

UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE CIÊNCIAS SOCIAIS E HUMANAS
PROGRAMA DE PÓS-GRADUAÇÃO EM ADMINISTRAÇÃO

Adriano Pereira

**UM SIMULADOR PARA ANÁLISE DE REDES DE PRODUÇÃO DA
INDÚSTRIA 4.0**

Santa Maria, RS
2019

Adriano Pereira

UM SIMULADOR PARA ANÁLISE DE REDES DE PRODUÇÃO DA INDÚSTRIA 4.0

Tese de doutorado apresentada ao Programa de Pós-Graduação em Administração da Universidade Federal de Santa Maria (UFSM), como requisito parcial para a obtenção do título de **Doutor em Administração**.

Orientador: Prof. Dr. Eugênio de Oliveira Simonetto

Santa Maria, RS
2019

Ficha gerada com os dados fornecidos pelo autor

Pereira, Adriano
Um Simulador para Análise de Redes de Produção da
Indústria 4.0 / Adriano Pereira.-2019.
128 p.; 30cm

Orientador: Eugênio de Oliveira Simonetto
Tese (doutorado) - Universidade Federal de Santa Maria,
Centro de Ciências Sociais e Humanas, Programa de Pós-
Graduação em Administração, Santa Maria, RS, 2019.

1. Simulador 2. Produção 3. Redes 4. Indústria 4.0 I.
Simonetto, Eugênio de Oliveira II. Pereira, Adriano III. Um
Simulador para Análise de Redes de Produção da Indústria 4.0.

Adriano Pereira

UM SIMULADOR PARA ANÁLISE DE REDES DE PRODUÇÃO DA INDÚSTRIA 4.0

Tese de doutorado apresentada ao Programa de Pós-Graduação em Administração da Universidade Federal de Santa Maria (UFSM), como requisito parcial para a obtenção do título de **Doutor em Administração**.

Aprovado em 16 de dezembro de 2019:

Eugênio de Oliveira Simonetto, Dr. (UFSM)
(Presidente/Orientador)

Guilherme Dhein, Dr. (UFSM)

Eliete dos Reis Lehnhart, Dra. (UFSM)

Goran D. Putnik, PhD. (UMINHO)

Wellington Furtado Santos, Dr. (IFFar)

Santa Maria, RS
2019

AGRADECIMENTOS

Embora muitas vezes pareça uma jornada solitária, a caminhada do doutorado só foi possível pela ajuda, incentivo, conselhos daqueles que estiveram comigo nesse caminho. Agradeço a todos que contribuíram para a concretização deste trabalho:

A Giséli, meu amor, que sempre me apoiou, me incentivou, me aconselhou, me deu forças em tudo que foi preciso.

A minha mãe, Leda, por toda a disposição em ajudar no que eu precisasse e, junto a minha irmã, Aline, sempre acreditarem em mim.

Ao meu orientador, prof. Eugênio de Oliveira Simonetto, por abrir as portas para mim no PPGA, por toda a orientação e por facilitar a realização deste trabalho.

Aos meus colegas do CPD, que permitiram o meu afastamento e, em especial, a equipe de desenvolvimento Java/Web, que segurarem as pontas quando foi preciso.

Aos colegas do PPGA, que vivenciaram os mesmos desafios. Em especial, agradeço ao Glauco e ao Daniel Arenhardt, pela companhia de grupo de pesquisa, pelos artigos, conversas sobre as pesquisas, e também pelos papos aleatórios.

Aos professores do PPGA, que de alguma forma participaram da minha formação como doutor. Aos professores Goran Putnik, pelo auxílio durante o trabalho e, junto aos professores Guilherme Dhein, Eliete dos Reis Lehnhart e Wellington Furtado Santos, pelas contribuições tanto na qualificação quanto na defesa.

Agradeço a todos que contribuíram, da maior à menor forma, para o desenvolvimento deste trabalho.

Muito obrigado!

RESUMO

UM SIMULADOR PARA ANÁLISE DE REDES DE PRODUÇÃO DA INDÚSTRIA 4.0

AUTOR: Adriano Pereira

ORIENTADOR: Eugênio de Oliveira Simonetto

A evolução das tecnologias e dos meios de produção levaram a revoluções na indústria. As três Revoluções Industriais foram marcos desses avanços, e continuam sendo estudados após o seu acontecimento. Em 2011, na Alemanha, surge o termo Indústria 4.0, referindo-se ao que seria a Quarta Revolução Industrial e que, ao contrário das revoluções predecessoras, vem sendo estudada *a priori*, isto é, antes de acontecer ou durante o seu acontecimento. A Indústria 4.0 faz uso de tecnologias como a Internet das Coisas e os Sistemas Ciber-Físicos para a construção de fábricas inteligentes, onde máquinas e humanos interagem e tomam decisões com base em informações obtidas em tempo real. A interação desses atores forma grandes redes complexas de produção, cujo estudo é importante para o avanço da Indústria 4.0, embora seja muito custoso em seu ambiente real. Neste contexto, técnicas de modelagem e simulação computacional podem ser utilizadas para analisar e avaliar diferentes cenários de produção, buscando prever situações futuras. Este trabalho teve como objetivo a construção de um *software* simulador para a análise de redes de produção da Indústria 4.0. Para tanto, fez-se uso da metodologia da *Design Science Research*, um método que prevê a elaboração de construtos para a solução de problemas relevantes, tendo por base o rigor científico e a extensão da solução para uma classe de problemas. A elaboração do simulador foi feita a partir de uma revisão da literatura e definição de um conjunto de requisitos. Foi construído um modelo de produção em redes da Indústria 4.0, e foi elaborado o problema de definição de nós para a execução de uma tarefa de produção nestas redes. Também foram propostos três algoritmos para a definição de nós e caminhos para a execução de tarefas de produção na rede (baseados em busca em largura, busca em profundidade e algoritmo genético); tanto o modelo quanto os algoritmos foram validados. O simulador foi implementado, e testes foram feitos para verificar o comportamento dos diferentes algoritmos e de diferentes estruturas de rede, bem como o comportamento das redes na presença de anomalias. Dentre os três algoritmos propostos, verificou-se que o algoritmo genético obteve melhores resultados quanto à execução de tarefas. Além disso, comprovou-se que o simulador pode ser utilizado para avaliar diferentes cenários, a partir da análise de produção em diferentes configurações de redes, bem como na presença de falhas. Com isso, destaca-se que o trabalho atingiu o seu objetivo, e pode ser utilizado em diferentes áreas para o estudo da produção em redes da Indústria 4.0, servindo como um arcabouço teórico para a análise dessas redes.

Palavras-chave: Simulador. Produção. Rede. Indústria 4.0.

ABSTRACT

A SIMULATOR TO INDUSTRY 4.0 PRODUCTION NETWORKS ANALYSIS

AUTHOR: Adriano Pereira

ADVISOR: Eugênio de Oliveira Simonetto

The evolution of technology and production methods has led to revolutions in industry fields. The first three Industrial Revolution were marks of these advances, and they keep being studied after their occurrence. In 2011, at Germany, the term Industry 4.0 was created. It refers to the Fourth Industrial Revolution, which has been studied before it happens, or while it is happening, differently of the previous three. Industry 4.0 uses technologies such as Internet of Things and Cyber-Physical Systems to build smart-factories, where machines and human interact and make decision based on real time information. The interaction of these actors makes huge and complex production networks, which study is important to the advance of Industry 4.0; nevertheless, this study is too expensive in the real environment. In this context, modeling and simulation techniques can be used to allow analysis and evaluation of different production scenarios. This work had as goal the development of a software simulator for the analysis of Industry 4.0 production networks. Therefore, Design Science Research method was used. This method is used when the research aims to build constructs to solve relevant problems, based on scientific rigor and the extension of the solution to a class of problems. The simulator development began with literature review and requirements definition. After that, an Industry 4.0 production network model was created and, then, three algorithms were proposed to define nodes and paths to execute production tasks in the network. The algorithms were based on depth-first search, breadth-first search and genetic algorithm metaheuristic. Both production model and the algorithms were validated. The simulator was implemented as a software, and tests were performed to check different algorithms and network structures behaviors, as well as networks behaviors with failures. Among the three proposed algorithms, genetic algorithm obtained better results regarding to tasks execution. Besides, it was proved that the simulator can be used to evaluate different scenarios, base on production analysis in different networks configurations, as well as in the presence of failures. Then, the work achieved its goal, and it can be used in different areas of Industry 4.0 production networks study, being a theoretical framework to these networks analysis.

Keywords: Simulation. Production. Network. Industry 4.0.

LISTA DE FIGURAS

Figura 1 – Exemplo de crossover.....	55
Figura 2 – Exemplo de mutação.....	55
Figura 3 – Fluxograma da Design Science Research.....	61
Figura 4 – Exemplo de rede de produção.....	64
Figura 5 – Nós testados com 0, 1 e 2 passo(s).....	69
Figura 6 – Busca em profundidade.....	71
Figura 7 – Busca em largura.....	72
Figura 8 – Exemplo de tarefa e solução como um cromossomo.....	74
Figura 9 – Exemplo de crossover uniforme.....	76
Figura 10 – Interface inicial do simulador com rede carregada.....	78
Figura 11 – Visualização de detalhes de um nó.....	79
Figura 12 – Informações de habilidades.....	80
Figura 13 – Distribuição de grau dos nós da rede.....	81
Figura 14 – Edição da rede.....	82
Figura 15 – Criação de pesos aleatórios para arestas.....	82
Figura 16 – Criação de habilidades aleatórias.....	83
Figura 17 – Remoção de nós.....	84
Figura 18 – Tela de alteração de habilidades.....	85
Figura 19 – Tela de execução de uma tarefa por busca em largura ou profundidade.....	86
Figura 20 – Tela de execução de tarefas aleatórias com os algoritmos de busca de largura ou profundidade.....	87
Figura 21 – Tela de execução de uma tarefa utilizando o algoritmo genético.....	88
Figura 22 – Tela de execução de tarefas aleatórias com o algoritmo genético.....	89
Figura 23 – Resultado da execução de uma tarefa utilizando o Algoritmo Genético.....	90
Figura 24 – Execução de tarefas aleatórias na rede.....	101
Figura 25 – Total de tarefas executadas por tamanho da rede e quantidade máxima de passos. Redes com grau médio 2.....	102
Figura 26 – Total de tarefas executadas por tamanho da rede e quantidade máxima de passos. Redes com grau médio 4.....	103
Figura 27 – Total de tarefas executadas por tamanho da rede e quantidade máxima de passos. Redes com grau médio 8.....	103
Figura 28 – Total de tarefas executadas por tamanho da rede e quantidade máxima de passos. Redes com grau médio 16.....	103
Figura 29 – Total de tarefas executadas por tamanho da rede e quantidade máxima de passos. Redes com grau médio 32.....	104

LISTA DE TABELAS

Tabela 1 – Comparação de algoritmos com solução ótima – valores médios de 5 tarefas.....	93
Tabela 2 – Resultado da execução de 1000 tarefas em três tipos de redes.....	97
Tabela 3 – Total de passos e tarefas executadas.....	100
Tabela 4 – Quantidade de tarefas completadas com remoção de 1% dos nós.....	105
Tabela 5 – Quantidade de tarefas completadas com remoção de 5% dos nós.....	105
Tabela 6 – Quantidade de tarefas completadas com remoção de 10% nós.....	106
Tabela 7 – Quantidade de tarefas completadas com remoção de 15% nós.....	106
Tabela 8 – Quantidade de tarefas completadas com remoção de 20% nós.....	106

LISTA DE ABREVIATURAS E SIGLAS

AG	Algoritmo Genético
CAD	<i>Computer-Aided Design</i>
CPS	<i>Cyber Physical System</i> (Sistema Ciber-Físico)
DS	<i>Design Science</i>
DSR	<i>Design Science Research</i>
DT	<i>Digital Twins</i>
ERP	<i>Enterprise Resource Planning</i> (Sistemas de Gestão Empresariais)
i4.0	Indústria 4.0
IoS	<i>Internet of Services</i>
IoT	<i>Internet of Things</i> (Internet das Coisas)
JSON	<i>JavaScript Object Notation</i>
SAD	Sistema de Apoio à Decisão
SED	Simulação de Eventos Discretos

SUMÁRIO

1	INTRODUÇÃO.....	21
1.1	PROBLEMA DE PESQUISA.....	23
1.2	OBJETIVO DO TRABALHO.....	23
1.3	JUSTIFICATIVA.....	24
1.4	ORGANIZAÇÃO DO TEXTO.....	28
2	REVISÃO BIBLIOGRÁFICA.....	29
2.1	INDÚSTRIA 4.0.....	29
2.2	REDES COMPLEXAS.....	33
2.2.1	Definição, Representação e Características Estruturais.....	36
2.2.1.1	Grau (<i>degree</i>) e distribuição de grau.....	37
2.2.1.2	Caminhos e distâncias.....	37
2.2.1.3	Coefficiente de aglomeração (clusterização).....	38
2.2.1.4	Resistência e robustez.....	38
2.2.2	Topologia de redes complexas.....	38
2.2.3	Modelos de Redes.....	39
2.2.3.1	Grafos aleatórios.....	39
2.2.3.2	Grafos de pequenos mundos (small-worlds).....	40
2.2.3.3	Grafos livres de escala.....	40
2.3	MODELAGEM E SIMULAÇÃO COMPUTACIONAL.....	41
2.4	TRABALHOS RELACIONADOS.....	44
2.5	HEURÍSTICAS E ALGORITMO GENÉTICO.....	52
3	MÉTODO DE PESQUISA.....	56
4	UM SIMULADOR DE PRODUÇÃO EM REDES COMPLEXAS DA INDÚSTRIA 4.0.....	62
4.1	O PROBLEMA DE DEFINIÇÃO DE NÓS PARA EXECUÇÃO DE UMA TAREFA DE PRODUÇÃO EM REDES DA INDÚSTRIA 4.0.....	67
4.2	EXECUÇÃO DA TAREFA UTILIZANDO BUSCA EM PROFUNDIDADE E BUSCA EM LARGURA.....	69
4.3	UMA ABORDAGEM BASEADA EM ALGORITMO GENÉTICO PARA O PROBLEMA DA DEFINIÇÃO DE NÓS PARA A EXECUÇÃO DA TAREFA.....	72
4.4	DESENVOLVIMENTO DO SIMULADOR.....	77
4.5	VALIDAÇÃO DOS ALGORITMOS PROPOSTOS.....	90
4.6	COMPARAÇÃO COM O SOLVER LINGO E ALGORITMO ÓTIMO.....	91
4.7	COMPARAÇÃO ENTRE ALGORITMOS.....	94
5	USO DO SIMULADOR PARA AVALIAÇÃO DE HIPÓTESES E DIFERENTES CENÁRIOS DE PRODUÇÃO DA INDÚSTRIA 4.0.....	96
5.1	EXECUÇÃO DE TAREFAS EM DIFERENTES TIPOS DE REDES.....	96
5.2	RELAÇÃO ENTRE O TAMANHO DA REDE E SUA CONECTIVIDADE COM A CAPACIDADE DE EXECUÇÃO DE TAREFAS.....	98
5.2.1	Análise da hipótese número 1 (H1).....	99
5.2.2	Análise da hipótese número 2 (H2).....	102
5.2.3	Análise da hipótese número 3 (H3).....	104
6	DISCUSSÃO DE RESULTADOS E CONSIDERAÇÕES FINAIS.....	108
6.1	GENERALIZAÇÃO PARA UMA CLASSE DE PROBLEMAS.....	110
6.2	COMUNICAÇÃO DE RESULTADOS.....	111

6.3	LIMITAÇÕES DA PESQUISA.....	112
6.4	TRABALHOS FUTUROS.....	112
	REFERÊNCIAS.....	114
	APÊNDICE A – EXEMPLO DE CÓDIGO JSON DE UMA REDE.....	125
	APÊNDICE B – CÓDIGO LINGO.....	127

1 INTRODUÇÃO

Sistemas tradicionais de manufatura estão ficando obsoletos, na medida em que dinâmicas modernas de produção forçam mudanças nos processos, tornando-os cada vez mais integrados em cadeias globais de recursos e *stakeholders* (PUTNIK et al., 2012). Agilidade, reações rápidas às mudanças e proatividade são características importantes para responder às exigências de competitividade do mercado moderno (PUTNIK et al., 2012). O panorama da manufatura está cada vez mais complexo e dinâmico, como consequência da globalização e da recessão econômica recente, aumentando a importância do projeto de redes de cooperação entre companhias para suportar a competição (MOURTZIS; DOUKAS; PSAROMMATIS, 2013).

Os paradigmas de produção têm sido alterados ao longo do tempo, dirigidos pelos avanços tecnológicos em eventos definidos como Revoluções Industriais, que usualmente foram estudados após acontecerem (LASI et al., 2014). A mais recente Revolução Industrial, que guiará (ou está guiando) as mudanças no processo produtivo, vem sendo chamada de Indústria 4.0 (i4.0) – ou Quarta Revolução Industrial – (HERMANN; PENTEK; OTTO, 2016) e tem como base os avanços das tecnologias digitais, combinando recursos da Internet e objetos “inteligentes” em sistemas modulares de produção capazes de gerirem seu processo de manufatura, permitindo tanto a customização de produtos quanto condições econômicas para uma produção em massa (LASI et al., 2014). Neste aspecto, humanos e máquinas devem estar integrados, formando uma rede com entidades que podem estar localizadas em posições geográficas diferentes, fornecendo produtos e serviços de forma autônoma (SILVA; SANTOS FILHO; MIYAGI, 2015).

Mudanças vem ocorrendo em processos de manufatura, projeto, produto, operações e sistemas relacionados com a produção, podendo aumentar o ciclo de vida dos produtos, bem como agregar valor a toda cadeia organizacional (FIRJAN, 2016). Diferentemente das outras três Revoluções Industriais, observadas e estudadas após ocorrerem, a Indústria 4.0 vem sendo prevista e estudada *a priori*, provendo uma série de oportunidades, tanto no ramo acadêmico quanto no empresarial (HERMANN; PENTEK; OTTO, 2016).

A Indústria 4.0 vem sendo apresentada como um conjunto de mudanças gerais em todas as áreas de uma organização, incluindo aí processos de manufatura, causadas pela digitalização e automatização (TRSTENJAKA; COSICA, 2017). Diversas áreas do conhecimento estão estudando a Indústria 4.0, como engenharia elétrica, administração, ciência da computação, engenharia de sistemas de informação e negócio, engenharia

mecânica, entre outros (LASI et al., 2014). A Indústria 4.0 é suportada pela introdução de tecnologias da Internet na indústria, e está intimamente ligada ao conceito de Sistema Ciber-Físico (*Cyber Physical System* – CPS) (DRATH; HORCH, 2014). No ambiente de manufatura, os CPS integram máquinas inteligentes, sistemas de armazenamento e instalações de produção capazes de trocarem informações e dispararem ações de forma autônoma, formando redes globais (KAGERMANN; WAHLSTER; HELBIG, 2013). CPS consistem na fusão dos níveis digitais e físicos, de forma que as representações não podem ser razoavelmente diferenciadas (LASI et al., 2014).

Ao utilizar recursos da Internet, com CPS interagindo entre si através de atividades proativas, a i4.0 institui a construção de grandes redes que conectam seus diferentes componentes (BRETTEL et al., 2014). Uma rede é um conjunto de vértices (ou nós) relacionados entre si através de arestas; diferentes sistemas têm a estrutura de rede, que também são chamadas de grafos (NEWMAN, 2003). Redes são estudadas tanto para investigar as partes que as compõem, quanto para entender o relacionamento entre essas partes e o seu comportamento como um todo (FIGUEIREDO, 2011). As redes que possuem estrutura irregular, complexa, alterada dinamicamente ao longo do tempo, compostas por milhares ou milhões de nós, são conhecidas como *redes complexas* (BOCCALETTI et al., 2006), como é o caso das redes formadas na Indústria 4.0. Sistemas complexos têm como característica a auto-organização, isto é, capacidade de arranjo (adaptação) sem nenhum princípio de organização externa sendo aplicada (AMARAL; OTTINO, 2004).

Em geral, não é possível estudar redes complexas diretamente em funcionamento, tendo em vista que, por exemplo, a inclusão de uma falha na rede pode gerar prejuízos financeiros catastróficos (BRASE; BROWN, 2009). Criar ambientes com componentes reais para teste também pode ser demasiadamente custoso; portanto, o uso de modelagem e simulação computacional é uma alternativa segura e de baixo custo para o estudo das redes complexas (BRASE; BROWN, 2009). Além disso, cada vez mais se tem estudado redes maiores (em número de nós e conexões), devido à melhoria das capacidades computacionais disponíveis: de redes com dezenas ou centenas de vértices, passou-se a estudar redes com milhares de nós (NEWMAN, 2003).

Questões de estudo podem variar de acordo com o tamanho das redes: em redes pequenas, por exemplo, pode-se estudar qual o nó que, quando removido, terá maior impacto na conectividade; porém, em grandes redes, isso passa a perder o sentido (NEWMAN, 2003). Nestes casos, uma questão interessante pode ser a porcentagem de vértices que precisam ser removidos para afetar, significativamente, a conectividade da rede (NEWMAN, 2003).

Os elementos básicos de redes complexas de produção são unidades de trabalho, conectadas entre si, que compartilham informações e permitem a resolução de demandas de forma integrada com a participação de mais de um elemento, o que pode ser visto como cooperação quando uma unidade em particular não está disponível para satisfazer uma demanda, mas consegue completar a tarefa com o auxílio de outras unidades interconectadas (PUTNIK et al., 2012). No contexto da Indústria 4.0, diferentes atores podem representar uma unidade de trabalho, desde um sensor até uma organização completa, passando por pessoas e máquinas de uma planta fabril. Esses atores estão inter-relacionados, formando uma rede complexa de produção. As demandas na rede podem ser por matéria-prima, produtos, serviços, informações, enfim, qualquer item necessário no processo produtivo.

1.1 PROBLEMA DE PESQUISA

A Indústria 4.0 é um marco de mudanças nos processos de produção, estudada *a priori*, que envolve estruturas de redes complexas, compostas por uma imensidão de unidades de produção e consumo, que interagem entre si. Diante do exposto, emerge a questão que orientou o desenvolvimento deste estudo: como é possível avançar os estudos de gestão da Indústria 4.0, em especial na análise de redes complexas de produção, possibilitando a predição de fenômenos e a avaliação de cenários?

1.2 OBJETIVO DO TRABALHO

Diante do problema elencado, esta pesquisa tem como objetivo a construção de um *software* simulador para análise de redes de produção da Indústria 4.0, que possibilite a predição e o estudo de fenômenos relacionados com o processo produtivo, contribuindo para o avanço do conhecimento relacionado à gestão na Indústria 4.0.

Para atingir o objetivo principal do trabalho, uma série de objetivos específicos deverão ser alcançados. Estes objetivos estão descritos a seguir:

1. Identificar e analisar *softwares* para a simulação de produção em redes complexas da Indústria 4.0.
2. Definir um conjunto de requisitos de um simulador de redes de produção da i4.0.
3. Definir um modelo para simular tarefas de produção em redes da Indústria 4.0.
4. Elaborar a definição do problema de determinação de nós para a execução de uma tarefa de produção em redes da i4.0.

5. Construir métodos para a definição de nós que devem atender a uma tarefa de produção nessas redes.
6. Desenvolver um *software* simulador de redes complexas de produção da i4.0.
7. Validar os modelos e algoritmos propostos, bem como o simulador desenvolvido.
8. Avaliar o funcionamento do simulador a partir da análise de diferentes cenários de produção em redes da Indústria 4.0.

1.3 JUSTIFICATIVA

O progresso tem levado à descentralização da produção: modelos centralizados das primeiras Revoluções Industriais não são mais tão aplicáveis e, agora, as fábricas vêm se tornando menores e estão mais próximas dos consumidores, utilizando cadeias de suprimento dinâmicas. A manufatura distribuída está utilizando tecnologias que vem se tornando mais maduras (SRAI et al., 2016).

A Indústria 4.0 será impulsionada pelas tecnologias da Internet, que possibilitam a comunicação entre humanos e máquinas em Sistemas Ciber-Físicos, através de grandes redes (BRETTEL et al., 2014). A Internet é um instrumento poderoso para o gerenciamento de sistemas distribuídos, capazes de obter informações individuais de produtos em toda a cadeia produtiva (BRETTEL et al., 2014).

Mudanças estão ocorrendo e ocorrerão com o avanço da i4.0. Profissões serão afetadas, como a de planejador de produção: a comunicação entre máquinas será cada vez mais necessária, levando à remoção (ou redução) da interação humana em partes cruciais do processo produtivo, automatizando a tomada de decisão para operações de seleção, sequenciamento e escalonamento da produção (TRSTENJAKA; COSIC, 2017). Dessa forma, segundo os autores, *softwares* serão necessários para a execução das tarefas outrora realizadas por um operador humano.

Softwares de planejamento da produção farão parte dos novos sistemas computacionais da Indústria 4.0, podendo estar relacionados aos Sistemas de Gestão Empresariais (*Enterprise Resource Planning – ERP*), utilizar tecnologia de computação em nuvem e Internet das Coisas para manipular grandes quantidades de dados em tempo real (TRSTENJAKA E COSIC; 2017). Neste contexto, modelagem e simulação permitem que unidades descentralizadas flexibilizem seus produtos, agilizando processos de inovação na produção (BRETTEL et al., 2014).

Softwares podem ser utilizados para auxiliar a tomada da decisão, que está cada vez mais complexa com a Indústria 4.0; técnicas de simulação podem ser utilizadas por

fornecerem uma análise rápida de dados a um baixo custo e com baixo risco. Se, por um lado, a indústria reconhece as limitações e desafios das práticas atuais, por outro, a academia vem estudando métodos e técnicas para testes de metodologias e tecnologias, que podem ser empregados na indústria, no contexto da i4.0 (MOURTZIS et al., 2019).

O número de publicações relacionadas à simulação vem crescendo ao longo dos anos; o uso de simulação para aprendizagem deve ser evoluído e difundido, entre diferentes áreas de aplicação, embora hoje esteja principalmente relacionado com os setores de automóveis e aviação (MOURTZIS; DOUKAS; BERNIDAKI, 2014). A modelagem de sistemas de produção da i4.0 auxilia a prever potenciais atrasos, interrupções e falhas, possibilitando seu tratamento em tempo, melhorando a produtividade e a eficiência desses sistemas (LONG; ZEILER; BERTSCHE, 2018).

Um conceito de simulação que pode ser aplicado em sistemas de manufatura é o de *Digital Twins* (DT); trata-se de uma técnica de representação do espaço físico em um ambiente computacional, sincronizando dados em tempo real, que pode ser utilizada para lidar com fábricas inteligentes (TAN et al., 2019), como as da i4.0. Trata-se, segundo Tan et al. (2019), de uma representação do sistema de produção, apta a utilizar dados de sensores, dispositivos inteligentes conectados e modelos matemáticos.

Ao virtualizar o processo de manufatura, desde o projeto até a produção, operacionalização e manutenção, a técnica de DT integra toda a fábrica no modelo virtual de simulação, facilitando o controle e a gestão no espaço físico, podendo ser utilizada para avaliação e otimização do processo (TAN et al., 2019).

DT é um conceito que aplica os CPS ao campo da manufatura; comparado aos sistemas tradicionais de monitoramento, DT não apenas representa o mundo real, como também realiza previsões, provê soluções e suporte à tomada de decisão antes da ocorrência de uma falha ou anomalia, além de simular possíveis eventos futuros. DT está sendo listado como um dos maiores equipamentos de manufatura para suportar a Indústria 4.0. (TAN et al. 2019).

Para Viera et al. (2018), a simulação de eventos discretos é um campo importante de estudo na Indústria 4.0; trata-se de uma técnica sólida, com vários anos de aplicação, que pode ser utilizada para resolução de diferentes tipos de problemas. Os autores propõem uma agenda para o uso de simulação de eventos discretos na pesquisa da i4.0, que engloba 6 itens, resumidos a seguir:

1. Simulação de eventos discretos como uma ferramenta de instrução: a simulação pode ser utilizado para demonstrar e ilustrar, através de animações, a dinâmica

associada aos negócios e processos das organizações, podendo ser utilizada como ferramenta de instruções. Realidade Virtual e a habilidade de testar cenários alternativos podem ser consideradas vantagens, especialmente em contexto de i4.0, caracterizado pela dinâmica de negócios e processos.

2. Utilização de simulação para verificar os ganhos da implantação da Indústria 4.0, o que pode ser feito a partir da comparação de cenários com e sem características da i4.0.
3. Modelos reutilizáveis: permitir a criação de submodelos que possam ser reutilizados e compartilhados.
4. *Digital Twin* através da simulação de eventos discretos: possibilitar testes de cenários com a virtualização de toda a fábrica em tempo real, utilizando submodelos para, combinados, modelar todos os processos.
5. Simulação de cadeia de suprimento em tempo real: devido aos altos níveis de incerteza e variabilidade da cadeia de suprimento, a simulação pode ser a técnica mais efetiva para a solução de problemas relacionados. Na i4.0, é importante haver a habilidade de visualização dessas complexas redes, incluindo logística interna, bem como a habilidade simular modelos de dados em tempo real.
6. Simulação de eventos discretos utilizando *big data*: tendo em vista que uma das consequências da implantação da i4.0 é a criação de grandes volumes de dados (*big data*), é importante possibilitar a simulação utilizando esses dados.

Além das fábricas, os produtos também estão se tornando “inteligentes”, sendo capazes de carregar informações e possuir inteligência para fornecer *feedback* para os sistemas de manufatura, que podem utilizar essa informação para otimizar o processo de produção; nas fábricas, recursos estão cada vez mais conectados, melhorando o controle do processo e da qualidade, a partir de trocas de informações (TRSTENJAKA E COSIC; 2017). A Indústria 4.0, portanto, necessita de incrementos em termos de *software*, e ligações entre as diversas partes do processo de manufatura e cadeias de suprimento (TRSTENJAKA E COSIC; 2017).

Os avanços em tecnologias de informação e comunicação possibilitam o monitoramento de grandes quantidades de dados em tempo real; isso se dá, entre outras coisas, pela diminuição do preço dos sensores, e permite a comunicação em diferentes níveis hierárquicos, utilizando a infraestrutura de rede da Internet (BRETTEL et al., 2014). A utilização de engenharia integrada com informações disponíveis em tempo real, ao longo da

cadeia de valores, possibilita a otimização dos processos, tornando cada vez *menos* importante qual unidade em particular está executando cada processo específico (BRETTEL et al., 2014).

Processos podem ser modelados de forma global, com centralização de dados, e soluções não ótimas podem ser permitidas em uma unidade para resolver gargalos em outras (BRETTEL et al. 2014). Se, hoje, decisões e adaptações em processos são feitas principalmente por humanos, no futuro este processo deve ser automatizado, em processos de decisões autônomas (BRETTEL et al., 2014). Dessa forma, espera-se que os resultados da simulação do sistema, utilizando informações de cada nó (unidade de trabalho), possam ser utilizados no processo de tomada de decisão.

A i4.0 traz uma série de desafios para pesquisa. Modelar um sistema com sensores globais e uma grande quantidade de dados é um desafio: enquanto simulações podem lidar com sistemas em todas as suas complexidades, dados de sensores devem manter-se atualizados – informações chegando e mudando – e a simulação de um sistema deve ser eficiente e dinâmica. A conexão e integração entre os diversos componentes de sistemas criam um novo desafio, com mais variáveis de decisão (XU et al., 2016).

Fábricas de semicondutores são exemplos de local onde pode ser aplicada simulação em um cenário de Indústria 4.0. Diferentes níveis podem ser utilizados para modelar e simular a fábrica: (i) a um nível de fábrica completa, para a qual dados na casa de *gigabytes* podem ser gerados pelos sensores por segundo; (ii) em nível de compartimento (no qual um compartimento é um grupo de máquinas compartilhando uma mesma zona de transportadores), pode-se analisar parte da fábrica com centenas de *megabytes* por segundo; bem como (iii) em nível de máquina, gerando poucos *megabytes* por segundo, com tomada de decisão local – sem grandes interações (XU et al., 2016).

A i4.0 oferece muitas oportunidades para melhorias de paradigmas, facilitando a visão e execução de automação inteligente, permitindo processos mais rápidos, flexíveis e eficientes, com redução de custos (XU et al., 2016). A partir de redes colaborativas, é possível modelar e simular os passos dos processos de produtos, levando em conta fronteiras interorganizacionais (BRETTEL et al., 2014).

Assim, pode-se dizer a i4.0 é um fenômeno em consolidação, que está modificando e modificará os processos produtivos. Se, por um lado, esta é a primeira vez que uma Revolução Industrial vem sendo estudada antes de seu acontecimento, ou enquanto ocorre, por outro, verifica-se a dificuldade em testar ambientes reais, tanto pelo seu custo elevado, quanto pela baixa quantidade de organizações que já operam com base nos princípios da i4.0. Dessa forma, uma maneira de estudar a i4.0 é através do uso de técnicas de modelagem e

simulação computacional, que possibilitam a construção e avaliação de cenários similares aos reais, sem riscos aos sistemas físicos representados.

