

**UNIVERSIDADE FEDERAL DE ITAJUBÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM
ENGENHARIA DE PRODUÇÃO**

Renato Pontes Rodrigues

**Aplicação da Simulação Híbrida no Sequenciamento da
Produção em Sistemas Job Shop Flexíveis**

Novembro de 2019

Itajubá – MG

UNIVERSIDADE FEDERAL DE ITAJUBÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM
ENGENHARIA DE PRODUÇÃO

Renato Pontes Rodrigues

Aplicação da Simulação Híbrida no Sequenciamento da
Produção em Sistemas Job Shop Flexíveis

Tese submetida ao Programa de Pós-Graduação em Engenharia de Produção como requisito parcial à obtenção do título de Doutor em Engenharia de Produção.

Área de Concentração: Engenharia de Produção.

Orientador: Prof. Dr. Alexandre Ferreira de Pinho.

Novembro de 2019

Itajubá – MG

DEDICATÓRIA

*Aos meus pais
José Altamiro Rodrigues e
Isabel Carvalho Pontes Rodrigues.*

EPÍGRAFE

“O sucesso é ir de fracasso em fracasso sem perder entusiasmo”

Winston Churchill

AGRADECIMENTOS

A Deus, por me dar forças para fazer o meu trabalho da melhor forma, a cada dia, com dedicação e responsabilidade.

Ao meu orientador Professor Alexandre Ferreira de Pinho, que é um grande orientador, sempre se importando com os passos essenciais do processo de pesquisa, que muitas vezes passam despercebidos pelos alunos. Muito obrigado pela paciência, compreensão, motivação e amizade fundamentais para conclusão desta pesquisa.

Por todo o apoio, eu agradeço aos meus pais José Altamiro Rodrigues e Isabel Carvalho Pontes Rodrigues, em especial a minha mãe pela determinação e amor em todos os momentos difíceis de nossas vidas.

À minha namorada Luana Gonçalves Ribeiro por me ouvir, me apoiar e me incentivar em meu trabalho.

Ao meu irmão Raul, pela amizade, e aos meus sobrinhos Miguel, Giovana e Caio.

Ao amigo David pelo apoio nas dúvidas técnicas em relação a este trabalho.

Aos professores do Núcleo de Estudos Avançados para Auxílio à Decisão (NEAAD): José Arnaldo, Fabiano Leal, Rafael Miranda, José Hamilton, José Antônio de Queiroz.

Aos amigos do NEAAD do Instituto de Engenharia de Produção e Gestão agradeço pela companhia agradável e pelas dicas úteis para o desenvolvimento desta pesquisa.

Aos coordenadores do programa de pós-graduação em engenharia de produção em meu período, professor Carlos Sanches e Carlos Mello, por todo auxílio prestado ao longo desses anos de doutorado.

A CAPES, CNPq e a FAPEMIG pelo apoio e incentivo à pesquisa brasileira, em especial a essa.

A todos vocês, muito obrigado!

RESUMO

Esta pesquisa estuda um dos mais complexos e importantes problemas na área de sequenciamento de produção: os sistemas *job shop* flexíveis, o qual pertence à classe de problemas matematicamente intratáveis, denominados *NP-hard*. O aludido sistema é de suma importância para a indústria que utiliza a estratégia de produção *make to order* e busca flexibilidade de *mix* e volume. Existe uma demanda atual pela robustez do método de sequenciamento. O mesmo deve ser eficaz perante a problemas inesperados do cotidiano, que afetam a eficiência da sequência inicial definida. Além disso, deve-se considerar tempos estocásticos e buscar atingir mais objetivos de produção simultaneamente. Sendo assim, esta pesquisa realizará o sequenciamento por meio da simulação, que deverá sequenciar as ordens de produção no início do processo e re-sequenciar quando necessário, a fim de se buscar um modelo mais robusto aos problemas cotidianos e procurando melhorar mais indicadores ao mesmo tempo, isso porque modelos tradicionais tendem a ser mais sensíveis a eventos inesperados. Para isso, o referido sistema precisará usar agentes dentro de modelos de Simulação a Eventos Discretos, formando um modelo de Simulação Híbrida. O objetivo é avaliar como o agente dentro do modelo pode ajudar na resposta a estes eventos, adicionando robustez ao mesmo, além de deixá-lo mais acessível aos usuários. A pesquisa é de caráter quantitativo, utilizando o método de modelagem e simulação, seguindo um modelo empírico normativo. A contribuição científica é gerada no modelo final que é mais robusto a eventualidades e tem um foco mais abrangente. Além disso, esta maneira de modelar traz uma interface amigável para inserir modificações, o que melhora a integração com os usuários mais despreparados. A validação do modelo consiste na comparação de seus resultados com os resultados do gráfico de Gantt. Depois disso, foram feitas comparações dos resultados obtidos com e sem re-sequenciamento. Primeiro, com o agente utilizando uma lógica de sequenciamento e depois também a utilizando, porém com ajustes na sequência durante a produção dos lotes, buscando responder a eventos e neutralizar os pontos negativos da lógica inicial através do re-sequenciamento. Frisa-se que essa programação garante que o agente “Gerente” reduza o *makespan* e aumente a utilização das máquinas quando aumenta sua interferência no modelo.

PALAVRAS-CHAVE: Sequenciamento; Simulação Baseada em Agentes; Simulação Híbrida; Sistemas *Job Shop* Flexíveis.

ABSTRACT

This research studies one of the most complex and important issues in production scheduling research: flexible job shop systems, which belongs to a class of NP-hard problems. These systems are extremely important for the industry which uses make to order production strategy and seeks mix and volume flexibility. There is a current demand for robustness of the sequencing method. The same should be effective in the face of unexpected quotidian problems that affect the efficiency of the initial sequence defined. In addition, it must be considered stochastic times and seek to achieve more production goals simultaneously. The scheduling will be performed through simulation that should sequence the orders at the beginning of the process and re-sequence it when necessary, in order to seek a more robust model to everyday problems and seeking to improve more indicators at the same time. That is because traditional models tend to be more sensitive to unexpected events. The model system will use agents within Discrete-Event Simulation models, generating a Hybrid Simulation model. The goal is to evaluate how the agent within the model can help in responding to these events, adding robustness to it, and making it more accessible to users. This is a quantitative research, using the modeling and simulation method and following a normative empirical model. The scientific contribution is generated in the final model that is more robust to eventualities and has a broader focus. Additionally, the model constructed in this work brings a friendly interface to insert modifications, what enhance model integration with unprepared users. The work consists in the validation of the model by comparing its results with the results of the Gantt chart. After that, comparisons of the results obtained with and without re-sequencing were made. First with the agent using one sequencing logic and then using the same logic, but with sequence adjustments during the batches production, seeking to neutralize the negative points of the initial logic through re-sequencing. It also stresses that this schedule ensures that the Manager agent reduces makespan and increases machine utilization while increasing its interference in the model.

KEYWORDS:

Scheduling; Agent Based Simulation; Hybrid Simulation; Flexible Job Shop Systems.

LISTA DE FIGURAS

Figura 2.1 - Fluxograma para revisão sistemática da literatura.....	24
Figura 2.2 - Artigos publicados por ano.	27
Figura 2.3 - Artigos publicados por país	27
Figura 3.1 - Características do agente	45
Figura 5.1 - Diagrama de estado do agente no modelo intermediário.....	66
Figura 5.2 - Diagrama de estado do agente no modelo final.	67
Figura 5.3 - Mapeamento do SJSF do modelo final.	68
Figura 5.4 - Modelo computacional do SJSF (parte discreta).	73
Figura 6.1 - Gráfico de <i>Gantt</i> das oito ordens produzidas.....	84
Figura 6.2 - Testes de normalidade dos dados gerados pelo modelo	86
Figura 6.3 - <i>Boxplot</i> dos testes <i>One Sample-T</i>	88
Figura 7.1 - Valor do <i>makespan</i> (minutos) médio durante as réplicas.	90
Figura 7.2 - <i>Boxplot</i> do <i>makespan</i> e utilizações dos setores com e sem re-sequenciamento. ..	94
Figura A - Símbolo adicional utilizado no mapeamento	107

LISTA DE QUADROS

Quadro 3.1 - Definições para SBA com base nas propriedades do agente	43
Quadro 3.2 - Comparação entre SED e SBA.	48
Quadro 3.3 - Outros atributos comparativos entre SED e SBA.	49
Quadro 4.1 - Estrutura do Protocolo ODD	60
Quadro 5.1 - Lógica MTP.	74
Quadro 5.2 - Lógica Menor Fila Adiante modificada.	75
Quadro 5.3 - Evento de entrada do lote no modelo	80
Quadro 5.4 - Quebra do lote em produtos.	80
Quadro 5.5 - Lógica de movimentação dos produtos no modelo.....	81

