



O USO DE INTELIGÊNCIA COMPUTACIONAL NA LÓGICA DE DECISÃO EM MODELOS DE SIMULAÇÃO - UMA APLICAÇÃO NO CÁLCULO DE EMISSÕES DE GASES EM SISTEMAS LOGÍSTICOS

Fábio Freitas da Silva^a, Eder Reis Tavares^a, Túlio Almeida Peixoto^a,

Ítalo de Oliveira Matias^a, João José de Assis Rangel^{a*}

Universidade Candido Mendes – UCAM, Campos dos Goytacazes – RJ, Brasil

Resumo

Este trabalho emprega redes neurais artificiais em modelos de simulação a eventos discretos para classificação de entregas de fornecedores em um sistema logístico. Os modelos de simulação foram resolvidos utilizando algoritmos de inteligência computacional para efetuar decisões baseadas nas emissões dos gases emitidos pela frota e custos gerados durante as entregas. Além disso, foi realizada uma comparação de dois *frameworks* de redes neurais artificiais para determinar o mais eficiente em termos de velocidade de execução. Foi utilizado o *software* livre e de código fonte aberto Ururau. Foram classificados as entregas dos fornecedores de regiões distintas para avaliar qual região tem maior número de aprovações. A biblioteca *Fast Artificial Neural Network* apresentou tempo de execução quatro vezes mais rápida, aproximadamente, em relação à ENCOG que estava sendo utilizada no *software*.

Palavras-chave: Simulação a Eventos Discretos, Ururau, Efeito Estufa, Monóxido de Carbono.

Abstract

This work employs artificial neural networks in discrete event simulation models to classify deliveries of suppliers in a logistic system. The simulation models were solved using computational intelligence algorithms to make decisions based on the gas emissions emitted by the fleet and costs generated during the deliveries. Besides, a comparison of two frameworks of artificial neural networks to determine the most efficient one in terms of execution speed was made. The free and open source *software* Ururau was used. Deliveries of suppliers of different regions were classified to evaluate which region has the highest number of acceptance. The Fast Artificial Neural Network library presented better performance concerning the one that has been used in the *software*.

Keywords: Discrete Event Simulation, Ururau, Greenhouse effect, Carbon Monoxide.

1. Introdução

Em recentes trabalhos, autores como Bergmann *et al.* (2014) e Silva *et al.* (2014) utilizaram redes neurais artificiais (RNA) para modelar os eventos associados a decisões tomadas por pessoas em modelos de simulação a eventos discretos (SED). Estes pesquisadores têm investigado alternativas para representar de forma mais realística e com maior nível de detalhes o comportamento das ações realizadas por pessoas em sistemas discretos. Isto se deve ao fato de que, na maioria dos casos, os *softwares* de SED utilizam regras básicas, com operadores lógicos, para representar decisões. Ou ainda, em outros casos mais simples, utilizam apenas porcentagens obtidas em dados históricos para representar as tendências realizadas pelo conjunto das decisões tomadas em algum processo. Assim, a inclusão de algoritmos de inteligência computacional, como RNA, que podem representar melhor uma ação proveniente de uma decisão humana em modelos de SED é um campo de pesquisa que pode ser explorado.

Diante desta questão, o presente trabalho tem como objetivo avaliar o desempenho de bibliotecas de RNA para atuarem na lógica de decisão de modelos de SED. O problema definido para os testes computacionais considerou um sistema de transporte de cargas com o cálculo de emissões de CO. Alguns trabalhos recentes têm demonstrado a aplicação de modelos de SED com cálculo de emissões gasosas em sistemas logísticos. Byrne *et al.*, (2010), por exemplo, apresentaram ferramentas utilizadas na análise ambiental de *supply chain*. Os autores investigaram o uso de SED para capturar a natureza dinâmica das operações e projetos de uma cadeia de suprimentos. Assim, eles analisaram a relação de compensação entre o viés econômico e ambiental. Já Jaegler e Burlat (2012) focaram sua pesquisa nas emissões de CO₂ ao longo da cadeia de suprimentos oriundas desde o gasto energético dos transportes até o armazenamento do estoque. No trabalho de Rangel e Cordeiro, (2015), por sua vez, foi demonstrado o cálculo das emissões de gases do efeito estufa provenientes do transporte de cargas em sistemas logísticos e a possibilidade de utilização do *software* Ururau nesta aplicação.

Os testes realizados consideraram um modelo de simulação de um sistema logístico onde foi avaliada as emissões de CO proveniente dos veículos da frota utilizada no transporte da carga. O modelo de simulação foi construído com o *software* Ururau, que é descrito em detalhes no trabalho de Peixoto *et al.* (2013). O referido *software* possibilita a inclusão de algoritmos de RNA durante a etapa de construção de um modelo SED. A condução dos testes buscou avaliar duas bibliotecas de RNA para realizarem a lógica de decisão nos modelos.

PESQUISA OPERACIONAL PARA O DESENVOLVIMENTO

A organização deste texto segue da seguinte forma. A segunda seção apresenta uma revisão de literatura sobre SED com aspectos ambientais, bem como o *software* Ururau no cálculo de emissões de CO e os *frameworks* das RNA. A terceira seção descreve o cenário do modelo de simulação proposto. A quarta seção descreve os testes e as análises dos resultados de simulação e a comparação das bibliotecas de RNA. Por fim, a quinta seção discute as conclusões obtidas com este trabalho.

2. Revisão de Literatura

A seção foi elaborada para fornecer ao leitor uma base sobre os assuntos discutidos neste trabalho. Assim, esta seção foi dividida nos seguintes itens. Primeiro, foram apresentadas referências bibliográficas sobre simulação discreta com aspectos ambientais. Em seguida foram descritas referências relacionadas com o *software* Ururau como ferramenta viável para o cálculo de emissões de CO. Por fim foram apresentadas referências sobre *Frameworks* de redes neurais artificiais.

2.1 Simulação a eventos discretos com aspectos ambientais

Uma revisão de literatura realizada por Fakhimi *et al.* (2013) forneceu uma visão sintetizada das abordagens de modelagem e simulação que tratavam de problemas ambientais. Neste trabalho, os autores agruparam as publicações por áreas de aplicação. De acordo com os resultados apresentados por eles, 42% dos estudos desenvolveram modelos para o setor de manufatura, 35% estão relacionados à modelagem de ecossistemas, 14% dos modelos estão relacionadas à modelagem regional e urbana, 5% dos artigos desenvolveram modelos para o setor de agricultura, e 3% também utilizaram técnicas de modelagem para abordar os aspectos ambientais no setor de transporte. Os autores ainda dividiram os trabalhos por técnica de simulação e identificaram que 12 % dos artigos utilizaram SED para a modelagem dos sistemas.