Embora o número de trabalhos de simulação venha aumentando, não foi encontrado, em literatura, simulador semelhante ao proposto nesta pesquisa. Dessa forma, este trabalho é relevante ao desenvolver um simulador para a análise de redes de produção da Indústria 4.0. Com ele, espera-se que diferentes cenários possam ser avaliados, atendendo a pontos da agenda de pesquisa em simulação para a i4.0 de Viera et al. (2018), contribuindo para o avanço da gestão da produção em ambientes da i4.0. O simulador serve como arcabouço para a pesquisa de produção em redes da Indústria 4.0, podendo ser utilizado em diferentes campos, e também estendido e adaptado para outros problemas semelhantes. Acredita-se que o simulador possa ser incluído, por exemplo, em ambientes de *Digital Twins*, em fábricas inteligentes, auxiliando a tomada de decisão ao receber informação de diferentes fontes e simular cenários alternativos.

1.4 ORGANIZAÇÃO DO TEXTO

Este texto está organizado em 6 capítulos, levando em conta esta Introdução. O capítulo 2 traz uma revisão da bibliográfica, abordando os principais tópicos envolvidos neste trabalho. O capítulo 3 aborda o método utilizado para a condução desta pesquisa. Na sequência, o capítulo 4 aborda o simulador de redes de produção desenvolvido neste trabalho. No capítulo 5 exibe-se a utilização do simulador, a partir da execução de tarefas em diferentes tipos de redes; e da discussão e análise de três hipóteses a respeito da relação entre o tamanho da rede e sua conectividade, e a execução de tarefas de produção. Por fim, o capítulo 6 traz as considerações finais do trabalho, analisando os resultados obtidos, elencando as limitações da pesquisa e indicando possíveis trabalhos futuros.

2 REVISÃO BIBLIOGRÁFICA

Neste capítulo, pontos relacionados ao trabalho são discutidos. A seção 2.1 traz conceitos relevantes em relação à Indústria 4.0; na seção 2.2, há uma discussão a respeito das redes complexas; já a seção 2.3 aborda a modelagem e simulação computacional; na seção 2.4 são discutidos trabalhos e sistemas semelhantes ao proposto; e, por fim, a seção 2.5 traz uma discussão sobre heurísticas e meta-heurísticas, em especial, o algoritmo genético.

2.1 INDÚSTRIA 4.0

O termo Indústria 4.0 se refere à Quarta Revolução Industrial, e foi utilizado pela primeira vez na Feira de Hanover, na Alemanha, em 2011, para definir o fenômeno que segue as três primeiras revoluções industriais, ocorridas em um período de 200 anos (DRATH; HORCH, 2014) como um resultado da (i) mecanização, (ii) eletricidade e (iii) tecnologias da informação (KAGERMANN; WAHLSTER; HELBIG, 2013). A primeira revolução, datada da década de 1780, foi orientada pelos teares mecânicos dirigidos por motores a vapor e deu início a uma significativa mudança, com a centralização de fábricas e o crescimento da produção; a segunda revolução aconteceu 100 anos após a primeira, e teve seu auge com a produção do Ford T nos Estados Unidos, sendo impulsionada pelas linhas de produção baseadas na divisão do trabalho; já a terceira revolução industrial tem como marco, em 1969, a apresentação do primeiro controlador lógico programável, permitindo a programação digital de sistemas autônomos (DRATH; HORCH, 2014).

A Indústria 4.0 está intimamente ligada aos Sistemas Ciber-Físicos, cujo entendimento é auxiliado a partir de 3 hipóteses apresentadas por Drath e Horch (2014):

1. A infraestrutura de comunicação será cada vez mais acessível e introduzida em qualquer lugar, auxiliando a engenharia, configuração, serviços, diagnósticos, operação, etc.
2. Dispositivos de campo, máquinas, fábricas e até mesmo produtos individuais estarão conectadas a uma rede, disponíveis como objetos, e armazenarão dados em tempo real. Isso permitirá que eles sejam acessados e analisados, na rede, a qualquer momento e em qualquer lugar.
3. Dispositivos de campo, máquinas, fábricas e até mesmo produtos individuais serão capazes de armazenar informação e conhecimento sobre eles mesmos dentro da rede, tendo, dessa forma, uma representação virtual na rede. Além dos dados,

outras ações podem ser realizadas nos objetos físicos, como funções de negociação e exploração.

Os autores, ainda, resumem os Sistemas Ciber-Físicos a partir de três níveis: (i) os objetos físicos; (ii) a representação dos objetos em modelos de dados em uma infraestrutura de rede; e (iii) serviços baseados nos dados disponíveis. Dessa forma, componentes da indústria poderão negociar entre si, sistemas poderão ser virtualmente testados e otimizados, dados de fábricas estarão disponíveis para qualquer pessoa autorizada, e planos de produção poderão ser revolucionados por algoritmos (DRATH; HORCH, 2014).

A Indústria 4.0 busca criar produtos, processos e procedimentos inteligentes, e tem como ponto chave as fábricas inteligentes, isto é, plantas com capacidades de tratarem maiores complexidades, que estão menos propensas a interrupções e têm humanos e máquinas comunicando-se entre si de forma natural, como em uma rede social (KAGERMANN; WAHLSTER; HELBIG, 2013). Uma fábrica inteligente (*smart-factory*) é composta por uma rede que inclui equipamentos de produção, robôs, transporte e logística (GONG; ZOU; KAN, 2019).

A i4.0 traz novas oportunidades para as fábricas inteligentes, já que cada máquina, veículo, estação de trabalho, enfim, entidade envolvida no processo produtivo pode ser autônoma, controlando e regulando-se (XU et al., 2016). A obtenção de dados em tempo real dessas ferramentas, através de sensores, pode ser usadas para prevenir defeitos ou danos; novas interfaces entre homem e máquina, assim como técnicas de impressão 3d, podem acelerar o processo de transformação de instruções digitais para produtos físicos (XU et al., 2016).

A Quarta Revolução Industrial entregará melhor qualidade em campos como engenharia, planejamento, manufatura, operações e processos logísticos, com maior flexibilidade e robustez; ela poderá levar a cadeias de valores auto-organizáveis, otimizadas com base em uma série de critérios (KAGERMANN; WAHLSTER; HELBIG, 2013).

Hermann, Pentek e Otto (2016) apontam quatro componentes para a Indústria 4.0, a partir de uma revisão de literatura: (i) Sistemas Ciber-Físicos; (ii) Internet das Coisas; (iii) Internet de Serviços; e (iv) Fábricas Inteligentes. Os CPS tratam da integração do mundo físico ao virtual, isto é, equipamentos capazes tanto de armazenarem dados de seu estado, quanto de realizarem operações. A Internet das Coisas (*Internet of Things – IoT*) trata da conexão dos objetos de nosso cotidiano, aumentando a ubiquidade da Internet através da integração desses objetos, e leva à construção de uma grande rede de comunicação entre dispositivos e humanos (XIA et al., 2012). A IoT é amplamente aceita como uma tecnologia

que já impacta sistemas de manufatura industriais (TAN et al., 2019). A Internet dos Serviços (*Internet of Services* – IoS) possibilita a oferta e demanda de serviços através da Internet, sendo composta por participantes, infraestrutura de serviços e os serviços (BUXMANN; HESS; RUGGABER, 2009). A Internet das Coisas e a de Serviços possibilitam a construção de redes que integram todo o processo de manufatura, levando a construção das Fábricas Inteligentes (*Smart Factory*) (KAGERMANN; WAHLSTER; HELBIG, 2013). Tratam-se de fábricas em que as tarefas são realizadas a partir de CPS, que se comunicam através da IoT (HERMANN; PENTEK; OTTO, 2016).

A partir destes componentes, Hermann, Pentek e Otto (2016) trazem uma definição da Indústria 4.0:

“Indústria 4.0 é um termo coletivo para tecnologias e conceitos de organização de cadeias de valor. Com a estrutura modular de Fábricas Inteligentes da Indústria 4.0, CPS monitoram processos físicos, criam uma cópia virtual do mundo físico e realizam decisões descentralizadas. Sobre a Internet das Coisas, CPS comunicam e cooperam entre si e com humanos em tempo real. Através da Internet de Serviços, tanto serviços internos quanto entre organizações são oferecidos e utilizados por participantes da cadeia de valor” (HERMANN; PENTEK; OTTO, 2016, p. 11, traduzido pelo autor).

Além dos conceitos de Fábricas Inteligentes e Sistemas Ciber-Físicos, Lasi et al. (2014) trazem outros conceitos relacionados à Indústria 4.0: (i) auto-organização, em virtude da descentralização dos sistemas de manufatura; (ii) novos sistemas de distribuição e aquisição, nos quais os processos de distribuição e aquisição serão mais individualizados, e processos conectados serão tratados pelo uso de diferentes canais; (iii) novos sistemas no desenvolvimento de produtos e serviços, tratando estes desenvolvimentos de forma individualizada, com abordagens de inovação aberta, inteligência e memória de produto; (iv) adaptação às necessidades humanas, ao invés do contrário; (v) responsabilidade social da corporação, já que o foco em sustentabilidade e gestão eficiente de recursos vêm crescendo nos processos de manufatura.

Rüßmann et al. (2015) discutem nove pilares tecnológicos que “sustentam” a Indústria 4.0. Segundo os autores, a conexão entre sensores, ambientes de trabalho, máquinas e sistemas de TI, utilizando protocolos padrões da Internet, permitirá a predição de falhas, auto-configuração e adaptação a mudanças, possibilitando processos mais eficientes com redução de custos. Os pilares indicados por Rüßmann et al. (2015) são:

1. *Big data* e análise de dados: consiste na análise de grandes quantidades de dados de manufatura, de diversas fontes – como equipamentos de produção, sistemas de gestão de empresas e clientes – permitindo a tomada de decisão em tempo real.

2. Robôs autônomos: embora robôs já sejam utilizados em indústrias, eles tendem a ser cada vez mais autônomos, trabalhando, inclusive, ao lado de humanos e aprendendo com eles, de forma segura, além de serem capazes de se comunicarem. Espera-se que robôs custem menos e tenham maiores capacidades.
3. Simulação: indica-se que simulações passarão a ser executadas utilizando informações de tempo real, auxiliando a tomada de decisão. Dessa forma, modelos virtuais serão utilizados para prever o mundo real, possibilitando testes de otimização de parâmetros, para que sejam implantados nas linhas de produção.
4. Integração de sistemas de forma horizontal e vertical: com a Indústria 4.0, haverá maior integração entre sistemas, em redes intercompanhias, permitindo uma maior automação. Atualmente, sistemas de informação de companhias não são totalmente integrados, tanto interna quanto externamente.
5. A Internet das Coisas Industrial: a partir da Internet das Coisas Industrial, mais equipamentos terão processamento embarcado e estarão conectados, permitindo maior interação entre eles, auxiliando a obtenção de respostas em tempo real.
6. Segurança cibernética: devido à maior conectividade dos sistemas na Indústria 4.0, maiores proteções contra ataques cibernéticos deverão ser tomadas, impulsionando a construção de ferramentas com este propósito.
7. A nuvem: companhias já vem utilizando tecnologias de computação em nuvem para aplicações empresariais e de análise de dados, o que será impulsionado pela Indústria 4.0, aumentando a performance das tecnologias em nuvem, podendo levar, inclusive, a sistemas de monitoramento e controle de processos para a ambientes baseados em nuvem. A computação em nuvem auxilia, também, questões entre companhias.
8. Fabricação de aditivos: tecnologias como impressão 3D vêm sendo adotadas principalmente para produção de componentes individuais e prototipação; com a Indústria 4.0, tecnologias de fabricação de aditivos permitirão a construção de produtos customizados, de forma descentralizada, reduzindo despesas com estoque, permitindo a criação de projetos complexos e leves.
9. Realidade aumentada: a realidade aumentada suporta uma grande variedade de sistemas e possibilita o auxílio na tomada de decisão e no desenvolvimento de procedimentos, provendo informações em tempo real.

Schlechtendahl et al. (2015) propõem uma forma de integração de sistemas de produção já existentes, para que possam atuar na Indústria 4.0. Os autores indicam que as fábricas na Indústria 4.0 têm todos os seus participantes interconectados, compartilhando informações entre si. Para atingir essa integração, o trabalho propõem um *CPPS (Cyber Physical Production System) Enabler*, que permite a descoberta e conexão desses sistemas, possibilitando o acesso e a distribuição de dados (SCHLECHTENDAHL et al., 2015).

Lee, Bagheri e Kao (2015) indicam uma arquitetura para CPS em sistemas de manufatura da Indústria 4.0. Os autores indicam que os CPS estão em estágio inicial de desenvolvimento e, por isso, é necessário definir metodologias e guias para sua implementação na indústria. A arquitetura proposta no trabalho de Lee, Bagheri e Kao (2015) é composta por uma estrutura com 5 níveis, e provê um guia passo-a-passo para o desenvolvimento e implantação de CPS para manufatura. Os 5 níveis são: (i) conexão inteligente; (ii) conversão de dados para informações; (iii) ciber; (iv) cognição; e (v) configuração.

Em Brettel et al. (2014), os autores indicam que a indústria está sofrendo modificações, que decorrem, entre outras coisas, da diminuição do ciclo de vida dos produtos e da maior customização exigida pelos consumidores. A Internet, neste contexto, pode ser utilizada para gerenciar sistemas distribuídos e tecnologias como rádio frequência, que auxiliam o processo de produção. No trabalho, os autores analisam temas que consideram promissores a trazerem grandes impactos à Indústria 4.0.

A estrutura da Indústria 4.0 é a de uma grande rede, formada pelos diversos elementos que a compõem, como as fábricas inteligentes e seus CPS. Dessa forma, uma maneira de estudar fenômenos que podem ocorrer na quarta revolução industrial, é através do estudo das redes, em especial, das redes complexas.

2.2 REDES COMPLEXAS

Redes representam várias estruturas na sociedade; são formadas por elementos “discretos”, separados, que possuem conexões ou interações com outros elementos. Para entender uma rede, é preciso entender tanto os elementos que a formam, quanto a sua estrutura. Exemplos de redes são a Internet, as redes sociais (na Internet ou não), a interação de proteínas e o cérebro humano, entre outros (LOVÁSZ, 2012).

Vivemos cercados por redes, o que justifica tanto o estudo das partes que formam uma rede, quanto o relacionamento entre essas partes; o comportamento individual pode

influenciar as demais partes e o comportamento da rede como um todo (FIGUEIREDO, 2011). Uma rede é “uma abstração que permite codificar algum tipo de relacionamento entre pares de objetos” (FIGUEIREDO, 2011, p. 304). Redes podem ser de indivíduos, computadores, objetivos. A relação entre os pares da rede tem algum significado. Diferentes tipos de relacionamentos podem gerar diferentes redes, em um mesmo grupo de indivíduos (ex: amizade e contato sexual) (FIGUEIREDO, 2011).

A estrutura de uma rede é dada pelas conexões que existem entre seus objetos. É importante estudar essa estrutura, pois há uma série de fenômenos que operam sobre ela. Também é importante caracterizar a estrutura da rede, isto é, “enumerar diversos aspectos que capturem e resumem a estrutura da rede” (p. 306), bem como entender as propriedades de uma rede, pois pode-se generalizar comportamentos com base nessas informações (FIGUEIREDO, 2011).

Cientistas vêm utilizando modelos matemáticos e computacionais para análise de sistemas físicos. Porém, sistemas complexos que envolvem redes, em geral, requerem uma análise diferenciada (BRASE; BROWN, 2009). Segundo os autores, para conseguir modelar uma rede complexa, é necessário:

- Entender a estrutura da rede: entender a topologia da rede. Pode ser necessário utilizar métodos de aprendizagem de máquina para inferir a estrutura indiretamente, quando a rede é muito complexa e desconhecida.
- Entender a dinâmica da rede: entender o comportamento da rede sob várias condições.
- Modelagem matemática e simulação: diferentes níveis de resolução podem ser utilizados – do mais alto grau de abstração ao mais baixo.
- Sensibilização à situação, projeto e controle de redes complexas: desenvolver métodos para entender e controlar eventos e anomalias em sistemas de redes complexas.

Sistemas simples são compostos por poucos componentes, e funcionam de acordo com leis bem conhecidas (ex: um pêndulo, que possui todas as partes enumeradas, e seu movimento é definido por equações conhecidas). Sistemas complicados possuem várias partes, mas todas são conhecidas, bem como seus papéis e relações (como um avião, que possui uma gigantesca quantidade de peças, mas todas elas são conhecidas e tem seu papel bem definido); nestes casos, a robustez (correto funcionamento do sistema na presença de falhas) é obtida através de redundância de partes. Os sistemas complexos, por sua vez,

possuem grande número de componentes, que agem de acordo com regras que podem mudar ao longo do tempo, e não são bem entendidas. A robustez, nestes casos, é obtida quando partes se adaptam, e cumprem outros papéis para manterem o correto funcionamento do sistema. Um exemplo de sistema complexo é a migração de gansos, cuja formação em “V” não é estática, e tem diferentes pássaros ocupando a ponta, como líder, em momentos diferentes (AMARAL; OTTINO, 2004). Ainda segundo os autores, “a característica mais comum de todos os sistemas complexos é que eles mostram organização sem qualquer princípio organizacional externo ser aplicado” (AMARAL; OTTINO, 2004, p. 2, traduzido pelo autor); “muitos detalhes de um sistema complexo podem ser irrelevantes, e seu comportamento é determinado por um pequeno número de parâmetros e mecanismos que, sim, são relevantes” (AMARAL; OTTINO, 2004, p. 13, traduzido pelo autor).

As redes complexas são redes com estrutura irregular, complexas e dinamicamente evoluídas ao longo do tempo, formadas por milhares ou milhões de nós; o estudo de redes complexas busca entender as propriedades das redes de unidades dinâmicas.

Amaral e Ottino (2004) elencam três desafios do estudo de sistemas complexos:

1. A natureza das unidades: os elementos que formam o sistema nem sempre são iguais ou vizinhos, podem ter estruturas internas complexas, e não possuem papéis estritamente definidos.
2. A natureza das interações: as interações podem ser complexas, e não conhecidas.
3. A natureza da força ou entrada de energia: sistemas complexos tendem a estar em um estado de desequilíbrio. Além disso, as perturbações externas são mal caracterizadas.

Em geral, duas topologias de rede são tipicamente consideradas: grafos *d*-dimensionais, em que cada nó está conectado a um número fixo de vizinhos (como uma grade); e grafos aleatórios, onde cada nó tem a mesma probabilidade de estar conectado com outro nó (AMARAL; OTTINO, 2004). Redes reais são agrupadas, com grande nível de conexões locais, e também são pequenos mundos, ou seja, são necessários pequenos números de passos para conectar quaisquer dois nós da rede (AMARAL; OTTINO, 2004).

A primeira prova da teoria das redes pode ser considerada a solução de Euler para o problema das pontes de Königsberg, em 1735. As redes também são estudadas extensivamente nas ciências sociais – na década de 1930, sociólogos estudavam as conexões entre pessoas para entender o funcionamento da nossa sociedade. Trabalhos típicos de estudo de redes em sociologia envolve a aplicação de questionários para que os respondentes detalhem sua interação com os outros. Nestes casos, os indivíduos são os vértices, e suas

interações são as arestas. Estudos típicos de redes sociais envolvem centralidade (quais os indivíduos que são mais conectados, ou têm mais influência) e conectividade (verificar como um indivíduo está conectado com outro através da rede) (NEWMAN, 2003).

Atualmente, cada vez mais se tem estudado redes maiores, devido às melhores capacidades computacionais disponíveis. De redes com dezenas ou centenas de vértices, passou-se a estudar redes com milhares ou milhões. Em redes pequenas, pode-se estudar, por exemplo, qual o nó que, quando removido, terá maior impacto na conectividade, porém em grandes redes, isso já não faz sentido. Nestes casos, uma questão interessante é: qual porcentagem de vértices precisa ser removida para afetar, significativamente, a conectividade da rede? (NEWMAN, 2003). As redes passaram a ser chamadas de “redes complexas”, pois a “topologia e a evolução das redes do mundo real apresentam propriedades organizacionais bastante robustas e distintas das redes aleatórias” (METZ et al., 2007, p3).

2.2.1 Definição, Representação e Características Estruturais

A Teoria dos grafos é a teoria matemática utilizada para o estudo de redes (LOVÁSZ, 2012). Assim, a definição formal de uma rede – ou um grafo – pode ser feita como definido por Figueiredo (2011). Uma rede é formada por um conjunto V de objetos (chamados de nós ou vértices); o número de objetos é dado pelo valor $n = |V|$. O relacionamento entre dois vértices da rede é chamado de aresta, e pode ser representado pelo par (i, j) . Nos relacionamentos simétricos, trata-se de um par não ordenado, isto é, (i, j) é igual a (j, i) . Já nos relacionamentos assimétricos, tem-se um par ordenado, em que o par (i, j) é diferente do par (j, i) . O conjunto de arestas (relacionamentos entre vértices de V) é definido pela equação (1):

$$E = \{(i,j) \mid i, j \in V, i \text{ está relacionado com } j\} \quad (1)$$

A representação de uma rede pode ser feita através de uma matriz A , quadrada, de n elementos. Cada elemento $A(i, j)$ representa o par de vértices (i, j) . Se o par possui relacionamento, seu valor será 1. Caso contrário, será 0 (FIGUEIREDO, 2011). A equação (2) traz a definição de $A(i, j)$.

$$A(i,j) = \{1, \text{ se } (i, j) \in E; 0 \text{ caso contrário}\} \quad (2)$$

Uma série de características podem ser utilizadas para descrever a topologia de uma rede complexa. A seguir, essas características são expostas:

2.2.1.1 Grau (degree) e distribuição de grau

O grau (ou conectividade) de um vértice é o número de arestas que estão relacionadas com ele. Define-se por d_i o grau do vértice $i \in V$. A soma do grau de todos os vértices é igual ao dobro do número de arestas ($2m$), pois cada aresta é contada duas vezes. O grau médio é dado pela média aritmética do grau de todos os vértices. Como o somatório do grau é dado por $2m$, tem-se que o grau médio é dado por $2m/n$ (FIGUEIREDO, 2011).

Em uma rede assimétrica (direcionada), o grau de um nó possui dois componentes: o número de arestas que saem daquele vértice, e o número de arestas que chegam no vértice. Assim, o grau total é dado pela soma desses dois componentes (BOCCALETTI et al., 2006).

A distribuição de grau de uma rede é a fração de vértices com determinado grau, e a distribuição complementar cumulativa do grau é a fração de vértices que têm grau maior ou igual a determinado valor k (FIGUEIREDO, 2011).

2.2.1.2 Caminhos e distâncias

Um caminho entre dois vértices em um grafo é uma sequência de vértices, sem repetição, onde existe uma aresta adjacente na sequência (FIGUEIREDO, 2011). Representar o tamanho dos caminhos mais curtos entre vértices de um grafo é interessante, e pode ser feito em uma matriz D , na qual um elemento d_{ij} representa o comprimento do caminho entre os vértices i e j (BOCCALETTI et al., 2006). O diâmetro de uma rede, por sua vez, é o maior valor de d_{ij} , isto é, o maior caminho entre dois vértices de uma rede (BOCCALETTI et al., 2006). Já a distância média é calculada pela média aritmética das distâncias entre todos os pares de vértices da rede (FIGUEIREDO, 2011).

Uma medida utilizada para calcular a centralidade de um nó (ou aresta), é o *betweenness* do nó (ou aresta), que se refere à proporção de caminhos mais curtos entre os vértices da rede que passam pelo nó (ou aresta) em questão (BOCCALETTI et al., 2006). Já a medida do *closeness* é dada pelo inverso da distância média a partir de todos os outros nós. Ou seja, quanto maior a centralidade (*betweenness*) de um nó, mais vezes ele aparecerá na lista de caminhos mais curtos entre os nós da rede; e quanto maior seu *closeness*, mais perto dos demais nós da rede ele está (BOCCALETTI et al., 2006).

2.2.1.3 Coeficiente de aglomeração (clusterização)

O coeficiente de aglomeração ou clusterização (*clustering*) de um vértice i é a fração de arestas que os vizinhos de i possuem entre si e o máximo de arestas que eles poderiam possuir entre si. O coeficiente de clusterização da rede é a média aritmética dos coeficientes de cada vértice (FIGUEIREDO, 2011). Também conhecido como transitividade, serve para medir a presença de um grande número de triângulos em um grafo (BOCCALETTI et al., 2006), isto é, quanto maior o coeficiente de aglomeração de uma rede, maior a chance de dois nós conectados a um determinado nó também estarem conectados entre si.

2.2.1.4 Resistência e robustez

A resistência de uma rede consiste na capacidade desta rede de resistir à perda de nós, sem que haja perda da sua funcionalidade; ela está ligada com a distribuição de grau dos nós, pois a remoção de vértices pode aumentar significativamente as distâncias dos caminhos mais curto, bem como remover conexões da rede (METZ et al., 2007).

Já a robustez está relacionada com a capacidade de a rede continuar funcionando na presença de falhas. Trata-se de um conceito amplo, que depende do contexto da rede, isto é, de suas funcionalidades e, em geral, a capacidade de operação da rede está relacionada com suas propriedades estruturais (FIGUEIREDO, 2011).

2.2.2 Topologia de redes complexas

Para generalizar resultados, é preciso trabalhar com os processos que deram origem às redes, que são determinados através de modelos matemáticos. Pode-se utilizar modelos aleatórios para a construção de redes com determinada propriedade estrutural de interesse. “Redes reais bem representadas por um modelo aleatório que sempre induz uma determinada propriedade estrutural vão sempre ter um determinado comportamento funcional que depende desta propriedade” (FIGUEIREDO, 2011, p.309).

Embora a representação de redes do mundo real como grafos compostos por vértices e arestas seja uma simplificação, tendo em vista que as propriedades da rede podem mudar ao longo do tempo e espaço, essa forma tem se mostrado uma representação informativa do sistema como um todo (BOCCALETTI et al., 2006). Ainda que representem diferentes estruturas do mundo real, desde sistemas de comunicação até sistemas biológicos e sociais, a maior parte das redes reais são caracterizadas por topologias semelhantes, como pequenos

caminhos e grandes coeficientes de aglomeração (*clustering*), que as tornam diferentes de estruturas regulares e grafos aleatórios – padrões de modelo utilizados em teorias de grafo matemáticas (BOCCALETTI et al., 2006). A seguir são discutidas algumas propriedades topológicas das redes (grafos).

- Pequenos mundos (*small-world*): A propriedade de pequenos mundos (*small-world*) indica que as redes têm caminhos (quantidade de arestas) relativamente pequenos entre dois nós, mesmo que elas tenham grande tamanho (número de nós) (ALBERT; BARABÁSI, 2002). Estudos de vários processos dinâmicos mostram a existência de caminhos curtos, “atalhos”, que ligam diferentes áreas de uma rede, podendo acelerar a comunicação entre nós distantes (BOCCALETTI et al., 2006).
- Distribuição de graus livre de escala: No passado, usualmente as redes eram representadas de forma regular, homogênea, de forma que quase todos os nós tinham topologia equivalentes, como em estruturas regulares ou grafos aleatórios, em que cada possível conexão entre dois nós da rede (aresta) tem a mesma possibilidade de existir ou não (BOCCALETTI et al., 2006). Verifica-se, porém, que a probabilidade de uma conexão entre dois nós existir depende do grau do nó, isto é, há uma “preferência no anexo”: nós com maior grau terão maior chance de estarem conectados a novos nós adicionados na rede (ALBERT; BARABÁSI, 2002)
- Alto coeficiente de aglomeração (transitividade): Em muitas redes, tem-se que, se um vértice A está conectado a um vértice B , e o vértice B está conectado ao um vértice C , há grande possibilidade de A estar conectado com C , ou seja, uma grande quantidade de triângulos de conexões (NEWMAN, 2003).

2.2.3 Modelos de Redes

Modelos podem ser utilizados para a construção de redes (grafos), com base em suas características – como as topológicas. A seguir, três modelos de geração de grafos são discutidos: grafos aleatórios, grafos de pequeno mundo e grafos livres de escala.

2.2.3.1 Grafos aleatórios

O estudo de grafos aleatórios teve seu início com Erdős e Rényi em 1959 (ERDÖS; RÉNYI, 1959), buscando estudar propriedades de grafos como uma função de acréscimo de

conexões aleatórias (BOCCALETTI et al., 2006). Os autores definiram um grafo como N nós conectados por n arestas escolhidas aleatoriamente, dentro de todas as possibilidades de conexões possíveis (ALBERT; BARABÁSI, 2002). O termo “grafo aleatório” faz referência à falta de ordem para o arranjo dos vértices; para a construção do grafo, parte-se de um conjunto de nós desconectados e conectam-se pares destes nós de forma aleatória, sendo que todas as conexões têm a mesma chance (probabilidade) de existir ou não (BOCCALETTI et al., 2006).

2.2.3.2 Grafos de pequenos mundos (*small-worlds*)

Watts e Strogatz (1998) construíram um modelo para a representação de redes com características de pequeno mundo e com alta aglomeração, através de um processo de reconstrução do grafo. O modelo parte de uma estrutura regular (anel) de n vértices, com k arestas por vértice e reescreve cada aresta com uma probabilidade p , permitindo variar o grafo entre totalmente regular, com $p = 0$, e totalmente aleatório, utilizando $p = 1$ (WATTS; STROGATZ, 1998).

Boccaletti et al. (2006) define o modelo de Watts e Strogatz partindo de um anel com N nós, cada um deles conectado simetricamente com seus $2m$ vizinhos próximos para um total de $K = mN$ arestas; para cada nó, cada uma de suas arestas é realocada para outro nó, escolhido aleatoriamente, com uma probabilidade p , ou preservada com uma probabilidade $1 - p$. A raiz do método está em sistemas sociais, nos quais muitas pessoas estão conectadas com seus vizinhos (grafo inicial) e possuem algumas conexões com pessoas de outros lugares (arestas reconstruídas) (ALBERT; BARABÁSI, 2002).

2.2.3.3 Grafos livres de escala

Barabási e Albert (1999) reportam a existência de grandes graus de auto-organização em redes complexas, com a característica de “livre de escala”, o que, segundo os autores, não é contemplado pelo modelo aleatório de Erdős e Rényi (1959), tampouco pelo modelo de pequeno mundo de Watts e Strogatz (1998). A origem da variação na escala (grau dos nós) é baseada em dois pontos negligenciados pelos modelos até então desenvolvidos: crescimento e anexo preferencial (BARABÁSI; ALBERT, 1999).

O algoritmo para a criação de uma rede livre de escala segue os seguintes passos: inicia com um pequeno número de nós e, a cada iteração, um novo nó é adicionado com

arestas conectando o nó adicionado a alguns nós já presentes na rede (crescimento); ao escolher os nós para conectar o novo nó, assume-se uma probabilidade de conexão (criação de uma aresta) que leva em conta o grau do nó já presente no grafo. Dessa forma, tem-se que nós com maior grau terão maior probabilidade de receber uma conexão com o novo nó adicionado à rede. Isso gera uma rede com alguns nós com maior número de conexões do que os demais (ALBERT; BARABÁSI, 2002).

2.3 MODELAGEM E SIMULAÇÃO COMPUTACIONAL

Modelagem é o processo de construção de modelos, que, por sua vez, são representações de algum sistema de interesse (MARIA, 1997). Segundo a autora, um modelo deve ser o mais próximo possível do sistema real que ele representa, embora seja uma simplificação, e não pode ser complexo a ponto de impossibilitar o seu entendimento e a sua experimentação.

A complexidade do modelo pode ser aumentada progressivamente, em processos iterativos, e um importante passo em sua construção é a validação, que pode ser feita através da comparação dos resultados obtidos pelo modelo com a saída do sistema real, utilizando, para isso, valores conhecidos (MARIA, 1997). Um modelo não pode ser mais complexo do que o sistema real que ele representa, pois, neste caso, será um problema novo, e não um modelo; a modelagem busca representar o que é mais importante no sistema real (CHWIF; MEDINA, 2006).

Modelos podem ser representações realistas do sistema original em forma de miniaturas, desenhos técnicos, diagramas funcionais, e podem ser expressados através de fórmulas matemáticas ou como um programa de computador, possibilitando, neste caso, simular o comportamento dinâmico do sistema (BOSSSEL, 2013). O avanço da tecnologia eletrônica possibilita a simulação em *software*, sem a necessidade da construção de parte do sistema real, ou de uma miniatura dele (BOSSSEL, 2013).

A simulação busca prever o comportamento de um sistema perante valores de entradas, tendo por base um conjunto de regras de funcionamento; ela pode ser composta por uma série de fórmulas matemáticas, mas não pode ser reduzida a apenas uma equação (CHWIF; MEDINA, 2006). Uma simulação é a imitação de um processo de um sistema real, ao longo do tempo; ela pode ser feita em um computador, ou manualmente, e trata da construção de uma história artificial, que permite a inserção de inferências nas características de operação do sistema real (BANKS et al., 2010).

A simulação computacional, em especial, é aquela que depende de um computador para ser realizada e é a simulação abordada nesta tese (CHWIF; MEDINA, 2006). A simulação computacional é um método para replicar um sistema real em um computador, conduzindo experimentos para entender o comportamento do sistema e/ou avaliando estratégias sobre o seu funcionamento (TAN et al., 2019).

A simulação pode ser vista como a operação de um modelo de sistema, e permite a reconfiguração e a experimentação dele, o que geralmente é impossível, impraticável ou demasiadamente custoso no ambiente real (MARIA, 1997). A partir da simulação, pode-se inferir comportamentos do sistema, em diferentes cenários analisados, nos quais parâmetros são alterados; a simulação também possibilita a análise do sistema ao longo do tempo (MARIA, 1997). Ainda segundo a autora, simulação pode ser utilizada antes da realização de alterações em um sistema, reduzindo prováveis problemas.

