

**UNIVERSIDADE DE SÃO PAULO  
ESCOLA DE ENGENHARIA DE SÃO CARLOS**

**João Victor Marmo**

**Aplicação de redes neurais no problema de  
sequenciamento de máquinas**

**São Carlos**

**2017**



**João Victor Marmo**

**Aplicação de redes neurais no problema de  
sequenciamento de máquinas**

Monografia apresentada ao Curso de Curso de Engenharia Elétrica com Ênfase em Sistemas de Energia e Automação, da Escola de Engenharia de São Carlos da Universidade de São Paulo, como parte dos requisitos para obtenção do título de Engenheiro Eletricista.

Orientador: Prof. Dr. Ivan Nunes da Silva

**São Carlos  
2017**

AUTORIZO A REPRODUÇÃO TOTAL OU PARCIAL DESTE TRABALHO,  
POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS  
DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

M351a Marmo, João Victor  
Aplicação de redes neurais no problema de  
sequenciamento de máquinas / João Victor Marmo;  
orientador Ivan Nunes da Silva. São Carlos, 2017.

Monografia (Graduação em Engenharia Elétrica com  
ênfase em Sistemas de Energia e Automação) -- Escola de  
Engenharia de São Carlos da Universidade de São Paulo,  
2017.

1. Redes Neurais. 2. Sequenciamento. 3.  
Heurísticas. I. Título.

# FOLHA DE APROVAÇÃO

Nome: João Victor Marmo

Título: “Aplicação de redes neurais artificiais no problema de sequenciamento de máquinas”

Trabalho de Conclusão de Curso defendido e aprovado  
em 30/11/2017,

com NOTA 9,0 (NOVE, ZERO), pela Comissão Julgadora:

*Prof. Associado Ivan Nunes da Silva - Orientador - SEL/EESC/USP*

*Mestre Wylliam Salviano Gongora - Doutorando - SEL/EESC/USP*

*Mestre Tiago Henrique dos Santos - Doutorando - SEL/EESC/USP*

Coordenador da CoC-Engenharia Elétrica - EESC/USP:  
Prof. Associado Rogério Andrade Flauzino



## RESUMO

Este trabalho analisou a aplicação de redes neurais para estimar um parâmetro de pesquisa  $k$  em problemas de sequenciamento de máquinas. Foi explorado o caso de máquina única para minimizar a *Average Weighted Tardiness* (AWT), ou seja, o atraso médio ponderado. A rede neural estima o parâmetro de pesquisa  $k$  para duas heurísticas diferentes, a *Average Tardiness Cost* (ATC) e a *Weighted Processing Time and Minimum Slack* (WPT-MS). Para testar a efetividade do método desenvolvido, a rede foi treinada com diferentes números de neurônios na camada oculta, assim como diferentes tamanhos de amostras e de entradas para analisar a influência destes parâmetros em sua efetividade. No total milhões de atividades foram sequenciadas e comparadas com o resultado de um algoritmo de busca exaustiva, e o resultado sugere que a rede neural pode gerar estimativas bastante precisas, variando menos de 1% do valor de *benchmark* para a AWT. Esse método pode aprimorar o tempo de processamento para empresas que fazem uso intensivo de sequenciamento, pois com a rede neural treinada, o tempo de resposta é muito mais rápido que os algoritmos de busca extensiva, além de se obter um resultado mais preciso do que ao utilizar parâmetros definidos manualmente.

**Palavras-chave:** heurística, Matlab, redes neurais, backpropagation, otimização, programação, sequenciamento, métodos de busca.





## ABSTRACT

The objective of this paper is to analyse the application of neural networks to estimate a look-ahead parameter  $k$  in scheduling problems. The single machine case was explored in order to minimise the average weighted tardiness. The neural network estimated the look-ahead parameter  $k$  for two different heuristics, the ATC rule and the WPT-MS. To test its effectiveness the network was trained using different number of neurons in the hidden layer, sample sizes and inputs. In total millions of jobs were scheduled and benchmarked against an exhaustive search method, the results suggests that the neural network can generate very accurate results, varying less than 1% from the benchmark AWT. This method can improve the time performance for the industry, since after the neural network is trained it have a fast time response than an exhaustive search method and is more accurate than a manually chosen parameter.

**Keywords:** back propagation, heuristics, Matlab, neural networks, optimization, programming, scheduling, search methods.



## LISTA DE FIGURAS

Figura 1 – Exemplo do layout de uma rede neural . . . . .	28
Figura 2 – Algoritmo de comparação . . . . .	31
Figura 3 – Layout da Rede Neural . . . . .	32
Figura 4 – Características da RNA . . . . .	33
Figura 5 – Performance do treino da RNA . . . . .	34
Figura 6 – Valores de referência de $k$ . . . . .	38
Figura 7 – Valores de <i>benchmark</i> para AWT nas regras ATC e WPT-MS . . . . .	39
Figura 8 – Erro quadrático médio para $k$ . . . . .	40
Figura 9 – Regras ATC e WPT-MS - Valores de $k$ estimados . . . . .	41
Figura 10 – Regras ATC e WPT-MS - AWT Estimado . . . . .	41



## LISTA DE TABELAS

Tabela 1 – Características das atividades . . . . .	37
Tabela 2 – Entradas da Rede Neural . . . . .	37



## LISTA DE ABREVIATURAS E SIGLAS

RNA	Rede Neural Artificial
ATC	<i>Apparent Tardiness Cost</i>
WPT-MS	<i>Weighted Processing Time and Minimum Slack</i>
AWT	<i>Average Weighted Tardiness</i>





## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>17</b>
<b>1.1</b>	<b>Objetivo do Trabalho</b>	<b>17</b>
<b>1.2</b>	<b>Organização do Trabalho</b>	<b>18</b>
<b>2</b>	<b>REVISÃO BIBLIOGRÁFICA</b>	<b>19</b>
<b>2.1</b>	<b>Visão Geral do Sequenciamento de Máquinas</b>	<b>19</b>
<b>2.2</b>	<b>Tipos de Redes Neurais</b>	<b>20</b>
2.2.1	Redes Neurais de Hopfield	20
2.2.2	Perceptrons Multicamadas	20
2.2.3	Redes Neurais Competitivas	21
2.2.4	Abordagens Híbridas	21
<b>2.3</b>	<b>Evolução das Heurísticas na Literatura</b>	<b>22</b>
<b>2.4</b>	<b>Aplicações de Redes Neurais na Literatura</b>	<b>23</b>
<b>3</b>	<b>METODOLOGIA</b>	<b>25</b>
<b>3.1</b>	<b>Teoria</b>	<b>25</b>
3.1.1	Métodos de Sequenciamento Heurísticos: a regra ATC	25
3.1.2	Métodos de Sequenciamento Heurísticos: a regra WPT-MS	26
3.1.3	A Rede Neural	27
<b>3.2</b>	<b>Método de análise</b>	<b>29</b>
3.2.1	Gerar as características das atividades e as entradas	29
3.2.2	Programa de <i>Benchmark</i>	31
3.2.3	A Rede Neural	32
3.2.4	Comparação e Análise	34
<b>4</b>	<b>RESULTADOS E DISCUSSÕES</b>	<b>37</b>
<b>4.1</b>	<b>Pesquisas Futuras</b>	<b>42</b>
<b>5</b>	<b>CONCLUSÃO</b>	<b>45</b>
	<b>REFERÊNCIAS</b>	<b>47</b>
	<b>APÊNDICES</b>	<b>49</b>
	<b>APÊNDICE A – OUTPUTS DA REDE NEURAL</b>	<b>51</b>



# 1 INTRODUÇÃO

A manufatura movimentada cerca de 10% do PIB do Brasil, e é uma área que vem exponencialmente ganhando complexidade, devido à diversificação de produção e inserção de novas tecnologias. Com esse ganho cada vez maior de escala, surgem preocupações sobre a otimização dos processos produtivos.

O sequenciamento de máquinas consiste em um desses problemas fundamentais. Sua forma de resolução impacta na eficiência bem como nos índices de produtividade do processo. Tal dificuldade varia de uma máquina com um pequeno número de atividades para serem realizadas à várias máquinas com diversas características, restrições e objetivos.

Um dos possíveis objetivos do sequenciamento pode ser minimizar o atraso e a precocidade ponderada do que foi fabricado. Ou seja, cada tarefa tem um peso relacionado a sua importância e o fabricante deseja completar essas tarefas o mais perto do prazo definido quanto possível. Assim, as entregas são mais precisas e os custos, reduzidos.

Problemas de sequenciamento são considerados NP-Complexos. Logo, não é possível solucioná-los em computadores convencionais em tempo polinomial, o que dificulta sua resolução. Por exemplo, uma fábrica com  $m$  máquinas e  $n$  atividades pode ser sequenciada de até  $(n!)^m$  formas diferentes, o que significa que se tivermos  $n = 30$  e  $m = 25$ , teremos  $3.9 \times 10^{810}$  possíveis soluções. Como parâmetro de comparação, sabe-se que o número de átomos em todo o universo observável é estimado estar entre  $10^{78}$  e  $10^{82}$ . No entanto, indústrias reais processam e analisam milhares de peças e centenas de máquinas, fato que comprova a complexidade do sequenciamento dessas atividades. (JAIN; MEERAN, 1996).

Para lidar com problemas desta natureza são adotadas regras heurísticas que visam encontrar soluções boas o suficiente que podem ser resolvidas sem um esforço tão grande quanto o da busca exaustiva. Com isso em mente, autores como Vepsalainen (1987) desenvolveram algoritmos como o *Apparent Tardiness Cost* (ATC) e o *Weighted Processing Time and Minimum Slack* (WPT-MS) para lidar com problemas envolvendo a minimização do atraso ou mesmo a precocidade da produção.

## 1.1 Objetivo do Trabalho

Este trabalho teve por objetivo investigar o uso de uma rede neural multi-camadas, a fim de estimar o valor de pesquisa  $k$ , que é usado em algumas heurísticas de sequenciamento como as regras ATC e WPT-MS. Diferentes conjuntos de entradas foram usadas para que o impacto nos resultados possa ser analisado.

A pesquisa exaustiva pode consumir muito tempo, porém, com o emprego de uma rede neural devidamente treinada para estimar o valor ótimo para  $k$  em qualquer

conjunto de atividades, o tempo consumido para otimizar qualquer sequenciamento pode ser drasticamente reduzido. Para validar a efetividade desse método, os valores também foram calculados por um algoritmo de busca exaustiva e comparados com as saídas da rede neural.

## **1.2 Organização do Trabalho**

Este trabalho de conclusão de curso está organizado em 5 capítulos principais.

O Capítulo 1 apresenta o contexto em que este trabalho está inserido, assim como sua relevância, seus objetivos e justificativas de seu desenvolvimento.