LISTA DE TABELAS

Tabela 5.1 - Dados de entrada do modelo.	79
Tabela 6.1 - Resultados calculados por <i>Gantt</i> e os obtidos no modelo.....	85
Tabela 6.2 - Comparações estatísticas entre resultados do modelo e gráfico de <i>Gantt</i>	87
Tabela 7.1 - Tempos de reparo e tempos para quebras iniciais (minutos).	91
Tabela 7.2 - Ordens a serem produzidas.....	92
Tabela 7.3 - Comparações estatísticas das variâncias das respostas com e sem re-sequenciamento.	93
Tabela 7.4 - Comparações estatísticas do modelo com e sem re-sequenciamento.....	93
Tabela A - Simbologia utilizada na técnica IDEF-SIM	107

LISTA DE ABREVIATURAS

SED	Simulação a Eventos Discretos
SBA	Simulação Baseada em Agentes
SH	Simulação Híbrida
JSF	<i>Job Shop</i> Flexível
SJSF	Sistemas <i>Job Shop</i> Flexíveis
SP	Sequenciamento da Produção

SUMÁRIO

1. INTRODUÇÃO	14
1.1. Contextualização	14
1.2. Problematização da pesquisa	16
1.3. Objetivo	19
1.4. Contribuição científica	19
1.5. Estrutura da tese.....	21
2. REVISÃO SISTEMÁTICA	23
2.1. Revisão sistemática da literatura	23
2.2. Casos na literatura de SP em SJSF.....	28
3. REVISÃO BIBLIOGRÁFICA.....	35
3.1. Simulação a eventos discretos	35
3.2. Simulação baseada em agentes.....	39
3.2.1. Agentes	44
3.3. Simulação Híbrida	46
3.4. Sequenciamento da Produção.....	50
3.4.1. Medidas de desempenho da produção	52
3.4.2. Regras heurísticas de sequenciamento.....	54
4. MÉTODO DE PESQUISA	58
4.1. Classificação da pesquisa	58
4.2. Protocolo ODD.....	59
4.3. Passo a passo para a condução da pesquisa.....	62
5. DESCRIÇÃO DO MODELO HÍBRIDO PARA SEQUENCIAMENTO DA PRODUÇÃO EM SISTEMAS <i>JOB SHOP</i> FLEXÍVEIS	63
5.1. O Protocolo ODD do modelo proposto	63
5.1.1. Proposta	64
5.1.2. Entidades, variáveis de estado e escalas	65
5.1.3. Visão geral do processo	68
5.1.4. Programação	72
5.1.5. Projeto.....	77
5.1.6. Inicialização	78

5.1.7. Dados de entrada.....	78
5.1.8. Submodelos.....	79
6. VALIDAÇÃO DO MODELO COMPUTACIONAL.....	83
7. ANÁLISE DOS RESULTADOS: TESTES DO MODELO FINAL.....	90
7.1. Passo a passo para execução do sequenciamento.....	95
8. CONCLUSÕES.....	96
8.1. Verificação dos objetivos e solução da questão de pesquisa.....	96
8.2. Contribuições da tese.....	98
8.3. Sugestões para trabalhos futuros.....	99
ANEXO A – Simbologia usada no IDEF-SIM.....	107
APÊNDICE A - Artigos publicados.....	108

1. INTRODUÇÃO

1.1. Contextualização

O sequenciamento da produção (SP) é uma área com muitos obstáculos a serem resolvidos ou melhorados. Um deles é o sequenciamento em Sistemas *Job Shop* Flexíveis (SJSF), pois a alta complexidade desse tipo de sistema pede uma resolução por meio da simulação, que tem a grande vantagem de tornar mais simples a análise de sistemas complexos (BANKS *et al.*, 2009).

Yu e Lee (2018) afirmam que, no aspecto prático, o problema de SP tem muitas aplicações industriais, especialmente em sistemas de manufatura e serviços de alta variedade e baixo volume. O SP está entre os problemas mais complexos dos sistemas de produção, sendo uma área com muitos obstáculos a serem resolvidos ou, pelo menos, melhorados.

Segundo Asadzadeh (2015), o problema de SP em SJSF é NP-hard, que é uma classe de problemas matematicamente intratáveis. Este sistema é altamente complexo e, portanto, requer uma resolução por meio de simulação, que tem a grande vantagem de simplificar a análise de sistemas complexos (LAW e KELTON, 2007).

De acordo com Krajewski *et al.* (2016), sistemas *Job Shop* se constituem em um tipo de manufatura direcionada à produção de *jobs*, que são ordens de produção customizadas. Esse sistema usa o *layout* funcional ou por processo, em que as máquinas ficam agrupadas por tipos. A diferença entre sistemas *Job Shop* e SJSF é que, no primeiro, há apenas uma máquina de cada tipo ou a peça é destinada a uma máquina pré-determinada dentro do setor. Já no segundo, há várias máquinas do mesmo tipo com capacidades iguais de produção, que juntas formam um setor, também denominado de *shop floor*, em que a peça pode ser destinada a qualquer máquina que esteja disponível nesse setor.

A crescente complexidade dos sistemas de manufatura é evidente nos sistemas atuais, uma vez que estes não estão mais dispostos apenas em linhas simples e fáceis de manipular. De acordo com Jacobs e Chase (2018), os sistemas de manufatura modernos consistem de muitas operações que ocorrem de modo não ordenado e não-linear. Nesses casos, a modelagem por meio da simulação torna-se um dos mais eficientes meios de estudo desses sistemas. A maioria dos sistemas atuais de manufatura são complexos demais para serem resolvidos analiticamente. Porém, estes modelos podem ser melhor analisados por meio de simulação, tornando o problema muito mais simples. Isso acontece pois, na simulação, um *software* é usado para

relacionar as mais diversas variáveis de um modelo e os dados são coletados para estimar as verdadeiras características desejadas no mesmo.

Segundo Chwif e Medina (2010), a simulação é uma poderosa ferramenta em que um modelo de simulação pode capturar características de natureza dinâmica e aleatória com mais precisão, tentando reproduzir, em um computador, o mesmo comportamento que o sistema apresentaria quando submetido às mesmas condições no mundo real. Essa dinâmica proporcionada pela simulação é muito importante para a classe de problemas de SP em que os sistemas são dinâmicos e quando novos eventos podem acontecer a qualquer momento durante a execução da produção programada.

Este trabalho buscou estudar o uso de simulação no SP, analisando as características e dificuldades apresentadas para solução dos referidos problemas. Por meio de uma análise da literatura de SP, é possível notar que a simulação é uma das importantes ferramentas para solucionar esse tipo de problema, pois considera a análise de todo o sistema. No entanto, a Simulação a Eventos Discretos (SED) ainda encontra obstáculos a serem superados. Klemmt *et al.* (2011) utilizam simulação para testar diferentes abordagens para resolver problemas de SP e comparar os resultados dessas estratégias. Eles afirmaram que as técnicas heurísticas isoladas têm menor impacto do que quando usadas em conjunto com a simulação e que trabalhos futuros devem explorar conexões entre diferentes técnicas.

Alguns problemas simples de SP podem ser resolvidos analiticamente ou por meio de *softwares* específicos para este fim. Porém, segundo Slack *et al.* (2009), o número de programas de produção possíveis aumenta à medida que o número de processos aumenta, visto que o número de combinações possíveis é $(n!)^m$, sendo “n” o número de trabalhos a serem realizados e “m” o número de máquinas. Percebe-se que esse número cresce consideravelmente quando o problema se torna mais complexo. Na prática não se tem apenas duas máquinas em série ou apenas algumas em paralelo, o que faria possível uma solução analítica, e sim sistemas de produção mais elaborados, sendo quase sempre complexos demais para a resolução analítica ou para a utilização de *softwares* para este fim.

Segundo Lustosa *et al.* (2008) a simulação é bastante utilizada para resolução de problemas mais complexos de SP, principalmente àqueles do tipo JSF, pois esse sistema não possibilita o encontro de uma sequência ótima de produção, mas sim uma boa sequência para determinado objetivo, baseado em uma heurística. Geralmente são sistemas muito complexos onde cada ordem possui um roteiro diferente com tempos de fabricação diferentes e em quantidades distintas. Os mesmos autores citaram que uma alternativa é simular o desempenho do sistema sob diversas regras de prioridade diferentes.

Sharma e Jain (2015) provaram através da simulação que, estatisticamente, as regras de SP são muito relevantes no desempenho dos sistemas JSF. Concluíram ainda que essas melhorias trazidas pelo SP elevam a satisfação dos clientes, devido ao aumento na rapidez e redução de atrasos.