Simulação é uma ferramenta já estabilizada para avaliar e prever o desempenho de sistemas complexos, que são analiticamente intratáveis; desde a sua concepção, na década de 1950, a simulação vem sendo empregada com sucesso para melhorar sistemas complexos (XU et al., 2016). Segundo os autores, simulações poupam tempo e recursos, e podem criar provas de conceitos anteriores à construção de um protótipo físico; é possível utilizar o modelo de simulação para testar diversas alternativas do projeto de um sistema e, assim, escolher aquele com melhor performance.

Modelagem e simulação pode ser visto como um método para o uso de modelos de sistemas reais ou imaginários, ou ainda como um processo que possibilita o melhor entendimento (ou ainda a predição) do comportamento de um sistema ou processo (RODIČ, 2017). Segundo o autor, o uso de modelagem e simulação está bem estabilizado tanto nos campos da ciência quanto de engenharia; auxiliando a redução de custos, reduzindo o tempo de ciclos de desenvolvimento, aumento a qualidade de produtos e facilitando a gestão do conhecimento.

Modelagem e simulação computacional é o processo de criação e experimentação de um sistema físico, utilizando um modelo matemático computadorizado (CHUNG, 2004). Um modelo é uma representação simplificada das interações entre as partes de um sistema; é uma abstração da realidade que busca se aproximar do comportamento do sistema real, sendo mais simples do que ele (CHWIF; MEDINA, 2006).

Embora seja uma simplificação, resultados de simulações podem trazer estimativas confiáveis para a tomada de decisão; a simulação precisa capturar uma grande quantidade de variáveis – detalhes da operação, que precisam estar disponíveis – podendo torná-la complexa

e difícil de ser executada computacionalmente (XU et al., 2016). Uma vez que a Indústria 4.0 faz uso de sensores e outros dispositivos altamente conectados, espera-se que dados estejam amplamente disponíveis.

Ainda que a simulação tradicionalmente tenha sido empregada na fase de projeto, atualmente vem sendo utilizada em processo de otimização de um sistema, envolvendo tomada de decisão em menor espaço de tempo; o grande grau de interconectividade dos sistemas na i4.0 resultou num aumento drástico de processos em tempo real (XU et al., 2016).

Prever o comportamento de um sistema sobre certas condições é uma tarefa difícil: conhecer o que vai (ou pode) acontecer pode ser um caso de vida ou morte, e não apenas de curiosidade (BOSSEL, 2013). O autor traz cinco razões para o uso de simulação computacional, em detrimento de outros tipos de simulação:

- A simulação computacional possibilita o uso de metodologias comuns e *softwares* aplicados, independentemente do sistema abordado.
- O custo da simulação computacional geralmente é apenas uma fração do custo para a construção de um simulador real, análogo ao sistema.
- O curso de tempo pode ser alterado na simulação, ou seja, é possível tanto acelerar quanto desacelerar os processos do sistema simulado computacionalmente.
- É possível realizar simulações que levem à destruição do sistema, sem consequências, já que a simulação pode ser repetida sistematicamente.
- Não há risco para o sistema real, pois o ambiente de simulação é isolado do real.

Modelos de simulação podem ser classificados com base em três dimensões (MOURTZIS; DOUKAS; BERNIDAKI, 2014):

1. Tempo de mudança: divide as simulações em estáticas (independentes do tempo), e dinâmicas (que evoluem ao longo do tempo). Simulações dinâmicas ainda podem ser categorizadas em discretas, onde as mudanças ocorrem em pontos discretos do tempo; ou contínuas. Simulações discretas são divididas entre as escalonadas por tempo, em que as alterações são realizadas após determinado intervalo de tempo definido; e dirigidas a eventos, nas quais as mudanças estão vinculadas a eventos escalonados, com intervalos de tempo irregulares.
2. Aleatoriedade: define se a simulação é determinística ou estocástica. Uma simulação determinística sempre gerará um mesmo resultado, dado um mesmo

conjunto de dados de entrada. Por outro lado, uma simulação estocástica não garante a repetição do resultado para um mesmo conjunto de entradas.

3. Organização dos dados: divide as simulações entre aquelas baseadas em grade, nas quais os dados estão associadas a células em locais específicos de uma tabela, e mudanças afetam cada célula de acordo com seu estado anterior e o estado de seus vizinhos; e simulações livres de malha (*mesh-free*), que estão vinculadas a dados de participantes individuais, cujas mudanças são direcionadas para cada par de participantes.

Maria (1997) elenca uma série de possibilidades do uso da modelagem e simulação: (i) melhor entendimento do sistema, ao desenvolver um modelo matemático a partir da observação do sistema real em detalhe; (ii) teste de hipóteses de viabilidade sobre o sistema; (iii) comprimir ou expandir o tempo de um sistema para análise; (iv) estudar os impactos da alteração do modelo, sem afetar o sistema real; (v) experimentar situações novas ou desconhecidas, sobre as quais apenas informações fracas estão disponíveis; (vi) identificar as variáveis com maior impacto no sistema, e a relação entre elas; (vii) identificar gargalos no fluxo de entidades do sistema; (viii) utilizar múltiplas métricas de análise de performance; (ix) empregar uma abordagem de sistema para a solução de problemas; e (x) desenvolvimento de sistemas mais robustos e bem projetados, reduzindo o tempo de desenvolvimento.

2.4 TRABALHOS RELACIONADOS

Para verificar se outros estudos abordaram uma temática semelhante à desta pesquisa, foram buscados trabalhos semelhantes em bases de dados científicas. Muhuri et al. (2019) elaboraram um estudo bibliométrico sobre a Indústria 4.0, utilizando como bases de dados os repositórios Web of Science¹ e Scopus². Entre os trabalhos analisados, alguns envolviam *softwares* e simuladores sobre a Indústria 4.0, elencados a seguir:

- Wang et al. (2016) propuseram um novo algoritmo para operações em uma fábrica inteligente (*smart-factory*), simulando produtos capazes de se comunicarem e cooperarem entre si, sem intervenção humana.
- Lee e Yang (2017) propuseram uma arquitetura de *software* para câmeras inteligentes, com um ambiente de simulação.

1 <https://www.webofknowledge.com/>

2 <https://www.scopus.com/>

- Hsieh et al. (2017) abordaram a máxima utilização de equipamentos em fotolitografia, a partir de uma abordagem de simulações ligadas a conhecimentos de *big data*.
- Tao et al. (2017) abordaram áreas menos exploradas de serviços de manufatura em correspondência de oferta e demanda em computação em nuvem, a partir da construção de um simulador que utiliza técnicas baseadas em rede para análise estatística e previsões.
- Li et al. (2017) propuseram um sistema de embalagem automático para *e-commerce*, baseado na Internet das Coisas, e um simulador robótico foi utilizado para demonstrar a eficácia do sistema.
- Turner et al. (2016) realizaram uma revisão sobre a fusão de pesquisas em áreas industriais de simulação de eventos discretos e realidade virtual.
- Bohács e Rinkács (2017) trataram de avanços da modelagem e simulação com a introdução de Sistemas Ciber-Físicos e Indústria 4.0; os autores propuseram um novo componente baseado em ontologia e explicaram sua funcionalidade.
- Wanyama (2017) trouxe uma aplicação prática da Indústria 4.0 em um sistema de educação. Em seu trabalho, um laboratório virtual *online* foi desenvolvido para possibilitar que estudantes realizassem múltiplos experimentos para um melhor entendimento da teoria.

Para verificar outros trabalhos que envolvam simulação computacional na Indústria 4.0, realizou-se uma pesquisa na base de dados do Web of Science, utilizando as seguintes palavras-chave: (i) *Industry 4.0*, (ii) *simulation*, e (iii) *software*. Optou-se pela busca na Web of Science por ser um indexador de diferentes periódicos acadêmicos relevantes. A pesquisa foi realizada em 25 de julho de 2019, e foram selecionados trabalhos posteriores a 2011, quando o termo “Indústria 4.0” foi lançado. Setenta artigos foram encontrados, sendo dois deles repetidos – totalizaram-se, portanto, sessenta e nove artigos para análise. Os resumos (*abstracts*) desses artigos foram lidos e, quando necessário, aprofundou-se em seu texto.

Dos sessenta e nove artigos analisados, um deles era puramente teórico (CARUSO, 2018), um artigo não abordava nenhum caso de simulação (HOFER, 2018), um trabalho encontrado tratava-se de uma apresentação (KRANZ; ZUPICK, 2017), nove não possuíam relação com Indústria 4.0 – (DE MARCO; RIEMMA; IANNONE, 2018), (YOUJUN; VAFAI, 2017), (GRASSO et al., 2016), (HASSAN et al., 2015), (MARZANO et al., 2015), (NIE et al., 2015), (DENKENA; WINTER, 2015), (FONTANA et al., 2015) e (GRASSO et

al., 2014). Outros cinco estavam relacionados a questões de energia ou robótica – (OUN; BENABDALLAH; CHERIF, 2019), (BUKATA et al., 2016), (LINS; OLIVEIRA, 2017), (ČEJKA; ČERNOHORSKÝ, 2016) e (LIHONG, 2015). Cinco artigos tinham relação com educação ou ensino – (MOURTZIS; TSAKALOS; ZOGOPOULOS, 2019), (MOURTZIS et al., 2019), (LECROQ; GRIEU; BOUKACHOUR, 2018), (KAZIMIROV, 2018) e (ANAND et al., 2018). Nove dos artigos abordavam assuntos de comunicação entre dispositivos, como redes de computadores e/ou sensores – (SADIKI et al., 2019), (AHMED et al., 2018), (RUPPERT; ABONYI, 2018), (LI et al., 2018), (ESPOSITO et al., 2017), (RAIZER et al., 2018), (ALI et al., 2018), (KADERA; NOVÁK, 2016) e (SEBASTIAN et al., 2017). Cinco artigos tratavam de simulações com realidade virtual, renderização 3D e/ou CAD (*Computer-Aided Design*, ou Desenho Assistido por Computador) – (ROMÁN-IBÁÑEZ et al., 2018), (LACOMBLEZ et al., 2018), (DE GIORGIO et al., 2017), (KOCSI; OLÁH, 2017) e (VELÍŠEK et al., 2017). Três artigos abordavam o *software* WorkDesigner – (GUST et al., 2017), (FELLER et al., 2017), (MÜLLER et al., 2015) – para projeto de ambientes de trabalho. Dentre os demais artigos, dezenove abordavam produção e/ou desenvolvimento de *software*, e outros onze – (YAZDI; AZIZI; HASHEMIPOUR, 2019), (RAUCH; SPENA; MATT, 2019), (CAPUTO et al., 2018), (SHINDE et al., 2018), (POZDNEEV et al., 2018), (CORALLO et al., 2018), (PENAS et al., 2017), (FRICKE; SCHÖNEBERGER, 2017), (TAO et al., 2016), (MOSTERMAN; ZANDER, 2016), (PERNICA; TOMÁŠKOVÁ, 2016) – tratavam de assuntos diversos, que não puderam ser agrupados, mas que não estavam relacionados com desenvolvimento de *software* ou produção. A seguir, são elencados e descritos, de forma breve, cada um dos dezenove artigos que possuem relação com produção e/ou desenvolvimento de *software*.

- *An online machine learning framework for early detection of product failures in an Industry 4.0 context* (CARVAJAL SOTO; TAVAKOLIZADEH; GYULAI, 2019): o trabalho apresenta um *framework* para detecção de falhas de montagem em superfícies de dispositivos em uma produção. Simula uma linha de produção utilizando o *software* comercial Siemens Plant Simulation.
- *Application of Digitization Procedures of Production in Practice* (SUJOVÁ; ČIERNA, 2019): os autores trazem a aplicação de digitalização em processos de produção, a partir de modelagem realizada em *software* de simulação. No artigo, foram analisadas 8 linhas de produção, com uma média de 15 estações de trabalho. No trabalho, foi abordado o conceito de *Digital Twin*, isto é, buscaram

representar o sistema real em um ambiente virtual na forma mais parecida possível.

- *Digital Value Stream Mapping Using the Tecnomatix Plant Simulation Software* (TREBUNA, P.; PEKARCIKOVA, M.; EDL, M., 2019): apresenta um método para otimizar fluxo material e de informação em uma companhia de produção, tendo como objetivo mostrar o potencial do uso de ferramentas de *software*. Utiliza caso de estudo no *software* de simulação Plant Simulation.
- *Modeling and Optimization for Automobile Mixed Assembly Line in Industry 4.0* (GONG; ZOU; KAN, 2019): realiza uma simulação de linha de montagem automotiva, no contexto da Indústria 4.0, com o objetivo de otimizá-la. Utiliza, para isso, o *software* FlexSIM.
- *Communication mechanism of the discrete event simulation and the mechanical project softwares for manufacturing systems* (LAURINDO; PEIXOTO; RANGEL, 2019): Apresenta um mecanismo de integração para comunicação *online* entre um simulador de eventos discretos e um *software* de dinâmica de sistemas.
- *Realistic modelling of flexibility and dependence in production systems in Industry 4.0 for analysing their productivity and availability* (LONG; ZEILER; BERTSCHE, 2018): no artigo é apresentado um conceito de sistema de produção flexível, e dois modelos estocásticos de coloração de redes de Petri são construídos e simulados usando o *software* REALIST, para investigar as características de flexibilidade e dependência em sistemas de produção na i4.0.
- *Knowledge Acquisition Using Computer Simulation of a Manufacturing System for Preventive Maintenance* (KŁOS, 2018): propõe uma metodologia para adquirir e manter conhecimento relacionado a falhas de sistemas de manufatura, usando métodos de simulação computacional, tendo em vista que a manutenção preventiva é importante para a i4.0, e que um sistema deve conseguir se reestruturar em caso de falhas.
- *Design flow for the development of optimized LED luminaires using multi-domain compact model simulations* (MARTY et al., 2018): neste trabalho, a i4.0 é vista como um processo de digitalização; no artigo, é proposto uma abordagem para simulação de produção de tecnologias de LED (iluminação).

- *Development of Medical Props Production Towards Industry 4.0* (PURNAMA; TONTOWI; SOPHA, 2018): realiza a simulação da produção de *medical props* (ambientes controlados de estudo para estudantes de medicina), através da comparação de cenários, buscando encontrar gargalos. Para tanto, faz uso do *software* Arena.
- *Integration of manufacturing operations management tools and discrete event simulation* (KRENCZYK et al., 2018): apresenta um método/ferramenta para a geração semiautomática de modelos de simulação, integrados com sistemas de Simulação de Eventos Discretos (SED), orientados a dados. A abordagem apresentada permite o uso de simulação e visualização para rápida verificação de plantas de fluxo de produção, e foi ilustrada através de uma implementação prática utilizando o *software* simulador FlexSim.
- *A heuristic and simulation hybrid approach for mixed and multi model assembly line balancing* (KRENCZYK; SKOLUD; HEROK, 2017): no artigo, é apresentada uma abordagem baseada em simulação e heurística para balanceamento de linha de montagem multi-modelo. Uma implementação da solução proposta é realizada utilizando o *software* FlexSim.
- *A Reliable Turning Process by the Early Use of a Deep Simulation Model at Several Manufacturing Stages* (URBIKAIN et al., 2017): no trabalho, é proposta uma metodologia para reduzir problemas em usinagem através da criação de uma ferramenta de simulação, que utiliza as principais variáveis do sistema e processa dados de entrada, gerando resultados que auxiliam o processo de planejamento e tomada de decisão. Um *software* é construído para isso, chamado Dynpro.
- *Integration and Deployment of a Distributed and Pluggable Industrial Architecture for the PERFoRM Project* (ANGIONE et al., 2017): o trabalho descreve a implementação de uma solução de arquitetura distribuída, adaptável e flexível, focada em vários blocos de construções, que podem ser usados para conectar sistemas legados com ferramentas de simulação, visualização de reconfiguração. Trata da integração de CPS heterogêneos.
- *Advancing the Performance of Complex Manufacturing Systems Through Agent-Based Production Control* (LÜDER; ZAWISZA; BECKER, 2017): no trabalho, é proposta uma arquitetura de controle baseada em agente, a qual permite comunicação entre uma oficina de reparos automotivos (com recursos e ordens de

consumidores), com uma loja de carros, de forma que eles possam negociar a melhor alternativa de escalonamento e sequência de ordens no caso de distúrbios. A arquitetura permite melhorias de performance no sistema de produção, em termos de saída, utilização de recursos, entre outros. O trabalho também discute a implementação e simulação do sistema, utilizando o *JADE framework*.

- *Optimization of auction mechanisms Agent-based coupling of cooperative production plants* (GEHLHOFF; LINNENBERG; FAY, 2017): no artigo, é apresentado um conceito para geração de benefícios adicionais para uma companhia, a partir de uma otimização baseada em um algoritmo de aprendizagem capaz de estimar as probabilidades para uma ordem futura, com base em um histórico.
- *ASID: Advanced System for Process Control towards Intelligent Specialization in the Power Engineering Field* (STAMATESCU et al., 2017): os autores apresentam uma arquitetura, o desenvolvimento e a implementação de um protótipo de um sistema integrado para a simulação de processos industriais automatizados, utilizando tecnologias avançadas de acordo com os princípios da i4.0. Trata-se de um simulador físico, real, que pode ser utilizado em campo de ensino.
- *Towards Industry 4.0 in Corporate Energy Management* (JUNKER; DOMANN, 2017): apresenta o desenvolvimento de um sistema de gestão de energia capaz de coletar dados de consumo no nível de plantas de produção, em formato de planilha, e um sistema de comunicação auto-organizável e modular – como uma combinação de *hardware* e *software*, que lê e utiliza esses dados em tempo real.
- *Autonomous and Flexible Multiagent Systems Enhance Transport Logistics* (GATH; HERZOG; EDELKAMP, 2014): no trabalho, é apresentado um sistema multiagente autônomo que otimiza o planejamento e o escalonamento de processos industriais utilizando como exemplo serviços de correio.
- *Agent-based Planning and Control for GroupageTraffic* (GATH; HERZOG; EDELKAMP, 2013): os autores tratam de um projeto de pesquisa e transferência de tecnologia, com um parceiro industrial. No trabalho, apresentam uma abordagem baseada em agente para modelar os processos atuais da organização, buscando explorar potenciais de otimização encontrados. Os autores utilizam

simulação para verificar o desempenho e a aplicabilidade do sistema, a partir de sua aplicação tendo por base o *framework* PlaSMA.

Apesar de haver trabalhos relacionados à Indústria 4.0 com abordagens de simulação, notou-se que nenhum trabalho abordado por Muhuri et al. (2019), tampouco encontrado na pesquisa realizada na Web of Science, propunha-se a analisar redes de produção da Indústria 4.0, como abordado nesta pesquisa. Passou-se, então, a buscar e analisar *softwares* e pacotes que se julgou poderem ser empregados para responder o problema desta pesquisa, em parte ou totalmente. A seguir, são discutidos o R (pacote *igraph*), Gephi e Wolfram Mathematica. A busca por estes programas foi feita tanto a partir de trabalhos acadêmicos da área, quanto pelas características oferecidas pelos *softwares*.

O R é um ambiente computacional de *software* voltado à computação estatística e aos gráficos (IHAKA; GENTLEMAN, 1996). Trata-se de uma suíte de *software*, que facilita a manipulação de dados, o cálculo e a visualização de gráficos. O R pode ser utilizado para a manipulação de redes (grafos), especialmente através do pacote *igraph*. O pacote fornece uma série de ferramentas para pesquisas com redes, capaz de lidar com grafos compostos por milhões de vértices e arestas (CSARDI; NEPUSZ, 2006). O *igraph* possibilita a criação, manipulação, e visualização de redes, bem como o cálculo de propriedades da topologia da rede e dos nós e arestas que a compõe, como diâmetro, distância média, *betweenness*, centralidade, entre outros. Embora o R, através do pacote *igraph*, seja capaz de manipular grafos (redes) complexos, não é possível simular a produção neste ambiente de forma nativa – isto é, seria necessário uma programação utilizando ele para a inserção destas funcionalidades. O *software* também necessita um conhecimento de sua sintaxe para operação, uma vez que sua interface é feita com linha de comando. Por esses motivos (interface por linha de comando e necessidade de implementação da parte de produção), o R não foi utilizado para a construção do simulador.

Um outro programa de manipulação de redes analisado foi o Gephi. Trata-se de um *software* de código aberto para análise de redes (grafos), que utiliza um renderizador 3D (BASTIAN; HEYMANN; JACOMY et al., 2009). Ainda segundo os autores, o *software* pode lidar com grandes redes – mais de 20.000 nós – com suporte a processamento paralelo. Trata-se de um *software* com interface gráfica, o que facilita sua manipulação, principalmente por pesquisadores de áreas não familiarizadas com linha de comando. O Gephi também possui funcionalidades para cálculos de informações da rede, como grau médio, grau médio ponderado, diâmetro da rede, densidade da rede, modularidade, coeficiente de aglomeração médio, centralidade, entre outros. Embora possua interface gráfica, optou-se por não utilizar o

Gephi neste trabalho, pois ele não traz funcionalidade de simulação de produção na rede – assim como o R, seria necessária a implementação desta funcionalidade. Como vantagens em relação ao iGraph, tem-se que a utilização de interface gráfica, que pode ser considerada mais amigável. Uma forma de utilizar o Gephi seria através da alteração de seu código fonte, para que incluísse as funcionalidades necessárias da simulação da produção. Todavia, isso demandaria tempo de estudo do funcionamento interno do *software*, e, além disso, algumas funcionalidades do sistema ficariam sem utilidade, como a visualização da rede, já que as redes analisadas nesta pesquisa são muito grandes para uma renderização.

Em Putnik et al. (2015), os autores fazem uma simulação de produção em uma rede com 8192 nós e aproximadamente 50.000 arestas. Três hipóteses são testadas em relação à robustez da rede e seu tamanho. Para a simulação da produção, foi utilizado o *software* Wolfram Mathematica, que é um programa pago e de código fechado. O Wolfram Mathematica é utilizado para computação técnica e possui mais de 5000 funções; ele é baseado na Wolfram Language, que permite uma programação envolvendo linguagem simbólica, computação matemática, análise de dados, cálculo de grafos, visualização em gráficos, entre outras funções (WOLFRAM, 2018). Apesar de ser utilizado para um propósito semelhante ao deste trabalho, o *software* Wolfram Mathematica foi descartado tanto por ser um *software* pago, quanto por sua interface com o usuário ser através de uma linguagem de programação, o que dificultaria a utilização por pessoas sem conhecimento nesta área.

Além destes, foram analisados três *softwares* de modelagem computacional, que possibilitam simulação da produção: Simio³, Arena⁴ e Flexsim⁵. A escolha por estes *softwares* foi feita devida à sua ampla utilização no campo de modelagem e simulação computacional. A análise de cada ferramenta foi feita tanto a partir do acesso aos seus sites e documentação, quanto pela instalação e testes de suas funcionalidades. Apesar de disponibilizarem suporte para a simulação de plantas produtivas, os *softwares* não foram construídos para trabalharem com redes grandes e complexas, como as da Indústria 4.0. Suas funcionalidades estão principalmente relacionadas com escalonamento e previsão de eventos futuros, tendo como base a definição de uma planta industrial completa, com seus detalhes – isto é, através da inclusão manual de cada item que compõe a cadeia produtiva. Por esses motivos, Simio, Arena e Flexsim não puderam ser utilizados nesta pesquisa.

3 <https://www.simio.com/>

4 <https://www.arenasimulation.com/>

5 <https://www.flexsim.com/>

2.5 HEURÍSTICAS E ALGORITMO GENÉTICO

Redes complexas, como as da i4.0, são formadas por milhares ou milhões de vértices e conexões. Dessa forma, torna-se difícil o uso de abordagens exatas para problemas envolvendo essas redes, que encontrariam uma solução ótima, devido ao altíssimo custo computacional envolvido; pode-se, então, utilizar uma abordagem heurística para buscar uma solução aproximada ao ótimo, a um custo e tempo menor (BOFFEY, 1982).

Métodos heurísticos podem ser utilizados em três situações: (i) quando há uma explosão de possíveis valores para as variáveis de decisão; (ii) quando há dificuldade em avaliar a função objetivo, devido à presença de variáveis estocásticas; e (iii) quando há condições que mudam acentuadamente com o tempo, sendo necessário uma série temporal de soluções, em vez de apenas uma única solução em um ponto do tempo (SILVER, 2004).

Quando o modelo acaba se afastando do ideal pretendido para a otimização, por ser muito complexo ou por não haver alternativas computacionais aceitáveis, técnicas heurísticas, não exatas, podem ser utilizadas para a solução de problemas, mesmo que não tenham garantia de sucesso (GOLDBARG; LUNA, 2005). Um método heurístico “admite sacrificar o ideal de perfeição sugerido pela otimização em troca de melhorar a eficiência de um processo de busca” (GOLDBARG; LUNA, 2005, p. 16). Métodos heurísticos podem ser utilizados para a obtenção de soluções para problemas que não teriam respostas ótimas em tempo hábil, sendo capazes de apresentar soluções de boa qualidade em tempo aceitável (ARROYO, 2002).

Silver (2004) traz uma série de categorias para as heurísticas, não excludentes:

- Soluções geradas de forma aleatória: consiste em gerar um conjunto de soluções factíveis, aleatoriamente, para o problema, e escolher a melhor.
- Decomposição do problema: busca particionar (decompor) um problema em um número de partes mais simples de solucionar.
- Métodos indutivos: algoritmos que generalizam a solução a partir de uma versão do problema que é menor, mais simples, e/ou parecida; ou ainda através de uma analogia.
- Métodos que reduzem o espaço de soluções: nestes algoritmos, a ideia é cortar o número de soluções disponíveis/consideradas, buscando não afetar a qualidade da solução obtida. Isso pode ser feito através da inserção de novas restrições, ou do “aperto” daquelas já existentes.

- Métodos de aproximação: consiste na manipulação do modelo matemático ou na utilização de um modelo relacionado mais simples.
- Métodos construtivos: utilizam os dados do problema para a construção da solução, passo a passo. Usualmente, a solução só é obtida de forma completa ao fim do processo. Uma abordagem especial do método construtivo é a abordagem gulosa, que sempre busca adicionar o próximo elemento com o melhor benefício imediato.
- Métodos de busca em vizinhança local: estes métodos partem de uma solução possível para o problema (obtida, por exemplo, a partir de um método construtivo), e escolhe a melhor solução disponível dentro de um conjunto de vizinhos – caso não haja solução melhor, o processo chegou ao final. A vizinhança de uma solução X é o conjunto de soluções que podem ser obtidas a partir de X com alguma transformação simples.

Além dessa classificação, Silver (2004) também traz o conceito de metaheurística: trata-se de um conjunto de algoritmos heurísticos de maior nível, projetados para guiar métodos ou processos a encontrar soluções factíveis para problemas difíceis de otimização matemática. As metaheurísticas, segundo o autor, estão preocupadas em evitar estacionar em ótimos locais (soluções que são as melhores dentro de uma vizinhança, mas que não necessariamente são as melhores para o problema), e na redução do espaço de busca. Cada metaheurística contém um conjunto de parâmetros ajustáveis, o que permite flexibilidade.

Dentre as metaheurísticas, tem-se os Algoritmos Genéticos (AG), que são métodos de otimização inspirados nos mecanismos de evolução, seguindo princípios da seleção natural e da sobrevivência dos mais aptos introduzidos por Charles Darwin; os algoritmos genéticos foram introduzidos por John Holland, em 1975, e popularizados por David Goldberg, um de seus estudantes, em 1989 (LACERDA; CARVAHO, 1999).

Ao seguir os princípios de Darwin, os AG privilegiam indivíduos com maior capacidade e maior probabilidade de reprodução; indivíduos com maior número de descendentes têm maiores chances de perpetuarem seus códigos genéticos que, no caso de um AG, trata-se da solução para o problema em questão (PACHECO, 1999).

Uma série de analogias é feita entre o sistema real e o algoritmo genético: chama-se *cromossomo* uma possível solução para o problema tratado (PACHECO, 1999), que geralmente é formada por uma cadeia de valores (*genes*) (LACERDA; CARVALHO, 1999). O funcionamento do AG consiste em, primeiramente, definir uma *população* inicial de cromossomos (soluções para o problema a ser otimizado), que pode ser criada de forma

aleatória. A partir dessa população, tem-se uma *evolução*, que inicia com a avaliação de cada cromossomo integrante da população a partir da definição de uma nota (chamada de *aptidão*). Essa nota pode ser definida com base na função objetivo do problema em questão. Usualmente, cromossomos mais aptos são selecionados como *membros* de uma *população intermediária*, enquanto que os menos aptos são descartados (*seleção natural*). Após a definição da população intermediária, realizam-se os processos de *crossover* (*reprodução*) e *mutação*. O *crossover* é realizado em um determinado número de pares de cromossomos, a partir da combinação dos seus valores: pode-se definir um ponto de corte no cromossomo, e unir a “cabeça” de um dos pais com a “cauda” do outro, gerando, assim, dois filhos que são adicionados à nova população, conforme a Figura 1. O *crossover*, em geral, somente é realizado em determinada probabilidade de pares de cromossomos, de forma que algumas soluções permaneçam inalteradas de uma população para outra. A escolha dos cromossomos para formar cada par para a reprodução (*crossover*) pode seguir uma série de algoritmos – por exemplo, uma escolha completamente aleatória, ou uma escolha por torneio, em que é selecionado um número pré-definido de cromossomos e escolhido, dentre eles, o de melhor aptidão para ser um dos elementos envolvidos na reprodução. Após o *crossover*, aplica-se o operador de mutação em cada um dos filhos gerados. A mutação consiste na possível alteração dos valores de cada um dos genes do cromossomo, com base em uma taxa de probabilidade (geralmente baixa, entre 0,1% a 5%), como ilustrado na Figura 2. A mutação, dessa forma, ajuda na diversificação das soluções geradas pela reprodução. Além do *crossover* e da mutação, uma estratégia usualmente aplicada é o *elitismo*, que consiste em manter a(s) melhor(es) solução(ões) de uma população para outra, tendo em vista que ela(s) pode(m) ser perdida(s) com a reprodução. Com isso, tem-se uma nova população, na qual o procedimento pode ser repetido, até determinado ponto de parada definido (LACERDA; CARVALO, 1999). O pseudo-código de um modelo de algoritmo genético está elencado em Algoritmo 1.

Figura 1 – Exemplo de *crossover*

Pai 1	0	0	1	1	1	0	1	0	1	0	1
Pai 2	1	1	0	1	0	1	0	1	1	0	1
Filho 1	0	0	1	1	1	0	0	1	1	0	1
Filho 2	1	1	0	1	0	1	1	0	1	0	1

Fonte: Elaborado pelo autor (2019)

Figura 2 – Exemplo de *mutação*

Filho 1	0	0	1	1	1	0	0	1	1	0	1
Filho 2	1	1	0	1	0	1	1	0	1	0	1

Após mutação

Filho 1	0	<u>1</u>	1	1	1	<u>1</u>	0	1	<u>0</u>	0	1
Filho 2	1	1	<u>1</u>	1	0	1	1	0	1	<u>1</u>	<u>0</u>

Fonte: Elaborado pelo autor (2019)

Algoritmo 1 – Modelo de Algoritmo Genético

população ← criar população aleatória

enquanto critério de parada não for satisfeito **faça**

 população_intermediária ← selecionar indivíduos mais aptos da população

 população ← realizar *crossover* usando indivíduos da população intermediária

 população ← realizar mutação nos indivíduos da população

fim enquanto

Fonte: Elaborado pelo autor (2019)

3 MÉTODO DE PESQUISA

As ciências naturais buscam estudar objetos ou fenômenos do mundo, em especial, verificando suas características ou propriedades, ou ainda, como estes fenômenos ou objetos interagem entre si (SIMON, 1996). As ciências tradicionais, como as ciências sociais e naturais, buscam explicar, descrever, explorar ou prever fenômenos e suas relações (DRESCH; LACERDA; ANTUNES JÚNIOR, 2015).

O mundo onde vivemos atualmente, por outro lado, é muito afetado pela ação do ser humano, com a criação de artefatos artificiais, que obedecem às leis da natureza, e são adaptados aos propósitos humanos (SIMON, 1996). Assim, em muitas ocasiões, o interesse está em estudar o projeto, a construção ou a criação de um artefato, isto é, objetiva-se solucionar problemas através de uma pesquisa científica. Para tanto, pode-se utilizar o paradigma epistemológico da *Design Science* (DRESCH; LACERDA; ANTUNES JÚNIOR, 2015), a “Ciência do Artificial”, no sentido de artefato construído pelo homem (SIMON, 1996).

Van Aken (2004) divide as disciplinas científicas em três categorias:

1. Ciências formais, como filosofia e matemática: tem por missão construir sistemas de proposições com validação, que está na consistência lógica interna.
2. Ciências explanatórias, como ciências naturais e a maior parte das ciências sociais: buscam descrever, explicar, e possivelmente prever fenômenos observáveis em seu campo. Suas proposições devem ser verdadeiras, com base nas evidências observadas, e em modelos de causa e consequência.
3. *Design sciences*, como ciências de engenharia, medicina e psicoterapia moderna: têm como objetivo desenvolver conhecimento para o projeto e a construção de artefatos, resolvendo problemas, ou ainda melhorando soluções existentes.

A *Design Science* (DS) busca a prescrição em detrimento da explicação, isto é, seu foco está na solução e não no problema em si; a DS atenta, ainda, não apenas ao rigor, mas também à relevância da pesquisa (DRESCH; LACERDA; ANTUNES JÚNIOR, 2015). Von Alan et al. (2004) indicam que a DS é um paradigma distinto, porém complementar, ao paradigma comportamental, sendo que este busca desenvolver e justificar teorias que explicam ou preveem fenômenos organizacionais.