Ainda neste tema, segundo dados recentes da *International Energy Agency* (IEA), os dois setores responsáveis pela maior parcela das emissões de CO₂ no mundo são: geração de eletricidade e aquecimento, com 42%, seguidos pelo setor de transporte, com 23%. Além disso, o setor de transportes apresentou uma taxa de crescimento considerada alta (64%), entre os anos de 1990 a 2012, impulsionado, principalmente, pelas emissões do setor rodoviário (IEA Statistics 2014).

Dada a importância dos aspectos ambientais na área de transporte e o baixo número de publicações nessa área identificados no trabalho de Fakimi *et al.* (2013), foi realizada então

PESQUISA OPERACIONAL PARA O DESENVOLVIMENTO

uma busca bibliográfica a base SCOPUS. Isto foi feito para mapear com mais detalhes os trabalhos de SED envolvendo aspectos ambientais na área de logística e cadeia de suprimentos. Esta busca então retornou um total de 10 artigos, apresentados no Tabela 1.

Tabela 1 - Artigos na base SCOPUS no período de 1960 a janeiro de 2015.

Assuntos	ARTIGO(S) IDENTIFICADO(S) NAS BASES SCOPUS
<i>Discrete event simulation / Sustainable / Logistics</i>	(Acaccia, Micheline, and Qualich 2007); (Byrne, Heavey, Ryan, and Liston 2010); (González and Echaveguren 2012);(Jaegler and Burlat 2012); (Jaegler and Burlat 2013); (Seay and Badurdeen 2014); (Tromp, Rijgersberg, and Franz 2010); (van der Vorst, Tromp and Zee 2009); (Rios, Stuart, and Grant 2003); Rangel and Cordeiro (2015)

O conjunto de trabalhos apresentados neste Tabela mostram as aplicações de SED em sistemas logísticos relacionados à questão ambiental. Dentre esses artigos, destacam-se os trabalhos de Byrne et al.(2010) Jaegler e Burlat (2012) , Jaegler e Burlat (2013) e Rangel e Cordeiro (2015) com maior aderência ao escopo do tema aqui tratado. Note ainda que, apesar da busca ter sido realizada no período de 1960 a janeiro de 2015, o primeiro artigo encontrado sobre o assunto data de 2007.

Byrne et al. (2010), em seu trabalho documentam ferramentas usadas na análise da cadeia de suprimentos e investiga o potencial usos de SED como método de capturar a sua natureza dinâmica de projeto e operações. Os autores fazem uma revisão e avaliação do uso de análises quantitativas para apoiar os tomadores de decisão na escolha de um projeto de cadeia de suprimentos mais amigável ambientalmente. Isto foi feito através da modelagem e SED para capturar os fatores de entrada dinâmicos e ilustrar a relação entre os aspectos econômicos e ambientais de um projeto de rede de suprimentos.

Jaegler e Burlat (2012) estudaram em seu trabalho as emissões de CO₂ ao longo da cadeia de suprimentos, desde as emissões do uso de energia do transporte até o armazenamento de estoque. O modelo proposto simula uma cadeia de suprimento de três camadas e quatro variáveis: capacidade de fornecimento, locais, tipo de transporte, e tipos de produtos. O objetivo é comparar os níveis de emissão de CO₂ para as diferentes configurações de cenários. Assim, foi possível apoiar a tomada de decisões e promover redes de colaboração sustentáveis, auxiliando os gestores a selecionar uma cadeia de suprimentos mais sustentável ambientalmente.

Jaegler e Burlat, (2013) também concentram sua pesquisa nas emissões de CO₂ ao longo da cadeia de suprimentos. Assim como o trabalho anterior, foi elaborado um modelo

PESQUISA OPERACIONAL PARA O DESENVOLVIMENTO

baseado na SED. Porém, este usa política de gestão de produção com ponto de reabastecimento e estratégia de resposta a demanda de produção para estoque. A capacidade de produção também foi levada em consideração. Além disso, foram modelados a localização das empresas e seus tipos de produtos. O objetivo foi comparar as emissões de CO₂ equivalentes de diferentes configurações de redes de abastecimento.

Por último, Rangel e Cordeiro (2015) demonstraram como o cálculo das emissões dos gases do efeito estufa provenientes do transporte em sistemas logísticos podem ser analisados com modelos SED. Os autores compararam as variáveis econômicas e ambientais, em contraste com que é feito usualmente nestes tipos de sistemas, onde é feito somente a análise das variáveis econômicas.

Os trabalhos citados abordaram a emissão de gases do efeito estufa na cadeia de suprimentos analisando as variáveis econômicas e ambientais. Assim, estes trabalhos serviram de guia para a elaboração e modelagem do sistema proposto.

2.2 O software Ururau no cálculo de emissões de CO

O Ururau é um *software* para SED livre e de código aberto, que utiliza como base a biblioteca de simulação JSL (*Java Simulation Library*) (ROSSETTI, 2008). Este *software* permite representar sistemas reais por meio da construção de modelos de simulação em interface gráfica (GUI - *Graphic User Interface*) como também em uma API - *Application Programming Interface* (PEIXOTO *et al.*, 2013). Para construir os modelos basta arrastar os módulos para a área de trabalho e os conectar, conforme o fluxo do sistema. Estes elementos possuem um conjunto de funções para utilizar no desenvolvimento do modelo de simulação, maiores explicações são apresentadas no trabalho de Peixoto *et al.* (2015). Silva *et al.* (2012) realizaram uma extensão ao código fonte do *software* Ururau partindo de suas camadas inferiores de forma a possibilitar a comunicação com um módulo de inteligência computacional. Posteriormente, Silva *et al.* (2014) realizaram a implantação de um *framework* no código fonte do Ururau, permitindo executar uma RNA no modelo de forma mais facilitada para pessoas menos especializadas em programação de computadores. Para desenvolvimento e utilização do *software*, é possível acessá-lo livremente em: <http://ururau.ucam-campos.br>.

Atualmente, a equipe de desenvolvimento do *software* construiu um módulo que permite ao *software* calcular a emissão de CO em entidades que trafegam nos modelos. Ou seja, a partir deste recurso, tornou-se possível desenvolver um modelo de um sistema logístico, por exemplo, em que a emissão dos gases emitidos pela descarga dos veículos pode

PESQUISA OPERACIONAL PARA O DESENVOLVIMENTO

ser computada durante as simulações. Para a utilização do referido módulo é necessária a inserção de dados como: o nome da variável que acumulará as emissões e os parâmetros para o cálculo propriamente dito (tempo, coeficiente de emissões, potência do motor, carga do veículo e peso bruto total do veículo).