Apesar de existir uma tendência de linearização dos SJSF para um sistema híbrido denominado célula, que é intermediário entre os sistemas *flow shop* e SJSF, segundo Muda e Hendry (2003) o SJSF é ideal nos casos de produção por encomenda de uma grande diversidade de produtos, em pouca quantidade cada e de clientes distintos, não justificando a alocação de recursos específicos para um tipo de produto. Portanto, esse sistema não perderá sua vocação produtiva, que é voltada ao alto índice de flexibilidade, tanto em termos de *mix* de produção quanto para volume produzido.

Segundo Shen *et al.* (2006), a manufatura moderna pede grande flexibilidade, conseqüentemente, estudos de PS em FJSS têm sido necessários para atender à nova geração de requisitos de sistemas de manufatura.

De acordo com Biagio (2015), o arranjo físico por processo é o mais comum que existe, devido ao fato de as micro e pequenas empresas muitas vezes não possuem recursos para dedicarem a um produto específico, além de produzirem em pequenas quantidades, deixando os grandes volumes para empresas de grande porte. Foi constatado que o tipo de configuração JSF é o mais clássico e o mais frequente nas organizações e empresas, apesar de ser um sistema de difícil gestão.

Amiri *et al.* (2018) também afirmam que o SJSF é um dos mais comuns sistemas de produção do mundo e o SP nesses sistemas é de extrema importância para a indústria. Shen *et al.* (2018) afirmam que devido à ampla aplicabilidade dos SJSF para o mundo real, esse sistema é um dos mais populares entre os problemas de SP.

1.2. Problematização da pesquisa

O SP ainda é uma área com grandes problemas a serem resolvidos devido a sua complexidade e aplicabilidade nos mais variados sistemas de produção. Dentro deste contexto, a simulação entra como uma grande aliada para problemas com maiores quantidades de máquinas e fluxos mais complexos de produção (LUSTOSA *et al.*, 2008).

O sequenciamento de uma linha de produção no mundo real pode ser altamente complexo, ainda mais quando se sequenciam ordens com tempos de preparação dependentes,

restrições de processo e falhas que podem afetar a possibilidade de alcançar a meta de produção. Frantzén *et al.* (2011) afirmam que o programador, além de gerar sequências que são robustas e adaptáveis, deve ser capaz de re-sequenciar as ordens em tempo real, a fim de lidar com as incertezas do sistema de forma eficaz. Para contornar esses problemas, estes autores usaram a Simulação Baseada em Agentes (SBA) para o SP em tempo real de uma linha de usinagem complexa em um fabricante de automóveis.

A robustez de um modelo acontece quando este modelo pode reagir a eventos inesperados que comumente acontecem do cotidiano da produção, tornando o mesmo mais realista. O oposto de um modelo robusto é um sensível, em que não são previstas situações inesperadas no mesmo. Sendo assim, quando acontece algum problema no sistema real, este tipo de modelo perde a capacidade de prever resultados confiáveis para os gestores de produção, pois estes não refletem a realidade (FRANTZÉN *et al.*, 2011; THORNBLAD *et al.*, 2015; LOU *et al.*, 2012).

A indústria atualmente exige um *mix* de produção cada vez maior e com tamanhos de lotes menores a fim de responder às necessidades dos consumidores. Com isso, a conversão do planejamento de produção está cada vez mais complexa, e pesquisas sobre programação da produção têm sido requisitadas (SUDO e MATSUDA, 2013). Segundo estes autores, uma vez que o espaço de solução é muito grande, algoritmos matemáticos para SP em SJSF tornaram-se impotentes para obter uma solução ideal. Estes autores afirmam que é essencial construir um plano de produção prático adotando soluções quase ótimas, que são boas soluções, mas não é garantido que sejam ótimas. Neste sentido, sistemas de manufatura autônomos utilizando multi-agentes têm sido propostos. Nestes sistemas, um plano de produção é gerado de forma autônoma e dinâmica, usando comunicação e negociação entre os agentes que correspondem aos componentes da fábrica.

De um modo geral, as abordagens de programação baseada em simulação são derivadas de métodos que usam regras de prioridade (KIRAN, 1998). Na programação baseada em simulações, várias regras de SP podem ser usadas em diferentes fases para a tomada de decisões. Basicamente, uma regra de despacho é uma regra que dá prioridade a uma ordem de produção dentre outras em um estágio específico do processo.

Kulkarni e Venkateswaran (2016) chegaram à conclusão em seu trabalho que as grandes vantagens do uso da SBA para o SP são a criação de cenários mais realistas como aleatoriedades de várias formas e a natureza dinâmica do sistema poder ser incorporada ao modelo. Os mesmos autores propuseram que trabalhos futuros na área considerem tempos de processos estocásticos,

quebra de máquinas e outros problemas de chão de fábrica, a fim de obter um programa de produção mais robusto que funcione bem ante às irregularidades do sistema.

Xiong *et al.* (2017) citam que futuros trabalhos deveriam desenvolver regras que incluam mais informações sobre o sistema de produção a ser programado, e mais tipos de combinações dessa informação. Isso ajudaria a superar a curta visão das regras de sequenciamento e ajudaria a otimizar vários objetivos ao mesmo tempo. Os autores também propuseram que o nível de utilização das máquinas deveria ser considerado como um parâmetro do modelo, com o objetivo de avaliar a performance das regras de prioridade sob condições de alta ou baixa utilização. Para finalizar, eles citaram que outras medições de desempenho devem ser consideradas para avaliar a efetividade das regras de sequenciamento em todas elas, ou seja, fazendo o SP buscar resultados multi-objetivos quando possível, buscando melhorar a performance do sistema como um todo e tornar o SP mais robusto.

Vinod e Sridharan (2011) fazem o SP em uma empresa que usa o ambiente JSF, usando diferentes técnicas de atribuição de datas devidas para os pedidos dos clientes, além de várias lógicas de sequenciamento diferentes. Ao usar a simulação para testar essas técnicas, os autores chegam a várias conclusões sobre quais técnicas de datas devidas e sequenciamento são melhores para cada tipo de objetivo a ser atingido. Para cada objetivo de produção, como tempo médio de atraso e tempo médio de fluxo, existe uma combinação de método de atribuição de data devida e lógica de sequenciamento melhor para cada caso. No entanto, os autores constatam que há uma necessidade de testar essas diferentes lógicas para ambientes de JSF onde há padrões variados de chegadas de pedidos e interrupções no sistema como paradas de máquinas.

Como é possível notar pelos trabalhos supracitados, existem avanços na utilização da simulação aplicada a problemas de SP. Entretanto, ainda existem dificuldades e limitações a serem vencidas. Alguns trabalhos recentes já mostram uma tendência da utilização da SBA integrada a SED, com o intuito de se resolver tais limitações e dificuldades.

Desta forma, esta pesquisa procura utilizar a SBA combinada a SED, formando uma simulação denominada Simulação Híbrida, para trabalhar com problemas de sequenciamento, especificamente problemas do tipo JSF, buscando sempre avanços nesses tipos de problemas através dessa relativa nova abordagem da simulação.

1.3. Objetivos

O objetivo principal desta pesquisa é desenvolver um modelo de Simulação Híbrida para o SP, buscando deixar o mesmo mais robusto e menos suscetível a problemas, tais como: quebra de máquinas, tempos estocásticos de processo e necessidades de balanceamento de máquinas entre os setores. Para isso, o modelo inclui, nas regras de SP, mais informações do sistema produtivo a ser programado, buscando atingir mais metas de produção ao mesmo tempo e propondo soluções em termos de modelagem.

Segmentando este objetivo, tem-se como objetivos específicos:

- Avaliar a vantagem que os agentes possuem de conter, em sua programação interna, a lógica de decisão que pode ser usada em qualquer momento no modelo, além de responder a qualquer evento novo que aconteça no mesmo;
- Melhorar a interação entre o modelo e seus usuários, buscando fazer o modelo pronto atender não somente aos especialistas em simulação, mas também a usuários menos preparados, buscando facilitar as modificações no mesmo por meio de uma planilha de interface do *software Excel*®. A criação dessa ferramenta de interface entre modelo e modelador pode ser vista como uma ferramenta de interação para tratar de problemas de sequenciamento. Essa interação faz parte dos conceitos de modelos *Digital Twins* usados na Indústria 4.0, que é considerada a quarta revolução na indústria.

