,76948	61590,62243
	3	201,59742	137,59597	97,48756	223,81699	180,47380
	4	149,07904	102,75863	84,08907	136,51187	393,11365
	5	237,27520	104,29400	74,99532	137941,65630	701,60660
Resultados do Algoritmo de Colônia Artificial de Abelhas						
Entradas	1	84,05939	93,61609	89,47687	126,94453	104,51787
	2	86,26094	106,39904	91,25031	81,44489	73,89119
	3	80,13649	115,86082	71,51967	100,43290	70,89900
	4	90,92064	107,94595	147,72644	79,50259	271,71314
	5	137,44709	142,09648	209,83626	77,60265	131,98017
Resultados do Algoritmo Glowworm						
Entradas	1	106,94875	536,63928	78,88150	1684,72364	1813,72144
	2	123,96123	103,63887	4617,03120	312,13878	174,88676
	3	128,65882	1262,23317	39844,05460	1211,30003	3160756,35269
	4	1429,57139	101069,49463	3163,91525	111469,93968	3168,03682
	5	152,93877	71,17508	264,08394	5987,12550	528527,17601
Resultados do Algoritmo de Busca Gravitacional						
Entradas	1	76,64338	74,74454	75,01644	76,28851	72,09633
	2	129,58318	83,74422	80,26113	78,36645	85,50194
	3	90,34719	71,00811	86,24897	83,39973	97,68690
	4	607,58825	92,66359	76,72839	78,78387	79,54569
	5	98,20976	76,62858	73,22980	72,90614	84,42829
Resultados do Algoritmo Imperialista Competitivo						
Entradas	1	78,09162	78,90398	85,73670	78,73490	76,45483
	2	177,39605	86,90540	153,06311	134,84363	85,82621
	3	106,52724	173,01655	79,55618	77,89399	81,24194
	4	193,42493	153,44257	92,00207	130,65074	357,08487
	5	107,33463	259,19599	4272,28983	42766,53488	73075,75404

Figura 5.12 - Melhores configurações dos métodos em *n* previsões à frente do grupo C.



5.4. *Previsões*

Neste capitulo contém as previsões realizadas pela WNN com cada método de treinamento. O Quadro 5.17 mostra os resultados obtidos para a previsão do preço da saca de soja dos melhores métodos e configurações apresentados acima para à previsão de 30 dias à frente. Pelo Quadro 5.17 o melhor método foi ICA, o maior erro foi de R$2,44, o segundo melhor resultado foi ABC com o maior erro na previsão de R$2,57, o terceiro método foi *Glowworm* sendo o maior erro R$2,50, o quarto método foi GSA com o maior erro de R$3,12 e o último método foi ED com o maior erro de R$ 5,99. Todas as previsões são potencialmente boas, pois segundo Lewis (1997 apud Veiga, C. P., 2014), as previsões com MAPE abaixo de 10% são potencialmente boas.

Quadro 5.17 – Resultado das 30 previsões a frente das melhores configurações de cada método para previsão do preço da saca de soja.