O termo *Design Science* faz alusão ao objetivo da pesquisa, que é a construção (projeto, *design*) de conhecimento para a resolução de problemas (VAN AKEN, 2004), como é o caso desta pesquisa. Ainda segundo o autor, não basta apenas entender um problema,

como nas ciências explanatórias: é preciso construir conhecimento útil ao desenvolvimento de soluções para problemas da área estudada. Entender o problema é apenas parte do caminho para resolvê-lo; é preciso, ainda, projetar, desenvolver e testar soluções (VAN AKEN, 2004).

As pesquisas orientadas pela DS buscam desenvolver conhecimentos científicos capazes de promover o projeto de artefatos ou intervenções por profissionais; isto é, ela não está preocupada com a ação em si, mas sim com a construção de conhecimento para o projeto de soluções (VAN AKEN, 2004).

Dresch, Lacerda e Antunes Júnior (2015) resumiram uma lista de conceitos relacionados à DS:

1. *Design Science*: procura consolidar conhecimentos sobre o projeto e o desenvolvimento de soluções para melhorar sistemas existentes, resolver problemas e criar artefatos. Busca projetar e não apenas aplicar.
2. Artefato: algo construído pelo homem; interface entre o ambiente interno e o ambiente externo de um sistema.
3. Soluções satisfatórias: soluções suficientemente adequadas para o contexto em questão. Devem ser viáveis e não necessariamente ótimas (solução ideal).
4. Classes de problema: organização que orienta a trajetória e o desenvolvimento do conhecimento no âmbito da Design Science. As soluções devem permitir a generalização das prescrições para determinada classe de problemas.
5. Validade pragmática: soluções devem ser úteis, considerando custo/benefício, particularidades do ambiente e reais necessidades dos interessados na solução.

O método de pesquisa utilizado no desenvolvimento desta tese é a *Design Science Research* (DSR), um “método que fundamenta e operacionaliza a condução da pesquisa quando o objetivo a ser alcançado é um artefato ou uma prescrição” (DRESCH; LACERDA; ANTUNES JÚNIOR, 2015, p. 67). Trata-se, portanto, do método de pesquisa para a ciência do artificial, isto é, para o paradigma epistemológico da *Design Science*.

A *Design Science Research* busca diminuir o distanciamento entre teoria e prática, sustentando o desenvolvimento de artefatos e contribuindo para o fortalecimento de bases de conhecimento (DRESCH; LACERDA; ANTUNES JÚNIOR, 2015). Segundo Van Aken (2004, p. 2, tradução do autor), “todas as boas teorias são práticas, mas algumas são mais práticas que outras”. Sete critérios fundamentais para a DSR são elencados por Dresch, Lacerda e Antunes Júnior (2015):

1. Design como artefato: as pesquisas desenvolvidas pelo método da DSR devem produzir artefatos viáveis, na forma de um construto, modelo, método ou de uma

instanciação. Para esta pesquisa, o artefato produzido será um simulador, isto é, um construto.

2. Relevância do problema: a DSR busca desenvolver soluções para problemas importantes e relevantes para as organizações.
3. Avaliação do design: a utilidade, qualidade e eficácia do artefato devem ser rigorosamente demonstradas por meio de métodos de avaliação bem executados. A avaliação do simulador proposto nesta pesquisa ocorreu através de testes de sua execução e da análise de especialistas.
4. Contribuições da pesquisa: deve-se prover contribuições claras e verificáveis nas áreas específicas dos artefatos desenvolvidos e apresentar fundamentação clara em fundamentos e/ou metodologias de design.
5. Rigor da pesquisa: a pesquisa precisa ser desenvolvida a partir de métodos rigorosos, tanto na construção quanto avaliação do(s) artefato(s).
6. Design como processo de pesquisa: deve-se utilizar os meios disponíveis para alcançar os fins desejados e satisfazer as leis que regem o ambiente onde se encontra o problema estudado.
7. Comunicação da pesquisa: é preciso publicar a pesquisa, com seus resultados, para o público interessado.

Dresch, Lacerda e Antunes Júnior (2015) definem uma série de passos que devem ser seguidos para o desenvolvimento de uma pesquisa baseada na metodologia do *Design Science Research*. Os passos são:

1. Identificação do problema: a primeira etapa do processo trata da identificação do problema a ser estudado, surgindo, especialmente, com base no interesse do pesquisador e justificado principalmente em termos de relevância.
2. Conscientização do problema e revisão sistemática da literatura: nesta etapa, é necessário compreender o problema, buscando informações sobre suas causas e o contexto no qual está inserido. Este passo é importante para se ter acesso ao conhecimento necessário à construção do artefato e resolução do problema.
3. Identificação dos artefatos e configuração das classes de problemas: a revisão sistemática da literatura poderá indicar ao pesquisador a existência de artefatos e classes de problemas existentes para o problema que se busca resolver. Nesta etapa, são procurados artefatos já existentes para o problema da pesquisa, ou problemas similares. Caso existam, eles devem ser verificados, conferindo se algum pode ser

adaptado ao problema em questão. Este passo também é útil para que se obtenha conhecimento a partir das soluções já desenvolvidas encontradas.

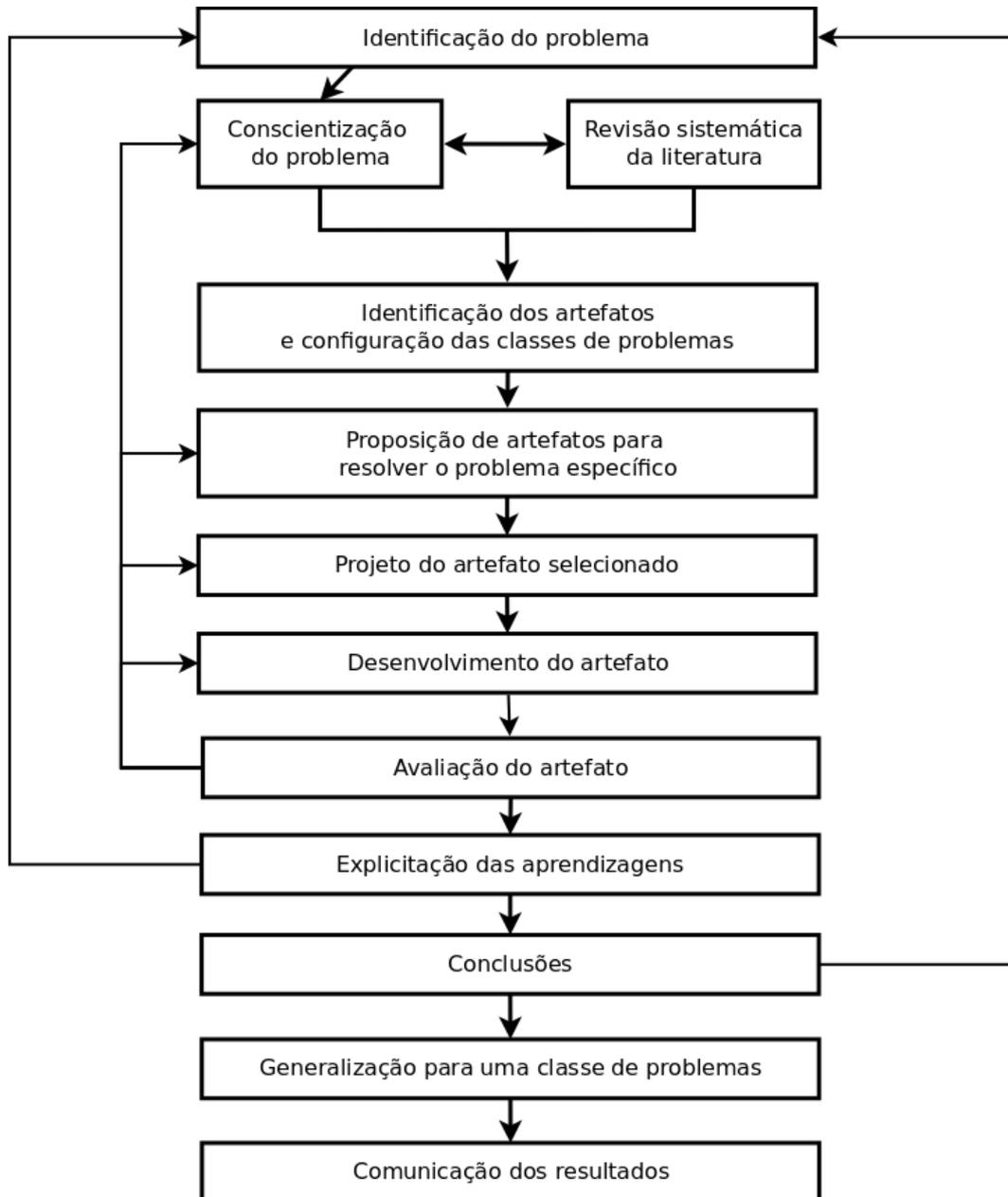
4. Proposição de artefatos para resolução de problemas: a partir da prévia identificação de artefatos similares e da classe de problema em que se enquadra a pesquisa, na quarta etapa é feita a proposição do artefato a ser construído para solucionar o problema. Trata-se de um passo de proposição de soluções, através de um processo criativo. Utiliza-se, essencialmente, um método abduutivo para propor soluções robustas, capazes de melhorarem a situação atual do problema.
5. Projeto do artefato: após a proposição do artefato, deve-se projetá-lo. É preciso considerar tanto as características internas, quanto o contexto no qual ele estará inserido. Essa etapa implica a descrição dos procedimentos para a construção e avaliação do artefato, considerando o desempenho esperado para garantir uma solução satisfatória para o problema, garantindo, assim, o rigor da pesquisa.
6. Desenvolvimento do artefato: de posse do projeto do artefato, passa-se a sua construção. O artefato pode ser construído utilizando diferentes abordagens, como algoritmos computacionais, protótipos, maquetes. Esta etapa terá como saídas tanto o artefato construído, quanto o processo de sua construção, sendo uma das contribuições para o avanço da geração de conhecimento.
7. Avaliação do artefato: a etapa seguinte ao desenvolvimento do artefato trata de sua observação e avaliação, verificando, principalmente, se ele atende aos requisitos previamente definidos para solucionar o problema. É possível realizar a avaliação tanto em um ambiente real, onde ocorre o problema investigado, quanto em um ambiente experimental. Caso sejam verificados problemas, por exemplo, se o artefato não atender a algum requisito, deve-se retornar aos passos iniciais da pesquisa, buscando incrementá-lo.
8. Explicitação das aprendizagens e conclusão: findada a avaliação do artefato e considerando que ele atingiu os objetivos esperados, a próxima etapa da pesquisa consiste na documentação e divulgação dos resultados. É importante, nesta etapa, apontar as limitações verificadas no decorrer do trabalho, bem como novos *insights* que tenham surgidos, que podem auxiliar a condução de novos trabalhos.
9. Generalização para uma classe de problemas e comunicação dos resultados: finalmente, deve-se generalizar a solução encontrada para a classe de problema na qual a pesquisa se enquadra. Isso permitirá a utilização do conhecimento obtido em outras situações similares, de maneira indutiva. Também é fundamental que haja

publicações do conhecimento adquirido em revistas, periódicos, congressos, para que seja utilizado por outros interessados na temática, seja no meio acadêmico ou não.

Esta pesquisa foi conduzida com base nos critérios fundamentais elencados por Dresch, Lacerda e Antunes Junior (2015), e seguindo a série de passos propostos por eles, ilustrados na Figura 3. As etapas de identificação e conscientização do problema estão englobadas na Introdução do trabalho (capítulo 1). A revisão da literatura foi realizada no capítulo 2 desta tese. Dentro da revisão, na seção 2.4, foi feita a identificação de artefatos, a partir da investigação de ferramentas semelhantes que poderiam ser utilizadas para alcançar o objetivo desta pesquisa. Os capítulos a seguir tratam dos demais passos da pesquisa.

Após a revisão da literatura e identificação de artefatos, passou-se a construção do simulador. Para tanto, inicialmente foi definido o modelo da rede de produção, e um conjunto de requisitos que o simulador deveria atender, com base na literatura, no problema tratado e nos objetivos desta pesquisa. Nesta etapa, também foi definido o problema de determinação de nós para atender a uma demanda dentro da rede de produção, e foram elaborados três algoritmos não-ótimos para solucionar este problema. Posterior às definições, o simulador foi implementado e validado. A finalização do trabalho aconteceu com a execução de testes dentro do simulador, comparando diferentes cenários de redes de produção, e a conclusão do trabalho, com a explicitação dos resultados e a generalização para a classe de problemas definida.

Figura 3 – Fluxograma da Design Science Research



Fonte: Elaborado pelo autor com base em Dresch, Lacerda e Antunes Júnior (2015)

4 UM SIMULADOR DE PRODUÇÃO EM REDES COMPLEXAS DA INDÚSTRIA 4.0

No estudo de redes complexas, simuladores são alternativas aos testes em ambientes reais que, em geral, são demasiadamente custosos (BRASE; BROWN, 2009), e muitas vezes impraticáveis, como é o caso das redes de produção da Indústria 4.0. Para atingir os objetivos desta pesquisa, fez-se necessário a construção de um *software* que possibilite emular, em parte, o comportamento da produção. O desenvolvimento do simulador envolve as etapas de proposição, projeto, e desenvolvimento de artefatos da metodologia de DSR proposta por Dresch, Lacerda e Antunes Júnior (2015).

A construção do simulador foi realizada em um processo iterativo e incremental, a partir de uma definição inicial de requisitos, desenvolvimento, avaliação e reinício do ciclo; em cada etapa de desenvolvimento, tem-se como saída um subconjunto do sistema final, funcional, e não apenas um protótipo (LARMAN, 2004). O que se apresenta aqui, nesta tese, é o resultado final deste processo incremental. Optou-se pela construção de um novo *software*, pois não se encontrou nenhum sistema ou ferramenta que satisfizesse o objetivo desta pesquisa, como visto na seção 2.4.

A construção do simulador iniciou com a modelagem do processo de produção em uma rede da i4.0. Para tanto, utilizou-se como base o trabalho de Škulj et al. (2014) – em que tarefas de produção são definidas como conjuntos de letras do alfabeto, e podem ser realizadas por um ou mais nós que possuam habilidades para tanto. Definiu-se que os nós da rede/grafos simbolizam unidades de produção e/ou consumo, e que estão relacionadas entre si através de arestas (conexões). Os nós podem representar qualquer item envolvido no processo de produção da fábrica, buscando virtualizar todo o processo, como um *Digital Twin*, atendendo ao quarto ponto da agenda de Viera et al. (2018) para a simulação da i4.0. Cada nó é identificado por um número único.

As letras do alfabeto (A, B, C, ..., Z) foram utilizadas para definir os itens de produção e consumo, como proposto em Škulj et al. (2014). Cada nó da rede pode possuir um conjunto de habilidades, formado por um ou mais itens de produção/consumo (letras) diferentes. Cada item de habilidade (letra) possui três características: (i) custo, (ii) capacidade (quantidade de itens disponíveis), e (iii) tempo para produção do item. Essas três características de cada habilidade foram escolhidas pelas suas importâncias no processo produtivo. A capacidade é uma restrição, e indica a quantidade máxima de atividade que uma organização pode atender; já o custo é o objetivo mais importante do desempenho de uma organização, e leva em conta gastos com funcionários, instalações, materiais, entre outros (SLACK et al., 1999);

finalmente, o tempo é uma variável necessária para definir o custo, a alocação de equipe, quando a tarefa será concluída, entre outras informações vitais para o gerenciamento de uma operação (SLACK; JOHNSTON; BRANDON-JONES, 2015).

No modelo proposto, as arestas simbolizam as conexões entre os nós da rede, e possuem um valor único de peso – que indica o somatório de todos os custos envolvidos para que um item saia de um nó e chegue até outro.

Cada nó da rede pode demandar uma ou mais tarefas. Uma tarefa é formada por um conjunto de itens (letras), que devem ser processados a partir dos nós conectados ao nó que demanda a tarefa. As letras de uma tarefa podem simbolizar qualquer item envolvido no processo produtivo, desde uma matéria-prima até um serviço. Um nó pode processar a tarefa em parte ou na totalidade, dependendo de seus itens de habilidade disponíveis (ainda com capacidade) e da limitação de tempo imposta. Definiu-se como “fornecedores” os nós que fornecem algum item para atender a uma determinada tarefa. Podem ser fornecedores tanto os nós que são vizinhos imediatos ao nó que demanda a tarefa, quanto outros nós, desde que haja um caminho a partir deles até o nó solicitante da tarefa.

Dessa forma, o atendimento a determinada demanda (tarefa), dentro de uma rede complexa de produção da i4.0, foi definido como a determinação de (i) um ou mais nós com habilidade(s) para atendê-la, respeitando requisitos de tempo e capacidade; e (ii) do caminho que deve ser percorrido desde os nós fornecedores até o nó que está solicitando a tarefa. O custo total de uma tarefa é definido como o somatório entre os custos de produção dos itens dos fornecedores, e dos pesos das arestas no caminho definido entre os nós fornecedores e o nó que está realizando a demanda. Embora seja possível conhecer os nós com habilidades e capacidades para atender a uma tarefa, com os menores custos de produção, a definição desses nós não é de fácil cálculo, uma vez que o custo total depende também do caminho a ser percorrido ao longo da rede.

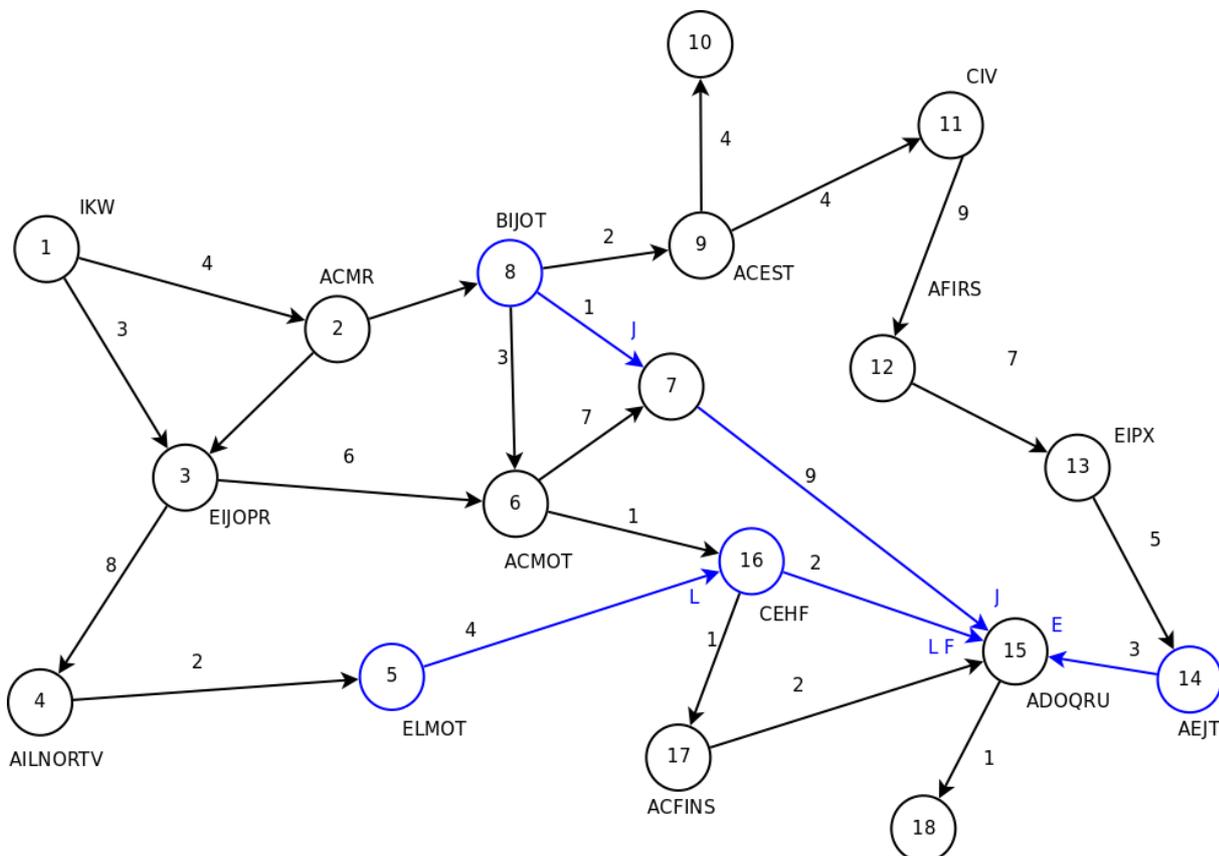
Ressalta-se, ainda, que as três características definidas (custo, tempo e capacidade) podem ser alterados por outras, desde que se mantenha a mesma semântica, isto é, pode-se substituir a variável custo por outra variável que se deseja minimizar no processo produtivo; e as variáveis tempo e capacidade por outras variáveis que sejam restrições.

Após a construção do modelo de produção, ele foi validado com um especialista em modelagem (doutorando em curso de Pós-Graduação em Administração) e um especialista em produção (Doutor em Administração), que aprovaram a definição realizada. A Figura 4 traz um exemplo de rede de produção construída neste trabalho, com os nós conectados e suas respectivas habilidades (conjunto de letras). Também é exibido, nesta figura, um exemplo de

tarefa (EFJL) sendo executada, demanda pelo nó 15. Verifica-se, neste caso, que o nó 14 fornece o item E, através do caminho (14, 15); o nó 16 fornece o item F, através do caminho (16, 15); o nó 8 fornece o item J, através do caminho (8, 7), (7, 15); e o nó 5 fornece o item L, através do caminho (5, 16), (16, 15).

Essa tarefa (EFJL) pode representar, como um exemplo de produção em alto nível de abstração, a construção de um computador pessoal. Neste caso, o item E simboliza um gabinete, o item F indica um monitor, o item J representa um teclado, e o item L um mouse. Cada nó, na rede, pode representar um fornecedor ou consumidor de peças de informática. As arestas indicam as conexões existentes entre esses participantes da rede, e seus pesos representam os custos de envio. A execução da tarefa consiste, portanto, em escolher os fornecedores de cada parte do computador, e o caminho que essas partes (peças) percorrerão até chegar ao nó que solicitou elas. Assim, ao final da execução da tarefa, o nó 15 terá um computador conforme solicitado, com os quatro itens.

Figura 4 – Exemplo de rede de produção



Fonte: elaborado pelo autor (2019)

A partir do objetivo do trabalho e da agenda proposta por Viera et al. (2018) para simulação em i4.0, foi elencado um conjunto de requisitos que o simulador deve atender, para que consiga realizar, de forma satisfatória, a simulação da produção em redes da i4.0 e, assim, contribuir para a melhoria da gestão da produção neste contexto. Os requisitos estão definidos a seguir:

1. Possibilitar a leitura de redes aleatórias em formato do *software* Pajek. Uma vez que o simulador trabalha com redes muito grandes, as quais usualmente não são reais, é necessário permitir a utilização de redes aleatórias. O Pajek é um programa para análise de grandes redes (BATAGELJ; MRVAR, 1998) que permite, entre outras coisas, a geração de redes aleatórias com características de pequenos mundos (*small-world*) ou distribuição de grau livre de escala. As redes geradas no Pajek podem ser salvas em arquivo-texto, em um formato específico do *software*. Julgou-se, dessa forma, como requisito do sistema a possibilidade de ler arquivos no formato gerado pelo Pajek, para que seja possível aproveitar essa funcionalidade de geração de redes aleatórias.
2. Permitir a edição da rede. Embora provavelmente a rede seja construída de forma automática, utilizando algoritmos aleatórios, pode ser necessário realizar alterações pontuais em seus nós – como inclusão de novas arestas, alteração de habilidades, entre outras. Assim, julgou-se necessário que o simulador permita a edição de nós, possibilitando a avaliação de diferentes cenários, o que é um ponto da agenda proposta por Viera et al. (2018) para a simulação da i4.0.
3. Possibilitar a criação de pesos aleatórios para as arestas: tendo em vista que todas as arestas dos grafos gerados de forma aleatória no Pajek tem peso igual a um, é preciso possibilitar que sejam definidos pesos aleatórios para as arestas das redes, sendo delimitados por valores mínimo e máximo definidos pelo usuário do simulador.
4. Criar habilidades aleatórias para os nós da rede: uma vez que os nós da rede precisam ter conjuntos de habilidades e que o número de nós analisados será grande, é necessário que o simulador consiga gerar essas habilidades de forma aleatória. Para possibilitar uma personalização da rede, definiu-se que as habilidades devem ser criadas para nós com grau (número de arestas) maior ou igual a determinado valor; e que devem ser indicados: (i) os números mínimo e máximo de habilidades que cada nó pode ter; (ii) os valores mínimo e máximo para a capacidade de cada habilidade; (iii) os limites inferior e superior de custo; e

- (iv) os limites inferior e superior de tempo de cada item de habilidade. Estes valores devem ser criados de forma aleatória, sem relação entre si.
5. Exibir a distribuição de grau da rede. O simulador deve exibir a distribuição de grau dos nós da rede, pois, a partir dessa informação, pode-se gerar habilidades diferentes para diferentes tipos de nós, bem como remover nós com maior ou menor grau, auxiliando a construção e avaliação de diferentes cenários.
 6. Permitir a remoção de nós. Com a exclusão de nós, é possível simular falhas na rede, permitindo a análise de características como robustez e resistência, entre outras. Foi definido que a remoção pode ser feita com base no identificador dos nós que devem ser removidos; com base em seu grau (isto é, remover nós com grau maior que determinado valor indicado); ou ainda aleatoriamente, definindo um número específico de nós que devem ser excluídos da rede. A exclusão de nós deve remover suas habilidades e as conexões que passam por eles.
 7. Possibilitar a criação de tarefas. O simulador deve possibilitar tanto a inserção de uma tarefa específica, quanto a construção de um número determinado de tarefas aleatórias. Dessa forma, pode-se analisar a execução dessas tarefas, verificando o comportamento da rede de produção.
 8. Permitir a alteração de habilidades em lote. A alteração de valores das habilidades pode impactar no comportamento da rede. Portanto, é necessário possibilitar que o usuário do simulador faça essa alteração de forma simples, sem que precise alterar nó a nó. Assim, foi definido como requisito a possibilidade de alterar os valores de custo, tempo ou capacidade de cada item de produção/consumo. A alteração pode ser feita de duas formas: (i) definindo um valor padrão para todos os valores daqueles itens de habilidade; ou (ii) aumentando/diminuindo cada valor em uma determinada porcentagem.
 9. Armazenar a rede completa em formato de arquivo. Tendo em vista que o arquivo gerado pelo Pajek traz apenas informações da rede (nós e vértices), o simulador deve ser capaz de manipular, em arquivo, dados de habilidades dos nós da rede. Assim, pode-se criar modelos reutilizáveis, outro ponto indicado por Viera et al. (2018).
 10. Possuir interface gráfica: uma vez que o estudo da indústria 4.0 pode ser realizado por pessoas de diferentes áreas do conhecimento, elencou-se a necessidade da construção de uma interface gráfica para o simulador. Assim, facilita-se o uso do

simulador como uma ferramenta de instrução, item também proposto na agenda de Viera et al. (2018).

Após a elaboração dos requisitos do sistema, partiu-se para a determinação do problema de definição de nós para atender a uma tarefa de produção em redes da i4.0.

4.1 O PROBLEMA DE DEFINIÇÃO DE NÓS PARA EXECUÇÃO DE UMA TAREFA DE PRODUÇÃO EM REDES DA INDÚSTRIA 4.0

Após a construção do modelo de rede de produção da i4.0 e dos requisitos para o simulador, foi feita a definição formal do problema de determinação de nós para a execução de uma tarefa dentro da rede, discutido nesta seção.

Uma rede complexa de produção da Indústria 4.0 pode ser definida como um grafo $G = (V, E)$, sendo V um conjunto de N vértices, $V = \{1, 2, 3, 4, \dots, N\}$, representando as unidades de produção e consumo; e E um conjunto de arestas $E = \{(i, j) \mid i, j \in V, \text{ quando existe a aresta de } i \text{ até } j\}$, representando suas conexões. O conjunto de pesos (custos) das arestas é dado pela matriz W , sendo que um elemento $w_{ij} \in W$ indica o peso para uma transação saindo de i e indo para j , para cada par $(i, j) \in E$. O peso das arestas indica todo e qualquer custo envolvido para a execução de parte de uma tarefa em um nó vizinho ao atual (exemplo: transporte). Cada nó n da rede possui um conjunto de habilidades, e a função $S(i)$ retorna o conjunto de habilidades de um determinado nó i ; cada habilidade é definida como uma letra do alfabeto (A, B, C, \dots, Z) . Para cada habilidade s , de um nó i , tem-se um custo, definido através da matriz C , cujos elementos $c_{i,s} \in C$ indicam o custo para executar a tarefa s no nó i . O tempo de resposta é definido pela matriz T , cujos elementos $t_{i,s} \in T$ indicam o tempo para a disponibilização de um item de habilidade s pelo nó i . A quantidade de um item de habilidade disponível é definido pela matriz Q , com elementos $q_{i,s} \in Q$, que armazenam a quantidade disponível da habilidade s para o nó i .

Dada uma tarefa X , oriunda de um vértice $v \in V$, definida como um conjunto não ordenado de letras, busca-se definir, para cada elemento de X , um vértice do grafo G (elemento de V) capaz de atender ao item da demanda (chamado de fornecedor), bem como o caminho deste vértice fornecedor até o vértice v , gerando o menor custo. Restrição de tempo deve ser aplicada, e a disponibilidade de recursos deve ser considerada conforme a execução das tarefas.

Dessa forma, a solução para o problema pode ser representada através de uma matriz N que relaciona a tarefa X com os nós de V , de forma que um elemento $n_{i,j} \in N$ assumo valor

1 quando o elemento $j \in V$ seja responsável pelo item de tarefa $i \in X$, e 0 caso contrário. Além disso, a matriz P também faz parte da solução, relacionando cada elemento da tarefa com o conjunto de arestas E . Um elemento $p_{i,j} \in P$ assume o valor 1 quando a aresta $j \in E$ faz parte do caminho da solução do item de tarefa i , e 0 caso contrário. Uma restrição de tempo, $TIME$, também se aplica ao problema como uma restrição. O problema, então, consiste em:

$$\min \sum_{x \in X} \sum_{v \in V} n_{x,v} \cdot c_{v,x} + \sum_{x \in X} \sum_{e \in E} p_{x,e} \cdot w_e \quad (3)$$

sujeito a:

$$\sum_{x \in X} \sum_{v \in V} n_{x,v} \cdot t_{v,x} \leq TIME \quad (4)$$

$$q_{v,x} \geq n_{x,v} \quad \forall v \in V, \forall x \in X \quad (5)$$

onde:

X = conjunto com os itens da tarefa

V = conjunto de vértices da rede

E = conjunto de arestas da rede

$TIME$ = restrição de tempo

$n_{i,j} \in \mathbb{N}$ assume o valor 1 quando o elemento $j \in V$ é responsável pelo item de tarefa $i \in X$

$c_{i,s} \in \mathbb{C}$ indica o custo para executar a tarefa s no nó i .

$p_{i,j} \in P$ assume o valor 1 quando a aresta $j \in E$ faz parte do caminho da solução da tarefa i , e 0 caso contrário

$w_{i,j} \in W$ indica o peso para uma transação saindo de i indo para j , para cada par $(i, j) \in E$

$t_{i,s} \in T$ armazena o tempo para a disponibilização de um item de habilidade s pelo nó i

$q_{i,s} \in Q$ armazena a quantidade disponível do item de habilidade s para o nó i .

Modelos de simulação convencionais usualmente utilizam parâmetros determinísticos, que são definidos uma só vez e não são mais modificados. Por outro lado, a manufatura e logística no mundo real trata dados que são alterados constantemente, o que pode levar a desencontros entre a informação do mundo real e a utilizada na simulação (TAN et al., 2019). Dessa forma, cada tarefa dentro do simulador é tratada de forma independente, possibilitando que a rede seja alterada, quando necessário, a cada execução. Isso também atende a um ponto da agenda proposta por Viera et al. (2018), que indica que os dados da simulação possam ser atualizados em tempo real.

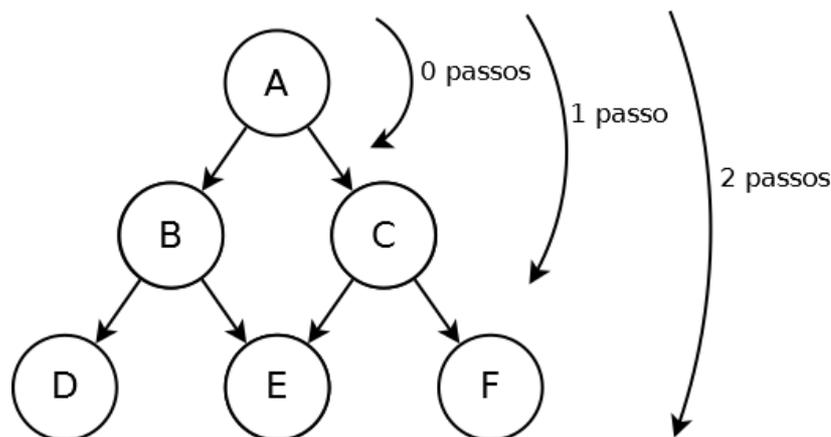
Tendo em vista a quantidade de variáveis envolvidas no problema, assim como a complexidade para a definição de caminhos em grafos, foram definidas três formas para a execução de uma tarefa na rede de produção proposta, que não necessariamente levam à

solução ótima: (i) realizando uma busca de nós em profundidade com política *first-fit* e limite de passos; (ii) realizando uma busca de nós em largura com política *first-fit* e limite de passos; e (iii) através de uma heurística baseada no algoritmo genético. As seções a seguir tratam dessas formas de execução da tarefa.