Por fim, destaca-se que em recente artigo, Dagkakis e Heavey (2015), analisaram o estado da arte de *softwares* livres de SED utilizados para apoio à decisão na área de Pesquisa Operacional. Neste trabalho, os autores citaram o *software* Ururau, dentre quarenta e quatro outros *softwares* e bibliotecas com propósito equivalente, e ressaltaram os seguintes fatos positivos:

- I. comparação dos resultados com o software ARENA;
- II. disponibilidade de interface gráfica para a construção de modelos; e
- III. disponibilidade de videos tutoriais na internet.

2.3 Frameworks de redes neurais artificiais

Segundo Silva *et al.* (2014), a avaliação das RNAs devem levar em consideração aspectos como a arquitetura da rede, algoritmo de treinamento, sistema operacional e tipo de licença do *software*. Desta forma, três dos principais *frameworks* não comerciais foram pesquisados em seu trabalho. São eles: *Java Object Oriented Neural Engine* (JOONE), NEUROPH e ENCOG. Os autores concluíram que o ENCOG mostrou-se mais adequado para criação de RNAs e não apresentou erros de funcionamento e incompatibilidades com o código do *software* Ururau.

Em outro trabalho, Matviyukiv & Faitas (2013) realizaram um estudo comparativo de RNAs com o intuito de selecionar a ferramenta mais adequada para classificação espectral. Os autores testaram o ENCOG 3.1, o JOONE 2 RC1, o NEUROPH 2.6 e *Fast Artificial Neural Network* 2.2 (FANN). Para resolver este tipo de problema, os autores concluíram que a biblioteca ENCOG melhor se adequava, pois apresentava melhor resultado com um número baixo de itens de treinamento. Em um teste preliminar em que um problema XOR foi proposto, ambas as bibliotecas atingiram as saídas esperadas conforme a Tabela 2, porém a FANN gerou resultados com menor número de épocas.

Da mesma forma o Codeproject (2010) realizou um estudo semelhante comparando o JOONE, NEUROPH e ENCOG criando uma RNA *feedforward* para reconhecer uma operação lógica XOR. O ENCOG foi o que melhor se destacou, pois fornece um ambiente limpo, com bom desempenho em termo de tempo total de processamento e uma interface de programação de aplicações fácil de usar.

PESQUISA OPERACIONAL PARA O DESENVOLVIMENTO

Os trabalhos citados anteriormente mostram que o ENCOG apresentou um melhor tempo de processamento em relação aos outros *frameworks*. E, somente o trabalho de Matviykviv & Faitas (2013) comparou este com a FANN. Apesar de apresentar um melhor desempenho de tempo de processamento, esta biblioteca se limita a estruturas de redes neurais multicamadas enquanto o ENCOG trabalha tanto com redes neurais como com outros tipos de aprendizagem de máquina. Como a estrutura da FANN atende aos requisitos para funcionar em conjunto com o código do Ururau e é dedicada exclusivamente a redes neurais optou-se por compará-la com o ENCOG.

ENCOG é um *framework* de aprendizado de máquina que suporta uma variedade de algoritmos avançados, bem como classes de apoio para normalizar e processar dados. Algoritmos de aprendizado de máquina, como máquinas de suporte vetorial, RNA, redes *bayesian*, modelos ocultos de *markov*, programação genética e algoritmos genéticos são suportados. A maioria dos algoritmos de treinamento ENCOG é *multi-threaded* e trabalha bem para *hardware multicore*. Também pode fazer uso de uma *Graphics Processing Unit* (GPU) para aumentar ainda mais a velocidade do tempo de processamento. A biblioteca está em desenvolvimento ativo desde 2008 (HEATON, 2015).

A FANN é uma biblioteca livre de código fonte aberto (LGPL - *Lesser General Public License*) dedicada a RNA enquanto a ENCOG engloba outras funções. Ela implementa RNA multicamadas com suporte para ambas as redes: totalmente conectada e não totalmente conectada. A execução multi-plataforma da FANN suporta tanto ponto fixo quanto flutuante. Esta inclui um quadro para facilitar o manuseio do conjunto de dados de treinamento. É fácil de usar, versátil, bem documentada, e rápida. Outra diferença é que a FANN está disponível em mais de 20 linguagens de programação e a ENCOG em apenas duas. Um artigo de introdução e um manual de referência acompanham a biblioteca com exemplos e recomendações sobre como utilizá-la. Várias interfaces gráficas também estão disponíveis para a biblioteca (FANN HOME PAGE, 2015).

Ao comparar as bibliotecas FANN v 2.2.0. e ENCOG v 3.1.0, Matviykviv & Faitas (2013) encontraram os seguintes resultados para o problema XOR conforme a Tabela 2.

Tabela 2: Comparação entre FANN v 2.2.0. e ENCOG v 3.1.0

Entradas	Saída esperada
{0,0}	0
{1,0}	1
{0,1}	1
{1,1}	0
ENCOG	FANN

PESQUISA OPERACIONAL PARA O DESENVOLVIMENTO

Saída da rede	
0,071271268850058	0,00325358138134
0,913933801069238	0,99557708421431
0,892294065748732	0,99623926833288
0,107168017702071	0,00985109952545
Erro	
0,0095349077961609	0,0010269
Épocas	
46	20

Fonte: adaptado (Matviykv & Faitas 2013).

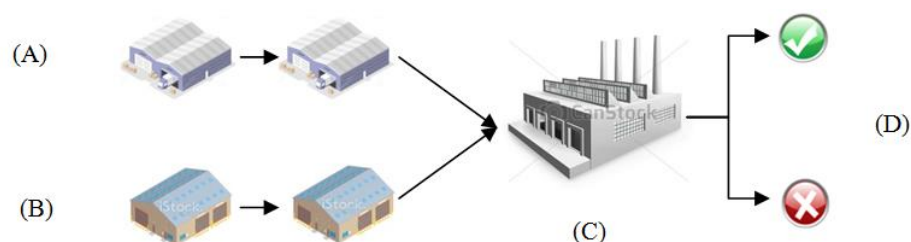
Na Tabela 2 é apresentada uma comparação entre os dois *frameworks* utilizando a mesma técnica de treinamento, *Resilient Propagation*. Este algoritmo apresenta alta velocidade de convergência, exatidão e robustez em relação a parâmetros de treinamento (RIEDMILLER; BRAUN, 1993). O objetivo era alcançar um erro menor que 1,0 %. Assim, o ENCOG alcançou esta meta na 46ª época enquanto a FANN atingiu na 20ª época.

3 Modelo de Simulação

A seguir foram descritos o sistema utilizado no estudo, a metodologia de simulação utilizada e o modelo conceitual. O problema proposto para a realização dos testes neste trabalho se refere a um sistema logístico hipotético, tipicamente utilizado no transporte de cargas. Nele, cada par de fornecedores será analisado separadamente.