No Capítulo 2 é feita uma revisão bibliográfica do panorama de sequenciamento de máquinas na literatura, do uso de heurística para resolver esses problemas e são introduzidas as redes neurais usadas em sequenciamentos.

O Capítulo 3 mostra o desenvolvimento da metodologia utilizada. Para isso são detalhadas as heurísticas analisadas e o tipo de rede neural utilizada, além de uma explicação do software de busca exaustiva desenvolvido para gerar os valores de referência.

No Capítulo 4 os resultados obtidos são discutidos e analisados, além de serem indicados alguns pontos para futuras pesquisas.

No Capítulo 5 são apresentadas as conclusões deste trabalho e comentários finais a respeito da pesquisa realizada.

Por fim, são apresentadas as referências bibliográficas

## 2 REVISÃO BIBLIOGRÁFICA

### 2.1 Visão Geral do Sequenciamento de Máquinas

A fim de se completar o lote de um produto o mais próximo possível do prazo de entrega definido, o sequenciamento consiste de um processo crítico na manufatura, por definir quando cada peça será produzida e em qual ordem. Para refletir esta realidade, ao se modelar matematicamente o problema são incluídas penalidades relacionadas ao atraso e à precocidade na produção de cada peça.

Segundo [Valente e Alves \(2005\)](#), a inclusão de parâmetros de custo para atividades feitas tanto antes do prazo quanto depois, é compatível com a filosofia da produção just-in-time, que enfatiza a produção de bens de consumo apenas quando eles são necessários. O custo de uma atividade feita antes do prazo pode ser interpretado como o custo de deterioração de produtos perecíveis ou os custos de estocagem de produtos finalizados. O custo de atraso representa o aumento nos custos de envio, perda de vendas e perda de credibilidade.

Como foi dito anteriormente, os problemas de sequenciamento são considerados NP-Complexos, ou seja, para sistemas maiores eles não podem ser resolvidos por computadores convencionais em tempo polinomial. O que significa que em casos reais o uso de algoritmos de busca exaustiva não é uma solução viável.

A fim de contornar o problema, são utilizadas regras heurísticas de priorização para reduzir o esforço computacional e obter uma performance mais robusta. Diversas pesquisas foram feitas comparando tais regras ([KANET; LI, 2004](#)). Apesar de eficientes, regras heurísticas não podem garantir soluções ótimas, porém elas fornecem soluções aproximadas quase tão boas quanto as soluções ótimas ([HO; CHANG, 1991](#)).

Para aprimorar essas heurísticas, o uso Redes Neurais Artificiais (RNA) foi crescendo em popularidade no campo de sequenciamento nos últimos anos. [Kohonen \(1986\)](#) definiu uma rede neural como “massivas redes de elementos simples interconectados de forma paralela e suas organizações hierárquicas que visam interagir com objetos do mundo real da mesma forma que sistemas nervosos biológicos o fazem”. Hoje, elas podem ser consideradas como soluções viáveis para os problemas de otimização de sequenciamento. Existem diversas vantagens em seu uso:

- Com sua capacidade de aprendizado, elas evitam a necessidade de se modelar matematicamente ou de se pesquisar a fundo o problema a ser resolvido;
- Alta resiliência a distorções dos dados de entrada e seu aprendizado;

- Pode fornecer saídas com uma precisão adequada utilizando-se de entradas limitadas;
- Usam muito menos processamento computacional do que outros métodos de modelagem.

As abordagens existentes para o uso de redes neurais em sequenciamento podem ser classificadas em quatro grupos, as redes tipo Hopfield, os Perceptrons multicamadas, as redes baseadas em competição e as abordagens híbridas. (AKYOL; BAYHAN, 2007).

## 2.2 Tipos de Redes Neurais

### 2.2.1 Redes Neurais de Hopfield

A rede neural de Hopfield foi introduzida por Hopfield em 1982, é uma rede de neurônios totalmente conectada que pode realizar tarefas computacionais. Eles servem como uma memória endereçável que permite a recuperação de dados com base no grau de similaridade.

Segundo Akyol e Bayhan (2007):

Nesta rede, a função objetivo e as restrições do problema são codificadas em termos de uma função de energia adequada. O objetivo é obter uma configuração que minimiza a função de energia. A tradução do problema de otimização em uma função de energia apropriada é, em geral, uma tarefa difícil

A vantagem desta abordagem é a sua capacidade computacional altamente paralela, enquanto que a desvantagem está relacionada à facilidade com que ela se prende em muitos mínimos locais.

### 2.2.2 Perceptrons Multicamadas

O perceptron multicamadas é uma rede *feedforward* com um conjunto de neurônios conectados por links ponderados. Consiste em uma camada de entrada, uma ou mais camadas ocultas e uma camada de saída.

Muitas vezes, é treinado por um método chamado *Backpropagation*, que é uma regra de aprendizagem que usa uma técnica de descida gradiente com propagação de erro para trás. Os perceptrons multicamadas treinados por essa regra são chamados de redes de *backpropagation*. Normalmente, eles não estão diretamente envolvidos no problema de otimização.

Uma das dificuldades desta técnica está no risco dela ficar presa no ótimo local. Dessa forma, o ponto de partida dos pesos das conexões tem majoritária importância na redução da ocorrência deste evento. Outra dificuldade com a construção deste tipo

---

de rede é a necessidade de gerar um conjunto de treinamento, que pode ser demorado. Portanto, nos últimos anos, o desempenho dessas redes foi sendo aprimorado combinado-a com diferentes heurísticas ou meta-heurísticas. (AKYOL; BAYHAN, 2007)

### 2.2.3 Redes Neurais Competitivas

Ao contrário das redes Hopfield, a estratégia do “vencedor leva tudo” forma a base das redes competitivas. Nesta rede não supervisionada, existe uma única camada de neurônios de saída totalmente conectados aos neurônios de entrada da rede. Na camada de saída, conhecida como camada competitiva, cada neurônio tenta inibir o neurônio com o qual ele está lateralmente conectado. Para um dado padrão de entrada apresentado à rede, o neurônio com o vetor de pesos com a menor distância do vetor de entrada é chamado de vencedor e sua saída está definida como um. (AKYOL; BAYHAN, 2007)

Para solucionar problemas de otimização, deve-se definir as equações de movimento com as restrições da situação-problema e uma função de energia que converge para os estados estáveis. Sua vantagem consiste na detecção de regularidades, correlação no vetor de entrada e adaptação das respostas futuras.

### 2.2.4 Abordagens Híbridas

O último tipo de rede neural aplicada ao sequenciamento são as abordagens híbridas. Geralmente ela é implementada para minimizar cada fraqueza dos métodos anteriores e melhorar seus pontos fortes. Atualmente, existe um crescente interesse em usar métodos evolutivos para minimizar o trabalho de tentativa e erro do desenvolvimento manual de redes neurais.

A abordagem genética tem vantagens em robustez e paralelismo, semelhantes às redes neurais, mas diferem porque possuem capacidades de busca globais, o que as torna, assim, menos propensas a ficarem presas em um mínimo ou máximo local. Isso é feito mantendo a diversidade na população e porque não é um método de informação gradiente.

De acordo com o trabalho de [Akyol e Bayhan \(2007\)](#), existem tendências atuais para usar redes neuronais evolutivas no sequenciamento em três níveis diferentes:

- No nível mais baixo, a evolução pode ser empregada para refinar o treinamento dos pesos.
- No próximo nível superior, a abordagem pode ser utilizada para aprimorar a arquitetura das redes neurais, o que afeta fortemente sua capacidade de processamento de informações. Isso ajuda a automatizar o design das RNAs, que é um trabalho de tentativa e erro tedioso e que depende da experiência humana.

- No nível mais alto, a evolução pode ser empregada para elaborar a regra de aprendizagem da RNA, que especifica como ajustar pesos no treinamento dos pesos. Como o treinamento dos pesos tem sido tradicionalmente considerado como um processo de aprendizagem, a evolução das regras de aprendizagem pode ser considerada como um processo de aprender a aprender os pesos (YAO; LIU, 1998).

Os algoritmos evolutivos tem inúmeras vantagens. Porém, há casos em que eles não são a melhor solução, como no caso de pesquisa com ajustes finos. Quando isso acontece, o algoritmo pode ser combinado com algoritmos de busca locais, como o *simulated annealing*, *tabu search*, algoritmo de *backpropagation*, etc., a fim de melhorar seu desempenho.

### 2.3 Evolução das Heurísticas na Literatura

Agora será apresentada uma breve visão geral da evolução das heurísticas para o sequenciamento na literatura. Com isso é possível identificar as tendências, as técnicas de ponta, e como essas técnicas foram aprimoradas com o tempo por muitos autores diferentes.

Em 1987, Vepsalainen (1987) desenvolveu regras de despacho para o problema de atraso ponderado em *job shops* e dois parâmetros de pesquisa diferentes foram identificados. Essas duas regras foram o *Apparent Tardiness Cost* (ATC) e o *Weighted COVERT*. Foram feitas comparações com as regras existentes anteriormente e os resultados mostraram que a regra ATC ocupa o primeiro lugar em todas as condições analisadas, exibindo robustez não alcançada por nenhuma regra dos estudos anteriores. A regra COVERT fica em segundo lugar e é confiável especialmente em fábricas congestionadas e com prazos apertados.

Este parâmetro de pesquisa, nomeado como  $k$ , é definido geralmente entre 1,5 e 4,5 e, como recomendado por Vepsalainen (1987),  $m = 2$  é bom para arranjos de fluxo estático e  $m = 3$  é razoável para arranjos dinâmicos. Os experimentos de Valente e Alves (2005) também mostraram que os melhores valores de  $k$  raramente estão abaixo de 0,5. Portanto, se a função retornar um valor inferior a 0,5, vale a pena definir  $k$  para 0,5 em vez disso.

Cicirello e Smith (2001) consideraram em 2001 o problema do sequenciamento com atraso ponderado usando configurações dependentes da sequência com tempos de chegada desconhecidos em um ambiente dinâmico e mostrou que a randomização de heurísticas de despacho neste caso pode resultar em uma melhoria do desempenho. Foi concluído que quanto "mais fácil" o problema, menores serão os benefícios obtidos pela randomização, e quanto "mais difícil", maiores os benefícios.

Em 2004, Kanet e Li (2004) introduziram uma nova regra heurística chamada *Weighted Modified Due Date* (WMDD) para minimizar o atraso ponderado. A eficácia desta regra foi testada em relação a outras regras concorrentes que foram desenvolvidas para o atraso ponderado. Os resultados mostraram que, em todos os casos, a regra da



WMDD que foi introduzida aqui está entre as melhores regras. WMDD, WCOVERT e ATC parecem gerar resultados muito semelhantes, mas uma vantagem clara que a WMDD tem sobre o ATC e o WCOVERT é a sua simplicidade.