1.4. Contribuição científica

Esta pesquisa propõe soluções para algumas das dificuldades e limitações existentes no sequenciamento JSF relatadas anteriormente. Estas estão relacionadas a fragilidades dos modelos tradicionais de sequenciamento perante problemas cotidianos, como: quebra de máquinas e necessidades de mudanças no plano de produção. Quando tais eventos acontecem o SP perde sua eficiência, prejudicando seus resultados. Vários são os autores que constataam a robustez como sendo a principal demanda dos modelos atuais de sequenciamento, estes serão mais detalhados durante a revisão da literatura desta tese (BURDETT *et al.*, 2019; XIONG e FU, 2018; NING e JIM, 2018; XIONG *et al.*, 2017; KULKARNI e VENKATESWARAN, 2016; THORNBLAD *et al.*, 2015; LOU *et al.*, 2012; RAJABINASAB e MANSOUR, 2011; FRANTZÉN *et al.*, 2011).

Por isso, será criado um modelo genérico que funcione para diversas situações dentro desse sistema, que realize o SP de maneira mais abrangente e menos sensível a problemas

cotidianos, ou seja, mais robusto. Espera-se demonstrar que a Simulação Híbrida chega como um avanço no tratamento desses tipos de problemas. O agente deverá se provar eficiente ao realizar um SP mais robusto e com foco mais abrangente no que tange às metas de produção. Burdett *et al.* (2019) comentam da importância de se construir um modelo genérico que funcione para diversas situações.

Xiong e Fu (2018) citam que, como os agentes reduzem a complexidade computacional, usam parâmetros simples e seus modelos têm baixa complexidade, os modelos que utilizam agentes são mais propícios a simulação em tempo real devido a fácil inserção e modificação de dados. Isso facilita a criação desse modelo genérico para utilizar dados atualizados de maneira fácil.

Adicionalmente, o modelo construído nesta pesquisa tem algumas características que o inserem dentro do contexto de Gêmeo Digital ou *Digital Twin* em inglês, que é uma cópia virtualizada de processos reais. Este termo significa que, ao mesmo tempo em que alterações no produto ou processo são desenvolvidas no plano físico, seu modelo correspondente é modificado em tempo real em todos os seus aspectos. O conceito também engloba que esses novos modelos devem ser mais dinâmicos no sentido de absorver modificações ao longo do ciclo de vida dos produtos, tornando o modelo mais fiel às mudanças que ocorrem na realidade (RODIČ, 2017). Este novo paradigma da modelagem e simulação encaixa-se melhor dentro do conceito dos modelos *Digital Twin*. Tal conceito, devido as suas características, insere-se bem dentro do contexto da Indústria 4.0.

Rodič (2017) ainda cita que a simulação na Indústria 4.0 requer a modelagem usando o conceito de fábrica virtual e o uso de inteligência artificial para ajustar de forma autônoma os sistemas de produção. Trata-se de uma construção e modificação automatizada de modelos, via algoritmos ou modificados via interfaces amigáveis. Entretanto, existe um longo caminho para tornar o conceito de gêmeo digital uma realidade em todos os aspectos mencionados. Embora o modelo desta tese não seja focado nisso, ele possui uma característica mencionada que é utilizada dentro desse conceito: a integração.

A simulação tradicional usa modelos chamados de *Stand-Alone*, com escopo e tempo de vida limitado. A tendência da simulação é mudar de uma situação de análises e otimizações pontuais para uma integração de modelos em que o uso possa ser recorrente. Os modelos tradicionais somente são acessíveis por especialistas em simulação. Estes deverão ser substituídos por modelos que possam ser conectados a outras fontes de dados e que possam ser controlados e modificados facilmente, por meio de uma interface amigável (BARLAS e HEAVEY, 2016). Entre as principais mudanças para a simulação deixar o status de *Stand-Alone*

para o *Digital Twin* está a construção e modificação de modelos automatizada, usando banco de dados que abastecem os modelos, podendo possuir alteração de lógicas automatizadas.

Esta pesquisa faz isso na medida em que estabelece uma conexão do modelo simulado com uma planilha em *Excel*, que será usada para colocar dados de entrada, como o tempo que cada peça passará em cada máquina e o roteiro de máquinas que essa peça deverá percorrer no processo de fabricação, bem como a quantidade de pedidos e a quantidade a ser produzida em cada pedido. A quantidade de máquinas em cada setor produtivo também pode ser facilmente modificada dentro deste modelo. Todas estas características reunidas fazem o modelo ter uma interface mais amigável e aumentam a integração do modelo com o usuário, podendo ser modificado facilmente e tornando-o mais genérico, para ser usado em diferentes situações dentro do SJSF. É importante frisar que estas características não fazem do modelo um Gêmeo Digital, ele apenas possui a característica de integração modelo e usuário, que é uma das exigências de tais modelos.

O modelo criado nesta pesquisa realizará sua produção procurando:

- Sequenciar a produção em um sistema com *layout* funcional;
- Utilizar a produção empurrada;
- Utilizar o carregamento infinito;
- Produzir sob encomenda usando a estratégia *make to order*;
- Usar a programação da produção para frente dentro do modelo;
- Buscar usar lógicas diferentes de sequenciamento;
- Usar como variáveis de resposta o *makespan* e utilizações das máquinas;
- Fazer com que o agente use interações com outros agentes buscando detectar problemas no chão de fábrica;
- Reagir diante destes problemas;
- Buscar re-sequenciamento em tempo real quando necessário.

1.5. Estrutura da tese

Esta tese está estruturada em mais sete capítulos. O Capítulo 1 apresentou a contextualização e introdução desta pesquisa, além de apontar os objetivos a serem alcançados nesta tese.

O Capítulo 2 apresenta uma revisão sistemática de literatura, realizando uma análise dos artigos encontrados em periódicos internacionais.

O Capítulo 3 apresenta a fundamentação teórica que apoia esta pesquisa. Neste capítulo, procura-se descrever o que há de mais recente na literatura sobre SBA, SP e SED. Para tal, uma revisão de literatura foi realizada, contemplando, principalmente, periódicos internacionais e nacionais, artigos de congressos, livros, teses e dissertações.

O Capítulo 4 apresenta o método de pesquisa Modelagem e Simulação, método este que guiou a condução desta pesquisa. Ao longo do capítulo, são apresentados os principais conceitos relacionados a este método de pesquisa, as etapas que envolvem a elaboração de um modelo de simulação e a classificação desta pesquisa segundo Bertrand e Fransoo (2002).

O Capítulo 5 descreve a pesquisa em geral, a construção do modelo, o mapeamento do processo, as lógicas de sequenciamento propostas utilizadas pelo agente, a planilha de dados de entrada utilizada pelo modelo, entre outros detalhes da construção do mesmo.

O Capítulo 6 valida estatisticamente os resultados do modelo simulado com o gráfico de Gantt, demonstrando a eficácia do modelo.

O Capítulo 7 faz um teste “*what if*”, comparando estatisticamente o SP no modelo com características mais avançadas de programação, a serem explicadas nesta tese.

Por fim, o Capítulo 8 encerra a tese apresentando as principais conclusões obtidas e propõe sugestões para trabalhos futuros.

2. REVISÃO SISTEMÁTICA

Este capítulo tem o objetivo de apresentar os problemas e soluções de SP que dão o suporte teórico a esta tese, com base em recentes artigos de periódicos importantes da área. Serão apresentados os principais problemas recentes do SP e as diferentes estratégias para resolvê-los, utilizando ferramentas diversas.

Foi feita uma revisão sistemática da literatura que servirá de base para a condução deste trabalho e os resultados dessa revisão serão apresentados no primeiro item deste capítulo. Os detalhes dos artigos encontrados ficam no último item do mesmo.

2.1. Revisão sistemática da literatura

A revisão sistemática da literatura segue o modelo proposto por Sampaio e Mancini (2007), que apresentam um fluxograma detalhado sobre como proceder com esse processo inicial da pesquisa, mostrado na Figura 2.1.