Assim, na primeira situação, na região A, o fornecedor 1 encaminha seus produtos para o fornecedor 2 que por sua vez abastece a empresa manufatureira. Esta classifica cada uma das entregas entre os elos da cadeia como aprovadas ou reprovadas. A classificação leva em consideração as emissões de CO dos veículos e o custo de transporte. Na segunda situação isto acontece de forma análoga para os fornecedores da região B, porém com diferenças na localização o que influencia nas emissões e custos. Esta classificação é feita através da função de decisão do Ururau com RNA.

A Figura 1 ilustra o referido sistema. Nela encontram-se os dois pares de fornecedores (A e B), a empresa manufatureira (C) e a classificação das entregas como aprovada ou reprovada (D).



PESQUISA OPERACIONAL PARA O DESENVOLVIMENTO

Figura 1: Ilustração do sistema proposto. (A) – Fornecedores da região A, (B) – Fornecedores da região B, (C) – Empresa Manufatureira, (D) - Entregas aprovadas ou reprovadas.

A metodologia proposta por Banks *et al.* (2010) foi utilizada para a construção dos modelos de simulação apresentados a seguir. Os seguintes passos foram utilizados: concepção (construção do modelo conceitual; coleta de macro-informações e dados, modelagem dos dados de entrada); implementação (construção do modelo computacional, verificação e validação); análise (modelo operacional; projeto experimental; experimentação; interpretação e análise estatística dos resultados; documentação e apresentação dos resultados). A validação e verificação do modelo de simulação seguiram as etapas sugeridas por Sargent (2013).

Os modelos conceituais foram descritos em IDEF-SIM (MONTEVECHI *et al.*, 2010). O sistema do problema proposto foi traduzido para um modelo computacional no Ururau 1.0 que pode ser observado na Figura 2.

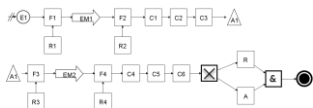


Figura 2: Modelo computacional no *software* Ururau.

O módulo E1 é responsável pela criação de veículos (entidades), estes são gerados no modelo a cada 3 hora. As funções F1 e F3 representam os processos de carregamento e F2 e F4 descarregamento. Os recursos R1 e R3 representam as equipes de carregamento e R2 e R4 as equipes de descarregamento. Os módulos A1 (*jump*) fazem a conexão do modelo. Os módulos EM1 e EM2 são responsáveis pelo transporte de entidades e pelo cálculo das emissões. Para o cálculo das emissões utilizou-se a fórmula que pode ser vista no trabalho de Rangel e Cordeiro (2015). Esta leva em consideração a potência do veículo, tempo de trajeto, carregamento, Peso Bruto Total (PBT) e coeficiente de emissões. Os módulos C1 e C4 geram os custos de cada entidade, C2 e C5 são responsáveis por contar as entidades e C3 e C6 calculam as emissões por entidade. Os módulos A e R fazem a contagem, respectivamente,

PESQUISA OPERACIONAL PARA O DESENVOLVIMENTO

das entregas aprovadas e reprovadas. O módulo marcado com “X” é o decisor em que a RNA atua. Durante o tempo de execução do modelo de simulação, a mesma é ativada sempre que uma entidade precisa passar pelo módulo Decisor e, com isso, precisa tomar uma decisão sobre sua rota. Esta decisão é tomada com base nas emissões por entidades geradas pelos módulos C3 e C6 e os custos obtidos em C1 e C4. O apêndice A mostra os detalhes sobre os parâmetros da simulação.

Para configuração da rede foram utilizados 4 neurônios na camada de entrada e uma camada oculta com 8 neurônios e 1 neurônio na camada de saída conforme a Figura 3.

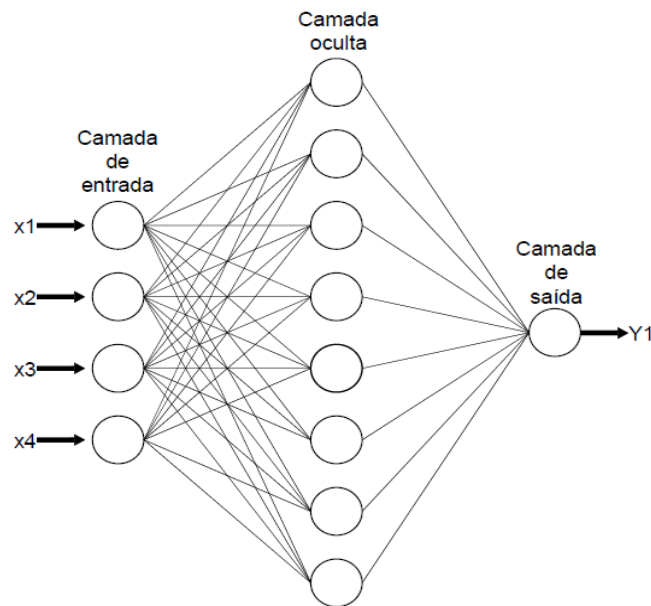


Figura 3: Arquitetura da rede proposta.

O primeiro e o terceiro neurônio da camada de entrada servem para receber as emissões e o segundo e o quarto para os custos gerados por cada entidade. Embora o problema seja linearmente separável, o *software* exige o uso mínimo de uma camada oculta. Assim, o número de neurônios na camada oculta foi determinado de forma experimental, uma vez que na literatura não há um consenso sobre um número ótimo. Quanto maior o número de neurônios na camada oculta, maior é o tempo de processamento. Por isso, buscou-se atingir o menor número de neurônios na camada oculta que não comprometesse a classificação. A camada de saída classifica as entregas como aprovado ou reprovado.

O conjunto de treinamento foi baseado nas emissões e custos gerados, através da entrega realizada por cada fornecedor. O arquivo de treinamento foi criado com os números aleatórios atribuídos pela simulação. Para tal, foi executado o modelo sem o módulo de decisão, apenas, para captar os valores da simulação. Estes valores foram armazenados em

PESQUISA OPERACIONAL PARA O DESENVOLVIMENTO

um arquivo de texto, pelo módulo *Write* do Ururau, que salvou os números gerados pelas variáveis de emissão e custo, para cada elemento criado na simulação durante as replicações. Posteriormente os valores das variáveis emissões e custo foram copiados para uma planilha eletrônica, no qual foram estabelecidos critérios para os mesmos. A estratégia utilizada para determinar os critérios foi arbitrária, onde foi determinado um limite para o custo e a emissão, caso um dos valores superassem essa faixa a entrega era reprovada. Em caso de reprovação atribui-se o valor zero e aprovação um. Por último, o arquivo foi convertido para o formato .txt e o diretório indicado no módulo decisor com RNA do *software*. Cabe ressaltar que o treinamento é executado, apenas uma vez pelo *software* antes das replicações da simulação.