Saca de Soja		ED		ABC		Glowworm		GSA		ICA	
Período (dias)	Demanda Real	Previsão	Erro	Previsão	Erro	Previsão	Erro	Previsão	Erro	Previsão	Erro
1	63,59	63,60	-0,01	63,09	0,50	62,99	0,60	63,02	0,57	62,82	0,77
2	63,56	63,54	0,02	62,93	0,63	62,78	0,78	62,94	0,62	62,64	0,92
3	63,20	63,42	-0,22	62,69	0,51	62,58	0,62	62,80	0,40	62,46	0,74
4	63,77	63,43	0,34	62,55	1,22	62,39	1,38	62,69	1,08	62,22	1,55
5	63,73	63,48	0,25	62,43	1,30	62,21	1,52	62,57	1,16	62,02	1,71
6	64,25	63,53	0,72	62,27	1,98	62,03	2,22	62,45	1,80	61,81	2,44
7	63,89	63,52	0,37	62,11	1,78	61,86	2,03	62,33	1,56	61,59	2,30
8	63,21	63,51	-0,30	61,95	1,26	61,70	1,51	62,21	1,00	61,39	1,82
9	62,95	63,53	-0,58	61,81	1,14	61,55	1,40	62,09	0,86	61,18	1,77
10	62,76	63,55	-0,79	61,66	1,10	61,40	1,36	61,98	0,78	60,98	1,78
11	62,33	63,57	-1,24	61,51	0,82	61,25	1,08	61,86	0,47	60,78	1,55
12	62,02	63,57	-1,55	61,36	0,66	61,12	0,90	61,74	0,28	60,58	1,44
13	61,71	63,59	-1,88	61,22	0,49	60,98	0,73	61,62	0,09	60,38	1,33
14	60,81	63,60	-2,79	61,08	-0,27	60,86	-0,05	61,50	-0,69	60,18	0,63
15	58,96	63,61	-4,65	60,94	-1,98	60,74	-1,78	61,39	-2,43	59,98	-1,02
16	59,35	63,63	-4,28	60,80	-1,45	60,62	-1,27	61,27	-1,92	59,79	-0,44
17	59,10	63,64	-4,54	60,66	-1,56	60,51	-1,41	61,16	-2,06	59,60	-0,50
18	58,69	63,65	-4,96	60,53	-1,84	60,40	-1,71	61,04	-2,35	59,40	-0,71
19	58,69	63,66	-4,97	60,39	-1,70	60,29	-1,60	60,93	-2,24	59,22	-0,53
20	57,69	63,68	-5,99	60,26	-2,57	60,19	-2,50	60,81	-3,12	59,03	-1,34
21	57,90	63,69	-5,79	60,14	-2,24	60,09	-2,19	60,70	-2,80	58,84	-0,94
22	57,90	63,70	-5,80	60,01	-2,11	60,00	-2,10	60,59	-2,69	58,66	-0,76
23	58,33	63,71	-5,38	59,89	-1,56	59,91	-1,58	60,48	-2,15	58,48	-0,15
24	58,20	63,73	-5,53	59,76	-1,56	59,83	-1,63	60,37	-2,17	58,30	-0,10
25	58,33	63,74	-5,41	59,64	-1,31	59,74	-1,41	60,27	-1,94	58,13	0,20
26	58,78	63,75	-4,97	59,53	-0,75	59,66	-0,88	60,16	-1,38	57,95	0,83
27	58,28	63,76	-5,48	59,41	-1,13	59,59	-1,31	60,06	-1,78	57,78	0,50
28	58,35	63,78	-5,43	59,30	-0,95	59,51	-1,16	59,95	-1,60	57,61	0,74
29	58,70	63,79	-5,09	59,19	-0,49	59,44	-0,74	59,85	-1,15	57,44	1,26
30	58,59	63,80	-5,21	59,08	-0,49	59,37	-0,78	59,75	-1,16	57,27	1,32
MSE		5,7863		1,9104		2,1008		2,8275		1,5082	
MAPE		3,3621		2,0710		2,2236		2,4782		1,7429	
R2		-0,0454		0,6549		0,6205		0,4892		0,7275	

O Quadro 5.18 apresenta os resultados para 12 previsões à frente do grupo A, segundo o quadro o melhor método foi ED, o maior erro foi de 97,07ton, o segundo melhor resultado foi *Glowworm* com o maior erro na previsão de 78,37ton, o terceiro

método foi GSA sendo o maior erro 99,35ton, o quarto método foi ABC com o maior erro de 109,27 e o último método foi ICA com o maior erro de 108,50ton.

Quadro 5.18 – Resultado das 12 previsões à frente das melhores configurações de cada método para previsão do Grupo A.

Grupo A		ED		ABC		Glowworm		GSA		ICA	
Período (Mês)	Demanda Real	Previsão	Erro	Previsão	Erro	Previsão	Erro	Previsão	Erro	Previsão	Erro
1	622,78	566,90	55,88	572,44	50,34	584,93	37,85	574,70	48,07	570,47	52,31
2	595,97	564,19	31,77	572,52	23,45	578,92	17,04	575,26	20,71	570,17	25,80
3	633,09	563,69	69,40	572,32	60,77	587,79	45,30	575,64	57,45	569,54	63,55
4	604,86	564,37	40,49	566,47	38,39	586,72	18,14	570,95	33,91	563,93	40,93
5	625,09	565,03	60,06	565,01	60,08	583,27	41,83	570,12	54,97	562,57	62,52
6	659,65	566,76	92,89	563,14	96,51	585,37	74,28	568,98	90,67	560,97	98,68
7	544,47	566,76	-22,29	560,80	-16,33	585,94	-41,47	567,47	-23,00	559,12	-14,65
8	567,13	566,67	0,47	559,34	7,80	584,78	-17,65	566,67	0,46	558,07	9,07
9	615,92	566,55	49,37	557,85	58,07	585,04	30,88	565,88	50,04	557,07	58,85
10	553,62	566,45	-12,84	556,45	-2,84	585,44	-31,82	565,16	-11,54	556,20	-2,58
11	627,77	566,39	61,38	555,29	72,48	585,15	42,62	564,62	63,16	555,55	72,23
12	663,48	566,41	97,07	554,21	109,27	585,10	78,37	564,13	99,35	554,98	108,50
MSE		1570,0852		3506,2177		1943,9432		2954,7812		3626,4880	
MAPE		5,4644		7,8779		6,4279		7,3301		8,0553	
R2		-0,1681		-1,6084		-0,4462		-1,1982		-1,6979	