Em 2005, [Valente e Alves \(2005\)](#) apresentou duas novas heurísticas para o problema de sequenciamento de máquinas com precocidade/atraso, uma regra de despacho e um *greedy procedure*. Ambos também usam um parâmetro de pesquisa. Em vez de usar um valor fixo para este parâmetro, como costumava ser feito nas pesquisas anteriores, eles desenvolveram funções que mapeiam algumas possibilidades estatísticas de valores apropriados para esse parâmetro. Os resultados computacionais mostraram que as versões baseadas em função das regras de despacho são superiores às suas contrapartes de valor fixo.

É interessante notar que, enquanto as abordagens e as heurísticas estão evoluindo continuamente, o parâmetro de pesquisa ainda está presente por mais de 25 anos nas melhores regras, destacando sua importância.

## 2.4 Aplicações de Redes Neurais na Literatura

De acordo com [Su e Shiue \(2003\)](#):

Redes neurais como ferramentas de aprendizagem, demonstraram sua capacidade de capturar a relação geral entre variáveis que são difíceis ou impossíveis de se relacionar analiticamente, aprendendo, lembrando e generalizando a partir dos padrões de treinamento.

Em outras palavras, eles são aproximadoras universais de funções e, portanto, são atraentes para aprender automaticamente a relação funcional (não-linear) entre as variáveis de entrada e de saída. ([RAAYMAKERS; WEIJTERS, 2003](#))

Um avanço em relação ao uso de redes neurais foi feito por [KIM, LEE e AGNIHOTRI \(1995\)](#), que sugeriram combinar o paradigma de propagação de erros para trás com a regra ATC proposta por [Vepsalainen \(1987\)](#).

Em 1996, [Jain e Meeran \(1996\)](#), depois de revisarem os principais sistemas de sequenciamento de *job-shops* baseados em redes neurais, usaram uma rede neural modificada com *backerror propagation* para lidar com isso. Eles também declararam que todos os indicadores apontam para o uso de múltiplas ferramentas, combinando redes neurais e algoritmos adicionais para resolver o problema do *job-shop* NP-Complexo.

[Franca et al. \(1999\)](#) explorou em 1999 o problema de sequenciamento de máquina única com restrições de tempo, a fim de minimizar o atraso total. Ele usou um algoritmo híbrido-genético, pertencente à classe dos algoritmos meméticos (MA). Este novo método foi comparado com muitos outros métodos diferentes, incluindo o ATCS proposto por [LEE, BHASKARAN e PINEDO \(1997\)](#). Concluiu-se que o MA era o melhor algoritmo, seguido

do algoritmo genético (GA) com o operador OX; O ATCS veio como próximo, seguido por outro algoritmo genético; O pior desempenho foi obtido pela norma *EDD Insertion*.

Em 2004, [Akyol \(2004\)](#) analisou o uso de redes neurais para modelar seis algoritmos heurísticos diferentes aplicados ao sequenciamento em *flowshops* de  $n$  tarefas e  $m$  máquinas, com o objetivo de minimizar o tempo total de produção. Este estudo mostrou a aplicabilidade de redes neurais artificiais a problemas reais de sequenciamento. Fábricas que tem uma dinâmica similar podem se beneficiar do modelo obtido neste estudo.

### 3 METODOLOGIA

#### 3.1 Teoria

Em 1987, [Vepsalainen \(1987\)](#) desenvolveu diferentes regras heurísticas de despacho para o problema de sequenciamento de atraso ponderado com *job shops*. Juntamente com outros achados, identificou-se um parâmetro de pesquisa diferente, o que ajudou a desenvolver a *Apparent Tardiness Cost* (ATC). Os *benchmarks* foram feitos com as regras já existentes e os resultados mostraram que a regra ATC ocupa o primeiro lugar em todas as condições testadas. Apresenta robustez única e jamais equiparada por qualquer regra nos estudos anteriores que visavam minimizar o atraso ponderado.

A regra WPT-MS, (*weighted processing time and minimum slack*) é outra heurística que difere do ATC, pois também considera penalidades por precocidade. Isso é interessante para algumas aplicações, pois poderia implicar em custos de armazenamento extra ou produtos perecíveis danificados, ambos com um grande impacto na penalidade por erros de produção.

Dado que o trabalho visa reduzir o atraso médio ponderado, ou seja, reduzir a penalidade por trabalhos concluídos em um tempo superior ao estipulado, se faz necessário definir uma função objetivo a ser minimizada.

$$\sum_{j=1}^N w_j (C_j - d_j)^+ \quad (3.1)$$

onde  $N$  representa a quantidade de atividades e, para uma determinada atividade  $j$ ,  $C_j$  é o tempo de conclusão (relacionado ao tempo de processamento  $p_j$ ) e  $w_j$  é o peso dessa atividade. A representação  $(y)^+$  representa  $\max(y, 0)$ .

##### 3.1.1 Métodos de Sequenciamento Heurísticos: a regra ATC

Como mencionado anteriormente, a regra de ATC é uma das mais robustas e simples para lidar com o problema de sequenciamento. Seu funcionamento se baseia na análise de cada atividade usando um critério chamado Importância. A importância de cada atividade é calculada para o tempo 0 e, em seguida, a atividade com maior pontuação é agendada primeiro e processada. Tendo em conta o tempo utilizado para processar a última atividade, uma nova importância é calculada para todas as atividades restantes e a com importância mais alta será agendada. O processo continua até que todas as atividades sejam sequenciadas e o tempo passado seja igual à soma de todos os tempos de

processamento. A equação para Importância é mostrada em (3.2)

$$I_j(t) = \frac{w_j}{p_j} \exp\left(-\frac{(d_j - p_j - t)^+}{k\bar{p}}\right) \quad (3.2)$$

onde  $\bar{p}$  é o tempo médio de processamento de todas as atividades restantes e  $k$  o parâmetro de pesquisa mencionado anteriormente.

O valor  $k$  foi historicamente definido de forma manual antes deste estudo, dependendo das características do conjunto de atividades a serem sequenciadas. Na literatura foi observado por [Vepsalainen \(1987\)](#) que este parâmetro de pesquisa era geralmente definido entre 1,5 e 4,5 e  $k = 2$  era geralmente bom para fluxos estáticos e  $k = 3$  era razoável para fluxos dinâmicos. Os experimentos de [Valente e Alves \(2005\)](#) também mostraram que os melhores valores de  $k$  raramente estão abaixo de 0,5. Portanto, se a função retornar um valor inferior a 0,5, é aconselhável definir  $k$  para 0,5 em vez disso.

A média ponderada de atraso (AWT) para esta regra pode ser calculada como:

$$AWT_{ATC} = \min\left(\frac{\sum_{j=0}^N w_j (d_j - t)^+}{N}\right) \quad (3.3)$$

Podemos ver que o AWT é calculado multiplicando o peso da atividade pelo atraso, quando acontece. Depois disso, a média é calculada.

Com (3.3) é possível notar que a desvantagem da regra do ATC é que ela não leva em conta penalidades para a precocidade nas atividades.

### 3.1.2 Métodos de Sequenciamento Heurísticos: a regra WPT-MS

Com a necessidade de levar em consideração a penalidade para a precocidade no sequenciamento, a regra WPT-MS foi desenvolvida melhorando outra regra chamada EXP-ET. Funciona de forma semelhante à regra ATC, sequenciando primeiro as atividades com a maior importância e repetindo o procedimento com as restantes até que todas as atividades sejam sequenciadas.

A Importância e o novo parâmetro utilizado são descritos em (3.4):

$$I_j(t) = \begin{cases} W_j, & \text{se } s_j \leq 1 \\ \frac{W_j}{s_j}, & \text{se } 1 \leq s_j \leq \frac{W_j}{H_j + W_j} k\bar{p} \\ -H_j \left[ 1 - \frac{(k\bar{p} - s_j)}{\left( k\bar{p} - \frac{W_j}{H_j + W_j} k\bar{p} \right)} \right]^2, & \text{se } \frac{W_j}{H_j + W_j} k\bar{p} \leq s_j \leq k\bar{p} \\ -H_j, & \text{caso contrário} \end{cases} \quad (3.4)$$

Os novos parâmetros utilizados,  $W_j$ ,  $H_j$  e  $s_j$  (a folga do trabalho  $j$  no momento  $t$ ) são definidos como:

$$W_j = \frac{w_j}{p_j} \quad (3.5)$$

$$H_j = \frac{h_j}{p_j} \quad (3.6)$$

$$s_j = d_j - t - p_j \quad (3.7)$$

Este cálculo é diferente da ATC, pois é condicional a casos diferentes e usa o valor de  $k$  mais de uma vez, sendo, portanto, mais sensível a ele.

A AWT para a regra WPT-MS é calculada como:

$$AWT_{WPT} = \min \left( \frac{\sum_{j=0}^N w_j (d_j - t)^+ + h_j (t - d_j)^+}{N} \right) \quad (3.8)$$

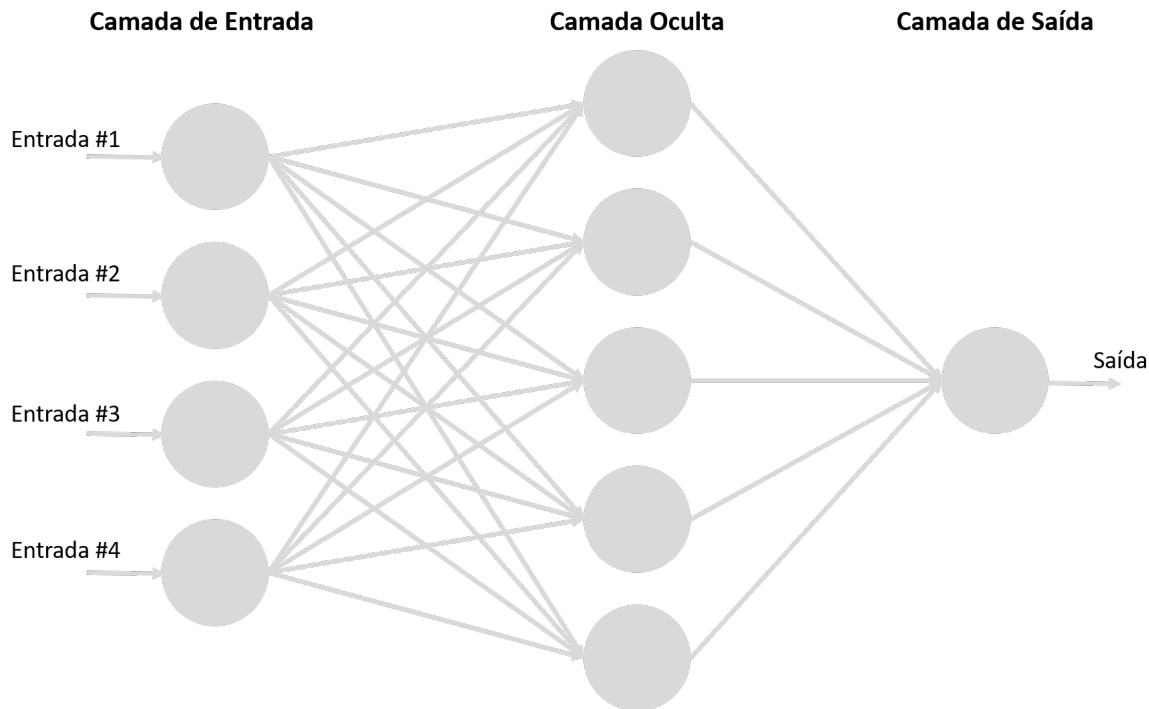
A equação (3.8) é diferente do AWT para a regra ATC, pois este também leva em conta a penalidade por precocidade, multiplicando-a pelo novo peso.