4. Testes e Análise dos Resultados

Para validar os modelos de simulação optou-se por usar funções determinísticas com seus valores arbitrados, de forma a gerar respostas previstas. Uma vez que o modelo apontou resultados esperados conforme as especificações estabelecidas, este foi validado.

Nos testes realizados foram utilizadas distribuições estocásticas, Normal e Triangular. Essas funções e seus parâmetros foram definidos de forma que fosse possível determinar a faixa de valores gerados. Assim, facilitando a delimitação dos limites utilizados para determinar a aprovação ou reprovação das entregas. Os testes e análises dos resultados foram divididos em duas etapas. Primeiro foram apresentados e discutidos os resultados da simulação. Em seguida foi feita a comparação entre as bibliotecas para determinar a de melhor tempo de execução.

Para o carregamento e descarregamento das cargas, foi estabelecido o valor que representa o tempo em que o recurso (equipe operacional) executa a operação. Para a equipe de carregamento foi considerado que a atividade era executada em 4.000 kg por hora, ou seja, para efetuar o carregamento de uma carga de 23.000 kg, seriam necessárias 5,75 horas. Já para a equipe do descarregamento, foi considerada que a atividade era desenvolvida em 6.000 kg por hora.

4.1 Resultados dos testes do modelo de simulação

Os parâmetros do experimento foram definidos com 10 replicações de 1000 horas para cada rodada de simulação. Esse número de replicações foi o suficiente para os resultados da simulação convergirem, dentro de um intervalo de confiança de 95%. Foi executado o teste para os fornecedores da região “A”. Já os fornecedores da região B, têm parâmetros de

PESQUISA OPERACIONAL PARA O DESENVOLVIMENTO

emissões e custos diferentes. Os parâmetros de ambas as regiões foram descritas no Apêndice A.

Os resultados encontrados na execução desses modelos podem ser observados na Tabela 3. Esta Tabela apresenta um resumo do relatório final gerado pelo simulador nestes cenários.

Tabela 3: resultado das simulações dos Fornecedores 1 e 2 de cada região no Ururau.

Região A		
	Fornecedor 1	Fornecedor 2
Variáveis	Média	Média
emissões totais (gCO)	259.377	257.637
custo (unidade monetária)	2.603	2.651
entidades (unidade)	170	167
aprovados (unidade)	116	
reprovados (unidade)	51	
Região B		
	Fornecedor 1	Fornecedor 2
Variáveis	Média	Média
emissões totais (gCO)	282.393	277.391
custo (custo unitário)	2.387	2.673
entidades (unidade)	170	167
aprovados (unidade)	142	
reprovados (unidade)	25	

A região “A” teve um total de emissões de CO equivalente a 517.014gCO (resultado da adição de 259.377gCO e 257.637gCO) e o custo médio por entrega desta região foi de 5.254 unidades monetárias (resultado da adição de 2.603u.m. com 2.651u.m.). Já região “B” teve um maior nível de emissão com 559.784gCO (resultado da adição de 282.393gCO e 277.391gCO) e um custo médio por entrega de 5060 unidades monetárias (resultado da adição de 2.387u.m. com 2.673u.m.). Assim, a região “B” teve um número maior de aproveitamento com 142 aprovações (85%) contra 116 da região “A” (69%).

Os limites utilizados para classificar a aprovação ou reprovação das entregas dos fornecedores pela RNA na Tabela 3 podem ser vistos no Quadro 1. No caso da extrapolação de algum desses valores a rede reprovou a entrega.

Quadro 1: Valores máximos de custos e emissões para aprovação das entregas.

Trecho	Custo (unidade monetária)	Emissão (gCO)
Fornecedor - Fornecedor	2.650	2.500
Fornecedor - Fábrica	2.690	2.200

PESQUISA OPERACIONAL PARA O DESENVOLVIMENTO

Foram analisados os processos de cada um dos fornecedores para verificar número e o tempo das entidades nas filas dos processos, conforme a Tabela 4.

Tabela 4: Resultado da simulação, número e tempo na fila dos processos.

Região A				
Fornecedor 1				
Processos	Número na Fila (unidade)	Desvio Padrão	Tempo na Fila (horas)	Desvio Padrão
F1	81,23	0,00	237,88	0,00
F2	0,01	0,00	0,06	0,02
Fornecedor 2				
Processos	Número na Fila (unidade)	Desvio Padrão	Tempo na Fila (horas)	Desvio Padrão
F3	0,38	0,05	2,23	0,31
F4	0,01	0,00	0,05	0,02
Região B				
Fornecedor 1				
Processos	Número na Fila (unidade)	Desvio Padrão	Tempo na Fila (horas)	Desvio Padrão
F1	81,23	0,00	237,88	0,00
F2	0,01	0,00	0,06	0,02
Fornecedor 2				
Processos	Número na Fila (unidade)	Desvio Padrão	Tempo na Fila (horas)	Desvio Padrão
F3	0,38	0,05	2,23	0,31
F4	0,06	0,00	0,41	0,05

Os processos de carregamento e descarregamento dos fornecedores de ambas as regiões foram considerados idênticos. A divergência no número e tempo nas filas desses processos se dá pelas diferenças nas funções de movimentação (EM1 e EM2).

A Figura 4 mostra as taxas de utilização dos recursos referentes aos processos dos fornecedores das Regiões A e B. Neste gráfico, pode ser verificado que os recursos utilizados nas funções de carregamento (R1 e R3) apresentam taxas de utilização maior do que os recursos utilizados no descarregamento (R2 e R4) para os dois fornecedores de ambas as regiões. Pois o fluxo de entidades nas funções de carregamento é maior comparado com o descarregamento. Estes recursos são utilizados nas funções F1, F2, F3 e F4.

PESQUISA OPERACIONAL PARA O DESENVOLVIMENTO

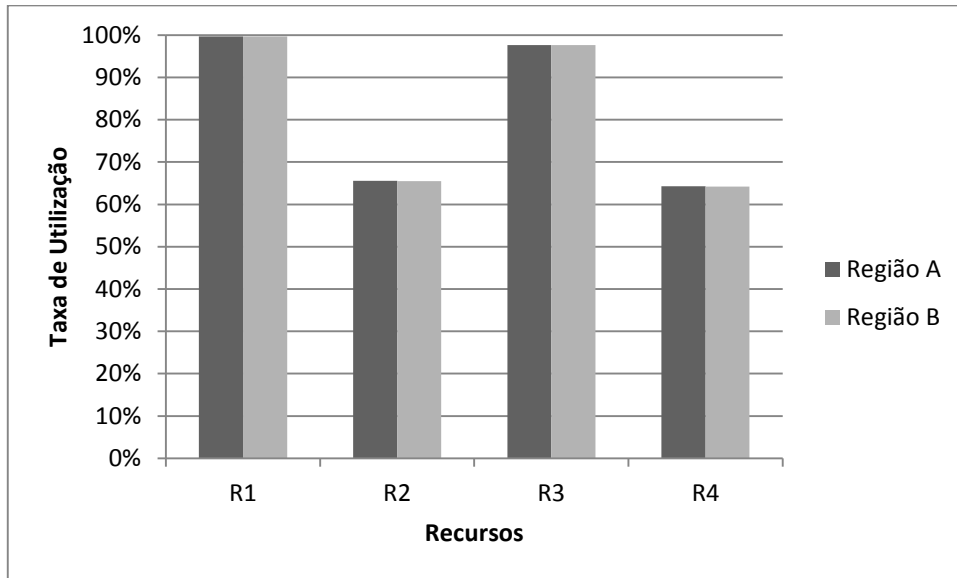


Figura 4: Taxa de utilização dos recursos nas regiões A e B.