O Quadro 5.19 apresenta os resultados para 12 previsões à frente do grupo B, segundo o quadro o melhor método foi ED, o maior erro foi de 15,08ton, o segundo melhor resultado foi do ABC com o maior erro na previsão de 15,75ton, o terceiro método foi *glowworm* sendo o maior erro 19,45ton, o quarto método foi ICA com o maior erro de 26,78ton e o último método foi GSA com o maior erro de 38,45ton.

O Quadro 5.20 apresenta os resultados para 12 previsões à frente do grupo C, segundo o quadro o melhor método foi ABC, o maior erro foi de 14,28ton, o segundo melhor resultado foi do GSA com o maior erro na previsão de 14,47ton, o terceiro método foi *glowworm* sendo o maior erro 14.31ton, o quarto método foi ED com o maior erro de 14,57ton e o último método foi ICA com o maior erro de 16,09ton.

Quadro 5.19 – Resultado das 12 previsões à frente das melhores configurações de cada método para previsão do Grupo B.

Grupo B		ED		ABC		Glowworm		GSA		ICA	
Período (Mês)	Demanda Real	Previsão	Erro	Previsão	Erro	Previsão	Erro	Previsão	Erro	Previsão	Erro
1	158,65	153,98	4,66	154,23	4,42	152,32	6,32	148,36	10,28	147,26	11,39
2	152,83	155,20	-2,36	155,65	-2,82	155,94	-3,11	149,46	3,37	149,41	3,42
3	157,97	156,53	1,43	157,16	0,81	159,72	-1,75	148,16	9,80	151,18	6,79
4	164,93	158,01	6,92	158,74	6,19	162,19	2,74	148,09	16,84	152,59	12,34
5	165,73	159,65	6,08	160,39	5,34	165,77	-0,04	147,56	18,17	152,52	13,22
6	161,38	161,47	-0,09	162,11	-0,73	169,35	-7,98	147,33	14,04	153,81	7,57
7	155,00	163,49	-8,50	163,90	-8,90	172,88	-17,88	147,04	7,96	154,89	0,10
8	160,62	165,76	-5,14	165,74	-5,12	176,80	-16,19	146,83	13,78	155,80	4,81
9	183,37	168,30	15,08	167,63	15,75	180,87	2,50	146,64	36,73	156,59	26,78
10	165,68	171,15	-5,47	169,55	-3,87	185,14	-19,45	146,48	19,20	157,59	8,10
11	173,59	174,38	-0,80	171,49	2,10	189,70	-16,11	146,34	27,24	158,51	15,08
12	184,67	178,05	6,62	173,44	11,23	194,53	-9,86	146,23	38,45	159,38	25,29
MSE		42,8869		49,5492		120,5970		434,6525		187,0657	
MAPE		3,1309		3,3104		5,2582		10,5518		6,5666	
R2		0,5594		0,4910		-0,2389		-3,4653		-0,9218	

Quadro 5.20 – Resultado das 12 previsões à frente das melhores configurações de cada método para previsão do Grupo C.