### 3.1.3 A Rede Neural

Uma rede neural é uma estruturação de elementos que simula um neurônio biológico. Tem a capacidade única de estimar os resultados com base apenas nas entradas, sem ser informada da equação que os relaciona. Isso permite que a rede neural descubra padrões subjacentes nos dados e preveja os resultados com uma boa precisão e velocidade.

A rede está organizada em camadas, como visto na Figura 2, ela possui a camada de entrada, a camada oculta e a camada de saída. Os dados alimentados à rede são apresentados através da camada de entrada, e então eles se comunicam com uma ou mais camadas ocultas, onde o processamento real é feito através de diferentes conexões ponderadas. A camada oculta é então vinculada à camada de saída.

Figura 1: Exemplo do layout de uma rede neural



Fonte: Próprio Autor

Na Figura 2, a camada de entrada possui  $N$  neurônios, 4 no exemplo, e a camada oculta possui 5 neurônios. As setas representam a troca de informações entre os neurônios que convergem na saída.

Essa rede normalmente possui uma regra de aprendizado para adaptá-la ao padrão apresentado. Esta regra de aprendizagem ajusta os pesos das conexões entre a camada de entrada, a camada oculta e a camada de saída para que as saídas calculadas da rede neural correspondam às saídas desejadas. Este processo é chamado de treinamento, o qual as entradas e saídas de destino são apresentadas à rede, e ela adapta seus pesos para dar resultados semelhantes a esse alvo.

Após o treinamento, a rede neural pode receber novas entradas com resultados alvo desconhecidos e estimar os resultados com base no processo de aprendizagem que passou.

Um método de treinamento comum é o *Backpropagation*; esta regra de aprendizagem usa uma técnica de descida gradiente com propagação de erro para trás. Ele funciona alimentando a rede com as entradas, passando-as então através da camada oculta usando o padrão inicial e comparando as saídas com os alvos. A diferença entre a saída esperada

e a saída alvo é medida e isso gera um valor  $\delta$  para cada nó oculto. Este valor é usado para treinar novamente a rede usando uma abordagem de gradiente (GURNEY, 1997). O processo continua e melhora-se até que o erro entre a saída e o alvo seja aceitável, ou seja, esteja dentro de uma margem definida pelo usuário.

### 3.2 Método de análise

Este estudo foi realizado com o uso da metodologia detalhada a baixo, com a finalidade de avaliar a eficácia do uso de redes neurais para esta aplicação:

- Geração de características, assim como as entradas das atividades;
- Processamento das entradas através de um programa de pesquisa e sequenciamento exaustivo;
- Processamento das entradas através da rede neural;
- Comparação das saídas da referência e da rede neural;

A geração de dados e o software de *benchmark* de pesquisa exaustivo foram desenvolvidos em C. Para a rede neural, foi utilizada uma *toolbox* específica do Matlab e a comparação de dados foi feita usando o Microsoft Excel.

As seções a seguir explicam detalhadamente o processo de cada etapa para o método mostrado acima.

#### 3.2.1 Gerar as características das atividades e as entradas

A geração de dados foi feita separadamente para o caso da regra ATC e a WPT-MS. Ambos foram programados em C para se ajustar a finalidade deste estudo.

Para a aplicação da regra ATC, existiam duas saídas diferentes, sendo elas:

- Um arquivo de texto com as características da atividade;
- Um com as entradas da rede neural.

Ambos se baseavam nas características das atividades geradas anteriormente.

Os dados foram gerados da seguinte forma:

- 5 simulações diferentes de 1.000 sequenciamentos, cada uma com 100 atividades a serem ordenadas;
- 5 simulações diferentes de 5.000 sequenciamentos, cada uma com 100 atividades a serem ordenadas.

Isso proporcionou 30.000 sequenciamentos diferentes de 100 atividades, ou seja, 3 milhões de atividades foram sequenciadas como referência.

Cada atividade tinha suas próprias características,  $p_j$  (o tempo de processamento),  $w_j$  (peso da atividade) e  $d_j$  (prazo). A regra WPT-MS apresentou uma característica extra,  $h_j$  (peso da precocidade).

Para cada atividade, o valor de  $p_j$  foi gerado como número aleatório entre 1 e 100; os valores de  $w_j$  e  $h_j$  também foram aleatórios, entre 1 e 10. O valor  $d_j$  foi definido como um número aleatório entre dois limites aleatórios que variam para cada trabalho. O limite superior foi definido como um número aleatório entre 10 e a soma de todos os tempos de processamento e o limite inferior como um número aleatório entre 1 e metade do limite superior. A razão para isso é garantir que as entradas da rede neural, que são calculadas com base nas características do trabalho, tenham uma variação suficientemente grande para treinar a rede neural. Os limites superior e inferior foram definidos de forma empírica para garantir a variação da entrada.

Depois que as características das atividades foram geradas, as entradas da rede neural foram definidas para representar cada conjunto de atividades (um pacote de 100 a serem sequenciadas). As sete diferentes entradas escolhidos foram:

$$\tau = 1 - \frac{\bar{d}}{C_{max}} \quad (3.9)$$

$$R = \frac{d_{max} - d_{min}}{C_{max}} \quad (3.10)$$

$$L = \frac{\sum_{j=1}^N (\bar{w} - w_j)^2}{N} \quad (3.11)$$

$$\bar{d} = \frac{\sum_{j=1}^N d_j}{N} \quad (3.12)$$

$$\bar{p} = \frac{\sum_{j=1}^N p_j}{N} \quad (3.13)$$

$$\bar{w} = \frac{\sum_{j=1}^N w_j}{N} \quad (3.14)$$

$$\bar{h} = \frac{\sum_{j=1}^N h_j}{N} \quad (3.15)$$



Onde  $C_{max}$  é a soma de todos os tempos de processamento  $p_j$  e  $\bar{d}$ ,  $\bar{p}$ ,  $\bar{w}$  e  $\bar{h}$  são as médias do prazo, tempo de processamento, peso e peso por precocidade de um dado conjunto de atividades. A entrada  $\tau$  é a criticidade do prazo,  $R$  é a distância dos prazos e  $L$  é o impacto de um determinado conjunto de atividades.

Após essa etapa, todos os dados e entradas brutos foram gerados para alimentar o programa de *benchmark* e a rede neural.

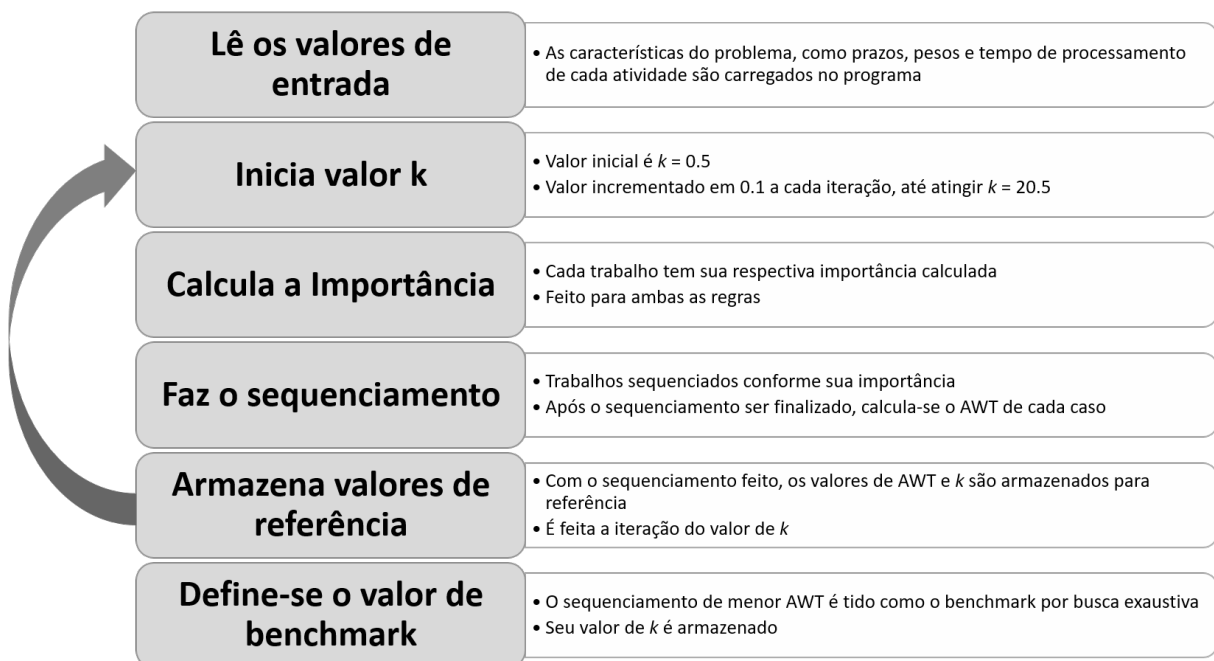
### 3.2.2 Programa de *Benchmark*

As características da atividade foram então processadas no software de *benchmark*, que fez uma busca exaustiva para encontrar o melhor valor de  $k$  para gerar um cronograma com o menor atraso médio ponderado. Como saída, este programa gerou o valor ideal para  $k$  e a AWT para cada conjunto de tarefas.

Dado um valor definido de  $k$ , para encontrar o melhor cronograma, tanto na regra ATC quanto na WPT-MS, o primeiro passo depois de receber as entradas é calcular a importância de cada trabalho. As equações (3.2) ou (3.4) foram usados de acordo com a regra, as atividades com a maior importância foram sequenciadas e o processo foi executado de novo para calcular a importância das atividades restantes.

Uma vez realizado o sequenciamento de todas as atividades, a AWT média foi calculada e armazenada, usando as equações (3.3) ou (3.8). A etapa de sequenciamento foi realizada para um único valor de  $k$  e o mesmo também foi armazenado. Em seguida, um novo ciclo de sequenciamento foi iniciado para um novo valor de  $k$ .