4.2 EXECUÇÃO DA TAREFA UTILIZANDO BUSCA EM PROFUNDIDADE E BUSCA EM LARGURA

A execução de uma tarefa na rede consiste na escolha de um conjunto de nós com habilidades disponíveis para processá-la, e da definição do caminho até o nó que está demandando a tarefa. Uma forma de definir esse conjunto é varrendo a rede a partir do nó solicitante, passando pelos nós conectados a ele em um método recursivo, até que encontre nós com habilidade para atender a tarefa, ou atinja um número máximo de passos pré-definido – profundidade em número de saltos para nós filhos. Se o número máximo de passos for 0, apenas o nó inicial será testado; caso seja 1, serão testados os filhos do nó inicial; caso seja 2, os filhos dos filhos também serão testados. A Figura 5 ilustra parte de uma rede, com os nós testados para cada número de passos definido.

Figura 5 – Nós testados com 0, 1 e 2 passo(s)



Fonte: elaborado pelo autor (2019)

Foram definidas duas formas para a busca por nós com habilidades para execução de uma tarefa na rede: busca por largura (*breadth-first search*) e busca por profundidade (*depth-first search*), dois métodos de busca de nós em grafos (CORMEN, 2009). A busca em largura parte de determinado nó origem s e testa, primeiramente, todos os nós que estão ligados à s pelas suas arestas. Feito isso, repete-se o procedimento elencando como novo nó origem um

dos nós ligados diretamente à s . Já a busca em profundidade tem como estratégia procurar o mais profundo possível na rede, isto é, varre-se o grafo a partir de determinado nó s , buscando sempre no primeiro nó conectado diretamente à s ($s1$) e, recursivamente, no primeiro nó conectado ao nó visitado anteriormente ($s1$). Ao chegar em um nó sem conexões com nós ainda não visitados, a busca retrocede e testa o próximo nó conectado à antiga origem. Nas duas buscas, deve-se guardar os nós já visitados para não passar por eles novamente (evitar ciclos) (CORMEN, 2009). As figuras 6 e 7 ilustram os dois métodos de busca enquanto que os Algoritmos 2 e 3 tratam da implementação dos algoritmos, levando em conta a definição do número de passos para busca.

Dada uma demanda, os algoritmos percorrem a rede e analisam cada nó visitado em busca de possíveis habilidades para atender a parte da tarefa. Sempre é escolhido o primeiro nó com habilidade e capacidade para atender a um item presente na tarefa que ainda não foi satisfeito, desde que o acréscimo deste item não extrapole o limite de tempo definido como restrição – política *first-fit*.

Algoritmo 2 – Busca em profundidade

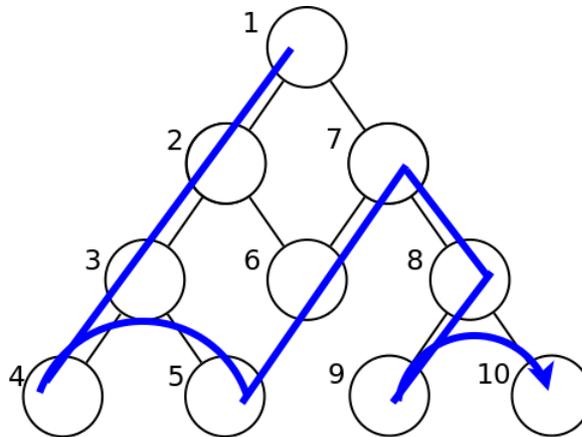
```

funcao busca em profundidade(no, tarefa, numero de passos)
  para cada item da tarefa faça
    se no possui habilidade disponivel então
      define habilidade para atender ao item de tarefa
      remove item da tarefa
      acrescenta tempo
      acrescenta custo da habilidade e transporte até o nó de solicitação
    fim se
  fim para
  se a tarefa não está completa e ainda não atingiu número máximo de passos então
    para cada vizinho conectado ao no atual como fornecedor que não foi analisado faça
      busca em profundidade(vizinho, tarefa, numero de passos)
    fim para
  fim se
  retorna tarefa
fim funcao

```

Fonte: elaborado pelo autor (2019)

Figura 6 – Busca em profundidade



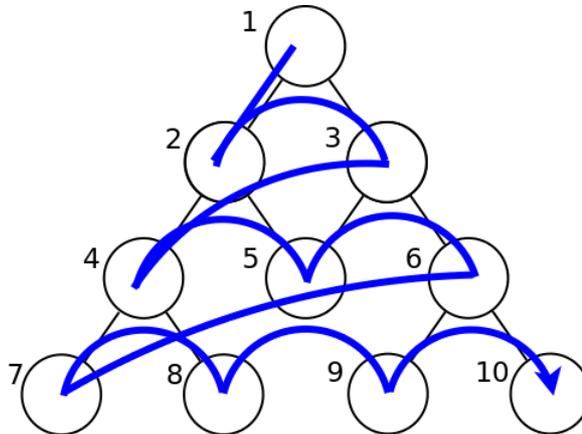
Fonte: elaborado pelo autor (2019)

Algoritmo 3 – Busca em largura

funcao busca em largura(*no*, *tarefa*, *numero de passos*)
 remove *no* da *lista de próximos*
 indica que *no* foi analisado
 se ainda não atingiu número máximo de passos **então**
 para cada *vizinho* que pode fornecer para o *no* e que não foi analisado **faça**
 adiciona *vizinho* na *lista de próximos*
fim para
fim se
para cada item da tarefa **faça**
 se não possui habilidade disponível **então**
 define habilidade para atender ao item de tarefa
 remove item da tarefa
 acrescenta tempo
 acrescenta custo da habilidade e transporte até o nó de solicitação
fim se
fim para
 se a tarefa não está completa **então**
 enquanto há algum *nó próximo* na *lista de próximos* **faça**
 busca em largura(*nó próximo*, *tarefa*, *numero de passos*)
 fim enquanto
fim se
 retorna tarefa
fim funcao

Fonte: elaborado pelo autor (2019)

Figura 7 – Busca em largura



Fonte: elaborado pelo autor (2019)

4.3 UMA ABORDAGEM BASEADA EM ALGORITMO GENÉTICO PARA O PROBLEMA DA DEFINIÇÃO DE NÓS PARA A EXECUÇÃO DA TAREFA

Redes complexas são formadas por milhares ou milhões de vértices e conexões. Por isso, torna-se inviável uma abordagem exata para o problema, que encontraria a solução ótima, devido ao altíssimo custo computacional envolvido; pode-se, então, utilizar uma abordagem heurística para buscar uma solução aproximada ao ótimo (BOFFEY, 1982).

Um algoritmo heurístico foi construído para prover uma solução ao problema de definição de nós para atender a uma demanda dentro de uma rede de produção da Indústria 4.0. O algoritmo foi dividido em duas etapas: na primeira, são selecionados os candidatos a fornecedores para a execução da tarefa, e, na segunda, são definidos os nós que fornecerão cada um dos itens da tarefa. Dessa forma, o algoritmo busca decompor em duas partes, mais simples de serem solucionadas, e diminuir a quantidade de possíveis fornecedores, reduzindo o espaço de soluções possíveis. Essas abordagens para uma heurística estão entre as categorias definidas por Silver (2004).

Para a definição dos candidatos a fornecedores, utilizou-se uma adaptação do algoritmo de Dijkstra (DIJKSTRA, 1959) para a definição do menor caminho entre dois nós de um grafo. O algoritmo de Dijkstra para a definição do menor caminho entre dois nós (P e Q) de um grafo parte da divisão dos nós em 3 conjuntos: A , contendo os nós para os quais sabe-se qual é o menor caminho a partir do nó origem P ; B , com os nós que estão conectados a pelo menos um nó de A , que não estejam em A ; e C com os demais nós do grafo. As arestas

também são divididas em 3 conjuntos, definidos por Dijkstra (1959) como: *I*, arestas que formam os caminhos mínimos de P para os nós em A; *II*, arestas que levam aos nós do conjunto B (uma única aresta para cada nó de B), de onde sairá a próxima aresta para I; e *III* com as demais arestas do grafo (DIJKSTRA, 1959).

O funcionamento do algoritmo consiste em dois passos (DIJKSTRA, 1959):

1. Passo 1: Consideram-se todas as arestas *r* que conectam o nó mais recentemente transferido para o conjunto A com os demais nós *R* dos conjuntos B e C. Para os nós *R* quem pertencem a B, deve-se investigar se o uso da aresta *r* leva a um caminho mais curto saindo de P a R, em relação ao caminho já encontrado no conjunto II. Caso leve a um caminho mais curto, então a aresta *r* substitui a aresta já adicionada ao conjunto II; caso contrário, ela é rejeitada. Se o nó *R* está no conjunto C, ele deve ser adicionado ao conjunto B e a aresta *r* no conjunto II.
2. Passo 2: Cada nó em B só pode estar conectado a P por um caminho se forem consideradas apenas as arestas do conjunto I com uma aresta de II. Dessa forma, cada nó no conjunto B possui uma distância a partir do nó P. Deve-se transferir o nó de B que possui a menor distância até P para o conjunto A, com a aresta correspondente sendo transferida do conjunto II para o conjunto I. A partir daí, volta-se ao Passo I até que o nó Q (destino) seja inserido em A.

Para este trabalho, o algoritmo de Dijkstra foi adaptado, uma vez que não se deseja encontrar a menor distância entre dois nós do grafo: o que se quer, neste caso, é delimitar um conjunto de nós (definido como *árvore de fornecedores*), mais próximos ao nó que demanda a tarefa, nos quais se espera encontrar candidatos aptos a resolver a tarefa em questão. Portanto, a metodologia de análise proposta em Dijkstra (1959) foi utilizada para varrer o grafo, a partir do nó que demanda a tarefa, até que determinado número de outros nós seja analisado. Utilizando o algoritmo baseado em Dijkstra, o caminho percorrido sempre será o menor entre os nós do grafo. Diferentemente do original, o algoritmo proposto para a geração da árvore de fornecedores, sumarizado em Algoritmo 4, termina quando um número pré-determinado de nós é analisado.

Algoritmo 4 – Construção da árvore de fornecedores

no_atual ← no que demanda a tarefa

adiciona *no_atual* com custo 0 na *lista_nos_alcançados*

enquanto número máximo de nós analisados não é atingido **faça**

no_atual ← nó com menor custo em *lista alcançados* que ainda não foi analisado

para cada *no_fornecedor* de *no_atual* **faça**

$custo_fornecedor \leftarrow custo_no_atual + custo_aresta(no_atual, no_fornecedor)$
 adiciona no_atual com custo $custo_fornecedor$ na lista $nos_alcançados$

fim para cada
fim enquanto

Fonte: elaborado pelo autor (2019)

Após a definição da árvore de fornecedores (conjunto de nós para a execução da tarefa), passa-se a determinar os nós responsáveis por cada um dos itens dessa execução, utilizando uma abordagem baseada na metaheurística Algoritmo Genético. A tarefa é um conjunto de letras que representam itens de produção/consumo (ex: ACEIJVT). Para cada item da tarefa, um nó fornecedor deve ser definido.

A representação da solução em um algoritmo genético é chamada de cromossomo. A representação do cromossomo foi definida como um vetor de nós, em que cada posição do vetor indica o nó que fornecerá o respectivo item da tarefa. A Figura 8 ilustra a representação de uma solução para a tarefa ACEIJVT.

Figura 8 – Exemplo de tarefa e solução como um cromossomo

A	C	E	I	J	V	T
8	30	210	8	36	54	101

Fonte: elaborado pelo autor (2019)

A partir da primeira parte do algoritmo (adaptado de Dijkstra), tem-se os custos das distâncias entre todos os possíveis fornecedores até o nó que solicita a tarefa. Dessa forma, a *função objetivo* pode ser definida como o somatório dos pesos dos caminhos entre os nós fornecedores até o nó solicitante da tarefa, e os custos para a produção de cada item. Caso um mesmo nó forneça mais de um item na solução, o custo do caminho é somado para cada item fornecido, pois não foi utilizada informação de capacidade de transporte entre itens das arestas no modelo proposto.

O cálculo de aptidão foi definido pela seguinte função (6):

$$\text{aptidão} = \frac{1}{(\text{custo} + \text{tempo} * \text{penalidade} + (\text{tempo} - \text{tempo limite}) * \text{penalidade excedente})} \quad (6)$$

Trata-se, portanto, do inverso do somatório entre: (i) o custo total da tarefa; (ii) o produto entre o tempo decorrido e uma penalidade definida na execução do algoritmo; e (iii) o produto entre o tempo que excede o limite definido e outro parâmetro de penalidade. As penalidades são parâmetros definidos ao executar o algoritmo; uma vez que soluções cujo tempo de execução é maior do que o limite não são removidas no processo, o uso de penalidades altas reduz as chances de essas soluções permanecerem nas populações subsequentes. Processos podem ser modelados de forma global, com centralização de dados, e soluções não ótimas podem ser permitidas em uma unidade para resolver gargalos em outras (BRETTEL et al. 2014). Por isso, optou-se por manter soluções que extrapolam o limite de tempo, deixando para o usuário do simulador a tarefa de tomar a decisão de utilizar, ou não, uma solução dessas quando julgar pertinente.

O tamanho da população é informado como parâmetro para a execução do algoritmo. A população inicial é criada de forma aleatória; para cada gene de um cromossomo da população inicial, escolhe-se o n -ésimo nó capaz de atender aquele item da tarefa, sendo n o número do cromossomo. Caso a população seja maior que o número de nós capazes de executar um item da tarefa, volta-se para o início da lista de possíveis fornecedores para o item, cada vez que se chega ao fim desta lista. Para cada item (letra) da tarefa, é armazenada uma lista com os nós que possuem o respectivo item entre suas habilidades. Essa lista é gerada no mesmo momento do processamento da árvore de fornecedores.

Para a reprodução, foi escolhido o método de *crossover* uniforme, trabalhando sempre com pares de pais que geram pares de filhos. O funcionamento do *crossover* uniforme, tendo por base Lacerda e Carvalho (1999), consiste em gerar um vetor aleatório de bits (valores 0 ou 1) para cada par de pais, com tamanho igual ao cromossomo de solução (mesmo número de itens da tarefa). Cada bit do vetor é analisado, posição a posição: caso seja o valor 0, o filho 1 terá o valor da solução (cromossomo) do pai 1, e o filho 2 terá a solução do pai 2. Caso o valor do bit seja 1, o filho 1 terá o valor da solução do pai 2, e o filho 2 terá a solução do pai 1. A figura 9 ilustra um exemplo de *crossover* uniforme. A seleção de nós para a reprodução foi feita utilizando o método de *torneio*: são escolhidos 3 cromossomos (dentro todos os disponíveis na população atual), de forma aleatória, cada um com a mesma probabilidade de escolha, e avaliados com relação à função aptidão. O cromossomo de melhor aptidão é escolhido para integrar a reprodução.

Figura 9 – Exemplo de *crossover* uniforme

Tarefa	A	C	E	I	J	V	T
Vetor de bits	1	0	1	0	0	1	0
Pai 1	8	30	210	8	36	54	101
Pai 2	115	296	233	230	42	148	55
Filho 1	115	30	233	8	36	148	101
Filho 2	8	296	210	230	42	54	55

Fonte: elaborado pelo autor (2019)

O algoritmo construído também faz uso do elitismo. Deve-se definir, como parâmetro, a percentagem das melhores soluções que devem ser alocadas na população intermediária, para que sejam avaliadas na seleção. Se o valor for 0, não há elitismo. Outros parâmetros que devem ser definidos para a execução do algoritmo são: o tamanho da população, a quantidade de iterações que haverá, e a taxa de mutação (percentual). O processo de mutação consiste na alteração do fornecedor de um item por outro, escolhido aleatoriamente, dentro da lista de nós que possuem habilidade para atender aquele item. O algoritmo genético construído está explicitado em Algoritmo 5.

Algoritmo 5 – Algoritmo genético para definição de nós para executar uma tarefa

```

população ← gera população inicial
enquanto não atinge limite de iterações faça
  se há elitismo então
    adiciona(população intermediária, melhores (população, TAXA ELITISMO))
  fim se
  enquanto i < tamanho(população) faça
    pai1 ← seleciona cromossomo por torneio (população)
    pai2 ← seleciona cromossomo por torneio (população)
    se realizar crossover(TAXA CROSSOVER) então
      vetor de bits ← gera vetor de bits aleatório
      filho1, filho2 ← crossover uniforme (pai1, pai2, vetor de bits)
      filho1 ← mutação(filho1, TAXA MUTAÇÃO)
      filho2 ← mutação(filho2, TAXA MUTAÇÃO)
      adiciona (população intermediária, filho1, filho2)
    senão
      adiciona (população intermediária, pai1, pai2)
    fim se
  população ← melhores (população intermediária, TAMANHO POPULAÇÃO)
fim enquanto
fim enquanto

```

4.4 DESENVOLVIMENTO DO SIMULADOR

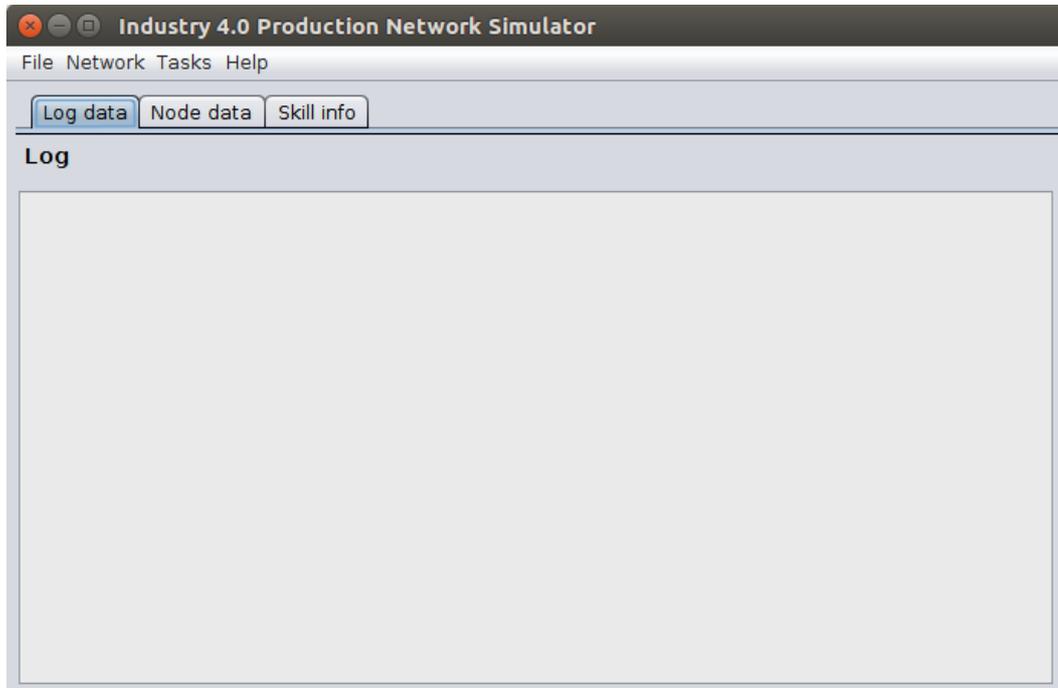
O simulador foi construído com base no conjunto de requisitos definidos, e nos métodos de execução das tarefas elaborados e descritos anteriormente. Utilizou-se a linguagem de programação Java para sua implementação, devido à familiaridade do pesquisador com essa tecnologia, a sua portabilidade entre sistemas operacionais, a facilidade em manipulação de arquivos e suporte a interface gráfica. A interface do *software* simulador foi feita em língua inglesa para facilitar o uso de pessoas de outros países.

Para armazenar os dados da rede em formato de arquivo, utilizou-se o formato JSON (*JavaScript Object Notation*), um formato leve para troca de dados, que tanto é facilmente lido e escrito por humanos, quanto é facilmente gerado e analisado por máquinas (EL-AZIZ; KANNAN, 2014). Trata-se de um formato de arquivo-texto, independente de linguagem de programação, que foi escolhido por ser um formato de troca de informações. O uso do formato JSON para representação da rede permite uma fácil geração de arquivos contendo a estrutura da rede. Dessa forma, espera-se que o simulador possa estar integrado à cadeia produtiva, em fábricas inteligentes, nas quais sensores obtêm informações dos nós e as lançam neste formato de arquivo. No arquivo JSON, além dos dados da estrutura da rede (nós, arestas e pesos), são armazenadas as informações de habilidades de cada nó, com seus valores de custo, tempo e capacidade (quantidade). No Apêndice A está um exemplo de parte do código de uma rede gerado pelo simulador, no formato JSON.

A interface do simulador é composta por um menu, com opções para a realização das atividades (Nova rede, editar rede, abrir arquivo, salvar arquivo, salvar arquivo com outro nome, salvar *log* de operações, sair; ler arquivo Pajek, ver distribuição de grau da rede, criar pesos aleatórios para as arestas, criar habilidades aleatórias, remover nós, alterar habilidades; criar uma tarefa – busca em profundidade ou largura, criar tarefas aleatórias – busca em profundidade ou largura, criar tarefa – algoritmo genético, criar tarefas aleatórias – algoritmo genético; e ajuda) e por três abas: na primeira, exibe-se um *log* com a saída de cada operação realizada no simulador; na segunda aba é mostrada a lista dos nós da rede, permitindo que sejam visualizadas informações de cada nó, como suas habilidades e arestas; e na terceira aba são ilustradas informações das habilidades, com o número de nós que possui cada uma delas, suas médias de custo e tempo.

A Figura 10 ilustra a tela inicial do simulador. Nela, pode-se verificar as três abas, e o menu principal, acima. Neste caso, a aba de *logs* está ativa.

Figura 10 – Interface inicial do simulador com rede carregada



Fonte: elaborado pelo autor (2019)

A Figura 11 traz a tela inicial do sistema, com a aba de dados dos nós. Para um nó selecionado, são exibidos sua descrição (*label*), seu tipo (*type*), quando definido, suas habilidades (*skills*) e suas arestas (*edges*).

Figura 11 – Visualização de detalhes de um nó

Skills

Skill	Capacity	Cost	Time
C	9999	11	1
S	9999	22	7
T	9999	3	10
F	9999	24	4
Z	9999	6	3

Edges

Origin	Destiny	Weight
62878	5	47
5	15438	24
5	14126	1
41398	5	39
33830	5	17
5	54784	41
52437	5	10
5	36308	27
28424	5	48
10822	5	24
5	62809	49
26110	5	18

Fonte: elaborado pelo autor (2019)

A Figura 12 mostra a aba de informações das habilidades dos nós da rede, ainda na tela inicial do simulador; para cada habilidade (letra do alfabeto), tem-se o número de nós com aquela habilidade, sua média de custo, média de capacidade e média de tempo. Ao selecionar uma habilidade, mostra-se uma lista com todos os nós que a possuem.

Figura 12 – Informações de habilidades

The screenshot shows the 'Industry 4.0 Production Network Simulator' window. It has a menu bar with 'File', 'Network', 'Tasks', and 'Help'. Below the menu bar are three tabs: 'Log data', 'Node data', and 'Skill info', with 'Skill info' being the active tab. The main content area is divided into two sections: 'Select skill:' and 'Nodes:'. The 'Select skill:' section contains a table with columns for Skill, Number of no..., Average Cost, Average Cap..., Average Capa..., and Average Time. The 'Nodes:' section contains a table with columns for Node id and Node label.

Skill	Number of no...	Average Cost	Average Cap...	Average Capa...	Average Time
A	23790	15,456	9.999	237876210	5,503
B	23530	15,524	9.999	235276470	5,511
C	23595	15,516	9.999	235926405	5,527
D	23770	15,505	9.999	237676230	5,503
E	24069	15,551	9.999	240665931	5,511
F	23800	15,373	9.999	237976200	5,5
G	23904	15,541	9.999	239016096	5,506
H	24095	15,609	9.999	240925905	5,465
I	24083	15,542	9.999	240805917	5,479
J	23911	15,475	9.999	239086089	5,511
K	23974	15,535	9.999	239716026	5,497
L	23839	15,494	9.999	238366161	5,512
M	23842	15,424	9.999	238396158	5,492
N	23592	15,445	9.999	235896408	5,488
O	23917	15,464	9.999	239146083	5,514
P	24026	15,545	9.999	240235974	5,527
Q	24165	15,496	9.999	241625835	5,521
R	23750	15,381	9.999	237476250	5,477
S	23873	15,472	9.999	238706127	5,493
T	23762	15,514	9.999	237606227	5,500

Node id	Node label
97756	v97756
32222	v32222
32223	v32223
64991	v64991
32225	v32225
32233	v32233
65004	v65004
65003	v65003
32237	v32237
65006	v65006
32238	v32238
97776	v97776
32240	v32240

Fonte: elaborado pelo autor (2019)

A Figura 13 mostra a tela com a distribuição de grau dos nós da rede. A primeira tabela traz o número de nós com cada grau, e a segunda traz a porcentagem em relação ao total de nós da rede.

Figura 13 – Distribuição de grau dos nós da rede

The screenshot shows a window titled "Degree Distribution" with two data tables. The first table, "Number of nodes", lists degrees from 0 to 13 and the corresponding number of nodes. The second table, "Degree distribution", lists the same degrees and their relative distribution values.

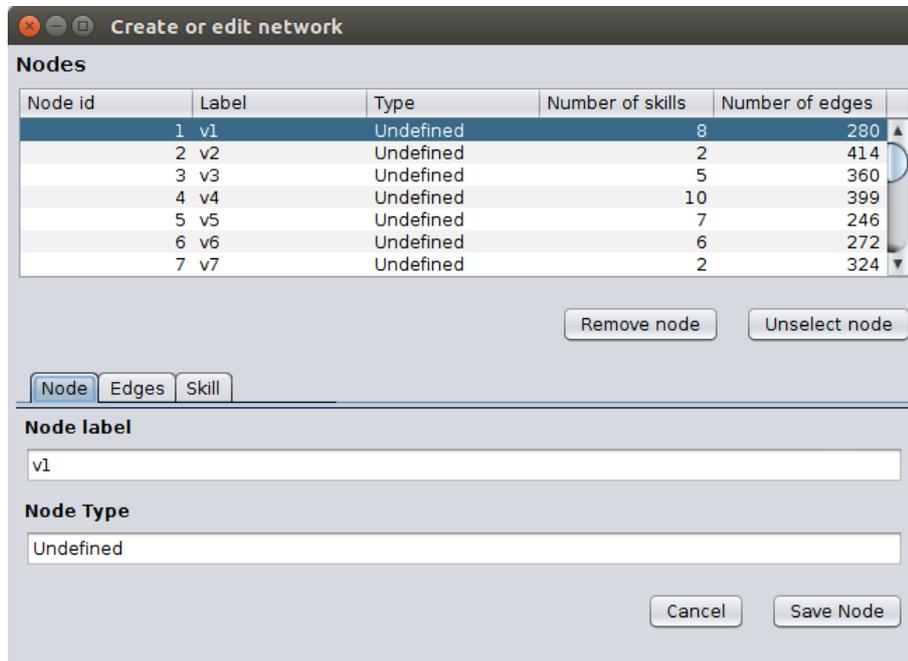
Number of nodes	
Degree	Number of nodes
0	4
1	39
2	178
3	472
4	1196
5	2442
6	4113
7	5695
8	7227
9	7967
10	7982
11	7239
12	5995
13	4649

Degree distribution	
Degree	Distribution
0	0,006
1	0,061
2	0,278
3	0,737
4	1,869
5	3,816
6	6,427
7	8,898
8	11,292
9	12,448
10	12,472
11	11,311
12	9,367
13	7,264

Fonte: elaborado pelo autor (2019)

A Figura 14 traz a tela de edição da rede. Através dela, é possível alterar o nome (*label*) do nó, seu tipo; alterar, adicionar ou remover suas arestas e habilidades.

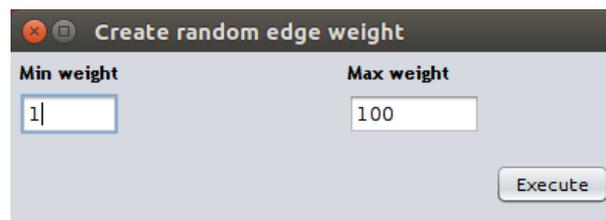
Figura 14 –Edição da rede



Fonte: elaborado pelo autor (2019)

A Figura 15 traz a tela de definição de pesos aleatórios para as arestas. Devem ser definidos os valores mínimo e máximo para os pesos.

Figura 15 – Criação de pesos aleatórios para arestas



Fonte: elaborado pelo autor (2019)

A Figura 16 traz a tela de criação de habilidades aleatórias. Deve-se definir o grau mínimo dos nós que serão considerados ao gerar as habilidades (se o valor for zero, serão geradas habilidades para todos os nós da rede); o número mínimo de habilidades que cada nó terá; o número máximo de habilidades que cada nó terá; os tamanhos mínimo e máximo de capacidade de cada item de habilidade em cada nó; os valores mínimo e máximo de custo de cada item de habilidade em cada nó; e os valores mínimo e máximo de tempo de cada item de habilidade em cada nó.

Figura 16 – Criação de habilidades aleatórias



Create Rand Skills

Min Degree
0

Min Skills
0

Max Skills
0

Min Capacity
0

Max Capacity
0

Min Cost
0

Max Cost
0

Min Time
0

Max Time
0

Create

Fonte: elaborado pelo autor (2019)

A Figura 17 traz a tela de remoção de nós. É possível remover os nós (i) por seus identificadores; (ii) com grau maior que o especificado; (iii) remover um número aleatório de nós; e (iv) remover nós por tipo. Através da remoção de nós, pode-se alterar a estrutura da rede, simulando, por exemplo, falhas.

Figura 17 – Remoção de nós

The image shows a software dialog box titled "Remove nodes". It has a standard window title bar with a close button (red X) and a maximize button. The dialog is divided into four sections, each with a text input field and a corresponding button:

- ID nodes to remove (separate by comma):** The input field contains the number "0". The button is labeled "Remove nodes by id".
- Remove nodes with degree greater than:** The input field contains the number "0". The button is labeled "Remove nodes by degree".
- Remove "N" random nodes:** The input field contains the number "0". The button is labeled "Remove random nodes".
- Remove nodes by type:** A dropdown menu is set to "Undefined". The button is labeled "Remove nodes by type".

Fonte: elaborado pelo autor (2019)

A Figura 18 traz a tela de alterações de habilidades. Para cada item de habilidade, são exibidas as seguintes informações: número de nós que contêm aquela habilidade; média de custo do item de habilidade; média de capacidade do item de habilidade; total de capacidade disponível para aquele item de habilidade; e média de tempo do item de habilidade. Para cada item de habilidade, é possível alterar os valores de capacidade, custo e tempo de duas formas: definindo um valor padrão para todos aqueles itens de habilidade em todos os nós; ou ainda aumentando (ou diminuindo) os valores em determinada porcentagem definida.

Figura 18 – Tela de alteração de habilidades

Update Skills

Select the skill

Skill	Number o...	Average C...	Average Capacity	Total ...	Avera...
A	23790	15,456	9.999	23787...	5,503
B	23530	15,524	9.999	23527...	5,511
C	23595	15,516	9.999	23592...	5,527
D	23770	15,505	9.999	23767...	5,503
E	24069	15,551	9.999	24066...	5,511
F	23800	15,373	9.999	23797...	5,5
G	23904	15,541	9.999	23901...	5,506
H	24095	15,609	9.999	24092...	5,465
I	24083	15,542	9.999	24080...	5,479
J	23911	15,475	9.999	23908...	5,511
K	23974	15,535	9.999	23971...	5,497
L	23839	15,494	9.999	23836...	5,512
M	23842	15,424	9.999	23839...	5,492
N	23592	15,445	9.999	23589...	5,488
O	23917	15,464	9.999	23914...	5,514
P	24026	15,545	9.999	24023...	5,527
Q	24165	15,496	9.999	24162...	5,521
R	23750	15,381	9.999	23747...	5,477
S	23873	15,472	9.999	23870...	5,493
T	23763	15,514	9.999	23760...	5,509
U	23824	15,562	9.999	23821...	5,512
V	23716	15,479	9.999	23713...	5,503
W	24053	15,484	9.999	24050...	5,509
X	23747	15,357	9.999	23744...	5,477
Y	23715	15,489	9.999	23712...	5,484
Z	23964	15,538	9.999	23961...	5,495

Select the field to change: Capacity

Node type: -- ALL --

Set default value: 2

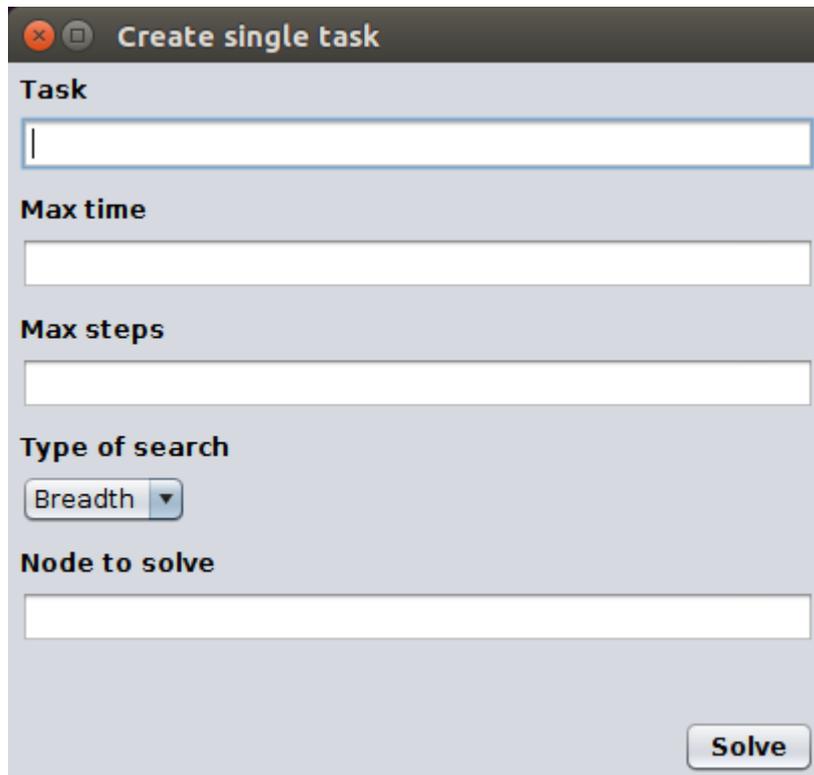
Increase or decrease X %: 0

Fonte: elaborado pelo autor (2019)

A Figura 19 traz a tela de criação de uma tarefa para execução com os algoritmos de busca em largura ou busca em profundidade. Como parâmetros, deve-se informar a tarefa (um conjunto de letras); o limite máximo de tempo; o limite máximo de passos que serão realizados na tarefa; o tipo de busca (largura ou profundidade); e o número de identificação do nó que está demandando a tarefa. O resultado da execução da tarefa traz o custo para a sua

execução, o tempo gasto para a execução, os nós que fornecerão cada item da tarefa e os nós que foram percorridos durante a busca.