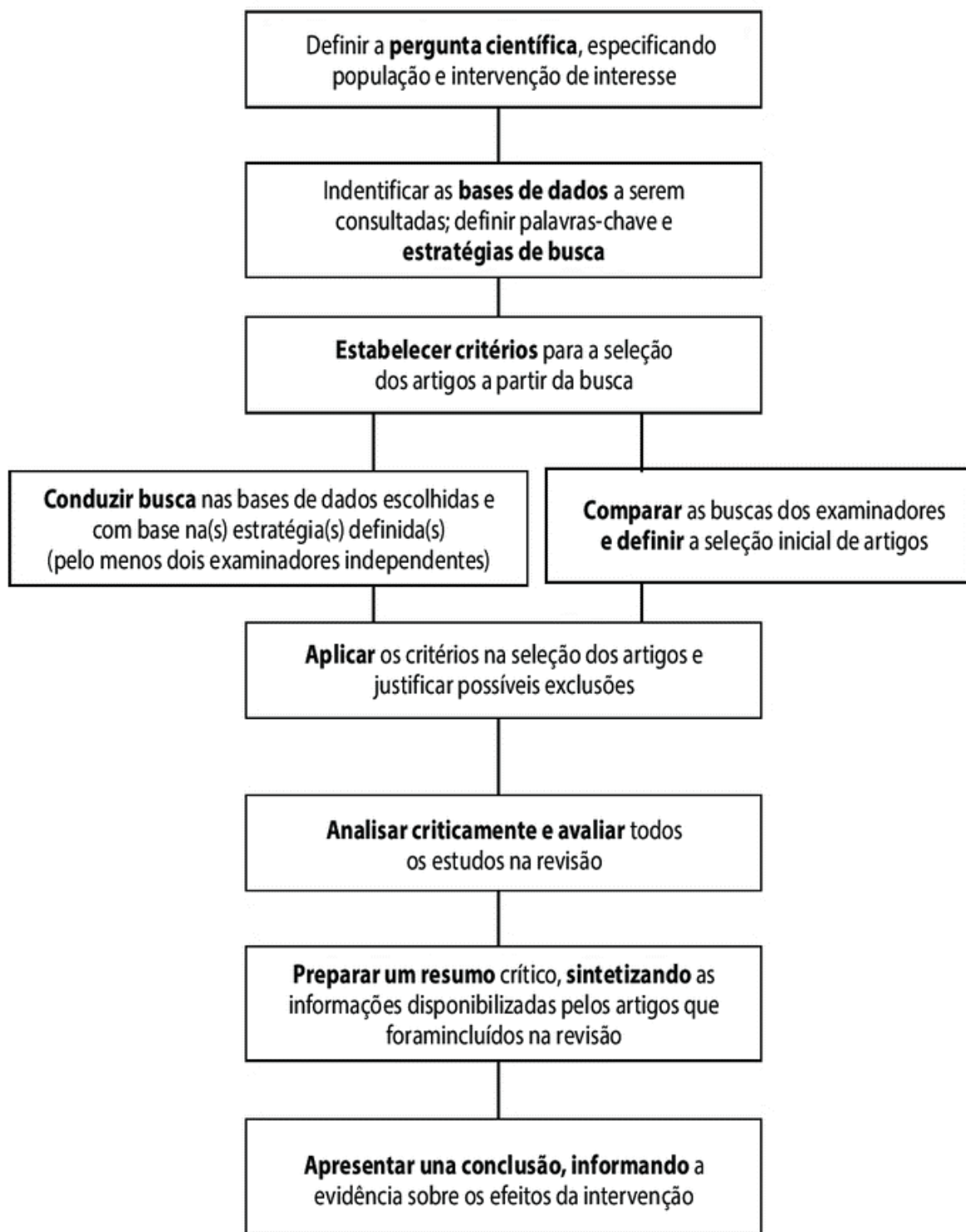


Figura 2.1 - Fluxograma para revisão sistemática da literatura.

Fonte: Sampaio e Mancini (2007)

O problema abordado por esta tese consiste em responder à seguinte pergunta científica: Quais os avanços em modelagem podem ser obtidos utilizando agentes em modelos de simulação voltados ao SP?

A revisão sistemática foi feita usando a base de dados do Scopus®. Essa escolha é justificada porque a Scopus® é uma das maiores e mais abrangentes base de dados multidisciplinares, incluindo aproximadamente 15.000 *journals* e mais de 4.000 editores (FRANCESCHINI *et al.*, 2014; JAHANGIRIAN *et al.*, 2010).

Foram adotados alguns critérios de busca para que o resultado da pesquisa na base de dados gere apenas artigos relevantes para este trabalho.

Os termos para procura utilizados nos títulos, resumos e palavras-chave foram: (*Flexible Job Shop*) E (*Scheduling OU Sequence*) E (*Simulation*). Essas foram as palavras que mais se adequaram a pergunta científica citada no começo deste item.

Os filtros adotados para a seleção da base teórica deste trabalho foram:

- Somente artigos;
- Artigos em inglês;
- Tipo de fonte: *Journals*;
- Artigos publicados nos últimos 5 anos;
- Artigos dentro das áreas: *Engineering, Computer Science, Decision Sciences*.

A busca com base nos termos e filtros utilizados resultou em 47 artigos. Sampaio e Mancini (2007) citam que o critério de leitura dos resumos é suficiente, nesta fase, para fazer a seleção dos artigos que serão realmente lidos por completo para a realização do trabalho. Todos os resumos foram lidos a fim de se selecionar os principais entre eles, o que resultou em dezesseis artigos:

- *A Simulation-Based Study of Dispatching Rules in a Dynamic Job Shop Scheduling Problem with Batch Release and Extended Technical Precedence Constraints* (XIONG *et al.*, 2017);
- *A local search genetic algorithm for the job shop scheduling problem with intelligent agents* (ASADZADEH, 2015);
- *A model for manufacturing scheduling optimization through learning intelligent products* (BOUAZZA *et al.*, 2015);
- *Multi-objective simulation optimization for uncertain resource assignment and job sequence in automated flexible job shop* (AMIRI *et al.*, 2019);
- *Effect of routing flexibility and sequencing rules on performance of stochastic flexible job shop manufacturing system with setup times: Simulation approach* (SHARMA e JAIN, 2017);

- *Optimization of dynamic and multi-objective flexible job-shop scheduling based on parallel hybrid algorithm* (YANG e GAO, 2018);
- *A new immune multi-agent system for the flexible job shop scheduling problem* (XIONG e FU, 2018);
- *A novel dynamic scheduling strategy for solving flexible job-shop problems* (NING et al., 2016);
- *A cloud based improved method for multi-objective flexible job-shop scheduling problem* (NING and JIM, 2018);
- *A memetic algorithm for multi-objective flexible job-shop problem with worker flexibility* (GONG et al., 2018);
- *A flexible job shop scheduling approach with operators for coal export terminals* (BURDETT et al., 2019);
- *Solving the flexible job shop scheduling problem with sequence-dependent setup times* (SHEN et al., 2018);
- *A novel hybrid method for solving flexible job-shop scheduling problem* (NING et al., 2016);
- *Solving the flexible job-shop just-in-time scheduling problem with quadratic earliness and tardiness costs* (REY et al., 2015);
- *Multiobjective joint optimization of production scheduling and maintenance planning in the flexible job-shop problem* (YE e MA, 2015);
- *Scheduling optimization of a stochastic flexible job-shop system with time-varying machine failure rate* (MOKHTARI et al., 2015).

Uma análise gráfica, mostrada na Figura 2.2, feita na base de dados do Scopus® com as palavras-chave citadas anteriormente, mostra a quantidade de artigos publicados anualmente na área, que serão de grande relevância para o conteúdo deste trabalho. Nota-se uma crescente tendência nas publicações da área, sendo que esta revisão foi realizada quando o ano de 2019 ainda estava no início. Para se obter uma análise mais completa, os dados foram considerados sem a aplicação dos filtros.

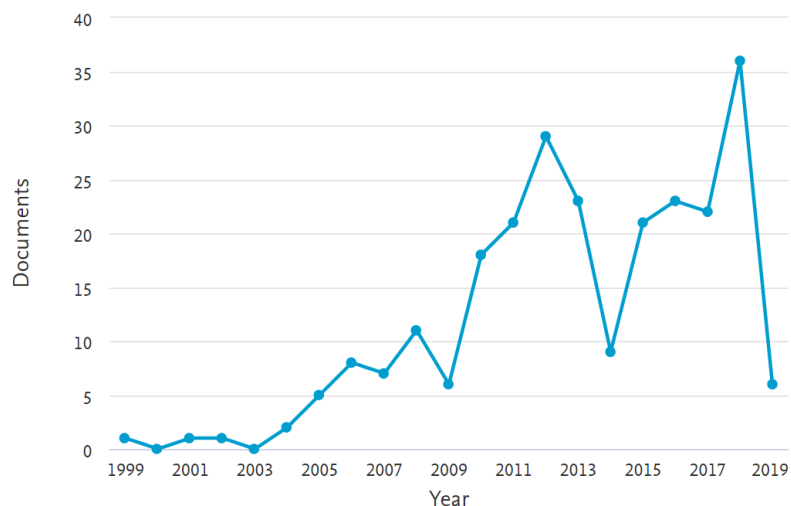


Figura 2.2 - Artigos publicados por ano.

Fonte: Scopus® (2019)

A Figura 2.3 mostra a distribuição de publicações desses artigos nos países, sendo possível notar que a China foi o país que mais abordou o assunto. Isso se deve as agências financiadoras do país que estão investindo nesta área de pesquisa, segundo dados dos próprios relatórios do Scopus®.