Grupo C		ED		ABC		Glowworm		GSA		ICA	
Período (Mês)	Demanda Real	Previsão	Erro	Previsão	Erro	Previsão	Erro	Previsão	Erro	Previsão	Erro
1	126,36	120,69	5,68	120,15	6,22	117,23	9,14	120,34	6,03	123,61	2,75
2	119,50	121,29	-1,79	119,67	-0,17	117,13	2,36	119,73	-0,24	122,60	-3,11
3	109,65	120,18	-10,53	119,01	-9,35	117,70	-8,04	118,98	-9,32	121,38	-11,73
4	104,97	119,54	-14,57	118,60	-13,63	117,91	-12,95	118,52	-13,56	121,05	-16,09
5	114,01	118,82	-4,81	118,40	-4,39	118,02	-4,01	118,28	-4,27	120,34	-6,33
6	116,09	118,43	-2,34	118,23	-2,15	118,04	-1,96	118,08	-1,99	119,77	-3,68
7	109,21	118,18	-8,98	118,13	-8,92	118,04	-8,83	117,95	-8,74	119,38	-10,17
8	132,35	117,85	14,49	118,06	14,28	118,04	14,31	117,87	14,47	118,98	13,36
9	126,62	117,60	9,02	118,02	8,60	118,04	8,58	117,81	8,81	118,66	7,96
10	112,21	117,39	-5,19	117,99	-5,78	118,04	-5,83	117,77	-5,57	118,41	-6,20
11	116,03	117,24	-1,21	117,97	-1,94	118,04	-2,01	117,75	-1,72	118,18	-2,15
12	128,92	117,11	11,81	117,95	10,97	118,04	10,88	117,73	11,19	117,99	10,93
MSE		74,9953		70,8990		71,1751		71,0081		76,4548	
MAPE		6,4923		6,1133		6,2610		6,0730		6,3483	
R2		-0,0645		-0,0064		-0,0103		-0,0079		-0,0853	

6. CONCLUSÃO E TRABALHOS FUTUROS

A previsão tem relevância estratégica em diversas áreas porém não faz parte da maior parte de empresas brasileira. É normal que se uma empresa utiliza de metodologias para a previsão, utilizem modelos qualitativos que possui grandes limitações, ou utilizam de ferramentas de computacionais simples e de grande dificuldade de modelagem.

Com o objetivo de encontrar uma ferramenta que consiga fornecer uma boa previsão foi testada a Rede neural de Wavelets com cinco diferentes métodos de treinamento. O teste da WNN foram utilizando dois estudo de casos reais: Previsão do preço da saca de soja que foi utilizado dados do banco de dados obtido do site do Centro de Estudos Avançados em Economia Aplicada – ESALQ/USP; e previsão de demanda de empresa de grande porte do ramo alimentício.

Uma boa previsão é uma importante ferramenta estratégia para tomada de decisão. Com a agricultura sendo de grande importância do Brasil, uma boa previsão é algo imprescindível. Do segundo estudo de caso, a previsão de demanda alimentícia torna-se interessante por serem produtos perecíveis.

A dissertação obteve resultados promissores, dois cinco métodos que foram testados não teve um que se destacou em relação aos estudos de caso, para cada estudo caso teve um melhor método e configuração de rede. Considerando para *n* previsões a frente, onde a dificuldade de previsão devido ao acumulo de erro, o método de Evolução Diferencial conseguiu ser melhor em dois testes, na previsão de demanda dos grupos A e B. Os métodos *Glowworm* e Busca Gravitacional não conseguiram obter o melhor resultado em nenhum dos testes para *n* previsões a frente, porém tiveram bons resultados também.

Outra situação importante é que os dados que seria previsto já eram conhecidos, assim buscou-se uma configuração que representasse-los melhor, encontrar essas configurações sem conhecer os dados é uma das maiores dificuldades de utilizar métodos como redes neurais, pois não há ainda uma metodologia consistente e que apresente a melhor configuração dos métodos para a série proposta reduzindo os erros de previsão, a decisão dessas configurações permanecem sob responsabilidade do usuário e a estimativa desses parâmetros tem influência direta com os resultados finais e no grau de acurácia. Esta afirmação fica clara devido ao fato que os melhores modelos de estimação, não foram os melhores modelos de previsão, isto deve ao fato que esses

modelos ficam mais sensíveis a modificação das entradas e se perdem facilmente na realização da previsão.

Vale ressaltar que não dá para afirmar qual é o melhor método dos testados e se são os melhores métodos para estas séries temporais, ainda exige novos testes e comparações empíricas para a escolha da melhor ferramenta para cada situação específica.

Como proposta para continuidade do trabalho, sugere-se um aprofundamento em cada método de otimização, buscando os melhores parâmetros para cada método, também a realização de testes com diferentes sementes de geração de números aleatórios, sendo neste trabalho todos os métodos foram testados com a mesma distribuição de números aleatórios. Uma outra linha de continuidade, é testar os métodos para diferentes funções de ativação ou até mesmo diferentes redes neurais.