4.2 Comparações das bibliotecas

Foi pesquisado o framework FANN para averiguar se o mesmo em conjunto com o Ururau apresenta melhor desempenho quando comparado ao ENCOG que está em uso atualmente no *software*. Esta comparação foi realizada em uma máquina *Dell Inspiron* com processador *Intel® Core™ i3-4130 CPU@3,4GHz*, Sistema operacional Windows 8.1 64bits. A Tabela 5 apresenta esta comparação.

Como pode ser visto na Tabela 5 os parâmetros tempo médio de entidades na fila (F1), quantidade média de entidades na fila (F1) e recursos ocupados (R1) apresentaram resultados similares. Isso mostra que a opção por uma ou outra rede não tem grande influência nos resultados da simulação. Tais resultados já eram esperados, pois ambas as bibliotecas baseiam-se em RNA, com diferenças em suas arquiteturas e algoritmos de funcionamento. Apesar da não influencia dos resultados o objetivo da comparação foi verificar qual delas apresenta o menor tempo total de execução. A biblioteca FANN apresentou um melhor resultado em termos de tempo total de execução 92,69s, sendo 75,68% mais rápido que a ENCOG. Observa-se que a maior influência no tempo total de execução vem do tempo de construção do modelo e treinamento da rede. Esse tempo foi de 380,13s para ENCOG que representa aproximadamente 99,76% do total, enquanto o da FANN foi de 91,49s que representa 98,71%. Já os tempos de simulação apresentam pouca divergência para as duas redes.

Assim a biblioteca FANN é aproximadamente quatro vezes mais rápida do que a ENCOG na aplicação de RNA com o *software* Ururau. Apesar de o problema apresentado ser

PESQUISA OPERACIONAL PARA O DESENVOLVIMENTO

simples, a biblioteca FANN foi inserida no *software*, pois apresentou um melhor tempo de execução. Os testes realizados não englobam todas as possibilidades de análise, assim é possível que em outros tipos de problema a biblioteca ENGOG seja mais indicada. Desta forma, manteve-se também a ENCOG que além de RNA possui outras funções que podem ser incorporadas futuramente. Portanto a versão 1.0 do Ururau possibilita a opção por uma ou outra biblioteca.

Tabela 5: Comparação do ENCOG x FANN no *software* Ururau para Região “A”.

Parâmetro	Experimento 1	Experimento 2
Ferramentas de RNA	ENCOG	FANN
Tempo médio das entidades na fila F1 (horas)	237,88	237,88
Quantidade média das entidades na fila F1(unidade)	81,23	81,23
Recursos ocupados R1(%)	99,7	99,7
Decisão com RNA (unidade)	167	167
Tempo de construção do modelo e treinamento da RNA (s)	380,13	91,49
Tempo de simulação (s)	0,92	1,20
Tempo total de execução	381,05 s	92,69 s
Decisões divergentes entre o experimento 1 e 2	3	
Percentual de decisões divergentes	1,8%	

A quantidade de decisões convergentes entre os experimentos foi alta, apresentando uma diferença média de 3 decisões (1,8%). Assim classificação das redes não apresenta diferença significativa pelo teste t considerando um intervalo de confiança de 95%. Contudo, é difícil afirmar se essa diferença foi consequência das características da simulação ou das bibliotecas como, por exemplo, a arquitetura de rede.

Tanto para o ENCOG quanto para a FANN o erro diminui gradativamente conforme o número de épocas aumenta até se estabilizar gerando uma assíntota. Os gráficos de aprendizagem das redes podem ser observados nas Figuras 5 e 6. Os erros se estabilizaram aproximadamente em 5% e 2%, respectivamente, para o ENCOG e a FANN, isso significa que as redes apresentaram boas taxas de classificação. A menor taxa de erro da FANN em relação ao ENCOG reforça a sua vantagem para aplicações específicas de RNA no *software* Ururau.

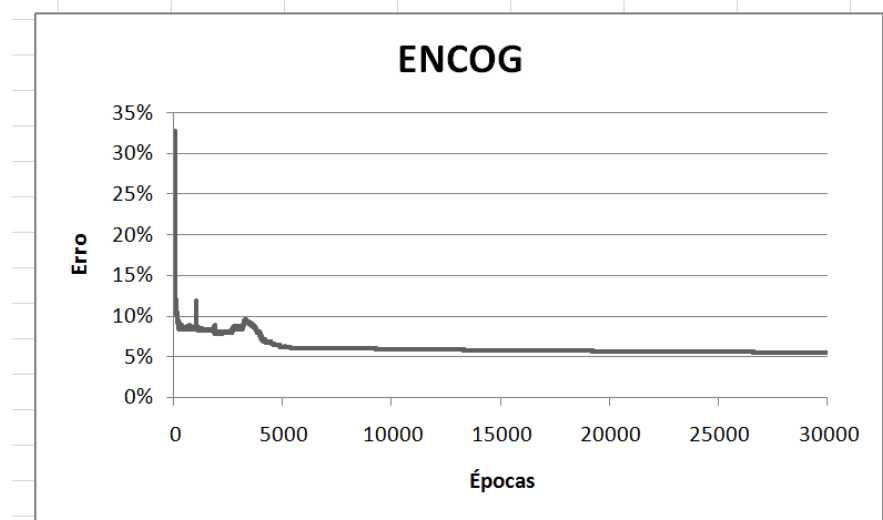


Figura 5: Curva de aprendizagem do ENCOG.