Figura 19 – Tela de execução de uma tarefa por busca em largura ou profundidade



The image shows a window titled "Create single task" with a standard OS title bar (close, maximize, and minimize buttons). The window contains the following fields and controls:

- Task:** A text input field.
- Max time:** A text input field.
- Max steps:** A text input field.
- Type of search:** A dropdown menu currently showing "Breadth".
- Node to solve:** A text input field.
- Solve:** A button located at the bottom right of the window.

Fonte: elaborado pelo autor (2019)

A Figura 20 é uma captura da tela de execução de tarefas aleatórias, utilizando o algoritmo de busca em largura ou profundidade. Deve-se informar o número de tarefas aleatórias que serão geradas e executadas; o número de identificação do nó para executar as tarefas, ou zero caso o nó seja escolhido aleatoriamente; os tamanhos mínimo e máximo de cada tarefa gerada (em quantidade de letras); o limite de tempo das tarefas; o número máximo de passos a serem percorridos; o tipo de busca a ser realizado na tarefa (profundidade ou largura); e o número de vezes que a execução deve ser repetida. Como resultados da execução, são mostradas as tarefas geradas com seu custo calculado, os nós responsáveis pelos itens da tarefa, o tempo gasto na execução da tarefa e os nós visitados. Além disso, é exibida uma média das execuções, com as informações de custo médio, tempo médio e número médio de tarefas completadas em sua totalidade.

Figura 20 – Tela de execução de tarefas aleatórias com os algoritmos de busca de largura ou profundidade

The image shows a software window titled "Random tasks" with a standard Windows-style title bar (close, maximize, minimize buttons). The window contains the following controls:

- Number of tasks:** An empty text input field.
- Node id - 0 if random:** A text input field containing the number "0".
- Min size of task:** An empty text input field.
- Max size of task:** An empty text input field.
- Max time:** An empty text input field.
- Max steps:** An empty text input field.
- Type of search:** A dropdown menu with "Breadth" selected.
- Repeat N times:** A text input field containing the number "1".
- Solve:** A button located at the bottom right of the window.

Fonte: elaborado pelo autor (2019)

A Figura 21 traz a tela de execução de uma tarefa utilizando o algoritmo genético desenvolvido. Deve-se informar a tarefa (conjunto de letras); o limite de tempo; o número de nós que serão analisados para a geração da árvore de fornecedores, no algoritmo baseado em Dijkstra; o número de identificação do nó que está demandando a tarefa; o tamanho da população do algoritmo genético; o valor da penalidade de tempo; a taxa de elitismo; a probabilidade de mutação; o número de iterações do algoritmo genético; a taxa de reprodução (*crossover*); e a penalidade para o tempo que excede o tempo limite definido. Como resultado, é exibida a população final, ordenada pela aptidão em ordem crescente. Para cada cromossomo da população final, exibe-se o custo, tempo calculado, valor de aptidão, e o nó responsável por cada item da tarefa.

Figura 21 – Tela de execução de uma tarefa utilizando o algoritmo genético

Task - like ABCDEF

Limit time	Max nodes analyzed - suppliers tree
<input type="text"/>	<input type="text"/>
Node from (node id)	Population size
<input type="text"/>	<input type="text"/>

Penalty time

Elitism Tax (0 to 1 - ex 0.04)

Mutation Probability (0 to 100 - ex 2)

Max iterations

Crossover Tax (0 to 100 - ex 75)

Penalty time excedent

Execute

Fonte: elaborado pelo autor (2019)

A Figura 22 traz a tela utilizada para a execução de tarefas aleatórias, a partir do algoritmo genético proposto. Deve-se informar, como parâmetros: o número de tarefas; os limites mínimo e máximo para o tamanho das tarefas que serão geradas; a penalidade de tempo geral; a probabilidade de mutação; o nó que demandará a tarefa (ou zero, se for selecionado um nó aleatório para cada tarefa); o número de iterações do algoritmo genético; o

limite de tempo da tarefa; o número de nós analisados na formação da árvore de fornecedores; o tamanho da população de cromossomos; a taxa de reprodução (*crossover*); a taxa de elitismo; e a penalidade para o tempo que excede o limite definido. Como resultado, são exibidos os valores médios de tarefas executadas, média de tarefas executadas dentro do limite de tempo, média de tempo da execução das tarefas e média de custo. Além disso, exibem-se as tarefas geradas em cada repetição, juntamente ao cromossomo de melhor aptidão obtido com a execução do algoritmo. Para este cromossomo, são exibidas as seguintes informações: custo da execução da tarefa, tempo de execução da tarefa, valor da aptidão e, para cada item da tarefa, os respectivos nós definidos para a sua execução.

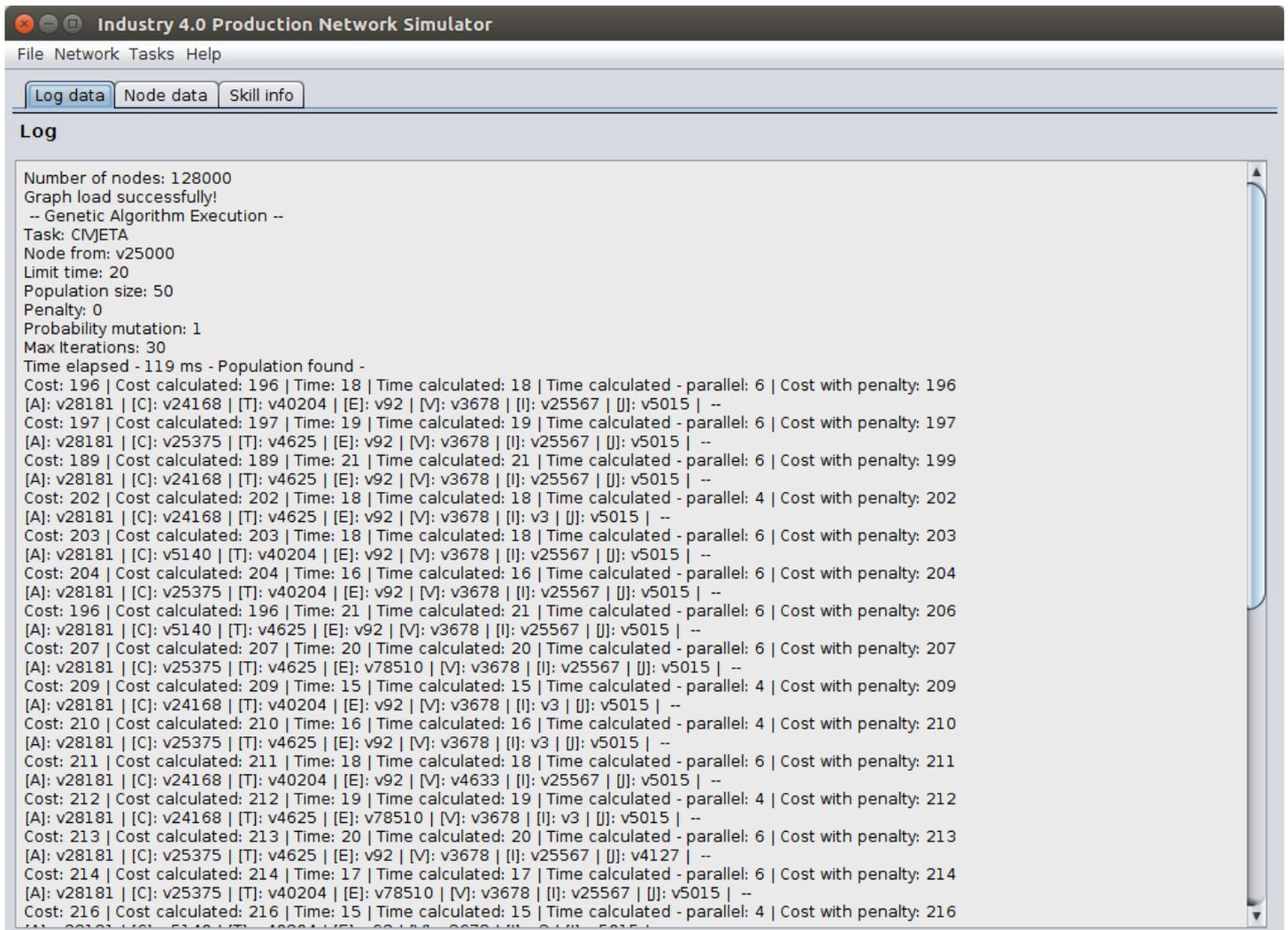
Figura 22 – Tela de execução de tarefas aleatórias com o algoritmo genético

Number of tasks	<input type="text"/>	Time penalty	<input type="text"/>
Min Size of task	<input type="text"/>	Max Size of task	<input type="text"/>
Node from (0 if random)	<input type="text" value="0"/>	Max iteration	<input type="text"/>
Limit time	<input type="text"/>	Max nodes analyzed - suppliers tree	<input type="text"/>
Population size	<input type="text"/>	Crossover tax (0 to 100)	<input type="text"/>
Elitism tax (0 to 1)	<input type="text"/>	Time penalty excedent	<input type="text"/>
Probability mutation (0 to 100)	<input type="text"/>	Repeat N times	<input type="text" value="1"/>

Fonte: elaborado pelo autor (2019)

Os resultados das execuções de tarefas no simulador são expostos na tela inicial do sistema, na aba de *logs*. A Figura 23 traz um exemplo do resultado da execução de uma tarefa de algoritmo genético.

Figura 23 – Resultado da execução de uma tarefa utilizando o Algoritmo Genético



Fonte: elaborado pelo autor (2019)

4.5 VALIDAÇÃO DOS ALGORITMOS PROPOSTOS

A oitava etapa da metodologia da DSR consiste na avaliação do artefato. Testes foram realizados para verificar o correto funcionamento dos algoritmos propostos e implementados no simulador. Para tanto, uma rede aleatória livre de escala com 128 mil nós e grau médio 5 foi construída, utilizando o Pajek, e importada para o simulador. No simulador, foram

definidos pesos aleatórios para suas arestas, variando os valores entre 1 e 50. Para cada nó que possuía pelo menos uma conexão (grau maior ou igual a 1), foi criado um conjunto de habilidades, com tamanho entre 1 e 10 itens. O tempo definido para cada habilidade foi gerado aleatoriamente, com valores de 1 a 10. Os custos foram gerados entre 1 a 30.

Em uma primeira etapa, foi realizado um teste de mesa, isto é, o funcionamento dos algoritmos foi verificado manualmente (MORGADO; ANDRÊO, 2013), a partir da execução de uma tarefa, criada aleatoriamente e atribuída a um nó escolhido ao acaso da rede criada anteriormente. A tarefa construída foi a seguinte: BGIJKMORSUTXZ, atribuída ao nó de índice 51266. O tempo máximo para a execução da tarefa foi 75, e o número máximo de passos foi indicado como 5 para os algoritmos baseados em busca em largura e profundidade construídos. Sua execução teve um custo no valor de 273, em um tempo de 74, para a execução do algoritmo de busca em largura. Para o algoritmo de busca em profundidade, o custo foi de 890 unidades e o tempo 71. Foram analisados os caminhos percorridos e os nós escolhidos para a execução da tarefa, manualmente, e os valores obtidos foram os esperados para os dois algoritmos

Também foi feita uma avaliação, através de um teste de mesa, da execução do algoritmo genético. A tarefa executada foi a mesma definida anteriormente, com o limite de tempo 75. O número máximo de nós analisados para montar a árvore de fornecedores foi 2; o tamanho da população para o algoritmo genético foi definido como 10 cromossomos. A penalidade geral de tempo foi definida como 0. A taxa de elitismo foi definida como 10%. A probabilidade de mutação foi definida como 1%. O número de iterações do algoritmo foi definido como 5. A taxa de *crossover* foi definida como 80%. Por fim, a penalidade para o tempo excedente foi definida como 10. Obteve-se, nesta execução, uma solução com o custo 296 e tempo 73.

Os três algoritmos, dessa forma, foram validados, verificando-se o correto funcionamento deles.

4.6 COMPARAÇÃO COM O SOLVER LINGO E ALGORITMO ÓTIMO

Problemas matemáticos que requerem um grande número de cálculos podem ser resolvidos através de *softwares* computacionais, como o LINGO, um programa que permite ao usuário inserir a formulação de um modelo e resolvê-lo, podendo, então, realizar modificações e novos testes (LINDO Systems, 2006). Além do teste de mesa, buscou-se utilizar o LINGO como parâmetro para testar o correto funcionamento dos algoritmos aqui

desenvolvidos. A versão utilizada foi a 18.0.44. Foi executada, no *software* LINGO, uma tarefa com uma rede de 10 nós e grau médio 3.

A rede foi construída com o *software* Pajek (rede livre de escala, direcionada, com tamanho de 10 nós e grau médio 3); foram atribuídos valores aleatórios para o peso das arestas (entre 1 e 5) e criadas habilidades para cada nó (mínimo de 1 item por nó e máximo de 10 itens; custo e tempo com valores entre 1 e 10) utilizando o simulador aqui desenvolvido. O tempo limite para execução da tarefa foi definido como 20. A tarefa foi criada aleatoriamente como DFQ, e atribuída ao nó de número 9. Após, foi construído um código no *software* LINGO, disponível no Apêndice B, para a solução da tarefa.

A solução ótima para o problema, obtida no *software* LINGO, foi a seguinte: para o item D, foi escolhido o nó 1; para o item F, foi escolhido o nó 4, e para o item Q, o nó escolhido foi o de número 8. O custo total calculado foi de 28, e o tempo ficou em 16. O mesmo problema foi executado no simulador, utilizando os algoritmos de busca em largura, busca em profundidade, e o algoritmo genético. Para o algoritmo genético, os parâmetros utilizados foram os seguintes: 10 nós analisados para formar a árvore de fornecedores, 10 cromossomos na população, sem penalidade de tempo, taxa de elitismo 10%, probabilidade de mutação de 1%, número máximo de iterações como 10, taxa de *crossover* de 80%, e penalidade excedente como 5. Para os algoritmos baseados em busca em largura e profundidade, utilizou-se como máximo de passos o valor 5.

Nos três casos de execução da tarefa, as soluções apresentadas foram as mesmas geradas pelo LINGO, isto é, a solução ótima. Isso pode ter ocorrido devido ao pequeno tamanho da rede. Dessa forma, também pode-se verificar o correto funcionamento dos três algoritmos propostos. Ressalta-se que redes com 100 nós ou mais não puderam ser executadas no *software* LINGO, devido às limitações do *software*.

Além da comparação com o LINGO, foi feita uma comparação com o algoritmo ótimo, isto é, aquele que verifica todas as soluções possíveis e utiliza a melhor. Para tanto, foi utilizada uma rede aleatória criada com característica de livre de escala no *software* Pajek, contendo 8000 nós e grau médio 5. A rede foi carregada no simulador e, nele, foram definidos pesos aleatórios para as arestas com valores entre 1 e 50; cada nó com grau médio maior ou igual a 1 recebem um conjunto entre 1 a 10 habilidades; os custos dos itens de habilidade foram definidos aleatoriamente entre 1 e 30; e os tempos de cada item de habilidade foram definidos aleatoriamente entre 1 e 10. Foram construídas 5 tarefas aleatórias com 3 itens cada, cada uma atribuída a um nó escolhido ao acaso. O limite de tempo definido foi 12. As tarefas foram executadas utilizando os três algoritmos não-ótimos propostos neste trabalho, e também

foram executadas por um algoritmo ótimo, desenvolvido para este teste – também utilizando a linguagem de programação Java.

Os parâmetros para a execução do algoritmo genético foram os seguintes: tamanho da população como 500; penalidade como 0; penalidade excedente como 20; probabilidade de mutação como 1%; taxa de *crossover* como 80%; máximo de iterações como 200. Para a busca em largura e profundidade, foi definido como 5 o número máximo de passos. Ainda que o AG gere soluções diferentes a cada execução, optou-se por utilizar a primeira solução gerada, pois o teste foi feito para 5 tarefas diferentes. A Tabela 1 traz o comparativo de resultados obtidos.

Tabela 1 – Comparação de algoritmos com solução ótima – valores médios de 5 tarefas

Tipo de algoritmo	Relação de custo com o ótimo	Relação de tempo com o ótimo
Ótimo	100%	100% (465 segundos)
Algoritmo Genético	107%	0,1774194% (0,825 segundo)
Busca em largura	278%	0,0021505% (0,01 segundo)
Busca em profundidade	396%	0,003871% (0,018 segundo)

Fonte: elaborado pelo autor (2019)

Com exceção do algoritmo de busca em profundidade, que não conseguiu executar completamente 2 tarefas, os demais executaram todas as 5 tarefas por completo, respeitando a restrição de tempo imposta. O Algoritmo Genético proposto teve um desempenho de custo médio na ordem de 7% a mais que o algoritmo ótimo; a busca em largura, neste caso, teve um custo de 278% do algoritmo ótimo; e a busca em profundidade, mesmo sem resolver duas tarefas, estava com o custo em 396% do ótimo. Por outro lado, o tempo de execução médio do AG foi inferior a 1% do algoritmo ótimo; enquanto que os algoritmos de busca em largura e busca em profundidade tiveram tempos de 0,002% e 0,0038%. Isso demonstra que os algoritmos aqui propostos são consideravelmente mais rápidos que o algoritmo ótimo e, ainda, o Algoritmo Genético gerou uma solução com um valor de custo pouco maior que o ótimo.

4.7 COMPARAÇÃO ENTRE ALGORITMOS

Além da comparação dos dois algoritmos baseados em busca e do AG proposto com a solução ótima, foi realizada uma comparação entre os três algoritmos construídos. Para tanto, foram executadas 100 tarefas em uma rede criada com 8 mil nós e grau médio 5. A rede foi criada no Pajek, como uma rede aleatória, livre de escala e direcionada. O peso das arestas, definido no simulador, variou entre 1 e 50. Para cada nó da rede com grau maior ou igual a 1, foi criado um conjunto de habilidades, contendo de 1 a 10 itens. O tempo definido de forma aleatória para cada item de habilidade teve valores entre 1 a 10. E, por fim, os custos de cada item de habilidade em cada nó foram gerados com valores entre 1 e 30.

As tarefas foram construídas de forma aleatória, contendo entre 3 e 15 itens. Para cada tarefa, foi designado um nó inicial, escolhido ao acaso entre os nós que possuem pelo menos uma conexão chegando até si. O tempo limite de cada tarefa foi definido como o triplo do tamanho da tarefa – buscando restringir a execução, tendo em vista que os valores de tempo variam entre 1 e 10, tem-se uma expectativa média em torno de 5 unidades de tempo para cada item. As mesmas, com os mesmos nós de demanda, foram executadas em cada algoritmo. Após as execuções, foi verificado se cada tarefa foi atendida totalmente, com a definição de fornecedores para todas os itens da tarefa, se o limite de tempo foi respeitado no caso do algoritmo genético, e os custos gerados.

Para a execução dos algoritmos de busca em largura e profundidade, foi definido um máximo de 5 passos. Para o algoritmo genético, os parâmetros foram os seguintes: número de nós analisados para a construção da árvore de fornecedores: 30; tamanho da população: 20; penalidade geral: 0; penalidade de tempo excedente: 5; probabilidade de mutação: 1%; máximo de iterações em cada tarefa: 20; taxa de elitismo: 10%; taxa de *crossover*: 80%. Para o AG, recuperou-se o cromossomo (solução) de melhor aptidão da população final gerada. Sabe-se que diferentes execuções do AG levam a diferentes resultados, porém optou-se por utilizar a primeira execução, tendo em vista que o teste foi repetido para 100 tarefas. A seguir, são discutidos os resultados obtidos com o teste.

Nas execuções do AG, apenas 2 tarefas não foram completadas, isto é, não foram encontrados nós fornecedores para todos os itens. Isso ocorreu em virtude do número de nós analisados na formação da árvore de fornecedores. Por outro lado, 48 tarefas foram executadas com tempo acima do limite definido, totalizando, portanto, 50 tarefas executadas totalmente e dentro do limite de tempo estipulado. Para a busca em largura, 24 tarefas foram concluídas totalmente, dentro do limite de tempo definido, enquanto que o algoritmo baseado

em busca em profundidade concluiu corretamente 22 tarefas. 15 tarefas foram completadas tanto pelo algoritmo de busca em largura quanto pelo AG; desses, em 12 casos o AG teve um custo menor, e em 3 a busca em largura teve menor custo. Comparando os algoritmos de busca em largura e profundidade, 17 tarefas foram completadas com sucesso em ambos algoritmos. Desses, apenas 1 caso teve um custo menor no algoritmo de profundidade. O mesmo ocorreu quando comparados o AG com o algoritmo de busca em profundidade: 17 tarefas foram completadas corretamente pelos dois, sendo que em apenas um caso a busca em profundidade teve custo menor que o Algoritmo Genético.

Em relação à correta execução das tarefas, dentro do limite de tempo estipulado, tem-se que o Algoritmo Genético completou 35 tarefas que a busca em largura não completou. Por outro lado, a busca em largura completou 9 tarefas que o AG não foi capaz de completar dentro do tempo. Comparando o AG com a busca em profundidade, tem-se que 35 tarefas foram completadas pelo AG que não foram completadas pela busca em profundidade; enquanto que 5 tarefas foram completadas pela busca em profundidade, mas não foram completadas pelo AG dentro do tempo definido. Em 5 ocasiões o algoritmo de busca em profundidade completou tarefas que o algoritmo de busca em largura não conseguiu completar; por outro lado, 7 tarefas foram completadas corretamente pelo algoritmo de busca em largura e não foram completadas pelo algoritmo de busca em profundidade.

Dessa forma, pôde-se perceber que o Algoritmo Genético obteve melhores resultados frente aos algoritmos de busca em largura e busca em profundidade. Entre esses, ainda, o algoritmo de busca em largura teve melhores resultados, conseguindo resolver mais tarefas e com menor custo – provavelmente devido à proximidade dos nós escolhidos com o nó que demanda a tarefa.

5 USO DO SIMULADOR PARA AVALIAÇÃO DE HIPÓTESES E DIFERENTES CENÁRIOS DE PRODUÇÃO DA INDÚSTRIA 4.0

Nesta etapa do trabalho, fez-se uso do simulador desenvolvido para avaliar diferentes hipóteses e cenários de produção dentro da indústria 4.0. As seções a seguir discutem a execução de tarefas em diferentes tipos de redes, e a relação entre o tamanho e a conectividade das redes com a capacidade de execução de tarefas.

5.1 EXECUÇÃO DE TAREFAS EM DIFERENTES TIPOS DE REDES

As redes de produção da Indústria 4.0 podem ter diferentes estruturas, o que pode impactar em seu funcionamento. Nesta etapa do trabalho, buscou-se analisar o comportamento, quanto à execução de tarefas, de três tipos de redes construídas a partir de diferentes algoritmos de geração de grafos aleatórios: (i) grafos de pequeno mundo (*small-world*); (ii) grafos livre de escala; e (iii) grafos aleatórios. As redes foram construídas com 64 mil nós cada, utilizando o *software* Pajek, seguindo os parâmetros listados abaixo:

1. Pequeno mundo (*small-world*): 64 mil vértices, 5 vizinhos ligados a cada vértice na rede inicial, probabilidade de troca de 0,5.
2. Livre de escala: rede direcionada, 64 mil vértices, grau médio 5.
3. Rede aleatória: rede com 64 mil nós e 320 mil arestas, utilizando o algoritmo de Erdos-Renyi.

Após a construção das redes, foram definidos pesos aleatórios para suas arestas, utilizando o simulador. Os pesos foram definidos entre 1 e 50, para as arestas das 3 redes. Além disso, foram definidas habilidades aleatórias com os seguintes parâmetros: número de habilidades entre 1 e 10 para cada nó que tinha pelo menos 1 conexão (grau maior ou igual a 1), tempo de cada item de habilidade definido entre 1 e 10, custo para cada item definido entre 1 a 30. A capacidade foi definida como 9999 para todos, tendo em vista que não seria alvo de investigação.

Feito isso, foram executadas 1000 tarefas aleatórias, utilizando o Algoritmo Genético proposto, em cada uma das três redes. As tarefas foram construídas com 15 itens, atribuídas a um nó aleatório (que possui pelo menos um nó conectado a si), e tiveram como limite de tempo o valor 45. A lista de fornecedores foi construída, para cada tarefa, com a análise de 50 nós. Os parâmetros do algoritmo genético foram:

- Probabilidade de mutação: 1%.
- Penalidade geral: 0.

- Tamanho da população: 50.
- Taxa de *crossover*: 80%.
- Taxa de elitismo: 2%.
- Penalidade para o tempo excedente: 10.

A tabela 2 resume os resultados obtidos, com a quantidade de tarefas completadas em sua totalidade, isto é, em quantas tarefas foram definidos fornecedores para todos os itens; a quantidade de tarefas completadas dentro do limite de tempo definido (45); o custo médio das tarefas; e o tempo médio das tarefas.

Tabela 2 – Resultado da execução de 1000 tarefas em três tipos de redes

Tipo de rede	Tarefas completadas	Tarefas completadas no tempo	Custo médio	Tempo médio
Pequeno mundo	1000	770	571	45
Livre de escala	985	965	545	43
Aleatória	1000	880	556	44

Fonte: elaborado pelo autor (2019)

Pode-se verificar que, embora na rede livre de escala 1,5% das tarefas não tenha sido executada em sua totalidade, 96,5% das tarefas foram executadas dentro do limite de tempo, enquanto que na rede com característica de pequeno mundo 77% das tarefas foram executadas no tempo, e na rede aleatória 88% foram executadas dentro do limite de tempo. O custo médio e o tempo médio também foram menores na com característica livre de escala: enquanto que o custo médio na rede livre de escala foi 545, os custos nas redes pequeno mundo e aleatória foram 571 e 556, respectivamente. Já o tempo médio foi 43 na rede livre de escala, 45 na rede pequeno mundo e 44 na rede aleatória.

Verificou-se, portanto, que a execução das tarefas foi mais efetiva na rede livre de escala. Embora uma pequena parcela das tarefas não tenha sido completada, a grande maioria delas foi atendida dentro do limite de tempo. O custo médio das tarefas nessa rede também foi menor que nas demais, assim como o tempo médio. Assim, na i4.0, redes com essa característica provavelmente serão mais efetivas que redes com característica apenas de pequenos mundos, ou redes totalmente aleatórias.

5.2 RELAÇÃO ENTRE O TAMANHO DA REDE E SUA CONECTIVIDADE COM A CAPACIDADE DE EXECUÇÃO DE TAREFAS

Diferentes cenários de redes de produção podem estar presentes na Indústria 4.0. Redes podem ser maiores ou menores, e seus nós podem estar mais ou menos conectados. A produção, dessa forma, pode ser afetada por essas características da rede. Nesta etapa do trabalho, buscou-se utilizar o simulador desenvolvido para analisar três hipóteses, construídas nesta pesquisa, tendo como base o trabalho de Putnik et al. (2015), trabalho em que a robustez de redes de produção e a taxa de produção foram analisadas com base no tamanho e o grau de conexões sociais da rede. As hipóteses criadas foram:

- Hipótese 1: em uma rede de Indústria 4.0, o número de tarefas de produção completas e o total de passos percorridos na busca por nós de execução estão relacionados ao tamanho e ao grau médio de conectividade da rede. O tamanho da rede pode fazer com que mais nós tenham habilidades para a execução de tarefas; e uma rede mais conectada tende a facilitar a busca por nós com habilidade para execução de uma tarefa.
- Hipótese 2: em uma rede de Indústria 4.0, o número de tarefas de produção completas está relacionado ao número máximo de passos que podem ser percorridos na busca por nós de execução, e varia com o grau e tamanho da rede. Espera-se que, ao percorrer mais passos, mais nós sejam atingidos e, conseqüentemente, mais tarefas possam ser concluídas. Esse comportamento pode variar com o tamanho da rede, já que, tendo mais nós, mais habilidades podem estar disponíveis, e com o grau de conectividade da rede, colaborando, também, para que mais nós sejam atingidos.
- Hipótese 3: Em uma rede de Indústria 4.0, a remoção de nós com maior conectividade afeta mais a execução de tarefas do que uma remoção aleatória, e está relacionada ao tamanho da rede e conectividade. A remoção de nós com maior conectividade, isto é, nós que tenham maior grau (número de arestas), possivelmente fará diminuir de forma mais acintosa a quantidade de tarefas completadas corretamente, em comparação com uma remoção aleatória, tendo em vista que contribuirá para uma maior diminuição do número de arestas da rede. Em redes mais conectadas, deve-se presenciar uma menor queda de produção, tendo em vista o número de conexões já existentes.

Para comprovar ou refutar as hipóteses elencadas, foram realizados testes utilizando o simulador desenvolvido e o *software* Pajek. Inicialmente, um conjunto de redes foi construído, a partir do Pajek, variando-se o número máximo de nós e o grau médio de cada rede. As redes foram construídas de forma aleatória, livres de escala, e tiveram como tamanho máximo 8 mil nós, 16 mil nós, 32 mil nós, 64 mil nós e 128 mil nós. Além disso, para cada tamanho, o grau médio foi variado entre 2, 4, 8, 16 e 32. Dessa forma, foram geradas 25 redes, combinando tamanho e grau médio. Para cada rede, foram criadas habilidades aleatórias, no simulador desenvolvido. O número mínimo de itens de habilidade que cada nó tem foi definido como 1, e o máximo indicado como 10. Para esta fase do trabalho, as habilidades dos nós foram determinadas como capacidade mínima e máxima igual a 1000, e os custos e tempos de resposta como 0, visto que estes valores não eram considerados nas hipóteses.

5.2.1 Análise da hipótese número 1 (H1)

A hipótese número 1 afirma que “em uma rede de Indústria 4.0, o número de tarefas de produção completas e o total de passos percorridos na busca por nós de execução estão relacionados ao tamanho e ao grau médio de conectividade da rede”. Para verificar essa hipótese, foram criadas e executadas 1000 tarefas em cada uma das 25 redes, com tamanho mínimo de 2 e máximo de 20 itens diferentes. A busca foi feita por largura, e o número máximo de passos definido foi 10. Cada teste foi executado 100 vezes, e os valores obtidos são a média dessas 100 execuções. A Tabela 3 sumarizam os dados obtidos, enquanto que a Figura 24 sintetiza esses valores graficamente.

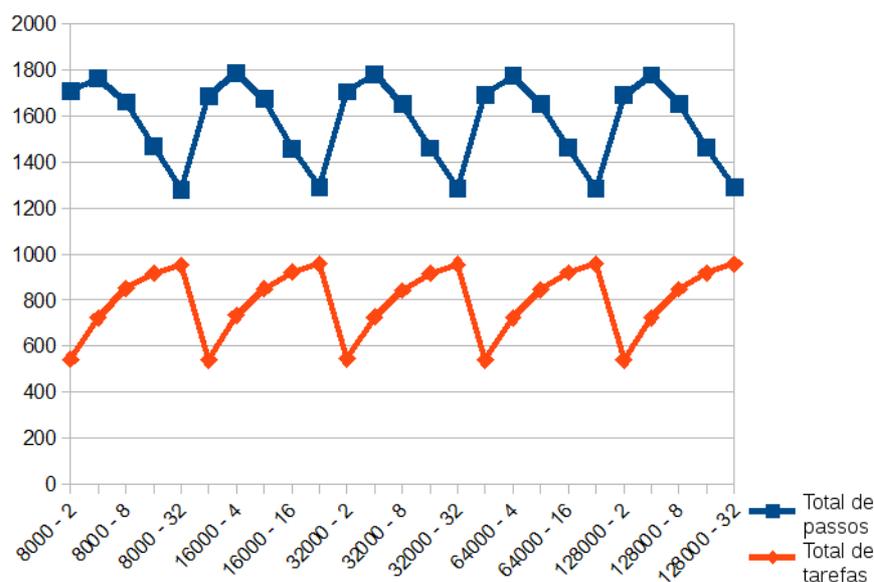
Tabela 3 – Total de passos e tarefas executadas

Número de nós	Grau médio	Total de passos percorridos	Tarefas executadas completamente
8000	2	1709	542
8000	4	1763	722
8000	8	1661	852
8000	16	1467	916
8000	32	1280	952
16000	2	1686	538
16000	4	1787	733
16000	8	1673	848
16000	16	1459	921
16000	32	1290	956
32000	2	1705	544
32000	4	1782	727
32000	8	1651	841
32000	16	1460	916
32000	32	1286	954
64000	2	1692	540
64000	4	1774	724
64000	8	1651	846
64000	16	1462	920
64000	32	1284	956
128000	2	1690	538
128000	4	1777	723
128000	8	1654	847
128000	16	1461	918
128000	32	1290	957

Fonte: elaborado pelo autor (2019)

A partir da tabela e do gráfico, percebe-se que tanto o número de tarefas completadas quanto o total de passos não tiveram grande variação com relação ao tamanho da rede, isto é, o comportamento foi semelhante para redes de tamanhos diferentes. Este comportamento pode ter ocorrido devido ao tamanho das redes e a quantidade de tarefas executadas, visto que o número de tarefa foi menor que o tamanho da rede.