7. REFERÊNCIAS

ABBASS, H. A., "*An evolutionary artificial neural networks approach for breast cancer diagnosis*". Artificial Intelligence in Medicine, v. 25, p. 265–281, 2002.

ADAMOWSKI, J., CHAN, H. F., "*A wavelet neural network conjunction model for groundwater level forecasting".* Journal of Hydrology, v. 407, p. 28-40, 2011.

AGUIRRE, L. A.; RODRIGUES, G. G. e JÁCOME, C. R. F., "*Identificação de sistemas não-lineares utilizando modelos NARMAX polinomiais - Uma revisão e novos resultados*". SBA Controle & Automação, v. 9, n. 2, p. 90-106, de 1998

AKAY, B., KARABOGA, D., "*A modified artificial bee colony algorithm for real-parameter optimization*", Information Sciences, v. 192, p. 120–142, 2012.

ALEXANDRIDIS, A., ZAPRANIS, A. D., "*Wavelet neural networks: A practical guide"*. Neural Networks, v. 42, p. 1–27, 2013.

ATASHPAZ-GARGARI, E., LUCAS, C., "*Imperialist competitive algorithm: An algorithm for optimizations inspired by imperialistic competition*". IEEE Congress on Evolutionary Computation (CEC), Singapore, p. 4661-4667, 2007.

BALANAGAMMAL, D. *et al***. "***Forecasting of agricultural scenario in Tamil Nadu – a time series analysis*", Journal of the Indian Society of Agricultural Statistics, v. 53. p. 273-286, 2000.

BASHIR, Z., EL-HAWARY, M. E., "*Short term load forecasting by using wavelet neural networks*". Electrical and Computer Engineering, 2000 Canadian Conference, Halifax, NS, v. 1, p. 163 – 166, 2000.

BASU, M., "Improved differential evolution for short-term hydrothermal scheduling", Electrical Power and Energy Systems, v. 58, p. 91–100, 2014.

BENAOUDA, D., MURTAGH, G., STARCK, J.-L., RENAUD, O., "*Wavelet-based nonlinear multiscale decomposition model for electricity load forecasting*". Neurocomputing, v. 70, p. 139–154, 2006.

BERNARD, C., MALLAT, S., SLOTINE, J.-J. "*Wavelet interpolation networks*". InPaper presented at the the proc. of ESANN'98, 1998.

BILLINGS, S. A., WEI, H.-L., "*A new class of wavelet networks for nonlinear system identification*". IEEE Transactions on Neural Networks, v. 16, p. 862–874, 2005.

CHAUHAN, N., RAVI, V., CHANDRA, D.K., "*Differential evolution trained wavelet neural networks: application to bankruptcy prediction in banks*". Expert Systems with Applications. v. 36, p. 7659-7665, 2009.

CHEN, Y., YANG, B. DONG, J. "*Time-series prediction using a local linear wavelet neural wavelet*". Neurocomputing, v. 69, p. 449–465, 2006.

CHIA-NAN, K., "*Identification of nonlinear systems with outliers using wavelet neural networks based on annealing dynamical learning algorithm*". Engineering Applications of Artificial Intelligence, v. 25, p. 533-543, 2012.

CEPEA, Centro de Estudos Avançados em Economia Aplicada, USP. Disponível em:< http://cepea.esalq.usp.br/soja/>. Acesso em: 25 de setembro de 2014.

DAUBECHIES I., "*Ten Lectures on Wavelets*". CBMS-NSF Regional Conference Series in Applied Mathematics, Philadelphia, Pennsylvania, 1992.

ESEN, H., OZGEN, F., ESEN, M., ABDULKADIR S., "*Artificial neural network and wavelet neural network approaches for modelling of a solar air heater*". Expert Systems with Applications, v. 36, p. 11240-11248, 2009.

FERRAZ, M. I. F., SÁFADI, T., LAGE, G., "*Uso de modelos de séries temporais na previsão de precipitação pluviais mensais no município de Lavras – MG*". Revista Brasileira de Agrometeorologia, v. 7, n. 2, p. 259-267,1999.

GAO, R., TSOUKALAS, H. I., "*Neural-wavelet methodology for load forecasting*". Journal of Intelligent & Robotic Systems, v. 31, p. 149–157, 2001.

GAO, W., LIU, S., HUANG, L., "*A global best artificial bee colony algorithm for global optimization*", Journal of Computational and Applied Mathematics, v. 236, p. 2741–2753, 2012.