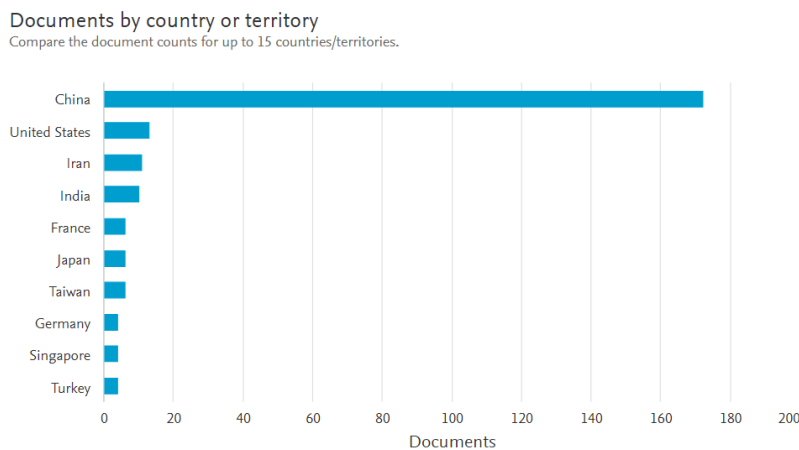


Figura 2.3 - Artigos publicados por país.

Fonte: Scopus® (2019)

A síntese das informações importantes utilizadas por esta tese originada destes artigos está disposta no próximo item. Além dos artigos oriundos desta revisão sistemática, o referido item também aborda alguns outros artigos encontrados aleatoriamente durante o processo de pesquisa.

Concluindo, apesar de serem apresentados vários artigos para a fundamentação teórica que se segue, a revisão sistemática se mostra importante pois apresenta uma forma organizada

de buscar exatamente o que se procura, ou seja, os artigos mais importantes para o trabalho que se pretende fazer, usando-os como base para a pesquisa desejada. A revisão também apresenta necessidades de pesquisa dentro do assunto, o que já foi relatado no Capítulo 1.

2.2. Casos na literatura de SP em SJSF

Este item tem o objetivo de sintetizar o estudo de trabalhos recentes e relevantes que usam agentes para resolver problemas de SP em sistemas de produção do tipo JSF. Muitas das ideias desta tese foram tiradas destes trabalhos, buscando-se resolver ou, pelo menos, melhorar os *gaps* encontrados na revisão sistemática da literatura de sequenciamento e durante todo o processo de pesquisa. As principais necessidades encontradas foram as de se realizar um sequenciamento mais robusto perante problemas, que busque atingir mais objetivos de produção ao mesmo tempo e que utilize tempos estocásticos nos modelos. Com o propósito de satisfazer estas necessidades, foram descritos abaixo os principais e mais importantes artigos encontrados para a realização deste trabalho.

O primeiro deles é o trabalho de Lou *et al.* (2010) que usa multi-agentes, sendo que estes interagem entre si para encontrar a melhor solução para o problema de sequenciamento. Eles usam agentes responsáveis pela tarefa, que contêm as restrições de procedência de cada ordem de produção, além dos agentes máquinas e recursos que contêm em si as informações das restrições de capacidade. Por meio de negociações entre os agentes, o modelo chega ao sequenciamento de forma descentralizada. Esse método se mostrou útil para atravessar problemas inesperados do chão de fábrica, pois os agentes são autônomos e podem tomar decisões locais, como quando a máquina quebra e manda as informações para o agente de tarefa, e este transfere a ordem de produção para outra máquina disponível.

Parthanadee e Buddhakulsomsiri (2010) utilizaram a SBA para programar a produção em uma indústria alimentícia, onde os insumos têm vencimento para serem usados. Os autores usaram o modelo para sequenciar, em tempo real, a produção, usando regras de sequenciamento simples, mas eficazes. Eles escolheram, dentro da simulação, a regra de sequenciamento mais eficaz entre várias outras utilizadas, com base em uma comparação estatística dos resultados baseado nos objetivos de produção, sendo eles o tempo médio de fluxo e tempo médio de atraso. Os resultados atingidos pelos autores demonstraram a eficácia do uso da simulação para regras de sequenciamento.

Sistemas de manufatura utilizando agentes autônomos têm sido propostos, como o trabalho de Atighehchian e Sepehri (2013), que realiza um SP dinâmico em uma única máquina tentando capturar características do ambiente para atuar nos pontos de decisão. Os autores fazem o SP mais próximo da realidade. Os modelos baseados em agentes podem executar o agendamento dinâmico porque os agentes podem executar ações inteligentes com base em sua lógica programada.

O trabalho de Rajabinasab e Mansour (2011) também usa agentes tarefa e agentes máquina, além do sistema de produção que é flexível em relação a rotas de processo. Em cada ponto do processo, a próxima máquina a ser utilizada é escolhida por meio do método que considera o tempo restante de processo e o tempo de folga das máquinas, chamado método baseado em ferormônios. Esse método de sequenciamento é baseado na troca de informações entre eles. Seus resultados são comparados com regras heurísticas tradicionais para o sequenciamento. Há também o agente administrador que tem acesso às informações de todos os agentes, como restrições dos mais diversos tipos. Com essas informações, ele analisa a viabilidade de produção das ordens recebidas. De início já existem os agentes máquina e o agente administrador, mas os agentes tarefa são inseridos e deletados do sistema de acordo com a entrada e saída de pedidos de produção. Os autores concluíram que o método baseado em agentes é mais eficiente e robusto em relação as heurísticas.

Thornblad *et al.* (2015) também se preocuparam em realizar um SP mais robusto e com objetivo de alcançar mais respostas. Eles desenvolveram uma solução prática para sequenciar utilizando dados reais que são aceitáveis na prática. Frantzén *et al.* (2011) afirmaram que o programador deve gerar modelos sequenciadores que sejam robustos e funcionem bem diante de situações diferentes.

Reisi-Nafchi e Moslehi (2015) utilizaram múltiplos agentes para lidar com diferentes exigências de clientes, visando aceitar mais pedidos destes. Os agentes trabalham com diferentes funções objetivas, isso porque os clientes também têm desejos diferentes. Dessa maneira os autores conseguiram atingir mais resultados ao mesmo tempo.

O trabalho de Weng *et al.* (2008) vai mais adiante e usa a programação de agentes não só para o sequenciamento, como também para estimar a data de entrega ao cliente e o momento de liberar o pedido para o chão de fábrica. A importância desse sistema ocorre em empresas que usam a estratégia de produção de fazer contra pedido, pois a estimativa eficaz do tempo de entrega ao cliente é um fator ganhador de pedidos. O momento de liberar o pedido tem grande influência no estoque em processo. A empresa usa um sistema JSF de produção, que citam ser o mais eficiente para empresas do tipo *make to order*, pois produz com mais flexibilidade e

customização do produto. Para atingir todos os objetivos, os autores usaram quatro tipos de agentes: o agente da entrada da ordem que define a data de entrega, o agente que libera a ordem para produção, o agente que calcula a rota e a sequência a ser produzida, além do agente que passa as informações aos outros agentes.

Em outro trabalho, Lou *et al.* (2012) usam agentes para fazer o sequenciamento em duas fases: uma fase proativa onde o sequenciamento é feito independente de eventuais problemas futuros; e outra fase reativa, em que é feito um re-sequenciamento frente às alterações ou perturbações no processo produtivo. Eles consideram em seu trabalho quebras de máquinas e ordens urgentes para serem terminadas. Esse re-sequenciamento citado é mais uma reparação da ordem gerada pelo sequenciamento proativo do que um sequenciamento totalmente novo. Isso evita a falta de continuidade no plano de produção, não desorganiza a entrega de matéria prima na linha e mantém a mão de obra nos locais planejados. O objetivo é fazer o novo sequenciamento resolver os problemas, porém ficando o mais próximo possível da ordem original. Os autores afirmaram que não buscam soluções ótimas, pois essas não existem na prática, mas sim soluções viáveis e robustas em relação às incertezas. Os autores também citaram a necessidade e a importância de seguir essa linha de pesquisa buscando um sistema genérico de sequenciamento em SJSF, que seja eficiente e atenda aos mais variados tipos de incertezas e problemas de chão de fábrica, que é algo parecido com o que esta tese pretende fazer.

Bouazza *et al.* (2015) usaram agentes com inteligência de aprendizado, ou seja, agentes que aprendem com base na experiência. A decisão do agente neste artigo é simples, consiste em escolher a melhor regra heurística de sequenciamento para cada situação, sendo que as regras são limitadas ao *first in first out* (FIFO) e ao Menor Tempo de Processamento (MTP). Em cada caso, uma regra se mostra melhor que a outra. Isso é medido, pelo agente, por meio de uma resposta que vem em forma de uma função objetivo. Apesar da simplicidade, os autores provaram que o agente é eficiente e consegue escolher a melhor regra para cada caso.