Figura 2: Algoritmo de comparação



Neste trabalho, os valores de  $k$  começaram com 0.5, valor correspondente ao mínimo

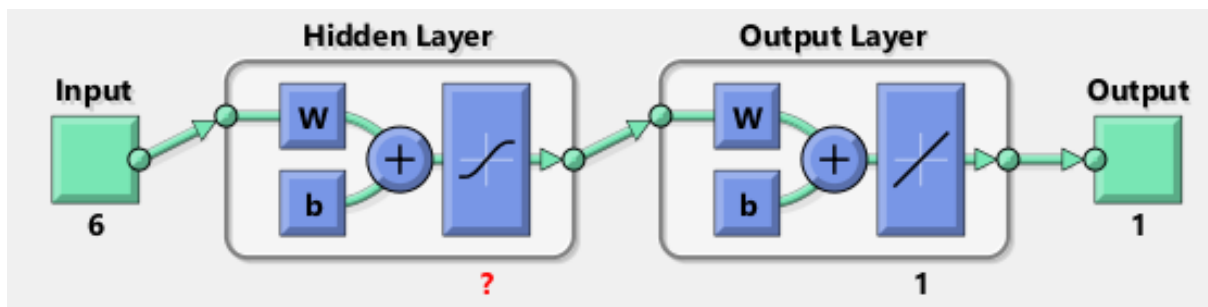
observado na literatura para ser efetivo (VALENTE; ALVES, 2005). De acordo com a metodologia, o incrementou-se  $k$  com uma razão de 0.1, até assumir o valor de 20.5. Para cada parâmetro de pesquisa utilizado, o valor de AWT e  $k$  foram armazenados e, no final, o software comparou todos os valores do atraso médio ponderado e o menor valor, com o seu correspondente  $k$ , foi definido como o melhor. Este processo foi feito para cada conjunto de atividades, com as duas regras diferentes. Ou seja, 30.000 conjuntos de 100 atividades foram sequenciados 200 vezes, o que resultou em 3 milhões atividades diferentes. Por fim, comparou-se tal resultado por meio de pesquisa exaustiva, processo que analisou 600 milhões possibilidades.

Através deste processo, este software de *benchmark* calculou o melhor parâmetro de pesquisa  $k$  para as regras ATC e WPT-MS para minimizar a AWT. Como entrada, recebeu as características da atividade  $p_j$ ,  $w_j$  e  $d_j$  para cada conjunto, e como resultado gerou dois arquivos, um lista o melhor valor de  $k$  para cada conjunto, usado para treinar a rede neural e o valor correspondente da AWT, para comparar o desempenho com os novos resultados gerados pela rede.

### 3.2.3 A Rede Neural

A rede neural utilizada neste trabalho foi desenvolvida através de uma das toolbox oferecidas pelo Matlab. Utilizou-se uma rede de *feed-forward*, de duas camadas, com número definido de neurônios ocultos e neurônios de saída linear. Esta rede tem a capacidade de desempenhar o mapeamento multidimensional arbitrariamente bem, contanto que existam dados consistentes e neurônios suficientes em sua camada oculta. O layout da rede neural é mostrado na Figura 3.

Figura 3: Layout da Rede Neural



Fonte: Toolbox Matlab

A Figura 3 apresenta a arquitetura da rede neural utilizada neste estudo. As entradas e o número de neurônios na camada oculta foram variando para cada regra usada. O número de neurônios da camada oculta, mostrados como '?' na imagem, foi testado com os valores 3, 6, 9 e 12.

Conforme indicado na documentação da toolbox, treinou-se a rede com o algoritmo de *backpropagation Levenberg-Marquardt*. Para casos de memória insuficiente, foi empregado

algoritmo de backpropagation com gradiente conjugado escalado.

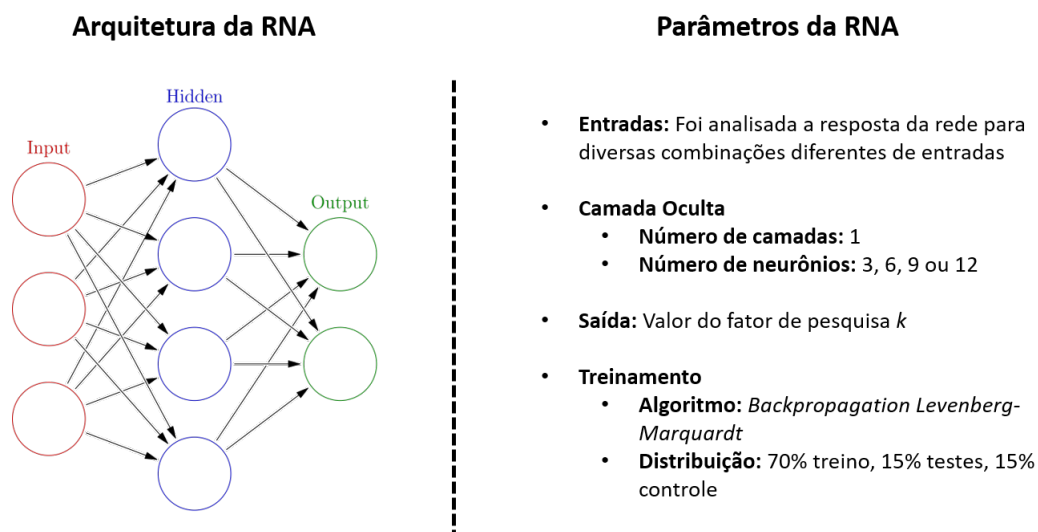
O fluxo da rede neural consistiu de sete etapas principais:

1. Levantamento dos dados;
2. Criação da rede;
3. Configuração da rede;
4. Inicialização dos pesos e as propensões;
5. Treinamento da rede;
6. Validação da rede (análise pós-treinamento);
7. Emprego da rede;

Para a regra ATC, a rede neural recebeu como entradas os resultados das equações (3.9) a (3.14) em um caso e (3.9) a (3.11) em outro, para analisar a diferença no comportamento da rede, variando o número de entradas. A WPT-MS teve entradas semelhantes, (3.9) a (3.15) em um caso e (3.9) a (3.11) no outro.

Os valores-alvo recebidos foram o  $k$  ideal gerado a partir do software de *benchmarking*. Isso foi feito para cada uma das 5 versões dos conjuntos de 1.000 e 5.000 atividades.

Figura 4: Características da RNA



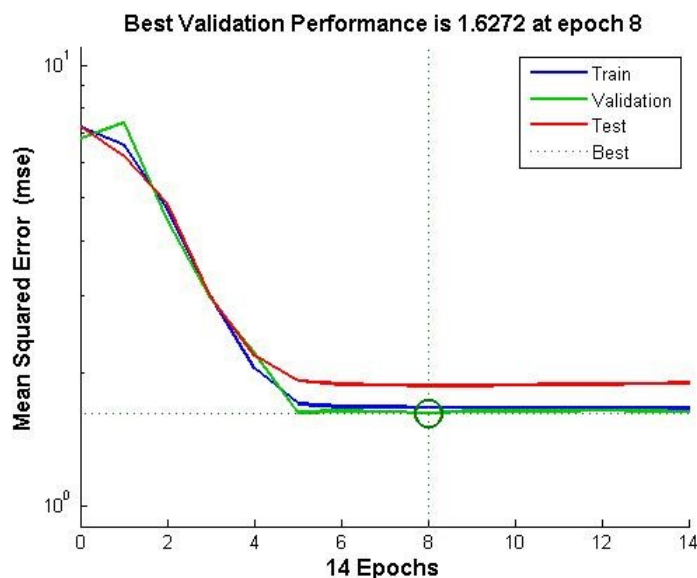
Dividiu-se cada um desses valores aleatoriamente, em que 70% deles foram empregados no treinamento da rede, ou seja, possibilitou o ajuste do erro. A etapa de validação consumiu 15% dos dados, em outras palavras, para medir a generalização da rede e interromper o treinamento quando a generalização parou de melhorar. Os últimos 15% foram

usados para testes. Tal grupo não teve efeito sobre o treinamento, portanto, forneceu uma medida independente do desempenho da rede durante e após o treinamento.

Quanto ao número de neurônios na camada oculta, testou-se com 3, 6, 9 e 12 neurônios para comparar o desempenho de cada caso.

Todos os treinamentos foram feitos de acordo com as definições acima, associando o emprego do algoritmo de *backpropagation Levenberg-Marquardt* e o erro quadrático médio para a medição de desempenho. Um exemplo dos treinamentos realizados pela rede neural está disposto na Figura 5.

Figura 5: Performance do treino da RNA



A Figura 5 apresenta o gráfico para o desempenho de um dos treinos. É possível ver a divisão de treinamento, validação e teste no gráfico. Ainda, é possível observar como o erro quadrático médio diminui ao longo do tempo, bem como a redução da sua taxa de melhora, momento correspondente ao fim do treinamento da rede.

É interessante considerar que, mesmo com as mesmas entradas, o treinamento de uma rede inúmeras vezes gera resultados diferentes devido a diferentes condições iniciais e amostragem.

Depois que todas as redes foram treinadas, validadas e testadas, os resultados - os valores estimados de  $k$  - foram armazenados.

### 3.2.4 Comparação e Análise

Por fim, os resultados fornecidos pela rede neural foram processados novamente através de uma versão adaptada do software de *benchmark*. Este, ao invés de gerar valores de  $k$  e fazer o sequenciamento, utilizou os valores de  $k$  estimados pela rede neural e calculou

o valor de AWT para cada conjunto de atividades. Esses valores foram então comparados à AWT de referência e o erro entre eles foi analisado.

A comparação foi realizada para o número de neurônios diferentes, 3, 6, 9 e 12, e também para uma amostra menor com número diferente de entradas, 6 e 3 entradas para o ATC e 7 e 3 para o WPT-MS. O contraste entre os valores de  $k$  e AWT para cada regra também foi levado em consideração e analisados.



## 4 RESULTADOS E DISCUSSÕES

Os primeiros resultados são as características das atividades. Uma vez que uma grande amostra foi utilizada, as variações dos dados de características ficaram próximas aos limites superior e inferior definidos para as variáveis:

Tabela 1: Características das atividades

$$\begin{aligned} 1 &\leq p_j \leq 100 \\ 0.1 &\leq w_j \leq 10 \\ 0.1 &\leq h_j \leq 10 \\ 1 &\leq d_j \leq 5935 \end{aligned}$$

É importante notar que atividades com prazo 1 eram possíveis, o que significa que quase certamente seriam sequenciadas com atraso. Isso foi feito para que a rede também possa ser treinada para esse tipo de situação. Os outros resultados foram esperados, pois são espalhados pelos limites superior e inferior previamente definidos.