GRASSI, L. H. M., "*Identificação e controle de sistemas dinâmicos utilizando redes wavelets*". Dissertação de Mestrado. Faculdade de Engenharia, Universidade Estadual Paulista, UNESP. Ilha Solteira, SP, 2004.

GUPTA, M. M.; RAO, D. H., "*Neuro-control systems: a tutorial. In: Neuro-Control Systems: Theory and Applications*". IEEE Press: Piscataway, NJ, USA. p. 1-43, 1994.

HAYKIN, S., "*Neural networks: a comprehensive foundation*", Prentice-Hall, Upper Saddle River, New Jersey, USA, 1994.

ILONEN, J., KAMARAINEN, J.-K., LAMPINEN, J. A., "*Differential evolution training algorithm for feed-forward neural networks*". Neural Processing Letters, v. 17, p. 93–105, 2003.

JIN, N., LIU D., "*Wavelet basis function neural networks for sequential learning*". IEEE Transactions on Neural Network, v. 19, p. 523-528, 2008.

KADAMBE, S., SRINIVASAN, P., "*Adaptive wavelets for signal classification and compression*". International Journal of Electronics and Communications, v. 60, p. 45–55, 2006.

KARABOGA, D., "*An idea based on Honey bee swarm for numerical optimization*", Technical Report TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, Kayseri, Türkiye 2005.

KARABOGA, D., AKAY, B., "*A comparative study of artificial bee colony algorithm*", Applied Mathematics and Computation, v. 214, p. 108–132, 2009.

KAVEH, A., TALATAHARI, S., "*Imperialist competitive algorithm for engineering design problems*". Asian Journal of Civil Engineering, v. 11, ed.6, p.675–697, 2010.

KAVEH, A., TALATAHARI, S., "*Optimum design of skeletal structures using imperialist competitive algorithm*". Computers & Structures, v. 88, p. 1220–1229, 2010.

KHABBAZI, A., "*Imperialist competitive algorithm for minimum bit error rate beamforming*" International Journal of Bio-Inspired Computation, v. 1, p. 125-133, 2009.

KHAJEHZADEH, M., TAHA, M. R., EL-SHAFIE, A., ESLAMI, M., "*A modified gravitational search algorithm for slope stability analysis"* Engineering Applications of Artificial Intelligence, v. 25, p. 1589–1597, 2012.

KOHONEN T. "*Self-organizing maps*", Springer Series in Information Sciences, 30. Springer Verlag, 2º ed., 1997.

KRISHNANAND, K. N., AMRUTH, P., GURUPRASAD, M.H., SHARSCHCHANDRA, BIDARGADDI, V., GHOSE,D., "*Glowworm-inspired robot swarm for simultaneous taxis towards multiple radiation sources*". IEEE International Conference on Robotics and Automation, Orlando, FL, p. 958-963, 2006.

KRISHNANAND, K. N., GHOSE, D., "*Detection of multiple source locations using a glowworm metaphor with applications to collective robotics*". IEEE Swarm Intelligence Symposium, Pasadena, CA, USA, p. 84-91, 2005.

KRISHNANAND, K. N., GHOSE, D., "*Glowworm swarm based optimization algorithm for multimodal functions with collective robotics applications*". Multi-agent and Grid Systems, v. 2, n. 3, p. 209-222, 2006.

KUÇUK, M., AGIRALIOGLU, N., "*Wavelet regression techniques for streamflow predictions*". Journal of Applied Statistic v. 33, n. 9, p. 943– 960, 2006.

KUMAR, N.; AHUJA, S.; KUMAR, V.; KUMAR, A. "*Fuzzy time series forecasting of wheat production".* International Journal on Computer Science and Engineering, v. 2, n. 3, p.635-640, 2010.

KUMAR, K.V., RAVI, V., CARR, M., KIRAN, N.K., "Software development cost estimation using wavelet neural networks". The Journal of Systems and Software, v. 81, p. 1853–1867, 2008.

LEKUTAI, G. "*Adaptive self-tuning neural wavelet network controllers*". Tese de Doutorado – Faculty of the Virginia Polytechnic Institute and State University, Blacksburg, Virginia, USA, 1997.

LELAND, W. E., TAQQU, M. S., WILLINFER, W., WILSON, D. V., "*On the self-similar nature of Ethernet traffic".* IEEE/ACM Transactions on Networking, v. 2, n. 1, p. 1-15, 1994.