Figura 24 – Execução de tarefas aleatórias na rede



Fonte: elaborado pelo autor (2019)

O aumento do grau médio da rede, por sua vez, teve influência direta tanto no número total de passos, quanto no número de tarefas executadas. Com exceção do grau médio 2 para 4, em que o total de passos aumentou, nos demais casos o aumento do grau médio diminuiu o total de passos. O número de tarefas completadas, por outro lado, sempre aumentou com o aumento do grau médio. A variação do grau, de 2 para 32, resultou em um aumento de cerca de 77% da quantidade de tarefas completadas, e em um decréscimo da ordem de 23% no número de passos.

As porcentagens de tarefas executadas em sua completude foram nas ordens de 53%, 72%, 84%, 91% e 95%, respectivamente, para redes com graus 2, 4, 8, 16 e 32. Percebe-se um avanço ao sair do grau 2 para o grau 4, e maior estabilização com o aumento da conectividade da rede. Os números de passos tiveram maiores quedas em redes com grau médio 16 e 32.

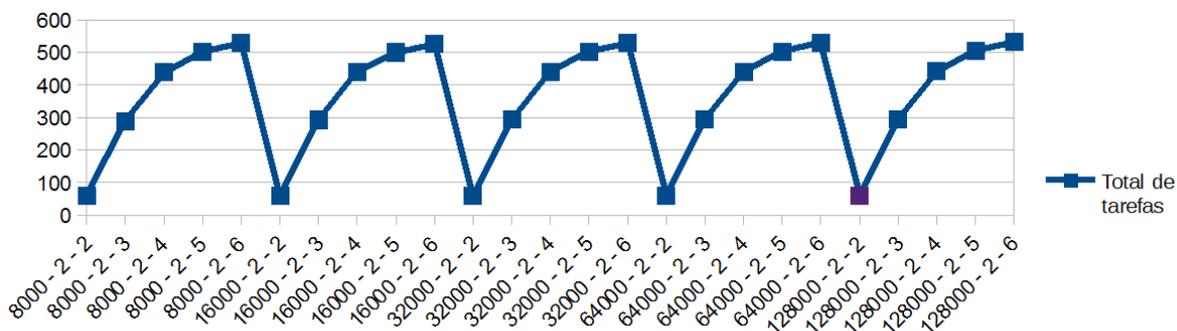
Verificou-se, portanto, que, para a simulação da execução de 1000 tarefas aleatórias, não houve diferenças quanto ao número de tarefas completadas em relação ao tamanho da rede. O aumento do grau médio da rede, por sua vez, influenciou a execução das tarefas. Verificou-se que redes mais conectadas conseguem completar mais tarefas, com menor número de passos, com exceção para as redes com grau médio 4, que tiveram mais passos que as redes com grau 2. Esse fato pode ter sido ocasionado pelo menor número de opções para busca em nós vizinhos nas redes com grau 2, em relação às redes com grau 4. A H1 foi, dessa forma, confirmada apenas em parte.

5.2.2 Análise da hipótese número 2 (H2)

A hipótese de número 2 indica que: “em uma rede de Indústria 4.0, o número de tarefas de produção completas está relacionado ao número máximo de passos que podem ser percorridos na busca por nós de execução, e varia com o grau e tamanho da rede”. Para verificar a H2, foram executadas 1000 tarefas com tamanho mínimo de 2 e máximo de 20 em cada uma das 25 redes, variando o número máximo de passos para a execução de 2 a 6. Cada simulação foi executada por 100 vezes, e os dados gerados referem-se à média dessas execuções. Foi utilizado o algoritmo baseado em busca em largura para a execução das tarefas.

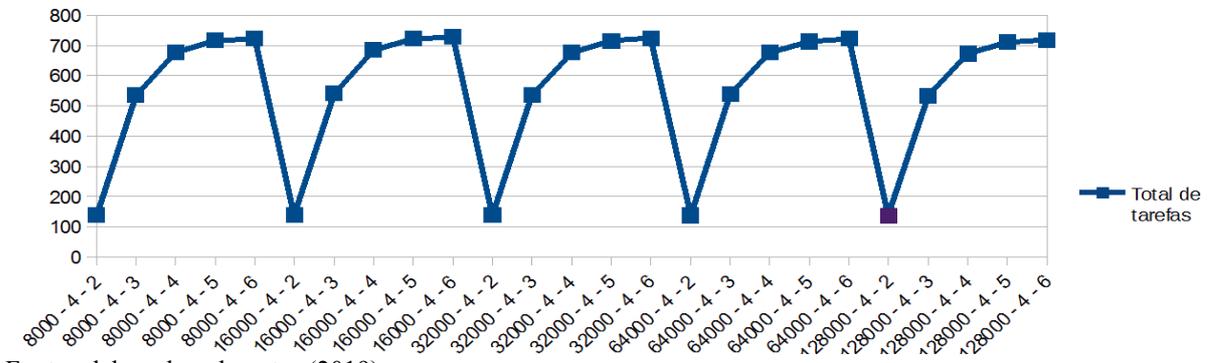
Assim como na H1, não foram verificadas alterações no número de tarefas executadas com o tamanho da rede. Por outro lado, o acréscimo de passos teve impacto direto na quantidade de tarefas executadas, assim como o grau médio dos nós das redes. Os gráficos das Figuras 25, 26, 27, 28 e 29 ilustram os resultados obtidos.

Figura 25 – Total de tarefas executadas por tamanho da rede e quantidade máxima de passos. Redes com grau médio 2



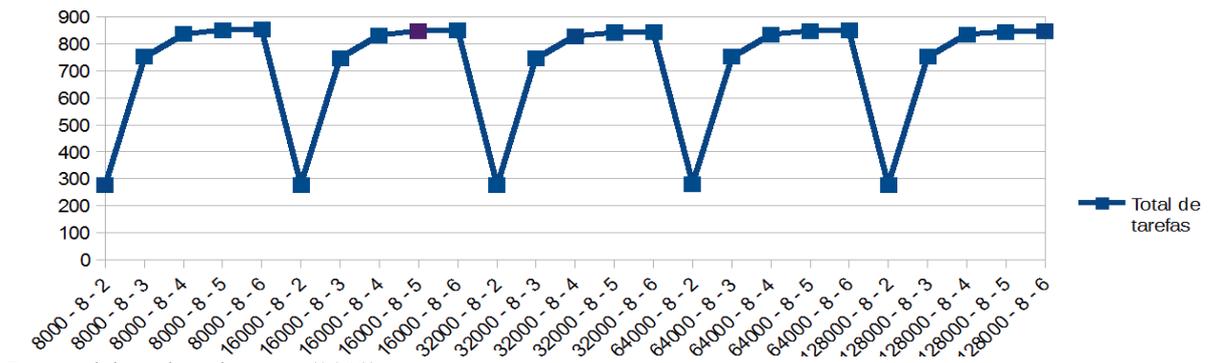
Fonte: elaborado pelo autor (2019)

Figura 26 – Total de tarefas executadas por tamanho da rede e quantidade máxima de passos. Redes com grau médio 4



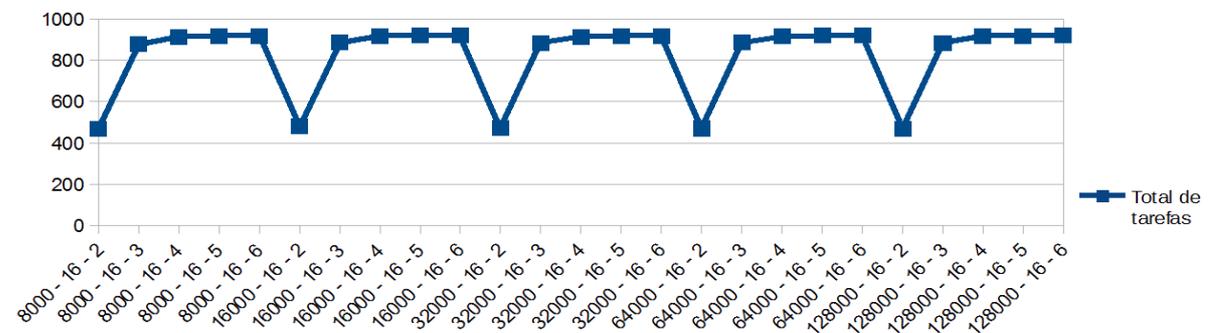
Fonte: elaborado pelo autor (2019)

Figura 27 – Total de tarefas executadas por tamanho da rede e quantidade máxima de passos. Redes com grau médio 8



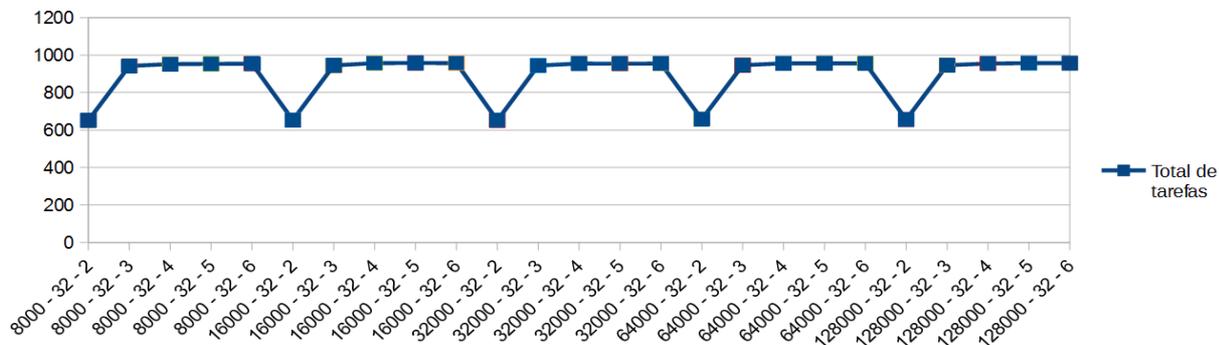
Fonte: elaborado pelo autor (2019)

Figura 28 – Total de tarefas executadas por tamanho da rede e quantidade máxima de passos. Redes com grau médio 16



Fonte: elaborado pelo autor (2019)

Figura 29 – Total de tarefas executadas por tamanho da rede e quantidade máxima de passos. Redes com grau médio 32



Fonte: elaborado pelo autor (2019)

Verifica-se, a partir dos gráficos, que o aumento do número máximo de passos tem influência sobre a quantidade de tarefas executadas, tendo em vista que mais nós serão visitados e, conseqüentemente, maior a chance de execução das tarefas. Para redes de grau 2 e 4, esse acréscimo ocorre de forma mais acentuada, o que pode ocorrer, pois estas redes são menos conectadas. Dessa forma, o acréscimo de passos faz com que a rede seja percorrida em maior amplitude. Já nas redes de grau médio 8, 16 e 32, há um menor ganho em relação ao número de tarefas executadas com o aumento de passos, o que pode ocorrer por elas já serem mais conectadas. Assim, o acréscimo de passos não tem tanto impacto, pois elas são percorridas em maior amplitude com um menor número de passos.

Dessa forma, pode-se concluir que uma maior varredura da rede levará a um número maior de tarefas completadas, principalmente em redes com menor conectividade. A hipótese H2, portanto, é confirmada em parte, já que o tamanho da rede não influenciou o número de tarefas completadas, mas o grau e o número de passos percorridos sim.

5.2.3 Análise da hipótese número 3 (H3)

A terceira hipótese indica que “em uma rede de Indústria 4.0, a remoção de nós com maior conectividade afeta mais a execução de tarefas do que uma remoção aleatória”. Para testar essa hipótese, foram executadas tarefas em redes que contavam, inicialmente, com 128 mil nós, e graus 2, 4, 8, 16 e 32. Para cada rede, foram removidos nós em porcentagem de 1%,

2%, 5%, 10%, 15% e 20%, de forma aleatória, ou escolhendo os nós com maior grau (conectividade). Foram executadas 1000 tarefas em cada rede, com tamanho mínimo de 2 e máximo de 20. A execução da tarefa foi feita como busca em largura, e o número máximo de passos definido foi de 5. Repetiu-se cada teste por 100 vezes, e os valores obtidos são a média dessas 100 execuções. As Tabelas 4, 5, 6, 7 e 8 ilustram a quantidade de tarefas completadas em cada rede, para a remoção de 1%, 5%, 10%, 15% e 20% dos nós respectivamente.

Tabela 4 – Quantidade de tarefas completadas com remoção de 1% dos nós

Grau	Tipo de remoção		Relação Grau/Aleatória
	Aleatória	Por grau	
2	505	451	89,31%
4	713	699	98,04%
8	853	843	98,83%
16	926	923	99,68%
32	964	964	100,00%

Fonte: elaborado pelo autor (2019)

Tabela 5 – Quantidade de tarefas completadas com remoção de 5% dos nós

Grau	Tipo de remoção		Relação Grau/Aleatória
	Aleatória	Por grau	
2	511	289	56,56%
4	730	630	86,30%
8	875	853	97,49%
16	950	942	99,16%
32	984	981	99,70%

Fonte: elaborado pelo autor (2019)

Tabela 6 – Quantidade de tarefas completadas com remoção de 10% nós

Grau	Tipo de remoção		Relação Grau/Aleatória
	Aleatória	Por grau	
2	516	98	18,99%
4	750	496	66,13%
8	899	836	92,99%
16	969	955	98,56%
32	993	990	99,70%

Fonte: elaborado pelo autor (2019)

Tabela 7 – Quantidade de tarefas completadas com remoção de 15% nós

Grau	Tipo de remoção		Relação Grau/Aleatória
	Aleatória	Por grau	
2	520	45	8,65%
4	765	356	46,54%
8	918	784	85,40%
16	979	962	98,26%
32	996	994	99,80%

Fonte: elaborado pelo autor (2019)

Tabela 8 – Quantidade de tarefas completadas com remoção de 20% nós

Grau	Tipo de remoção		Relação Grau/Aleatória
	Aleatória	Por grau	
2	520	16	3,08%
4	778	171	21,98%
8	930	662	71,18%
16	986	960	97,36%
32	997	995	99,80%

Fonte: elaborado pelo autor (2019)

Verificou-se que, na maioria dos casos, a remoção de nós com maior conectividade (grau) resulta em menor número de tarefas completadas. Ainda, este comportamento sofreu grande influência do grau médio da rede: em redes mais conectadas, o impacto da remoção de nós é menor do que nas redes menos conectadas. O aumento da porcentagem de remoção de nós resultou em uma diminuição do número de tarefas executadas, principalmente para as redes com grau médio 2 e 4. Nas redes com grau médio 16 e 32, o impacto foi consideravelmente menor. Nas redes com grau médio 8, o impacto foi maior com o aumento da porcentagem removida.

Na rede com grau médio 2, a remoção de 20% dos nós mais conectados fez com que apenas 16 tarefas fossem executadas em sua completude. Este número corresponde a 3,08% do total de tarefas completadas com a remoção de 20% dos nós da mesma rede, de forma aleatória. Ainda para as redes com grau médio 2, comparando a quantidade de tarefas completadas nas redes cujos nós removidos foram os mais conectados, em relação com as redes geradas com a remoção de nós de forma aleatória, verificou-se que a porcentagem de tarefas foi de 8,65% para a remoção de 15% dos nós; 18,99% para remoção de 10% dos nós; 56,56% para a remoção de 5% dos nós; e 89,31% para a remoção de 1% dos nós.

Dessa forma, a hipótese 3 (H3) foi confirmada, tendo em vista que a remoção de nós com maior conectividade afetou mais a execução de tarefas do que a remoção de tarefas de forma aleatória. Em especial, redes com menor grau de conectividade foram mais afetadas pela remoção dos nós de maior grau. Redes com maior grau médio terão mais nós com muitas conexões; nelas, mesmo removendo os nós mais conectados, ainda restarão outros com vários vizinhos, o que pode contribuir para a menor diminuição do total de tarefas completadas.

6 DISCUSSÃO DE RESULTADOS E CONSIDERAÇÕES FINAIS

A Indústria 4.0, ou Quarta Revolução Industrial, vem sendo tratada como um fenômeno que está alterando, e alterará, fortemente os processos produtivos, com a expansão da utilização e a integração de Tecnologias da Informação e Comunicação em todos os processos de manufatura. Pela primeira vez, uma Revolução Industrial vem sendo estudada antes de ocorrer, ou enquanto está acontecendo, e isso permite uma série de investigações para melhorar os processos ao longo de toda a cadeia produtiva.

Na i4.0, a utilização da Internet das Coisas, o uso de CPS e a proliferação de sensores, em fábricas inteligentes, acaba por conectar os mais diversos atores do processo produtivo, gerando grandes redes. Nessas redes, os nós trabalham em conjunto para atender tarefas de produção. O estudo das redes de produção da i4.0, em seu ambiente real, é difícil, tanto pela falta de casos práticos, quanto pelos custos e riscos envolvidos – a injeção de falhas e a reconfiguração da rede, para análise de cenários alternativos, podem ser catastróficos ao correto funcionamento do processo. Assim, o uso de técnicas de modelagem simulação computacional surge como uma opção viável para a análise das redes de produção da i4.0.

Esse trabalho tratou do desenvolvimento de um *software* simulador para análise de redes de produção, buscando contribuir para o avanço dos processos de gestão da Indústria 4.0. O trabalho foi conduzido utilizando a metodologia da *Design Science Research*, tendo em vista que seu objetivo era a construção de um artefato para a solução de um problema relevante, mantendo o rigor de uma pesquisa científica.

O desenvolvimento do artefato iniciou com estudos da literatura, a partir dos quais foram verificados conceitos relacionados com a Indústria 4.0, redes complexas (como as de produção da i4.0), modelagem e simulação computacional, heurísticas e Algoritmo Genético. Também foram analisados trabalhos relacionados com simulação na Indústria 4.0 e *softwares* para esta finalidade. Verificou-se que nenhum dos trabalhos encontrados propunha-se a resolver o mesmo problema tratado nesta pesquisa, tampouco os *softwares* possuíam abordagem semelhante à tratada aqui, corroborando com a relevância deste trabalho.

Posteriormente, foi definido o modelo de produção em redes da i4.0. O modelo considera os nós da rede como unidades de produção ou consumo, que podem possuir um conjunto de itens de habilidades de produção. Cada item de habilidade de um nó possui três características: custo, capacidade e tempo; e esses itens são representados pelas letras do alfabeto. Uma tarefa (demanda) de produção na rede é definida como um conjunto não ordenado de itens (letras), e é oriundo de um nó. O processo de resolução de uma tarefa

consiste em definir nós com habilidade para executar cada item da tarefa, e do caminho desde esses nós fornecedores até o nó que está demandando. Dessa forma, a tarefa pode ser realizada de forma colaborativa por vários nós da rede.

Foram determinados os requisitos que o simulador deveria atender para atingir ao objetivo do trabalho e contribuir para o avanço da gestão de redes de produção da i4.0: (i) permitir a leitura de arquivos no formato do *software* Pajek; (ii) permitir a edição da rede; (iii) possibilitar a criação de pesos aleatórios para as arestas; (iv) criar habilidades aleatórias para os nós da rede; (v) exibir a distribuição de grau da rede; (vi) permitir a remoção de nós; (vii) possibilitar a criação de tarefas; (viii) permitir a alteração de habilidades em lote; (ix) armazenar a rede completa em formato de arquivo; e (x) possuir interface gráfica.

Também foi elaborado e estruturado formalmente o problema de definição de nós para atender a uma tarefa de produção dentro da rede, e foram construídos três algoritmos não-ótimos para a resolução deste problema, dois deles baseados em busca em grafo, e um baseado na metaheurística de Algoritmo Genético. Feito isso, o simulador foi implementado como um *software*, utilizando a linguagem de programação Java, e seu comportamento foi validado. Os processos de avaliação (validação) mostraram que o artefato desenvolvido nesta pesquisa atinge seu propósito, ou seja, possibilita a análise de redes de produção da Indústria 4.0 através da simulação da execução de tarefas.

Após a avaliação do artefato, foram realizados testes em redes aleatórias, construídas a partir de diferentes algoritmos de geração de redes e/ou com diferentes características topológicas. Utilizou-se o simulador desenvolvido para verificar comportamento quanto à execução de tarefas. Estes testes mostram como o simulador pode ser utilizado como uma ferramenta para avaliação de cenários e teste de hipóteses, contribuindo para o avanço dos estudos em gestão da i4.0.

Como resultados desta pesquisa, pode-se elencar (i) o modelo de produção em rede que foi construído; (ii) os requisitos levantados para o simulador; (iii) a determinação do problema de definição de nós para a execução de uma tarefa de produção em redes da i4.0; (iv) os algoritmos para execução da tarefa; (v) a implementação do sistema; e (vi) os testes realizados em diferentes cenários de rede. Dessa forma, além do artefato em si, foi construído um arcabouço teórico para a análise de redes de produção da Indústria 4.0. A pesquisa, portanto, atingiu os seus objetivos de forma satisfatória.

6.1 GENERALIZAÇÃO PARA UMA CLASSE DE PROBLEMAS

A metodologia da DSR indica a generalização da aprendizagem para uma classe de problemas como um dos passos finais da pesquisa. A construção da classe de problemas, segundo Lacerda et al. (2013), é feita após a conscientização do problema, revisão sistemática da literatura e localização de artefatos para solucioná-lo. Dessa forma, definiu-se a classe de problemas na qual esta pesquisa está inserida como *Sistemas de Apoio à Decisão para a produção na Indústria 4.0*.

Sprague Jr. e Watson (1991) discutem a definição de um Sistema de Apoio à Decisão (SAD) trazendo tanto uma definição específica, como “sistemas computacionais interativos, que ajudavam os responsáveis pela tomada de decisões a utilizar dados e modelos para resolver problemas não-estruturados” (SPRAGUE JR; WATSON, 1991, p. 10), quanto uma definição ampla, que inclui qualquer sistema que contribui a um processo decisório. Os autores, porém, concluem que as definições podem ser ambíguas, uma vez que pessoas de diferentes áreas de conhecimento terão visões diferentes destes sistemas – um cientista de computação e um gerente, por exemplo. Pode-se utilizar, todavia, a definição dada por Rosini e Palmisiano (2003): para eles, um sistema de apoio à decisão é um sistema de informações que atende tanto a gerência quanto outros usuários de organizações, auxiliando a tomada de decisões semiestruturadas, de forma interativa. Além deles, Sauter (2014) define um SAD como um sistema computacional capaz de organizar informações e modelar resultados para auxiliar uma escolha.

O SAD foca na decisão, buscando respostas rápidas, e faz uso de modelos de análise e modelagem de dados (ROSINI; PALMISANO, 2003). Para os autores, os sistemas de apoio à decisão são ferramentas de grande potencial no auxílio a aumentar a eficácia dos recursos humanos. Eles trabalham com manipulação de dados (filtragem, resumo, combinação) agilizando o uso de determinada informação (ROSINI; PALMISANO, 2003). Dentre os auxílios que um SAD dá, destaca-se a geração de melhores alternativas, resposta rápida a problemas, solução a problemas complexos e uso de dados mais apropriados (SAUTER, 2014).

A arquitetura de um Sistema de Apoio à Decisão usualmente é composta por três subsistemas (ROSINI; PALMISANO, 2003):

1. Subsistema de diálogo: trata da interação com o usuário do sistema, ou seja, é o local de entrada de dados do usuário, quando necessário, e também o local de saída das informações que auxiliam a tomada de decisão.

2. Subsistema de dados: é parte da arquitetura responsável pelo armazenamento de dados, oriundo de uma ou mais fontes, internas ou externas. Usualmente é explorado por um banco de dados.
3. Subsistema de modelos: abrange o conjunto de modelos de decisão necessários para a resolução do problema. A utilização de vários modelos permite a resolução de partes distintas de um problema por cada um deles.

Assim, verifica-se que o simulador desenvolvido nesta pesquisa pode ser enquadrado como um Sistema de Apoio à Decisão, voltado para a análise de redes de produção da Indústria 4.0. O subsistema de diálogo, no simulador, consiste na interface gráfica desenvolvida para receber a entrada de dados do usuário, e devolver a saída dos processamentos; o subsistema de dados consiste na manipulação dos arquivos e demais estruturas com as informações das redes; e o subsistema de modelos consiste em todo o conjunto de representação da rede e da execução de tarefas.

A partir da simulação, o operador do sistema pode avaliar cenários distintos, alterando parâmetros e recebendo resultados que auxiliam a sua tomada de decisão. Outros artefatos, como *softwares*, podem fazer parte da classe de problemas de SAD para a produção na i4.0, auxiliando a tomada de decisão de outras formas, como, por exemplo, através do escalonamento de atividades, roteamento de processos, entre outros. Assim, tem-se que o conhecimento gerado por essa pesquisa não fica restrito apenas a um problema pontual. Ele pode ser generalizado e utilizado em outros contextos, contribuindo para o avanço da pesquisa em gestão da i4.0.

6.2 COMUNICAÇÃO DE RESULTADOS

A metodologia da DSR também traz como um de seus últimos passos a comunicação dos resultados alcançados. Esta etapa está englobada em todo este texto. Além disso, até a construção deste texto, foram publicados dois artigos resultantes do trabalho. São eles:

1. PEREIRA, Adriano; SIMONETTO, Eugênio de Oliveira. Indústria 4.0: Conceitos e Perspectivas Para o Brasil. **Revista da Universidade Vale do Rio Verde**, v. 16, n. 1, 2018. Artigo teórico com o levantamento inicial desta pesquisa, abordando conceitos e as perspectivas da i4.0 para o cenário brasileiro.
2. PEREIRA, Adriano; SIMONETTO, Eugênio de Oliveira; PUTNIK, Goran; CASTRO, Hélio. How connectivity and search for producers impact production in Industry 4.0 networks. **Brazilian Journal of Operations & Production**

Management, v. 15, p. 528-534, 2018. Artigo abordando os resultados encontrados e discutidos na seção 5.2 desta tese.

6.3 LIMITAÇÕES DA PESQUISA

Apesar de atender ao objetivo proposto, esta pesquisa possui algumas limitações que foram levantadas ao longo de seu desenvolvimento. São elas: (i) a falta da disponibilidade de dados reais de redes de produção da Indústria 4.0; e (ii) a simplificação do processo no modelo de simulação construído, o que é inerente a pesquisas que envolvam modelagem e simulação computacional, como visto na seção 2.3.

A falta de dados reais de redes de produção da i4.0 fez com que os testes fossem realizados utilizando redes geradas aleatoriamente, utilizando, para isso, algoritmos de construção de redes com característica de pequeno mundo, redes com distribuição de grau livres de escala, e redes aleatórias. Além disso, o simulador foi utilizado para a definição das habilidades de cada nó produtor da rede, e para a geração das tarefas. Havendo dados reais, eles poderiam ser preprocessados e aplicados no simulador.

Além disso, o processo de produção envolve uma série de itens que não foram considerados neste trabalho. O modelo de produção foi simplificado, a fim de envolver os principais pontos considerados. Consideraram-se itens como custo, tempo e capacidade (disponibilidade) de produção. Sabe-se, porém, que outros itens poderiam ter sido envolvidos, como a capacidade de transporte – capacidade nas arestas da rede. Essa informação, todavia, não foi utilizada, pois (i) aumentaria a complexidade da resolução da tarefa, e (ii) seria mais um item a ser gerado de forma aleatória para a rede. Dentre as limitações do modelo, também tem-se o número máximo de itens de produção e consumo, e o tamanho da tarefa. Foram utilizados as letras do alfabeto (A, B, C, ..., Z) tendo com base o trabalho de Škulj et al. (2014), limitando a 26 o número de itens distintos. Em uma rede de produção, este número tende a ser muito maior. Dessa forma, porém, o simulador facilita a visualização das tarefas, e dos itens de produção/consumo.

6.4 TRABALHOS FUTUROS

A continuação desta pesquisa pode ser feita em diferentes aspectos, discutidos nesta seção. Um dos pontos a serem explorados é a expansão do modelo de produção, englobando características não inseridas neste momento. Nesse sentido, é possível aumentar o número de

itens de habilidade/consumo, não restringindo eles apenas às letras do alfabeto. Isso pode ser feito utilizando identificadores numéricos para cada item de produção e consumo, e criando um “glossário” de relações entre cada item e o seu significado.

Além disso, o simulador serve como um arcabouço teórico, isto é, um alicerce para a pesquisa em produção da i4.0. Dessa forma, espera-se que ele possa ser englobado em outros sistemas, como em ERP, ou outros *softwares* utilizados em *smart-factories*, gerando *Digital Twins*. Assim, o simulador pode ser utilizado para prever situações, avaliando cenários, englobado como uma ferramenta de apoio à tomada de decisão dentro das fábricas autônomas.

Outro ponto que pode ser trabalhado, no futuro, é em relação à interface do simulador. Embora ele tenha sido construído com uma interface gráfica, ela pode ser melhorada para facilitar a usabilidade dos usuários, principalmente se for utilizada para fins empresariais (e não acadêmicos). Além disso, outros algoritmos para a definição de nós para a execução de uma tarefa também podem ser propostos e avaliados em relação aos algoritmos aqui construídos.

Por fim, a continuidade do trabalho também pode incluir a utilização do *software* aqui desenvolvido para analisar outros cenários de produção em redes da Indústria 4.0, analisando o comportamento em frente a diferentes condições.

REFERÊNCIAS

AHMED, K.; BLECH, J.; GREGORY, M.; SCHMIDT, H. **Journal of Sensor and Actuator Networks**, v. 7, n. 3, p. 33, 2018.

ALBERT, R.; BARABÁSI, A.-L. Statistical mechanics of complex networks. **Reviews of modern physics**, APS, v. 74, n. 1, p. 47, 2002.

ALI, C. B.; KHAN, M. B.; REHMAN, M.; TANOLI, S. A. K. NI-USRP based experimental evaluation of cooperative communication over direct communication in lab environment. In: **2018 International Conference on Electronics, Information, and Communication (ICEIC)**. IEEE, 2018. p. 1-5.

AMARAL, L. A.; OTTINO, J. M. Complex networks. **The European Physical Journal B**, Springer, v. 38, n. 2, p. 147–162, 2004.

ANAND, S.; GHALSASI, O.; ZHANG, B.; GOEL, A.; REDDY, S.; JOSHI, S.; MORRIS, G. Additive Manufacturing Simulation Tools in Education. In: **2018 World Engineering Education Forum-Global Engineering Deans Council (WEEF-GEDC)**. IEEE, 2018. p. 1-6.

ANGIONE, G. et al. Integration and deployment of a distributed and pluggable industrial architecture for the PERFoRM project. **Procedia Manufacturing**, v. 11, p. 896-904, 2017.

ARROYO, J. E. C. **Heurísticas e metaheurísticas para otimização combinatória multiobjetivo**. 2002. Tese – Unicamp.

BANKS, J.; CARSON, J. S.; NELSON, B. L.; NICOL, D. M. **Discrete-Event System Simulation**. Pearson, 2010.

BARABÁSI, A.-L.; ALBERT, R. Emergence of scaling in random networks. *Science*, American Association for the Advancement of Science, v. 286, n. 5439, p. 509–512, 1999.

BASTIAN, M.; HEYMANN, S.; JACOMY, M. et al. Gephi: an open source software for exploring and manipulating networks. **Icwsm**, v. 8, p. 361–362, 2009.

BATAGELJ, V.; MRVAR, A. Pajek-program for large network analysis. **Connections**, v. 21, n. 2, p. 47–57, 1998.

BOCCALETTI, S.; LATORA, V.; MORENO, Y.; CHAVEZ, M.; HWANG, D.-U. Complex networks: Structure and dynamics. **Physics reports**, Elsevier, v. 424, n. 4-5, p. 175–308, 2006.

BOFFEY, T. **Graph theory in operations research**. Scholium International, 1982.

BOHÁCS, G.; RINKÁCS, A. Development of an ontology-driven, component based framework for the implementation of adaptiveness in a Jellyfish-type simulation model. **Journal Of Ambient Intelligence And Smart Environments**, v. 9, n. 3, p. 361-374, 2017.

BOSSEL, H. **Modeling and simulation**. Springer-Verlag, 2013.

BRASE, J. M.; BROWN, D. L. Modeling, simulation and analysis of complex networked systems-a program plan. **Lawrence Livermore National Laboratory**, 2009.

BRETTEL, M.; FRIEDERICHSEN, N.; KELLER, M.; ROSENBERG, M. How virtualization, decentralization and network building change the manufacturing landscape: An Industry 4.0 Perspective. **International Journal of Mechanical, Industrial Science and Engineering**, v. 8, n. 1, p. 37–44, 2014.