Em seguida, analisou-se as entradas da rede neural, geradas com base nas características acima. Os resultados são mostrados abaixo:

Tabela 2: Entradas da Rede Neural

$$\begin{aligned} 0 &\leq R \leq 0.98 \\ 5.38 &\leq L \leq 11.72 \\ 0.23 &\leq \tau \leq 1.00 \\ 38.20 &\leq \bar{p} \leq 62.60 \\ 3.77 &\leq \bar{w} \leq 6.13 \\ 3.78 &\leq \bar{h} \leq 6.17 \\ 5.27 &\leq \bar{d} \leq 4139.09 \end{aligned}$$

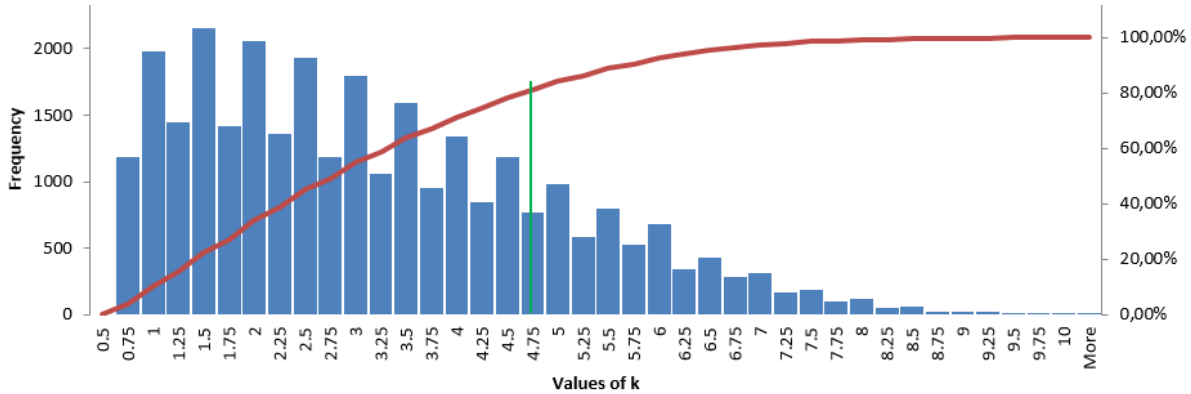
A variação desses valores tem um papel importante no treinamento da rede neural, uma vez que diferentes padrões resultam no aprendizado e ajuste da rede nas condições mais diversas. A maioria dos resultados provou ter uma variação bastante ampla, principalmente  $R$ ,  $\tau$  e  $L$ , pois foram calculados com base em características completamente aleatórias e com uma grande amostra.

Os valores de  $\bar{p}$ ,  $\bar{w}$ ,  $\bar{h}$  e  $\bar{d}$  tiveram uma variação relevante, e mesmo que não tenham abrangido todas as possibilidades, verificou ser suficiente para treinar a rede neural, como será discutido mais adiante. O valor de  $\bar{w}$  teve uma variação muito grande, já que os limites superior e inferior eram aleatórios.

Com as características das atividades, o programa de *benchmark* calculou o parâmetro de pesquisa ideal  $k$  e o AWT. A Figura 6 mostra um histograma com os valores de  $k$  para a regra ATC, a frequência de cada intervalo e a porcentagem acumulada. Todos os

valores do *benchmark* ATC foram utilizados, 5 versões de 1.000 conjuntos de dados e 5 com 5.000 conjuntos de dados, ou seja, 30.000 amostras.

Figura 6: Valores de referência de  $k$



Fonte: Próprio Autor.

Ainda na Figura 6, é possível observar a linha verde, delimitador da porcentagem acumulada dos resultados para cerca de 80% dos casos, neste caso 4.75.

Como esperado, nenhum valor de  $k$  estava entre 0 e 0.5, pois o valor foi ajustado para 0.5 se isso acontecer, como explicado anteriormente. Com o gráfico, é possível notar que, usando o Princípio de Pareto, 80.88% dos valores estavam entre 0.5 e 4.75, o que significa que menos de 20% estavam acima de 4.75. A média da amostra é de 3.11. A mesma análise foi feita para a regra WPT-MS e 78.7% dos valores estavam entre 0.5 e 10.0, sendo a média 6.7.

Em comparação, a regra ATC requer um número menor de  $k$  para otimizar o cronograma, 3.11 comparado com o 6.7 do WPT-MS. A variação dos dados na regra ATC também é menores, uma vez que cerca de 80% dos valores foram inferiores a 4.75, o equivalente para a WPT-MS foi 10.0.

Isso indica que é mais fácil obter resultados aceitáveis estabelecendo valores fixos para o parâmetro de pesquisa na regra ATC, uma vez que eles têm uma variação menor do que a WPT-MS. Ao considerar a precocidade para os cálculos do AWT, e também pelo fato de que na equação (3.4) o valor de  $k$  foi usado mais de uma vez, a regra WPT-MS é mais sensível a variações em  $k$ , portanto, o uso de redes neurais pode trazer uma solução mais viável e efetiva para isso.

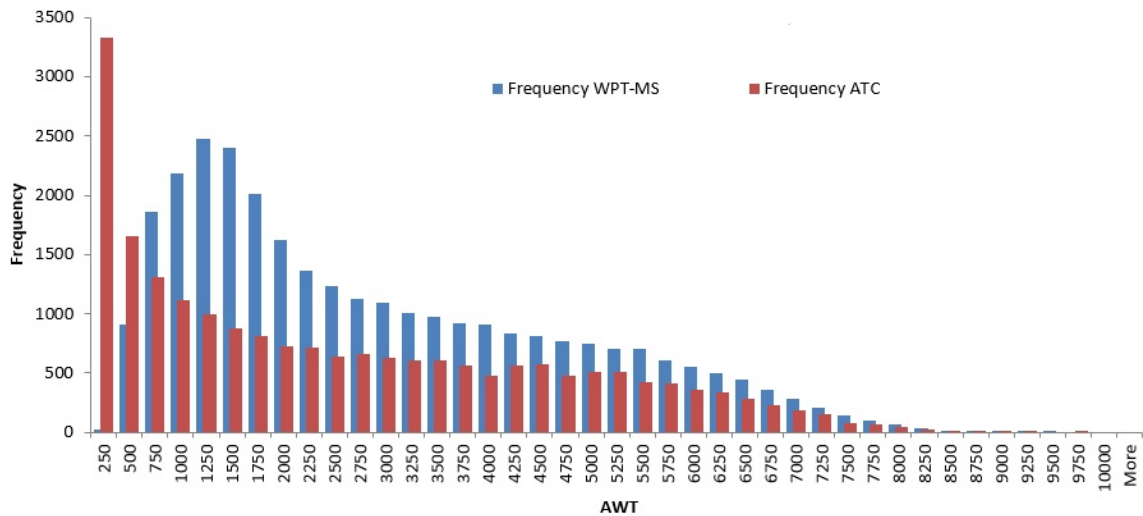
O valor equivalente do AWT foi calculado com base nos valores de  $k$  para ambas as regras. E os resultados são mostrados na Figura 7.

Como esperado, os valores de AWT para a WPT-MS são mais elevados, uma vez que a precocidade também adiciona penalidade à média. A maioria dos valores de ambas as regras está abaixo de 2.500. A média para o ATC era 2.412 e para WPT-MS era 2.794.

A partir daqui todos os dados gerados foram comparados e é hora de processá-los



Figura 7: Valores de *benchmark* para AWT nas regras ATC e WPT-MS



Fonte: Próprio Autor.

através da rede neural. A próxima análise foi determinar se o número de entradas afetou o desempenho da rede neural ou não. Para analisar isso, foi calculado o erro quadrático médio (MSE) do valor  $k$  para cada regra com 3, 6, 9 e 12 neurônios tanto para 1.000 como para 5.000 conjuntos.

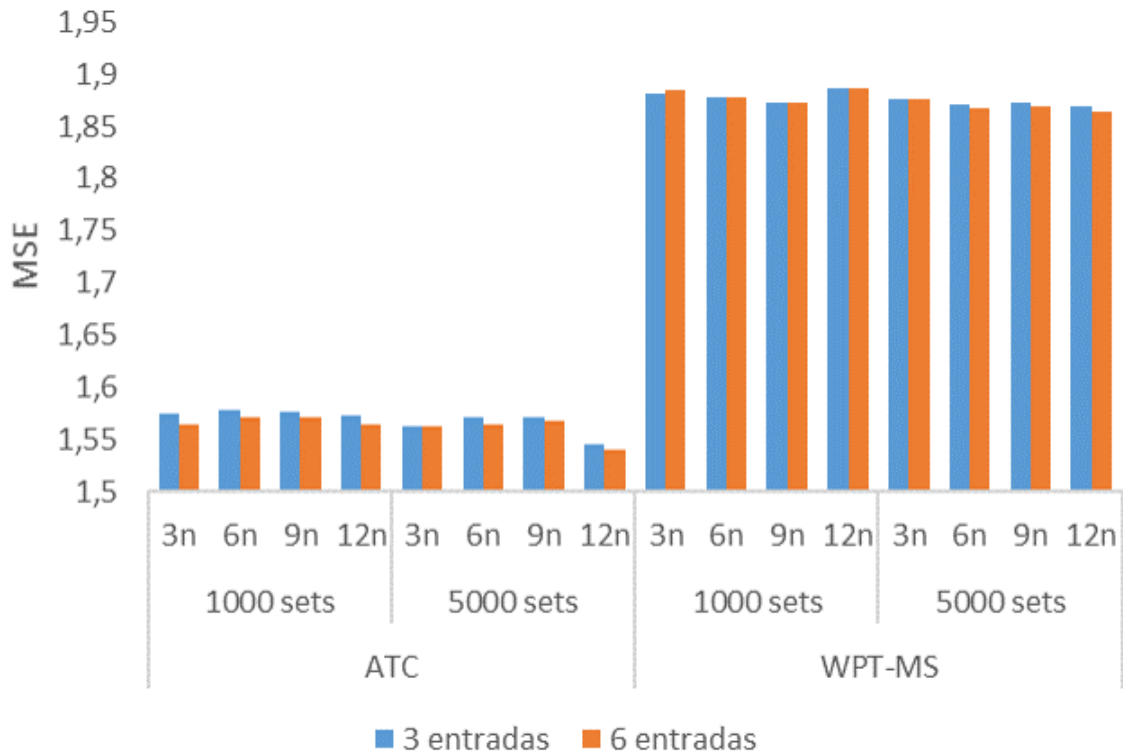
A Figura 8 expõe o impacto no erro médio quadrático de  $k$  de usar um número diferente de entradas na rede neural. O erro para 6 entradas é um pouco menor em ambos os casos.

O MSE médio entre todos os neurônios para 5000 conjuntos na regra ATC para 6 entradas foi 1,556 e para 3 entradas 1,562, uma mudança de menos de 0,5%. O mesmo comportamento ocorreu com 1000 conjuntos, com 6 entradas, o erro foi de 1,564 e 1,594 para 3 entradas, uma mudança de 1,9%. Em ambos os casos, um maior número de entradas gera um erro menor.