LIMA, R. C.; GÓIS, M. G.; ULISES, C., "*Previsão de preços futuros de Commodities agrícolas com diferenciações inteira e fracionária, e erros heteroscedásticos*". Revista de Economia e Sociologia Rural, v.45, n. 3 p. 621-644, 2007.

LING. S. H., IU, H. H., LEUNG, F. H. F., CHAN, K. Y., "*Improved Hybrid Particle Swarm Optimized Wavelet Neural Network for Modeling the Development of Fluid Dispensing for Electronic Packaging*" IEEE Transactions on Industrial Electronics, v. 55, p. 3447-3460, 2008.

MADALA, G., "*Introduction to Econometrics". Wiley, NY, USA, 2001.*

MARTIN, C., *"Prognóstico de demanda de potência elétrica para planejamento e operação de sistemas elétricos".* Tese de Doutorado. Universidade Federal de Santa Catarina, Florianópolis, SC, 2005.

MARTINES-MORALES, J. D., PALACIOS, E., VELÁZQUEZ-CARRILLO, G. A., "*Wavelet Neural Networks for Predicting Engine Emissions".* The 2013 Iberoamerican Conference on Electronics Engineering and Computer Science, p. 328-335, 2013.

MATKO, D; ZUPANCIC, B. e KARBA, R., "*Simulation and Modelling of Continuous Systems: A Case Study Approach*". Prentice Hall, London, England, 1992.

MAYER, D. G., KINGHORN, B.P., ARCHER, A. A., "Differential evolution – an easy and efficient evolutionary algorithm for model optimization". Agricultural Systems, p. 315–328, 2005

MCCULLOCH, W.S., PITTS, W., "*A Logical Calculus of the Ideas Immanent in Nervous Activity*", Bulletin of Mathematical Biophysics, v.5, p. 115 – 133, 1943.

MEYER, Y., "*Wavelets-Algorithms and Applications*". SIAM; Philadelphia, USA, 1993.

MINSKY, M., PAPERT, S., "*Perceptrons*". MIT Pess, Cambridge, MA, 1969.

MIRJALILI, S., HASHIM, S. Z. M., SARDROUDI, H. M., "*Training feedforward neural networks using hybrid particle swarm optimization and gravitational search algorithm*". Applied Mathematics and Computation, v. 218, p. 11125–11137, 2012.

MULLER, B., REINHARDT J. "*Neural networks: an Introduction*". Springer-Verlag, Berlin, Germany, 1990.

NELSON, M. M., ILLINGWORTH, W. T., "*A practical guide to neural nets*". Addison-Wesley: Reading, MA, USA, 1991.

PARTAL, T., CIGIZOGLU, H.K., "*Estimation and forecasting of daily suspended sediment data using wavelet–neural networks*", Journal of Hydrology, v. 358, p. 317– 331, 2008.

PELLEGRINI, F., FOGLIATTO, F. "*Passos para a implantação de sistemas de previsão de demanda: técnicas e estudo de caso"*. Revista Produção, v. 11, n. 1, 2001.

POURTAGHU, A., LOTFOLLAHI-YAGHIN, M.A., "*Wavenet ability assessment in comparison to ANN for predicting the maximum surface settlement caused by tunneling*", Tunnelling and Underground Space Technology, v. 28, p. 257–271, 2012.

PRICE, K. V., "*Differential evolution vs. the functions of the 2nd ICEO*", IEEE International Conference on Evolutionary Computation, IEEE Press, Indianapolis, USA, p. 153–157, 1997.

RAMIREZ, O. A., FADIGA, M. "*Forecasting Agricultural Commodity Prices with Asymmetric-Error GARCH Models*", Journal of Agricultural and Resource Economics, v.28, p.71-85, 2003.

RASHEDI, E., NEZAMABADI-POUR, H., SARYAZDI S., "*GSA: A gravitational search algorithm*", Information Sciences, v. 179, p. 2232-2248, 2009.

RUMELHART D. E., HINTON G. E., WILLIAMS R. J., "*Parallel Distributed Processing: Explorations In The Microstructure Of Cognition*". The MIT Press: Cambridge, MA, USA, 1986.

SARAFRAZI, S., NEZAMABADI-POUR, H., SARYAZDI, S., "*Disruption: A new operator in gravitational search algorithm*" Scientia Iranica D, v. 18, p. 539–548, 2011

SILVA, P. O. M. P., GOLDSCHMIDT, R. R., SOARES, J. A., FERLIN, C., Previsão de Séries Temporais Utilizando Lógica Nebulosa, 4º CONTECSI - Universidade deSão Paulo, 2007.