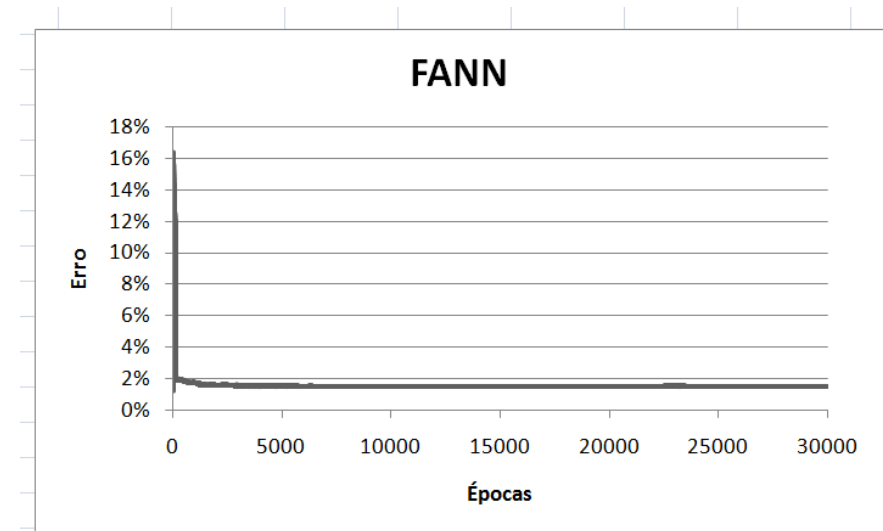


Figura 6: Curva de aprendizagem da FANN.

5. Conclusões

O estudo apresentou uma aplicação do *software* Ururau com um modelo de simulação com lógica de decisão através de uma rede neural. Para isso, foi desenvolvido e descrito um modelo hipotético de simulação de um sistema logístico, onde foram avaliadas as emissões de CO e custos provenientes dos veículos da frota utilizada no transporte da carga, para testar a função de decisão do *software*. A função de decisão "inteligente" realizou a verificação das variáveis avaliadas classificando as entregas dos fornecedores. Dentre as duas regiões apresentadas (região A e B, com distâncias diferentes entre fornecedor e fábrica), a região B (mais próxima) teve um maior número de aprovações nas suas entregas sendo escolhida como alternativa mais viável para o fornecimento, como era o esperado no modelo.

Após as análises, a biblioteca FANN foi acoplada ao *software* Ururau, pois apresentou

PESQUISA OPERACIONAL PARA O DESENVOLVIMENTO

desempenho, em relação ao tempos de execução, quatro vezes mais rápida do que a ENCOG. Além disso, em relação à classificação das redes, as bibliotecas não apresentaram diferença significativa entre elas. Esta modificação passou então a estar presente na versão atual do *software* (Ururau 1.0).

Agradecimentos

Os autores gostariam de agradecer à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e à Fundação de Amparo à Pesquisa do Estado do Rio de Janeiro (FAPERJ) pelo suporte financeiro para esta pesquisa.

Referencias

- Acaccia, G. M. & Micheline, R. C. & Qualich, N. (2007). Sustainable engineering management: end-of-life vehicles with recovery in mind. *World Review of Science, Technology and Sustainable Development*, 4(2/3), 105-125.
- Banks, Jerry & Carson John S. & Barry, Nelson L. & David M-Nicol (2010). *Discrete – Event System Simulation*. Fifth Edition, United States of America, Editora: Pearson Education.
- Bergmann, S. & Stelzer, S. & Strassburger, S. (2014) On the use of artificial neural networks in simulation-based manufacturing control. *Journal of Simulation*, 8, 76-90.
- Byrne, P. J. & Heavey, C. & Ryan, P. & Liston, P. (2010) Sustainable *supply chain* design: capturing dynamic input factors. *Journal of Simulation*, 4(4), 213–221.
- Codeproject. Benchmarking and Comparing ENCOG, NEUROPH and JOONE Neural Networks (2010). Disponível em: <http://www.codeproject.com/Articles/85487/Benchmarking-and-Comparing-ENCOG-NEUROPH-and-JOONE>. Acesso em: 27/03/2015.
- Dagkakis, G. and Heavey, C. (2015). A review of open source discrete event simulation *software* for operations research. *Journal of Simulation advance online publication*, 19 June 2015; doi:10.1057/jos.2015.9.
- Fakhimi, M. & Mustafee, N. & Stergioulas, L. & Eldabi, T. (2013). A review of literature in modeling approaches for sustainable development. In: *Proceedings of the 2013 Winter Simulation Conference*, edited by R. Pasupathy, S. - H. Kim, A. Tolk, R. Hill, and M. E. Kuhl, 282-290. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- FANN home page (2015). Introduction to FANN . Fast Artificial Neural Network Library, I. Disponível em: <http://leenissen.dk/FANN/wp/>. Acesso: 23 de Janeiro de 2015.
- González, V. & Echaveguren, T. (2012). Exploring the environmental modeling of road construction operations using discrete-event simulation. *Automation in Construction*, 24, 100–110.
- Heaton, J. (2015). Encog: Library of Interchangeable Machine Learning Models for Java and C#. *Journal of Machine Learning Research*, 16, 1243-1247.

PESQUISA OPERACIONAL PARA O DESENVOLVIMENTO

- IEA Statistics (2014). CO₂ emissions from fuel combustion-highlights. IEA, Paris. Disponível em: <http://www.iea.org/publications/freepublications/publication/CO2EmissionsFromFuelCombustionHighlights2014.pdf>. Acesso: em 27 de Fevereiro de 2015.
- Jaegler, A. & Burlat, P. (2012) Carbon friendly *supply chains*: a simulation study of different scenarios. *Production Planning & Control*, 23(4), 269–278.
- Jaegler, A. & Burlat, P. (2013). What is the impact of sustainable development on the re-localisation of manufacturing enterprises? *Production Planning & Control*, 25(11), 902–911.
- Matviyukiv, O.M & Faitas, O.I. (2013). Data Classification of Spectrum Analysis Using Neural Network. Lviv Polytechnic National University, Computer-Aided Design Department.
- Montevechi, J.A.B. & Leal, F. & Pinho, A.F. & Costa, R.F.S & Oliveira, M.L.M. & Silva, A.L.F. (2010). Conceptual modeling in simulation projects by mean adapted IDEF: An application in a Brazilian tech company. In: *Proceedings of the 2010 Winter Simulation Conference*, edited by B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, and E. Yücesan, 1624-1635. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Peixoto, T. A. & Rangel, J. J. A. & Matias, I. O (2015). Free and Open-Source Simulation Software "Ururau". In: *Simpósio Brasileiro de Pesquisa Operacional, Porto de Galinhas - PE. XLVII - SBPO - SOBRAPO*, 1, 1-12.
- Peixoto, T. A. & Rangel, J. J. D. A. & Matias, Í. D. O. & Montevechi, J. A. B. & Miranda, R. D. C. (2013). Ururau-um ambiente para desenvolvimento de modelos de simulação a eventos discretos. *Pesquisa Operacional para o Desenvolvimento*, 5(3), 373-405.
- Rangel, J.J.A. & Cordeiro, A.C.A. (2015). Free and Open-Source *Software* for Sustainable Analysis in Logistics Systems Design. *Journal of Simulation (Print)*, 9, 27-42.
- Riedmiller, Martin & Braun, Heinrich (1993). A direct adaptive method for faster backpropagation learning: The RPROP algorithm. In: *Neural Networks, IEEE International Conference on. IEEE*, 586-591.
- Rios, Pedro & Julie Ann Stuart & Ed Grant (2003). Plastics Disassembly versus Bulk Recycling: Engineering Design for End-of-Life Electronics Resource Recovery. *Environmental Science & Technology* 37(23), 5463–5470.
- Rossetti, M. D. (2008). Java Simulation Library (JSL): An Open-Source Object-Oriented Library for Discrete-Event Simulation in Java. *International Journal of Simulation and Process Modelling*, 4, 69-87.
- Sargent R G (2013). Verifications and validation of simulations models. *Journal of Simulation*, 7, 12–24.
- Seay, J. R. & Badurdeen, F. F. (2014). Current trends and directions in achieving sustainability in the biofuel and bioenergy *supply chain*. *Current Opinion in Chemical Engineering*, 6, 55-60.
- Silva, D.V.C. & Rangel, J.J.A. & Matias, I.O. & Vianna, D.S. & Peixoto, T.A. (2012). Modelos de Simulação a Eventos Discretos com Aspectos de Decisão Humana: Uma Aplicação com o Ururau. *PODes - Pesquisa Operacional para o Desenvolvimento*, 4(3), 339-355.