Para a regra WPT-MS, ocorreu um comportamento semelhante, com 5000 conjuntos e 7 entradas, o MSE médio foi de 1,869 e para 3 entradas 1,881, uma variância de 0,7%. Para 1000 conjuntos e 7 entradas, o MSE foi de 1,809 e para 3 foi 1,800, uma variância de 0,05%.

Isso mostra que, em ambos os casos, o uso de um número maior de entradas gera um erro menor. Por esta razão, para os outros testes com a amostra completa foram utilizados 6 entradas para a regra ATC e 7 para o WPT-MS.

Depois de decidido que o maior número de entradas seria usado, os dados completos foram executados através da rede neural para obter um valor estimado de  $k$ , as 5 versões de 5000 e 1000 conjuntos. O valor gerado foi avaliado e o valor AWT foi calculado com base no equivalente  $k$ . As figuras 9 e 10 mostram os resultados.

Figura 8: Erro quadrático médio para  $k$ 

Fonte: Próprio Autor

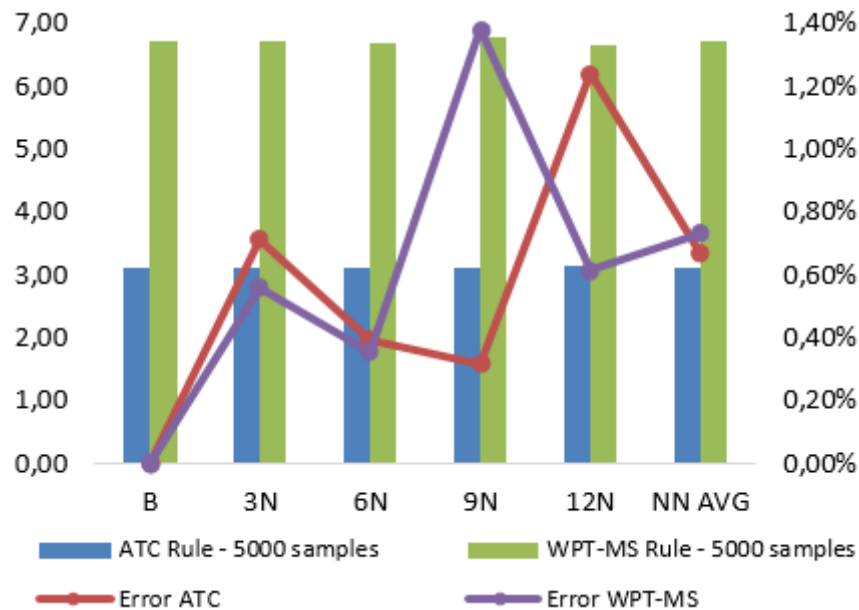
A Figura 9 apresenta as saídas  $k$  e os erros para 5000 amostras usando ambas as regras. B representa as referências a serem atingidas.

Para a regra ATC, em relação aos valores  $k$ , o *benchmark* apresentou uma média de 3,10 com 1000 amostras e 3,12 com 5000 amostras. Os valores médios para 1000 amostras geradas pela rede neural variam de 3,08 a 3,13, dependendo do número de neurônios. O mesmo se aplica a 5000 amostras, variando agora de 3,11 a 3,15. Os valores gerados pela rede têm um erro médio de 1,19% com 1000 amostras e 0,67% com 5000 amostras. Isso mostra que a amostra de treinamento maior melhorou as previsões da rede neural.

Ainda na regra ATC, os valores médios de referência para o AWT são muito semelhantes para 1000 e 5000 amostras, 2404,95 e 2399,89, respectivamente. Comparando os resultados médios da rede neural com o *benchmark*, eles tiveram 0,91% de erro médio para AWT com 1000 amostras e 0,22% com 5000 amostras, o que é esperado porque os valores de  $k$  gerados a partir de amostras maiores tiveram uma precisão melhor.

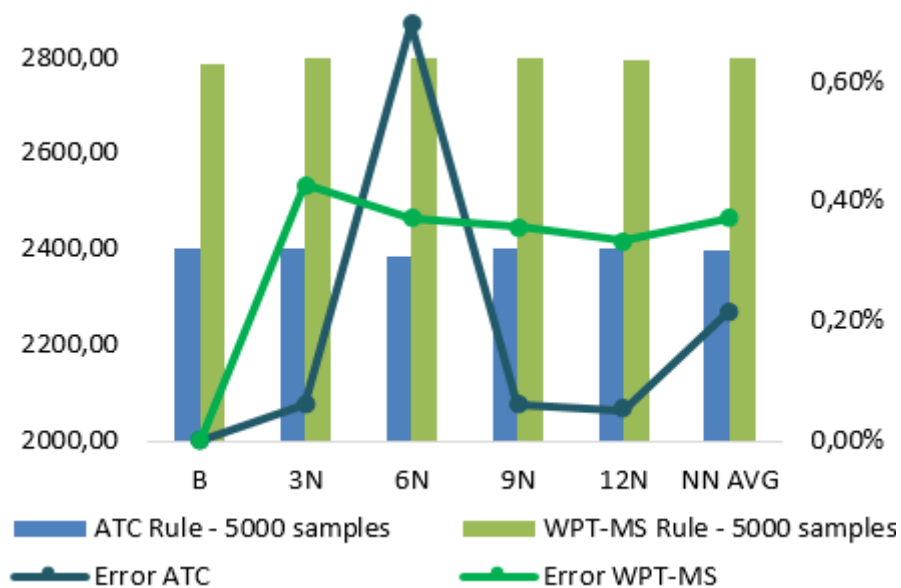
A Figura 10 mostra as saídas AWT e os erros para 5000 amostras usando ambas as regras. B representa as referências a serem atingidas.

O resultado com 12 neurônios na camada oculta apresentou a menor precisão. Este resultado foi obtido com 5000 amostras. Em contrapartida, os mesmos neurônios,

Figura 9: Regras ATC e WPT-MS - Valores de  $k$  estimados

Fonte: Próprio Autor.

Figura 10: Regras ATC e WPT-MS - AWT Estimado



Fonte: Próprio Autor.

quando em uma condição com 1000 amostras, geraram a melhor precisão. Encontrou-se um comportamento similar com número diferente de neurônios na camada oculta. Assim, acredita-se que, usando neurônios 3, 6, 9 e 12, não há uma relação clara entre o número de neurônios na camada oculta e o desempenho.

Usando os valores de  $k$  gerados pela RNA para calcular o AWT, com 1000 amostras o erro foi de 1,19% com  $k$  para 0,91% com o AWT. Com 5000 amostras, passou de 0,67% para 0,22%. Isso mostrou que o AWT tem uma sensibilidade menor às mudanças no

parâmetro de pesquisa, então o erro ao invés de ser propagado é na realidade reduzido. Esse fato melhorou o desempenho das previsões da rede neural.

A regra WPT-MS, por sua vez, tinha um valor médio de *benchmark* de referência de 6,64 para 1000 conjuntos de atividades e 6,71 para 5000. Os valores gerados a partir da rede variaram de 6,60 a 6,68 com 1000 conjuntos e de 6,69 a 6,77 com 5000 conjuntos. Os valores  $k$  gerados pela RNA têm um erro médio de 2,99% com 1000 amostras e 0,73% com 5000 amostras. Isso é semelhante ao comportamento na regra ATC, onde uma amostra maior implicava em um treinamento melhor.

Os valores médios para o AWT também foram semelhantes para 1000 e 5000 amostras, sendo eles 2813,8 e 2813,4, respectivamente. O erro médio foi de 0,34% para AWT com 1000 amostras e 0,37% com 5000 amostras em média.

O desempenho também não teve uma mudança significativa variando o número de neurônios na camada oculta, semelhante ao que aconteceu com a regra ATC. A precisão do AWT também foi melhor do que a precisão para os valores de  $k$ ; com 1000 amostras, o erro foi de 2,99% com  $k$  para 0,34% com AWT. Com 5000 amostras, passou de 0,73% para 0,37%.

Todos os resultados, para ambas as regras, estavam dentro do esperado. Isso mostrou que, tanto para a regra ATC quanto para a WPT-MS, a abordagem da rede neural para estimar o parâmetro de pesquisa foi efetiva, gerando erros baixos no AWT, abaixo de 0,4% nos dois casos.

A regra ATC teve um erro menor em geral, fato já esperado, uma vez que a variação de valores de  $k$  era menor do que na outra regra, e também porque não considera penalidades para a precocidade. Em geral, ambas as regras funcionam bem com a rede neural.

Vale ressaltar que para ambas as regras, a utilização da rede neural foi cerca de 200 vezes mais rápida que o algoritmo de busca exaustiva, uma vez que não foi necessário iterar diversos sequenciamentos com valores de  $k$  diferentes.

## 4.1 Pesquisas Futuras

Diferentes pontos a serem investigados podem também ser considerados para analisar a relação entre sequenciamento e redes neurais.

O comportamento de diferentes heurísticas também pode ser estudado para ver o desempenho da rede neural em dinâmicas diferentes de operação de uma fábrica, com isso espera-se criar um conjunto de ferramentas de regras heurísticas e redes neurais visando um planejamento mais amplo e eficaz.

Diferentes tipos de redes neurais e regras de treinamento também devem ser

pesquisados no futuro.

Dada uma certa heurística, diferentes entradas para a rede neural podem ser estudadas para analisar sua eficácia. Deve ser assegurado que essas entradas tenham uma variação grande o suficiente para treinar a rede. Diferentes tipos de características de atividades também podem ser interessantes para análise.

Ganhar um conhecimento mais amplo nesses aspectos será muito importante e relevante para a área e o impacto na indústria pode ser significativo.



## 5 CONCLUSÃO

A necessidade de cálculos em tempo real e ao grande número de variáveis envolvidas na indústria hoje em dia coloca em destaque a pesquisa por métodos cada vez mais rápidos e eficazes de sequenciamento. Dessa forma, este trabalho possibilitou a investigação e compreensão das redes neurais na solução de problemas de sequenciamento de máquinas.

As heurísticas estudadas provaram ser eficazes usando o parâmetro de pesquisa estimado por uma rede neural previamente treinada.

As características e as entradas das atividades tiveram variação grande suficiente para treinar a rede neural, assim como um número maior de entradas no treinamento provou criar resultados ligeiramente melhores. O impacto do uso de diferentes neurônios na camada oculta da rede também foi analisado e não se encontrou uma relação clara entre o desempenho e número de neurônios.

O número de amostras apresentadas para treinar a rede neural mostra claramente que uma amostra maior geralmente resulta em previsões melhores para a rede.

A regra ATC tende a ter uma variação menor dos valores  $k$  do que a WPT-MS e também mostrou uma menor sensibilidade às mudanças neste parâmetro. Isso se refletiu nos valores de AWT, que tiveram variações menores e valores também geralmente menores.