BUKATA, L.; ŠŮCHA, P.; HANZÁLEK, Z.; BURGET, P. Energy optimization of robotic cells. **IEEE Transactions on Industrial Informatics**, v. 13, n. 1, p. 92-102, 2016.

BUXMANN, P.; HESS, T.; RUGGABER, R. Internet of services. **Business & Information Systems Engineering**, Springer, v. 1, n. 5, p. 341, 2009.

CAPUTO, F.; GRECO, A.; FERA, M.; CAIAZZO, G.; SPADA, S. Simulation Techniques for Ergonomic Performance Evaluation of Manual Workplaces During Preliminary Design Phase. In: **Congress of the International Ergonomics Association**. Springer, Cham, 2018. p. 170-180.

CARUSO, L. Digital innovation and the fourth industrial revolution: epochal social changes?. **Ai & Society**, v. 33, n. 3, p. 379-392, 2018.

CARVAJAL SOTO, J. A.; TAVAKOLIZADEH, F.; GYULAI, D. An online machine learning framework for early detection of product failures in an Industry 4.0 context. **International Journal of Computer Integrated Manufacturing**, v. 32, n. 4-5, p. 452-465, 2019.

ČEJKA, J.; ČERNOHORSKÝ, J. Optimization of robotic workplaces. In: **2016 17th International Carpathian Control Conference (ICCC)**. IEEE, 2016. p. 146-150.

CHUNG, Christopher A. **Simulation modeling handbook: a practical approach**. CRC press, 2004.

CHWIF, L.; MEDINA, A. C. **Modelagem e simulação de eventos discretos**. Afonso C. Medina, 2006.

CORALLO, A.; ESPOSTITO, M.; MASSAFRA, A.; TOTARO, S. A Relational Database Management System Approach for Data Integration in Manufacturing Process. In: **2018 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC)**. IEEE, 2018. p. 1-7.

CORMEN, T. H. **Introduction to algorithms**. MIT press, 2009.52

CSARDI, G.; NEPUSZ, T. The igraph software package for complex network research. **InterJournal Complex Systems**, v. 1695, n. 5, p. 1–9, 2006.

DE MARCO, I.; RIEMMA, S.; IANNONE, R. Uncertainty of input parameters and sensitivity analysis in life cycle assessment: An Italian processed tomato product. **Journal of cleaner production**, v. 177, p. 315-325, 2018.

DE GIORGIO, A.; ROMERO, M.; ONORI, M.; WANG, L. Human-machine collaboration in virtual reality for adaptive production engineering. **Procedia Manufacturing**, v. 11, p. 1279-1287, 2017.

DENKENA, B.; WINTER, F. Simulation-based planning of production capacity through integrative roadmapping in the wind turbine industry. **Procedia CIRP**, v. 33, p. 105-110, 2015.

DIJKSTRA, E. W. et al. A note on two problems in connexion with graphs. **Numerische mathematik**, v. 1, n. 1, p. 269-271, 1959.

DRATH, R.; HORCH, A. Industrie 4.0: Hit or hype? **IEEE industrial electronics magazine**, IEEE, v. 8, n. 2, p. 56–58, 2014.

DRESCH, A.; LACERDA, D. P.; ANTUNES JÚNIOR, J. A. V. **Design science research: método de pesquisa para avanço da ciência e tecnologia**. Bookman Editora, 2015.

EL-AZIZ, A. A. A.; KANNAN, A. JSON encryption. In: **2014 International Conference on Computer Communication and Informatics**. IEEE, 2014. p. 1-6.

ERDÖS, P.; RÉNYI, A. On random graphs, I. **Publicationes Mathematicae (Debrecen)**, v. 6, p. 290–297, 1959. Disponível em: < http://www.renyi.hu/%5C~%7B%7Dp%5C_erdos/Erdos.html%5C#1959-11 >.

ESPOSITO, C.; FICCO, M.; CASTIGLIONE, A.; PALMIERI, F.; LU, H. . Loss-tolerant event communications within industrial internet of things by leveraging on game theoretic intelligence. **IEEE Internet of Things Journal**, v. 5, n. 3, p. 1679-1689, 2017.

FELLER, N.; AMANN, A.; MÜLLER, U.; SCHIFFMANN, M.; KURSCHEID, O.; GORZELLIK, M. Application of a Simulation-Based Software Tool for the Prospective Design of IT Work Places. In: **Advances in Neuroergonomics and Cognitive Engineering**. Springer, Cham, 2017. p. 87-96.

FIGUEIREDO, D. R. Introdução a redes complexas. **Atualizações em Informática**, p. 303–358, 2011.

FIRJAN. **Indústria 4.0: Panorama da Inovação**. 2016.

FONTANA, G.; GRASSO, F.; LUCHETTA, A.; MANETTI, S.; PICCIRILLI, M. C.; REATTI, A. A new simulation program for analog circuits using symbolic analysis techniques. In: **2015 International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)**. IEEE, 2015. p. 1-4.

FRICKE, A.; SCHÖNEBERGER, J. C. Know-how Protection and Software Architectures in Industry 4.0. In: **Computer Aided Chemical Engineering**. Elsevier, 2017. p. 2287-2292.

GATH, M.; HERZOG, O.; EDELKAMP, S. Agent-based planning and control for groupage traffic. In: **2013 10th International Conference and Expo on Emerging Technologies for a Smarter World (CEWIT)**. IEEE, 2013. p. 1-7.

- GATH, M.; HERZOG, O.; EDELKAMP, S. Autonomous and flexible multiagent systems enhance transport logistics. In: **2014 11th International Conference & Expo on Emerging Technologies for a Smarter World (CEWIT)**. IEEE, 2014. p. 1-6.
- GEHLHOFF, F.; LINNENBERG, T.; FAY, A. Optimization of auction mechanisms Agent-based coupling of cooperative production plants. **ATP EDITION**, n. 9, p. 54-66, 2017.
- GOLDBARG, H. P.; LUNA, H. P. L. **Otimização combinatória e programação linear: modelos e algoritmos** Rio de Janeiro: Elsevier, 2005.
- GONG, Lixiong; ZOU, Bingqian; KAN, Zhiqun. Modeling and Optimization for Automobile Mixed Assembly Line in Industry 4.0. **Journal of Control Science and Engineering**, v. 2019, 2019.
- GRASSO, F.; LUCHETTA, A.; MANETTI, S.; PICCIRILLI, M. C.; REATTI, A. SapWin 4.0—a new simulation program for electrical engineering education using symbolic analysis. **Computer Applications in Engineering Education**, v. 24, n. 1, p. 44-57, 2016.
- GRASSO, F.; LUCHETTA, A.; MANETTI, S.; PICCIRILLI, M. C.; REATTI, A. SapWin 4.0'An enhanced tool for analysis and design of analog circuits. In: **2014 International Symposium on Fundamentals of Electrical Engineering (ISFEE)**. IEEE, 2014. p. 1-5.
- GUST, P.; MÜLLER, U.; FELLER, N.; SCHIFFMANN, M. Field Study on the Application of a Simulation-Based Software Tool for the Strain-Based Staffing in Industrial Manufacturing. In: **Advances in Applied Digital Human Modeling and Simulation**. Springer, Cham, 2017. p. 3-12.
- HASSAN, N. M. S.; RASUL, M. G.; HARCH, Carl A. Modelling and experimental investigation of engine performance and emissions fuelled with biodiesel produced from Australian Beauty Leaf Tree. **Fuel**, v. 150, p. 625-635, 2015.
- HERMANN, M.; PENTEK, T.; OTTO, B. Design principles for industrie 4.0 scenarios. In: IEEE. SYSTEM Sciences (HICSS), 49th Hawaii International Conference on. **Anais...** 2016. p. 3928–3937.
- HOFER, F. Architecture, technologies and challenges for cyber-physical systems in industry 4.0: a systematic mapping study. In: **Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement**. ACM, 2018. p. 1.
- HSIEH, L. Y.; HUANG, E.; CHEN, C. H. Equipment Utilization Enhancement in Photolithography Area through a Dynamic System Control Using Multi-fidelity Simulation Optimization with Big Data Technique. **IEEE Transactions on Semiconductor Manufacturing**, v. 30, n. 2, p. 166-175, 2017.
- IHAKA, R.; GENTLEMAN, R. R. a language for data analysis and graphics. **Journal of computational and graphical statistics**, Taylor & Francis, v. 5, n. 3, p. 299–314, 1996.
- JUNKER, H.; DOMANN, C. Towards industry 4.0 in corporate energy management. **WIT Transactions on Ecology and the Environment**, v. 214, p. 49-56, 2017.

KADERA, P.; NOVÁK, P. Performance modeling extension of directory facilitator for enhancing communication in FIPA-compliant multiagent systems. **IEEE Transactions on Industrial Informatics**, v. 13, n. 2, p. 688-695, 2016.

KAGERMANN, H.; WAHLSTER, W.; HELBIG, J. Securing the future of German manufacturing industry. **Recommendations for implementing the strategic initiative INDUSTRIE**, v. 4, 2013.

KAZIMIROV, A. N. Education at University and Industry 4.0. In: **2018 Global Smart Industry Conference (GloSIC)**. IEEE, 2018. p. 1-6.

KŁOS, S. Knowledge Acquisition Using Computer Simulation of a Manufacturing System for Preventive Maintenance. In: **International Conference on Information and Software Technologies**. Springer, Cham, 2018. p. 29-40.

KOCSI, B.; OLÁH, J. Potential connections of unique manufacturing and industry 4.0. **LogForum**, v. 13, n. 4, 2017.

KRANZ, R.; ZUPICK, N. Arena and industry 4.0. In: **2017 Winter Simulation Conference (WSC)**. IEEE, 2017. p. 4450-4450. (PALESTRA

KRENCZYK, D.; KEMPA, W. M.; KALINOWSKI, K.; GRABOWIK, C.; PAPROCKA, I. Integration of manufacturing operations management tools and discrete event simulation. In: **IOP Conference Series: Materials Science and Engineering**. IOP Publishing, 2018. p. 022037.

KRENCZYK, D.; SKOLUD, B.; HEROK, A. A heuristic and simulation hybrid approach for mixed and multi model assembly line balancing. In: **International conference on intelligent systems in production engineering and maintenance**. Springer, Cham, 2017. p. 99-108.

LACERDA, E. G. M.; CARVALHO, A. C. P. L. F. Introdução aos algoritmos genéticos. **Sistemas inteligentes: aplicações a recursos hídricos e ciências ambientais**, v. 1, p. 99-148, 1999.

LACERDA, D. P.; DRESCH, A.; PROENÇA, A.; ANTUNES JÚNIOR, J. A. V. Design Science Research: método de pesquisa para a engenharia de produção. **Gestão & produção**, v. 20, n. 4, p. 741-761, 2013.

LACOMBLEZ, M.; JEANNE, B.; HAVARD, V.; BAUDRY, D. Co-Simulation Architecture Between a Digital Twin and a Virtual Reality Environment in an Industrial Context. In: **Advances in Manufacturing Technology XXXII: Proceedings of the 16th International Conference on Manufacturing Research**. September 11–13, 2018, University of Skövde, Sweden. IOS Press, 2018. p. 429.

LARMAN, C. **Agile and iterative development: a manager's guide**. Addison-Wesley Professional, 2004.

LASI, H.; FETTKE, P.; KEMPER, H.-G.; FELD, T.; HOFFMANN, M. Industry 4.0. **Business & Information Systems Engineering**, Springer, v. 6, n. 4, p. 239–242, 2014.

LAURINDO, Q. M. G.; PEIXOTO, T. A.; RANGEL, J. J. A.. Communication mechanism of the discrete event simulation and the mechanical project softwares for manufacturing systems. **Journal of Computational Design and Engineering**, v. 6, n. 1, p. 70-80, 2019.

LEE, Shih-Hsiung; YANG, Chu-Sing. A real time object recognition and counting system for smart industrial camera sensor. **IEEE Sensors Journal**, v. 17, n. 8, p. 2516-2523, 2017.

LECROQ, F.; GRIEU, J.; BOUKACHOUR, H. Poster: Teaching Automation and Logistics with Virtual Industrial Process. In: **International Conference on Remote Engineering and Virtual Instrumentation**. Springer, Cham, 2018. p. 57-65.

LEE, J.; BAGHERI, B.; KAO, H.-A. A cyber-physical systems architecture for industry 4.0-based manufacturing systems. **Manufacturing Letters**, Elsevier, v. 3, p. 18–23, 2015.

LI, T. H. S. *et al.* A three-dimensional adaptive PSO-based packing algorithm for an IOT-based automated e-fulfillment packaging system. **IEEE Access**, v. 5, p. 9188-9205, 2017.

LI, X.; LI, D.; WAN, J.; LIU, C.; IMRAN, M. Adaptive transmission optimization in SDN-based industrial Internet of Things with edge computing. **IEEE Internet of Things Journal**, v. 5, n. 3, p. 1351-1360, 2018.

LIHONG, D. Research on New Intelligent Industrial Control and Communication System based on PLC Technique. In: **2015 Conference on Informatization in Education, Management and Business (IEMB-15)**. Atlantis Press, 2015.

LINDO Systems. **Optimization Modeling with LINGO**. 6^a ed. Chicago: 2006.

LINS, T.; OLIVEIRA, R. A. R. Energy efficiency in industry 4.0 using SDN. In: **2017 IEEE 15th International Conference on Industrial Informatics (INDIN)**. IEEE, 2017. p. 609-614.

LONG, Fei; ZEILER, Peter; BERTSCHE, Bernd. Realistic modelling of flexibility and dependence in production systems in Industry 4.0 for analysing their productivity and availability. **Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability**, v. 232, n. 2, p. 174-184, 2018

LOVÁSZ, L. Large networks and graph limits. **American Mathematical Soc.**, 2012. v. 60.

LÜDER, A.; ZAWISZA, J.; BECKER, A. Advancing the Performance of Complex Manufacturing Systems Through Agent-Based Production Control. In: **German Conference on Multiagent System Technologies**. Springer, Cham, 2017. p. 103-120.

MARIA, A. Introduction to modeling and simulation. In: IEEE COMPUTER SOCIETY. PROCEEDINGS of the 29th conference on Winter simulation. **Anais...** 1997. p. 7–13.53

MARTY, C.; YU, J.; MARTIN, G.; BORNOFF, R.; POPPE, A.; FOURNIER, D. Design flow for the development of optimized LED luminaires using multi-domain compact model simulations. In: **2018 24rd International Workshop on Thermal Investigations of ICs and Systems (THERMINIC)**. IEEE, 2018. p. 1-7.

MARZANO, A.; FRIEL, I.; ERKOYUNCU, J. A.; COURT, S. Design of a Virtual Reality framework for maintainability and assemblability test of complex systems. **procedia cirp**, v. 37, p. 242-247, 2015.

METZ, J.; CALVO, R.; SENO, E. R.; ROMERO, R. A.; LIANG, Z. Redes Complexas: conceitos e aplicações. **Relatórios Técnicos do ICMC-USP**, São Carlos, 2007.

MORGADO, E. M.; ANDRÊO, J. R. Um estudo sobre a utilização do software SCRATCH como ferramenta de apoio ao ensino da disciplina de Lógica de Programação, 2013.

MOSTERMAN, P. J.; ZANDER, J. Industry 4.0 as a cyber-physical system study. **Software & Systems Modeling**, v. 15, n. 1, p. 17-29, 2016.

MOURTZIS, D.; DOUKAS, M.; PSAROMMATIS, F. Design and operation of manufacturing networks for mass customisation. **CIRP Annals-Manufacturing Technology**, Elsevier, v. 62, n. 1, p. 467-470, 2013.

MOURTZIS, D.; DOUKAS, M.; BERNIDAKI, D.. Simulation in manufacturing: Review and challenges. **Procedia CIRP**, v. 25, p. 213-229, 2014.

MOURTZIS, D.; TSAKALOS, D.; ZOGOPOULOS, V.. Optimization of highly automated production line: An advanced engineering educational approach. **Procedia Manufacturing**, v. 31, p. 45-51, 2019.

MOURTZIS, D.; VASILAKOPOULOS, A.; ZERVAS, E.; BOLI, N. Manufacturing System Design using Simulation in Metal Industry towards Education 4.0. **Procedia Manufacturing**, v. 31, p. 155-161, 2019.

MUHURI, Pranab K.; SHUKLA, Amit K.; ABRAHAM, Ajith. Industry 4.0: A bibliometric analysis and detailed overview. **Engineering applications of artificial intelligence**, v. 78, p. 218-235, 2019.

MÜLLER, U.; GUST, P.; FELLER, N.; SCHIFFMANN, M. WorkDesigner: consulting application software for the strain-based staffing and design of work processes. **Procedia Manufacturing**, v. 3, p. 379-386, 2015.

NEWMAN, M. E. The structure and function of complex networks. **SIAM review**, SIAM, v. 45, n. 2, p. 167-256, 2003.

NIE, D. F.; CHEN, X. D.; FAN, Z. C.; WU, Q. G. ALE-Based Finite Element Simulation of Large Deformation in Bending and Flattening Tests. **Procedia Engineering**, v. 130, p. 775-783, 2015.

OUN, A.; BENABDALLAH, I.; CHERIF, A. Improved Industrial Modeling and Harmonic Mitigation of a Grid Connected Steel Plant in Libya. **International Journal of Advanced Computer Science and Applications**, v. 10, No. 2, 2019.

PACHECO, Marco Aurélio Cavalcanti et al. Algoritmos genéticos: princípios e aplicações. **ICA: Laboratório de Inteligência Computacional Aplicada. Departamento de Engenharia Elétrica. Pontifícia Universidade Católica do Rio de Janeiro. Fonte desconhecida**, p. 28, 1999.

- PERNICA, B.; TOMÁŠKOVÁ, H. Do we Need More Fire Protection Units as an Impact of Population Aging and Industry 4.0 in The Czech Republic? In: **21st International Scientific Conference Economics and Management**. Brno, Czech Republic, 2016. p. 723 – 728.
- PENAS, O.; PLATEAUX, R.; PATALANO, S.; HAMMADI, M. Multi-scale approach from mechatronic to Cyber-Physical Systems for the design of manufacturing systems. **Computers in Industry**, v. 86, p. 52-69, 2017.
- POZDNEEV, B.; OVCHINNIKOV, P.; BUSINA, F.; OBUHOVA, E. Virtual Enterprise Environment and a New Era in Simulating Digital Manufacturing Management Processes, In: **INTED2018 Proceedings**, 2018, pp. 2715-2720.
- PURNAMA, I. L. I.; TONTOWI, A. E.; SOPHA, B. M. Development of Medical Props Production Towards Industry 4.0. In: **2018 1st International Conference on Bioinformatics, Biotechnology, and Biomedical Engineering-Bioinformatics and Biomedical Engineering**. IEEE, 2018. p. 1-5.
- PUTNIK, G. D.; CASTRO, H.; FERREIRA, L.; RUI, B.; VIEIRA, G.; ALVES, C.; SHAH, V.; PUTNIK, Z.; CRUZ-CUNHA, M. M.; VARELA, L. **Advanced Manufacturing Systems and Enterprises – Towards Ubiquitous and Cloud Manufacturing**. University of Minho, School of Engineering, LabVE, 2012.
- PUTNIK, G. D.; ŠKULJ, G.; VARELA, L.; BUTALA, P. et al. Simulation study of large production network robustness in uncertain environment. **CIRP Annals**, Elsevier, v. 64, n. 1, p. 439–442, 2015.
- RAIZER, A.; DOS SANTOS, D. A. P.; RIBEIRO, J. V. T.; AVILA, L. R.; FONSECA, M. P.; NEVES, M. R. B. Characterization of Electromagnetic Near Fields in Industrial Systems. In: **2018 13th IEEE International Conference on Industry Applications (INDUSCON)**. IEEE, 2018. p. 144-148.
- RAUCH, E.; SPENA, P. R.; MATT, D. T. Axiomatic design guidelines for the design of flexible and agile manufacturing and assembly systems for SMEs. **International Journal on Interactive Design and Manufacturing (IJIDeM)**, v. 13, n. 1, p. 1-22, 2019.
- RODIČ, B. Industry 4.0 and the new simulation modelling paradigm. **Organizacija**, v. 50, n. 3, p. 193-207, 2017.
- ROMÁN-IBÁÑEZ, V.; JIMENO-MORENILLA, A.; PUJOL-LÓPEZ, F. A. Distributed monitoring of heterogeneous robotic cells. A proposal for the footwear industry 4.0. **International Journal of Computer Integrated Manufacturing**, v. 31, n. 12, p. 1205-1219, 2018.
- ROSINI, A. M.; PALMISANO, A. **Administração de Sistemas de Informação e a Gestão do Conhecimento**. São Paulo: Thomson, 2003.
- RUPPERT, T.; ABONYI, J. Software Sensor for Activity-Time Monitoring and Fault Detection in Production Lines. **Sensors**, v. 18, n. 7, p. 2346, 2018.

RÜßMANN, M.; LORENZ, M.; GERBERT, P.; WALDNER, M.; JUSTUS, J.; ENGEL, P.; HARNISCH, M. Industry 4.0: The future of productivity and growth in manufacturing industries. **Boston Consulting Group**, v. 9, 2015.

SADIKI, S.; RAMADANY, M.; FACCIO, M.; AMEGOUZ, D.; BOUTAHARI, S. Running Smart Monitoring Maintenance Application Using Cooja Simulator. In: **International Journal of Engineering Research in Africa**. Trans Tech Publications, 2019. p. 149-159.

SAUTER, V. L. **Decision Support Systems for Business Intelligence**. Wiley, 2014.

SEBASTIAN, E. J.; JOSE, J. M.; SCHAPPACHER, M.; SIKORA, A. Seamless test environment for distributed embedded wireless networks. In: **2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)**. IEEE, 2017. p. 681-686.

SCHLECHTENDAHL, J.; KEINERT, M.; KRETSCHMER, F.; LECHLER, A.; VERL, A. Making existing production systems Industry 4.0-ready. **Production Engineering**, Springer, v. 9, n. 1, p. 143–148, 2015.

SHINDE, P.; MACHADO, P.; SANTOS, F. N.; MCGINNITY, T. M. Online Object Trajectory Classification Using FPGA-SoC Devices. In: **UK Workshop on Computational Intelligence**. Springer, Cham, 2018. p. 291-302.

SILVA, R. M. da; SANTOS FILHO, D. J.; MIYAGI, P. E. Modelagem de Sistema de Controle da Indústria 4.0 Baseada em Holon, Agente, Rede re Petri e Arquitetura Orientada a Serviços. In: XII Simpósio Brasileiro de Automação Inteligente (SBAI 2015). **Anais...** UNICAMP, 2015.

SILVER, E. A. An overview of heuristic solution methods. **Journal of the operational research society**, v. 55, n. 9, p. 936-956, 2004.

SIMON, H. A. **The sciences of the artificial**. MIT press, 1996.

ŠKULJ, G; VRABIC, R.; BUTALA, P. Experimental study of work system networking in production environment. **CIRP Annals**, v. 63, n. 1, p. 401-404, 2014.

SLACK, N.; CHAMBERS, S.; HARLAND, C.; HARRISON, A.; JOHNSTON, R. **Administração da Produção**: Edição Compacta. Atlas, 1999.

SLACK, N.; JOHNSTON, R.; BRANDON-JONES, A. **Administração da Produção**. Atlas, 2015.

SPRAGUE JR, R. H.; WATSON, H. J. **Sistema de Apoio à Decisão**: Colocando a Teoria em Prática. Rio de Janeiro: Campus, 1991.

SRAI, J. S.; KUMAR, M.; GRAHAM, G.; PHILLIPS, W.; TOOZE, J.; FORD, S.; BEECHER, P.; RAJ, B.; GREGORY, M.; TIWARI, M. K.; RAVI, B.; NEELY, A.; SHANKAR, R.; CHARNLEY, F.; TIWARI, A. DistribLACERDA, Daniel Pacheco et al. Design Science Research: método de pesquisa para a engenharia de produção. **Gestão & produção**, v. 20, n. 4, p. 741-761, 2013. uted manufacturing: scope, challenges and

opportunities. **International Journal of Production Research**, v. 54, n. 23, p. 6917-6935, 2016.

STAMATESCU, I.; STAMATESCU, G.; FAGARASAN, I.; ARGHIRA, N.; CALOFIR, V.; ILIESCU, S. S. ASID: Advanced system for process control towards intelligent specialization in the power engineering field. In: **2017 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)**. IEEE, 2017. p. 475-480.

SUJOVÁ, E; ČIERNA, H.; ŽABIŇSKA, I. Application of Digitization Procedures of Production in Practice. **Management Systems in Production Engineering**, v. 27, n. 1, p. 23-28, 2019.

TAN, Y.; YANG, W.; YOSHIDA, K.; TAKAKUWA, S. . Application of IoT-Aided Simulation to Manufacturing Systems in Cyber-Physical System. **Machines**, v. 7, n. 1, p. 2, 2019.

TAO, F.; CHENG, J.; CHENG, Y.; GU, S.; ZHENG, T.; YANG, H. SDMSim: a manufacturing service supply-demand matching simulator under cloud environment. **Robotics and computer-integrated manufacturing**, v. 45, p. 34-46, 2017.

TAO, Q.; KANG, J.; SUN, W.; LI, Z.; HUO, X. Digital evaluation of sitting posture comfort in human-vehicle system under industry 4.0 framework. **Chinese Journal of Mechanical Engineering**, v. 29, n. 6, p. 1096-1103, 2016.

TRSTENJAK, Maja; COSIC, Predrag. Process planning in Industry 4.0 environment. **Procedia Manufacturing**, v. 11, p. 1744-1750, 2017.

TREBUNA, P.; PEKARCIKOVA, M.; EDL, M. Digital Value Stream Mapping Using the Tecnomatix Plant Simulation Software. **International Journal of Simulation Modelling (IJSIMM)**, v. 18, n. 1, 2019.

TURNER, C. J.; HUTABARAT, W.; OYEKAN, J.; TIWARI, A. Discrete event simulation and virtual reality use in industry: new opportunities and future trends. **IEEE Transactions on Human-Machine Systems**, v. 46, n. 6, p. 882-894, 2016.

URBIKAIN, G.; ALVAREZ, A.; LÓPEZ DE LACALLE, L.; ARSUAGA, M.; ALONSO, M.; VEIGA, F. A reliable turning process by the early use of a deep simulation model at several manufacturing stages. **Machines**, v. 5, n. 2, p. 15, 2017.

VAN AKEN, J. E. Management research based on the paradigm of the design sciences: the quest for field-tested and grounded technological rules. **Journal of management studies**, Wiley Online Library, v. 41, n. 2, p. 219–246, 2004.54

VELÍŠEK, K.; HOLUBEK, R.; SOBRINO, D. R. D.; RUŽAROVSKÝ, R.; VETRÍKOVÁ, N. Design of a robotized workstation making use of the integration of CAD models and Robotic Simulation software as way of pairing and comparing real and virtual environments. In: **MATEC Web of Conferences**. EDP Sciences, 2017. p. 05008.

VIEIRA, A. A. C.; DIAS, L. M. S.; SANTOS, M. Y.; PEREIRA, G. A. B.; OLIVEIRA, J. A. Setting an Industry 4.0 Research and Development Agenda for Simulation-A Literature Review. **International Journal of Simulation Modelling (IJSIMM)**, v. 17, n. 3, 2018.

VON ALAN, R. H.; MARCH, S. T.; PARK, J.; RAM, S. Design science in information systems research. **MIS quarterly**, Springer, v. 28, n. 1, p. 75–105, 2004.

WANG, S.; WAN, J.; ZHANG, D.; LI, D.; ZHANG, C. Towards smart factory for industry 4.0: a self-organized multi-agent system with big data based feedback and coordination. **Computer Networks**, v. 101, p. 158-168, 2016.

WANYAMA, T. Using industry 4.0 technologies to support teaching and learning. **The International journal of engineering education**, v. 33, n. 2, p. 693-702, 2017.

WATTS, D. J.; STROGATZ, S. H. Collective dynamics of ‘small-world’ networks. **Nature**, Nature Publishing Group, v. 393, n. 6684, p. 440, 1998.

WOLFRAM. **Wolfram Mathematica**: Computação técnica moderna. 2018. Disponível em: <<http://www.wolfram.com/mathematica>>.

YAZDI, P. G.; AZIZI, A.; HASHEMIPOUR, M. A Hybrid Methodology for Validation of Optimization Solutions Effects on Manufacturing Sustainability with Time Study and Simulation Approach for SMEs. **Sustainability**, v. 11, n. 5, p. 1454, 2019.

YOUJUN, J.; VAFAI, K. Analysis of pore scale fluid migration in a porous medium-application to coal rock seam. **International Journal of Numerical Methods for Heat & Fluid Flow**, v. 27, n. 8, p. 1706-1719, 2017.

XIA, F.; YANG, L. T.; WANG, L.; VINEL, A. Internet of things. **International Journal of Communication Systems**, v. 25, n. 9, p. 1101, 2012.

XU, J.; HUANG, E.; HSIEH, L.; LEE, L. H.; JIA, Q. S.; CHEN, C. H. Simulation optimization in the era of Industrial 4.0 and the Industrial Internet. **Journal of Simulation**, v. 10, n. 4, p. 310-320, 2016.

APÊNDICE A – EXEMPLO DE CÓDIGO JSON DE UMA REDE

```

{
  "directed": true,
  "nodes": [
    {
      "skills": [{
        "cost": 28,
        "skill": "X",
        "time": 9,
        "capacity": 9999
      }],
      "id": 1,
      "label": "v1"
    },
    {
      "skills": [
        {
          "cost": 23,
          "skill": "Q",
          "time": 7,
          "capacity": 9999
        },
        {
          "cost": 21,
          "skill": "A",
          "time": 5,
          "capacity": 9999
        },
        {
          "cost": 14,
          "skill": "S",
          "time": 1,
          "capacity": 9999
        },
        {
          "cost": 7,
          "skill": "D",
          "time": 4,
          "capacity": 9999
        },
        {
          "cost": 26,
          "skill": "F",
          "time": 10,
          "capacity": 9999
        },
        {
          "cost": 12,
          "skill": "X",
          "time": 6,
          "capacity": 9999
        },
        {
          "cost": 19,
          "skill": "H",
          "time": 10,
          "capacity": 9999
        },
        {
          "cost": 36,

```

```

        "skill": "I",
        "time": 3,
        "capacity": 9999
      },
      {
        "cost": 28,
        "skill": "J",
        "time": 5,
        "capacity": 9999
      },
      {
        "cost": 12,
        "skill": "L",
        "time": 9,
        "capacity": 9999
      }
    ],
    "id": 2,
    "label": "v2"
  },
  ...
],
"edges": [
  {
    "node1": 26285,
    "node2": 1,
    "weight": 18
  },
  {
    "node1": 1,
    "node2": 35196,
    "weight": 17,
  },
  {
    "node1": 46268,
    "node2": 1,
    "weight": 28,
  },
  ...
]
}

```

APÊNDICE B – CÓDIGO LINGO

```

MODEL:
SETS:
NOS/1..9/:CUSTO_TRANSPORTE;
ITEMS/A..Z/:TASK;
ARESTAS (NOS, NOS) /
1,5
3,2 3,4
4,7 4,1 4,2 4,3
5,7
6,9 6,3
7,8 7,9
8,9 8,2 8,3
/:PESO_ARESTA;

SKILLS (NOS, ITEMS) /
1,D 1,G 1,M
3,K
4,F 4,W
5,Q 5,B 5,S 5,T
6,U 6,X 6,Z 6,N
7,P 7,C 7,E 7,I 7,M 7,N
8,P 8,Q 8,C 8,E 8,V 8,G 8,H 8,K 8,M
9,P 9,E 9,V 9,G 9,X 9,J 9,Z 9,K 9,M 9,O
/:CUSTO_SKILL, TEMPO_SKILL, SOLUCAO;
ENDSETS
DATA:
TASK = 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0;
PESO_ARESTA =
1
3 2
1 2 2 2
1
1 4
2 3
1 2 2
;
!4 3 3 2;
CUSTO_SKILL=
4 6 5 9 6 8 9 10 6 3 4 5 9 5 6 8 5 10 8 10 9 8 8 5 4 8 8 10 6 9 6 6 5
8 3 6 7 8 4
;
TEMPO_SKILL=
4 7 8 9 9 9 4 1 7 7 7 2 9 10 2 4 6 1 4 5 5 3 2 5 7 6 4 6 9 5 5 5 5 10
5 10 9 5 10
;
ENDDATA

@FOR(NOS(i) | i #NE# 9:
CUSTO_TRANSPORTE(i) = @MIN(ARESTAS(i, j): PESO_ARESTA(i, j) +
CUSTO_TRANSPORTE(j))
);
CUSTO_TRANSPORTE(9) = 0;
@FOR(ITEMS(item): @FOR(NOS(no): @BIN(SOLUCAO(no, item))));
@FOR(ITEMS(item): @SUM(SKILLS(no, item): SOLUCAO(no, item)) =
TASK(item));
TEMPO_TOTAL = @SUM(SKILLS(no, item): SOLUCAO(no, item) *
TEMPO_SKILL(no, item));
TEMPO_TOTAL < 20;

```

```
      CUSTO_TOTAL = @SUM(SKILLS(no, item): CUSTO_SKILL(no, item) *  
SOLUCAO(no, item) + CUSTO_TRANSPORTE(no) * SOLUCAO(no, item) );  
      MIN=CUSTO_TOTAL;  
      END
```