PESQUISA OPERACIONAL PARA O DESENVOLVIMENTO

Silva, M. G. D. & Rangel, J. J. A. & Silva, D. V. C. & Peixoto, T. A. & Matias, I. O. (2014). Decisão com RNA em Modelos de Simulação a Eventos Discretos. *Pesquisa Operacional para o Desenvolvimento*, 6, 299-317.

Tromp, S. O. & H. Rijgersberg & E. Franz (2010). Quantitative Microbial Risk Assessment for *Escherichia Coli* O157:H7, *Salmonella* Enterica, and *Listeria Monocytogenes* in Leafy Green Vegetables Consumed at Salad Bars, Based on Modeling *Supply chain* Logistics. *Journal of Food Protection* 73(10), 1830–40.

Van der Vorst Jack & Seth-Oscar Tromp & Durk-Jouke van der Zee (2009). Simulation Modelling for Food *Supply chain* Redesign; Integrated Decision Making on Product Quality, Sustainability and Logistics. *International Journal of Production Research* 47(23), 6611–6631.

APÊNDICE A - Módulos do Ururau com descrição e dados de configuração referente à região A e B.

Quadro 1: Parâmetros dos fornecedores da região A.

Módulo	Nome	Descrição	Dados de Configuração
Criar	E1	Responsável pela criação de entidades.	T. Chegadas: Const. (3 h) T. Prim. Chegada: 0.0 h Máx. Chegadas: Infinity
Loading Process	F1	Executa um processo	Type: Expression Value: (23000/4000)
Unloading Process	F2	Executa um processo	Type: Expression Value: 23000/6000
Loading Process	F3	Executa um processo	Type: Expression Value: (23000/4000)
Unloading Process	F4	Executa um processo	Type: Expression Value: 23000/6000
Resource Team for Loading	R1	Adiciona um recurso	Capacity 1
Resource Team for Unloading	R2	Adiciona um recurso	Capacity 1
Resource Team for Loading	R3	Adiciona um recurso	Capacity 1
Resource Team for Unloading	R4	Adiciona um recurso	Capacity 1
Emissions	EM1	Calcula as emissões	Variable name: em1 Time(h): NORM(8.45,1.0) Emissions coefficient: 1.5 ; Power: 136 Load(kg): 20000 Legal Combined Total Whole Weight(kg): 23000
Emissions	EM2	Calcula as emissões	Variable name: em2 Time(h): NORM(7.45,1.0) Emissions coefficient: 1.5 ; Power: 136 Load(kg): 23000 Legal Combined Total Whole Weight(kg): 23000
Atribuição	C1	Adiciona uma variável ao modelo.	Tipo: Variável Nome: custo1 Valor: NORM(2600,100)
Atribuição	C2	Adiciona uma variável ao modelo.	Tipo: Variável Nome: entidades1 Valor: entidades1+1
Atribuição	C3	Adiciona uma variável ao modelo.	Tipo: Variável Nome: emissoes_ind1 Valor: em1/entidades
Atribuição	C4	Adiciona uma variável ao modelo.	Tipo: Variável Nome: custo2 Valor: TRIA(2600,2650,2700)
Atribuição	C5	Adiciona uma variável ao modelo.	Tipo: Variável Nome: entidades2 Valor: entidades2+1
Atribuição	C6	Adiciona uma variável ao	Tipo: Variável

PESQUISA OPERACIONAL PARA O DESENVOLVIMENTO

		modelo.	Nome: emissoes_ind2 Valor: em2/entidades2
Atribuição	A	Adiciona uma variável ao modelo.	Tipo: Variável Nome: aprovado Valor: aprovado+1
Atribuição	R	Adiciona uma variável ao modelo.	Tipo: Variável Nome: reprovado Valor: reprovado+1
X (Decisor)	X2	Desvia a execução de uma função.	Tipo: ANN com 2 caminhos Arquivo de Treinamento: treinamento1.txt Algoritmo: ResilientPropagation Exp. Neurônio 1: custo1 Exp. Neurônio 2: emissoes_ind1 Exp. Neurônio 3: custo 2 Exp. Neurônio 4: emissoes_ind2 Exp. Neurônio 5: 0 Nº Neurônios Camada Entrada: 4 Nº Neurônios Camada Oculta: 8 Nº Neurônios Camada Saída: 1

Quadro 2: Parâmetros de emissões e custos diferentes para os fornecedores da região B.

Módulo	Nome	Descrição	Dados de Configuração
Emissions	EM1	Calcula as emissões	Variable name: EM1 Time(h): NORM(9.2,1.0) Emissions coefficient: 1.5 Power: 136 Load(kg): 20000 Legal Combined Total Whole Weight(kg): 23000
Atribuição	C1	Adiciona uma variável ao modelo.	Tipo: Variável Nome: custo Valor: NORM(2400,200)
Emissions	EM2	Calcula as emissões	Variable name: EM1 Time(h): NORM(8.0,2.0) Emissions coefficient: 1.5 Power: 136 Load(kg): 23000 Legal Combined Total Whole Weight(kg): 23000
Atribuição	C4	Adiciona uma variável ao modelo.	Tipo: Variável Nome: custo Valor: TRIA(2500,2700,2800)