O trabalho de Xiong *et al.* (2017) também foi importante para esta pesquisa pois citou que futuros trabalhos deveriam desenvolver regras que incluam mais informações sobre o sistema de produção a ser programado, e não somente das ordens em si, o que esta pesquisa procurou fazer. Os autores também propuseram trabalhar considerando a utilização das máquinas e sequenciamentos com mais objetivos simultaneamente, algo que também foi considerado nesta tese.

Amiri *et al.* (2018) fez o SP multi-objetivo utilizando um metamodelo gerado pela otimização via simulação. Metamodelos são funções matemáticas geradas pelos resultados do

modelo de simulação, incluindo várias variáveis de resposta. Eles testaram vários cenários experimentais através da SED e provam a eficiência do método, utilizando o *Data Envelopment Analysis* para atingir os melhores cenários. Os autores constatam que o SJSF é um dos mais comuns sistemas de produção do mundo e o SP nesses sistemas é de extrema importância. Eles tentaram maximizar a eficiência e minimizar o tempo de espera dos produtos. Contudo, os autores citaram que a inclusão de incertezas no processo e eventualidades que aproximariam o modelo de situações reais são impossíveis na modelagem matemática, onde várias simplificações devem ser feitas. Citam também que avanços na simulação podem resolver esses problemas, levando a resultados mais realistas. Frisaram que a questão da produtividade desses sistemas necessita de mais atenção.

Burdett *et al.* (2019) aplicaram meta heurísticas para o sequenciamento de ordens de carregamento de navios com base em prioridades de importância das ordens. Cada carregamento usa múltiplos recursos, o que dificulta o problema. Os autores geraram uma plataforma genérica de sequenciamento nestes tipos de operações e frisaram que essa é uma de suas principais contribuições, constatando a importância de um modelo que possa ser reutilizado em outras situações similares.

Shen *et al.* (2018) realizaram o SP em SJSF considerando *setup* dependente da sequência utilizada. O objetivo dos autores foi minimizar o *makespan*. Para isso, um algoritmo matemático foi usado para o SP, talvez por isso a solução não atingiu múltiplos objetivos. Contudo, o algoritmo se mostrou eficiente perante a muitos outros utilizados para comparação. Eles constatam que devido à importância e aplicabilidade do SP a problemas reais nos SJSF, o SP nesses sistemas são os mais populares entre os problemas da área de sequenciamento. Como proposições às pesquisas futuras, citam a necessidade de considerar múltiplas respostas produtivas para este tipo de problema.

Gong *et al.* (2018) consideraram a flexibilidade de habilidade e versatilidade do ser humano para tornar o SP em SJSF mais realista. Citaram que os sistemas flexíveis são mais realistas que os tradicionais *job shop*, pois nos flexíveis permite-se utilizar qualquer máquina disponível naquele setor, não sendo necessário esperar por uma máquina específica que foi programada anteriormente. Os autores concluíram que é mais interessante se ter uma combinação de operadores com alta flexibilidade e outros sem nenhuma, do que todos operadores com igual e média flexibilidade.

Mokhtari (2015) trouxe, em seu trabalho, a necessidade de se criar um modelo mais realista em SJSF que considere quebras de máquinas com taxas variáveis entre quebras. Também usou datas de entrega e tempos de processo aleatórios. A otimização via simulação é

utilizada para resolver o problema de SP. Contudo, há várias outras simplificações como eventualidades do cotidiano da manufatura que não são consideradas no problema. O autor assumiu que as taxas de quebra de máquinas e tempos de reparo assumiram distribuições exponenciais.

Ning *et al.* (2016) realizou o SP em SJFS com um algoritmo híbrido de SP multi-fases *particle swarm* em junções com outros algoritmos. Eles utilizaram uma simulação dinâmica também fazendo o re-sequenciamento quando encontram interferências, como trabalhos que se tornam urgentes e falta de matéria prima. Os autores chamaram essas interferências de fatores de re-sequenciamento. Foram feitas programações bastante complexas, mas que conseguiram resolver o problema. Em outro artigo, Ning *et al.* (2016) realizaram um trabalho bastante parecido no mesmo tipo de sistema, mas com outro algoritmo de SP, visando outras respostas produtivas. Mais tarde os mesmos autores avançaram seu trabalho em Ning e Jim (2018), evoluindo em termos de complexidade de algoritmos e buscando um SP multi-objetivo.

Sharma e Jain (2015) utilizaram a SED para avaliar o impacto da flexibilidade de rotas e das regras de SP na performance em modelos estocásticos de SJSF. Concluem que o nível de flexibilidade de rotas e das regras de SP são estatisticamente relevantes na performance desses sistemas. Citam que trabalhos futuros devem considerar perturbações no sistema, como quebras de máquinas e cancelamento de ordens.

Yang e Gao (2018) realizaram o SP visando minimizar o *makespan* e o custo de produção. Fazem isso usando um algoritmo paralelo híbrido de otimização que faz o re-sequenciamento visando minimizar os impactos negativos das interferências no processo. Eles utilizaram como distúrbios ao SP a inclusão e exclusão de tarefas, faltas de matéria-prima e mudanças nas prioridades das ordens. Contudo, a preocupação dos autores foi com o tempo de convergência e o número de iterações do algoritmo em relação a outros utilizados. Não há uma descrição detalhada do sistema produtivo nem das ordens de fabricação utilizadas. Existem mais trabalhos parecidos com este, utilizando algoritmos bastante complexos para achar a melhor sequência, buscando otimizar um objetivo, como o de Zhang *et al.* (2014) que buscaram minimizar o *makespan*. O problema com estes trabalhos é a demasiada complexidade para entendê-los e ainda mais para replicá-los a outras aplicações práticas.

Xiong e Fu (2018) utilizaram modelos multi-agentes para realizar o SP em SJSF. Frisaram que os agentes reduzem bastante a complexidade da programação, aumentam a flexibilidade dos modelos e os tornam mais robustos que os algoritmos meta-heurísticos de SP utilizados na área. Citaram que os modelos com agentes podem ser muito úteis com modelos

de simulações em tempo real. Isso se deve a facilidade de se mudar dados, aos parâmetros simples utilizados e à baixa complexidade computacional.

Ye e Ma (2015) trouxeram a preocupação de se fazer em conjunto as atividades de SP e o plano de manutenção das máquinas. Por isso, consideraram como respostas o *makespan* e custo de manutenção. Os autores utilizaram algoritmos genéticos para o SP e fizeram múltiplos experimentos com simulação para provar a validade das técnicas adotadas. Eles provaram a superioridade dos resultados quando se considera em conjunto a manutenção preventiva fixa e o SP, em vez de considerar e programar separadamente estas atividades.

Rey *et al.* (2015) consideraram um SJSF com algumas características *just in time* para aumentar o índice de satisfação dos clientes. Em sua pesquisa, o foco foi achar a hora ideal de soltar a ordem para a produção, atendendo a demanda em relação às datas devidas. Como distâncias longas entre términos de lotes e datas devidas são evitadas, os autores têm como meta minimizar a média e o desvio padrão dessa diferença. O trabalho, como muitos outros, também utiliza regras meta-heurísticas bastante complexas para o SP, combinando técnicas de algoritmos genéticos e otimização *particle swarm*. Contudo, as heurísticas responderam bem aos testes realizados com alguns exemplos teóricos.

A pesquisa desta tese utiliza algumas das estratégias utilizadas nos trabalhos mostrados neste item. O primeiro é usar agentes tarefa responsáveis pela origem e saída de cada ordem de produção. Ele deve conter informações sobre cada ordem, tais como: tempos em cada máquina, quais máquinas ela passará, rota a seguir entre os setores e quantidade de produtos em cada ordem. O agente tarefa é o que coleta informações da planilha de dados de entrada e as insere no modelo. Além disso, são utilizados os agentes máquina que contêm informações de restrições de capacidade, como número de máquinas em cada setor e disponibilidade de cada setor. Além disso, existe o agente administrador que tem acesso a informações de todos os agentes, como tempos de processo e rotas. Este último agente calcula a sequência de produção e implementa essa sequência no processo de fabricação. Como a necessidade citada por Lou *et al.* (2012) e Burdett *et al.* (2019), esta tese gera um modelo genérico de SP em SFJS que é eficiente na implementação das ordens, ainda que possam ser necessárias algumas adaptações no modelo. O modelo criado nesta tese utiliza os agentes recurso para transformar tempos determinísticos em estocásticos, através da entrada desses tempos dentro da distribuição triangular. Comumente, nas atividades de SP os tempos são determinísticos porque devem ser calculados de acordo com seu projeto de processo, pois antes de serem produzidos os tempos ainda não foram cronometrados, especificamente nos sistemas que usam a estratégia de

produção sob encomenda. Esta tese ainda usa duas medidas de saída para provar a eficiência do agente em um foco mais abrangente.