STORN, R. M., "*Designing digital filters with differential evolution*", New Ideas in Optimization, Advanced Topics in Computer Science, McGraw-Hill Inc., London, UK, p. 109–125, 1999

STORN, R. M., PRICE, K. V. "*Minimizing the real functions of the ICEC 96 contest by differential evolution*". IEEE International Conference on Evolutionary Computation, IEEE Press, Nagoya, Japão, p. 842–844, 1996.

STORN, R. M., PRICE, K. V., "*Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces*", Technical Report TR-95-012, International Computer Science Institute (ICSI), University of California, Berkeley, CA,USA, 1995.

SWAIN, R. K., SAHU, N. C., HOTA, P.K., "*Gravitational Search Algorithm for Optimal Economic*", Procedia Technology, v. 6, p. 411–419, 2012.

TALATAHARI, S., AZAR, B.F., SHEIKHOLESLAMI, R., GANDOMI, A.H., "*Imperialist competitive algorithm combined with chaos for global optimization*", Communications in Nonlinear Science and Numerical Simulation, v.17, p. 1312–1319, 2012.

TORN, A., ZILINSKAS, A., "*Global Optimization*". Berlin: Springer, 1989.

ULAGAMMAI M., VENKATESH P., KANNAN P S., et al. "*Application of bacterial foraging technique trained artificial and wavelet neural networks in load forecasting*". Neurocomputing, v. 70, p. 2659-2667, 2007.

VEIGA, C. P., "*Previsão de demanda no desempenho do varejo: estudo comparativo entre modelos lineares e modelos baseados em computação natural*". Tese de Doutorado. Pontifícia Universidade Católica do Paraná, PUCPR, Curitiba, PR, 2014.

VEITCH, D., "*Wavelet Neural Networks and their application in the study of dynamical Systems*". Dissertação de Mestrado. University of York, UK, Helsington, UK, 2005.

WASSERMAN P. D., "*Advanced Methods in Neural Computing*". Van Nostrand Reinhold: New York, NY, USA, 1993.

WEIGEND, A. S., GERSHENFELD, N. A., "Time series prediction: forecasting the future and understanding the past". Addison-Wesley Publishing Company, Menlo Park, CA, USA, 1994.

WELSTEAD S. T., "*Neural network and fuzzy logic applications in C/C++*". New York, John Wiley and Sons. Inc., New York, 1994.

WU, B., QIAN, C., NI, W., FAN, S., "*The improvement of glowworm swarm optimization for continuous optimization problems*", Expert Systems with Applications, v. 39, p. 6335–6342, 2012.

YANG, Y., HUA, C. ZHANG, S., "*A prediction Modeon of the Number of Taxicabs Based on Wavelets Neural Network".* Internacional Conference on Environmental Science and Engineering (ICESE), Singapore, v. 12, 2012.

YILMAZ, S., OYSAL, Y., "*Fuzzy Wavelet Neural Network Models for Prediction and Identification of Dynamical Systems*". IEEE Transactions on Neural Network, v. 21, p. 1599-1609, 2010.

YONENAGA, W. H., FIGUEIREDO, R. S. "*Previsão do preço da soja utilizando redes neurais*". Encontro Nacional de Engenharia de Produção, Rio de Janeiro, Rio de Janeiro, 1999.

YOUSEFI, M., DARUS, A. N., MOHAMMADI, H., "*An imperialist competitive algorithm for optimal design of plate-fin heat exchangers*". International Journal of Heat and Mass Transfer, p. 3178-3185, 2012.

ZAINUDDIN, Z., PAULINE, O., "*Modified wavelet neural network in function approximation and its application in prediction of time-series pollution data*". Applied Soft Computing, v. 11, p. 4866-4874, 2011.

ZHANG Q., BENVENISTE A.; "*Wavelet Networks*" IEEE Transactions Neural Networks, v. 3, n. 6, p. 889-98, 1992.

ZHANG, Q., "*Using wavelet network in nonparametric estimation*". IEEE Transactions on Neural Networks, v. 8, p. 227–236, 1997.

ZHANG, Y., MA, X., GU, Y., MIAO, Y., "*A Modified Glowworm Swarm Optimization for Multimodal Functions"*, Chinese Control and Decision Conference, Mianyang, p. 2070-2075, 2011.