A regra WPT-MS, em contraste, apresentava uma maior variação nos valores de  $k$  e nos valores de AWT. Isso era esperado, uma vez que esta regra usa o parâmetro de pesquisa mais de uma vez no cálculo da Importância. Os valores de AWT foram mais variados porque a penalidade por precocidade também é considerada nesta heurística, tornando o AWT geralmente maior e menos concentrado.

Ressalta-se ainda que a previsão para os valores de AWT teve uma precisão melhor do que as previsões de  $k$ , apesar do AWT ter sido calculado com base na Importância, resultado este que utiliza  $k$  em seu cálculos. Portanto, o erro foi reduzido, ao invés de ser propagado. Isto significa que os valores de saída da rede neural foram mais precisos.

No geral, o desempenho das saídas da rede neural foi muito bom, variando menos de 0,5% do valor ótimo de AWT em ambos os casos, quando treinado com 5000 conjuntos de trabalho.

Tendo em vista os resultados obtidos assim como a perspectiva de melhoria, as redes neurais podem ser consideradas como uma alternativa viável na indústria quando o tempo de processamento é uma preocupação crítica.





## REFERÊNCIAS

- AKYOL, D. E. Application of neural networks to heuristic scheduling algorithms. **Computers Industrial Engineering**, v. 46, n. 4, p. 679 – 696, 2004. ISSN 0360-8352. Computers and Industrial Engineering Special Issue on Selected papers from the 29th. International Conference on Computers and Industrial Engineering. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S036083520400066X>>.
- AKYOL, D. E.; BAYHAN, G. M. A review on evolution of production scheduling with neural networks. **Computers Industrial Engineering**, v. 53, n. 1, p. 95 – 122, 2007. ISSN 0360-8352. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0360835207000666>>.
- CICIRELLO, V. A.; SMITH, S. F. Randomizing dispatch scheduling policies. In: **In Using Uncertainty Within Computation: Papers from the 2001 AAAI Fall Symposium, Technical Report FS-01-04**. [S.l.]: AAAI Press, 2001. p. 30–37.
- FRANCA, P. et al. Memetic algorithms to minimize tardiness on a single machine with sequence-dependent setup. In: **In Proceedings of the 5th International Conference of the Decision Sciences Institute**. [S.l.: s.n.], 1999. p. 1708–1710.
- GURNEY, K. **An Introduction to Neural Networks**. Bristol, PA, USA: Taylor & Francis, Inc., 1997. ISBN 1857286731.
- HO, J. C.; CHANG, Y.-L. A new heuristic for the n-job, m-machine flow-shop problem. **European Journal of Operational Research**, v. 52, n. 2, p. 194 – 202, 1991. ISSN 0377-2217. Disponível em: <<http://www.sciencedirect.com/science/article/pii/037722179190080F>>.
- JAIN, A. S.; MEERAN, S. **Scheduling A Job-Shop Using A Modified Back-Error Propagation Neural Network**. 1996.
- KANET, J. J.; LI, X. A weighted modified due date rule for sequencing to minimize weighted tardiness. **Journal of Scheduling**, v. 7, n. 4, p. 261–276, 2004. ISSN 1099-1425. Disponível em: <<http://dx.doi.org/10.1023/B:JOSH.0000031421.64487.95>>.
- KIM, S.-Y.; LEE, Y.-H.; AGNIHOTRI, D. A hybrid approach to sequencing jobs using heuristic rules and neural networks. **Production Planning & Control**, v. 6, n. 5, p. 445–454, 1995. Disponível em: <<http://dx.doi.org/10.1080/09537289508930302>>.
- KOHONEN, T. **Representation Of Sensory Information In Self-Organizing Feature Maps, And Relation Of These Maps To Distributed Memory Networks**. 1986. 248-259 p. Disponível em: <<http://dx.doi.org/10.1117/12.964017>>.
- LEE, Y. H.; BHASKARAN, K.; PINEDO, M. A heuristic to minimize the total weighted tardiness with sequence-dependent setups. **IIE Transactions**, v. 29, n. 1, p. 45–52, 1997. Disponível em: <<http://dx.doi.org/10.1080/07408179708966311>>.
- RAAYMAKERS, W.; WEIJTERS, A. Makespan estimation in batch process industries: A comparison between regression analysis and neural networks. **European Journal of**

**Operational Research**, v. 145, n. 1, p. 14 – 30, 2003. ISSN 0377-2217. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S037722170200173X>>.

SU, C.-T.; SHIUE, Y.-R. Intelligent scheduling controller for shop floor control systems: A hybrid genetic algorithm/decision tree learning approach. **International Journal of Production Research**, v. 41, n. 12, p. 2619–2641, 2003. Disponível em: <<http://dx.doi.org/10.1080/0020754031000090612>>.

VALENTE, J. M.; ALVES, R. A. Improved heuristics for the early/tardy scheduling problem with no idle time. **Computers Operations Research**, v. 32, n. 3, p. 557 – 569, 2005. ISSN 0305-0548. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0305054803002533>>.

VEPSALAINEN, A. P. J. Priority rules for job shops with weighted tardiness costs. **Manage. Sci.**, INFORMS, Institute for Operations Research and the Management Sciences (INFORMS), Linthicum, Maryland, USA, v. 33, n. 8, p. 1035–1047, set. 1987. ISSN 0025-1909. Disponível em: <<http://dx.doi.org/10.1287/mnsc.33.8.1035>>.

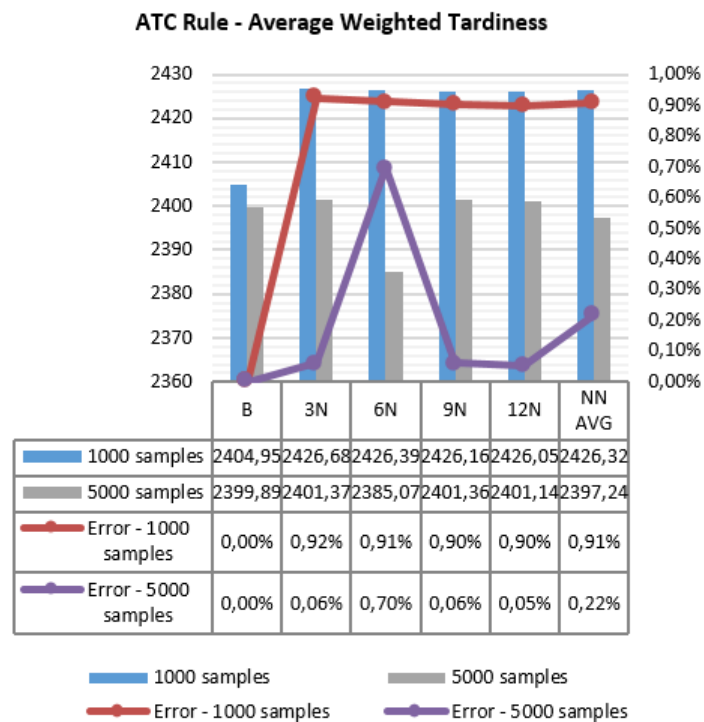
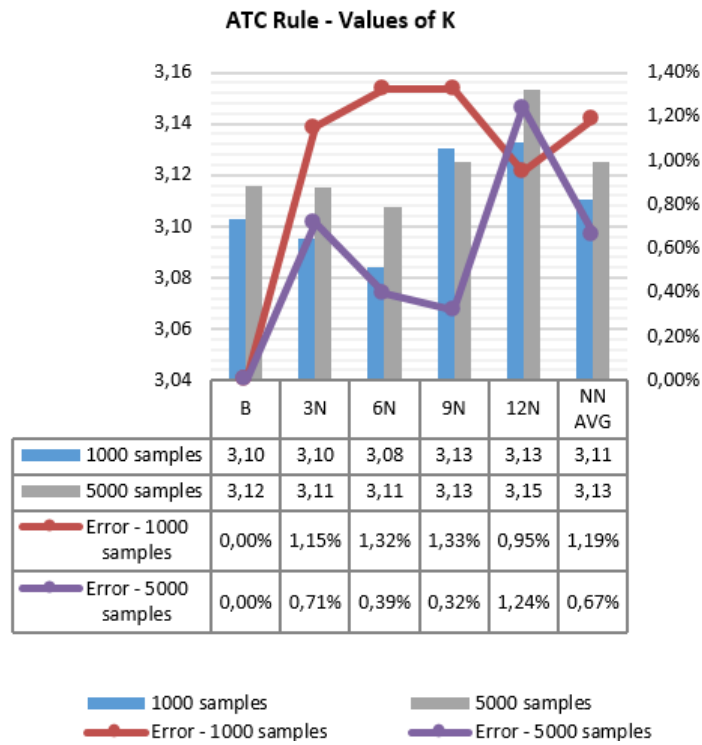
YAO, X.; LIU, Y. Towards designing artificial neural networks by evolution. **Appl. Math. Comput.**, Elsevier Science Inc., New York, NY, USA, v. 91, n. 1, p. 83–90, abr. 1998. ISSN 0096-3003. Disponível em: <[http://dx.doi.org/10.1016/S0096-3003\(97\)10005-4](http://dx.doi.org/10.1016/S0096-3003(97)10005-4)>.

## **Apêndices**

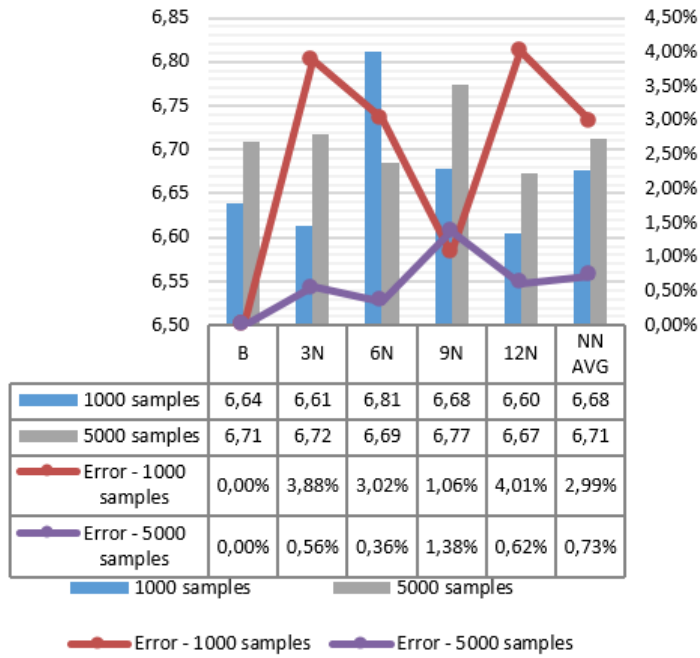


## APÊNDICE A – OUTPUTS DA REDE NEURAL

Aqui podem ser vistos em mais detalhes os resultados da rede neural discutidos durante o trabalho.



**WPT-MS Rule - Values of K**



**WPT-MS Rule - Average Weighted Tardiness**