As vantagens do uso dos agentes no SP estão, basicamente, na capacidade de resequenciamento dos modelos que utilizam agentes e na relativa simplicidade de programação. Conforme já comentado, essa capacidade vem da mudança de estado dos agentes por meio da reação de eventos inesperados ou por tempos pré-determinados. A mudança de estado aciona uma nova lógica de SP durante o tempo real de simulação. Esta lógica gera uma nova sequência. Esta funciona mais como um conserto na sequência original, adaptando-a aos problemas que ocorreram. Tudo isso é o que gera robustez ao modelo. A robustez é entendida como o contrário da sensibilidade a problemas cotidianos, que podem afetar fortemente os resultados de um sequenciamento comum. A nova abordagem do SP deve considerar um modelo que se adeque mais as características reais do dia a dia, buscando uma confiança maior nos seus resultados. Outra vantagem atingida com isso é a capacidade dos agentes de mesclar regras de SP atingindo, assim, mais objetivos de produção ao mesmo tempo.

3. REVISÃO BIBLIOGRÁFICA

Este capítulo tem o objetivo de apresentar os conceitos que dão o suporte teórico a esta tese, com base em livros acadêmicos, artigos de periódicos e artigos de congressos da área.

O tema inicial da fundamentação teórica a ser discutido neste capítulo é a SED, apresentando seus conceitos e quando ser ou não utilizada. Assim, será apresentado uma revisão do processo de criação de um modelo conceitual, a preparação da coleta de dados de entrada, construção do modelo computacional, e os processos contínuos de validação, verificação e estabelecimento de credibilidade. Ou seja, apresentam-se os passos para se conduzir um modelo computacional.

Depois disso, estuda-se a SBA, mostrando suas características, definindo o conceito de agentes e explicando seus diversos tipos. Este capítulo trará um entendimento melhor sobre essa área da simulação, também demonstrando as vantagens desse tipo de simulação.

O terceiro assunto utilizado nessa tese é a Simulação Híbrida, que é a união dos dois tipos de simulação citados anteriormente em um mesmo modelo. Ou seja, são modelos de SED que usam agentes em sua modelagem.

Logo após a fundamentação sobre os tipos de simulação utilizados aqui, será apresentada o estudo sobre o planejamento e controle da produção, especialmente o sequenciamento, trazendo suas mais diversas lógicas e os objetivos de produção procurados por cada uma. Também serão expostos pontos positivos e negativos de cada lógica, a fim de se escolher as mais propícias e úteis para esta pesquisa.

3.1. Simulação a eventos discretos

O desenvolvimento contínuo da tecnologia dos computadores e dos *softwares* têm levado à evolução de programas de simulação ainda mais potentes e complexos, de acordo com Banks *et al.* (2009). Isso tem viabilizado ainda mais o uso da simulação, sendo que ficou muito fácil e barato a aquisição de computadores capazes de desempenhar essas tarefas, o que tem contribuído muito para o uso da simulação.

Robinson (2005) afirmou que, desde o princípio, a Simulação a Eventos Discretos (SED) foi desenvolvida de forma significativa, pois seus conceitos foram bem estruturados, porém esta teve que acompanhar a evolução de desenvolvimentos em computação. Logo, no início o principal problema não estava no conhecimento em SED, mas sim na evolução tecnológica da época.

Um fato constatado por Kelton *et al.* (2007), foi que a simulação utilizada atualmente iniciou-se na década de 1970, quando os computadores se tornaram mais rápidos e baratos. Sendo assim, o valor da simulação começava a ser descoberto pela indústria de um modo geral, ainda que no começo fosse utilizada apenas pelas empresas de grande porte, que estão sempre abertas à inovação.

Para Saraiva Junior *et al.* (2011), a simulação é uma representação de um processo ou sistema real, num dado período de tempo, que envolve a observação desta para fazer inferências relativas às características do processo. Esta se inicia com o desenvolvimento de um modelo que represente o sistema a ser investigado, com o máximo de fidelidade possível.

De acordo com Banks (1998), a simulação funciona como a criação de um sistema artificial a partir de um sistema real ao longo do tempo, procurando imitar esse processo ou sistema real. Ela envolve a observação deste sistema artificial para fazer inferências a respeito das características da operação do sistema real representado.

A simulação, de acordo com Harrel *et al.* (2004), é como um meio de reproduzir o comportamento de sistemas dinâmicos usando um modelo computacional. Funciona como a imitação de um sistema dinâmico com o objetivo de avaliar o desempenho do sistema.

Ela traz vantagens como a de visualizar o sistema, promover mudanças nos dados de entrada e responder a questões do tipo "o que aconteceria se...?", também conhecida como *what if*, reduzindo assim gastos desnecessários e tempo, pois se não existisse a simulação, as situações teriam que ser testadas na prática. Essa característica será de suma importância para a realização desta tese, pois o teste de mudanças na ordem de produção poderia afetar bruscamente um sistema real.

Segundo Chwif e Medina (2010) é muito importante destacar de que maneira a simulação não pode ser vista ou definida. Os tópicos abaixo mostram algumas dessas situações:

- Uma bola de cristal, antecipando eventos futuros, ela não tem esse poder. Contudo, a simulação pode prever o comportamento de um sistema, baseado em dados de entrada específicos, com certa confiança;
- Uma ferramenta de otimização, visto que a simulação é uma ferramenta de análise de cenários. A simulação por si só não é capaz de identificar uma solução ótima. Para isso, é necessário o uso de técnicas complementares à simulação;
- Uma substituta do pensamento humano no processo de tomada de decisão, porém ela pode fornecer dados para auxiliar os gestores na tomada de decisão;

- Uma técnica de último recurso. Para usuários mais fluentes na ferramenta, pode ser usada até mesmo para problemas mais simples, contrariando as opiniões de muitos profissionais, que acham que a simulação deveria ser utilizada apenas quando todas as técnicas possíveis falhassem;
- Uma ferramenta que solucionará todos os problemas existentes. A simulação possui uma classe de problemas bem específicos nos quais se adapta bem, como a manufatura com processos discretos, por exemplo.

De acordo com Law e Kelton (2007), uma das maiores vantagens da simulação é o uso da SED para mostrar aos tomadores de decisão uma visão ampla de todo o sistema, sendo que estes tomadores de decisão, que são em sua maioria gerentes e diretores, não tem tanto contato com o chão de fábrica como os engenheiros em início de carreira.

Segundo Chwif e Medina (2010), a simulação é uma poderosa ferramenta, porém a mesma não serve para resolver todos os problemas, já que possui uma classe de problemas específicos nos quais se adapta bem. Esses mesmos autores explicam que um modelo de simulação pode capturar características de natureza dinâmica e aleatória com mais precisão, tentando reproduzir, em um computador, o mesmo comportamento que o sistema apresentaria quando submetido às mesmas condições no mundo real.

Os modelos de simulação são tão completos que, segundo Baines *et al.* (2004), eles podem, em alguns casos, até mesmo considerar aspectos característicos do trabalho humano em sua programação, em contraposição a modelos que assumem o comportamento humano como o de uma máquina. O autor sugere que o efeito da idade do trabalhador e a variação no desempenho provocada de acordo com o horário do dia, podem ser inseridos em modelos de simulação através de algumas funções e parâmetros da programação.

Então pode-se perguntar o porquê de se usar a simulação, concluindo que para responder a esta pergunta deve-se conhecer as vantagens da simulação. Harrel *et al.* (2004) indicam que a simulação é uma grande alternativa em vez de usar a abordagem de tentativa e erro, que é cara, consome muito tempo e esforço, além de perturbar o sistema sob avaliação, uma vez que faz modificações no sistema real. Já a simulação traz uma maneira de se tomar as melhores decisões com uma certa confiança satisfatória (se o modelo for bem validado).

Dentre as várias vantagens possibilitadas pela simulação, apresentadas em Banks *et al.* (2009), algumas são importantes para este trabalho em específico:

- Uma das principais vantagens da simulação é que esta pode promover mudanças no sistema sem interrompê-lo ou modificá-lo, explorando novas políticas, procedimentos, regras de decisão e fluxos de informação;

- Hipóteses de como e porque certos fenômenos ocorrem podem ser testados;
- Pode ser empregada a análise de gargalo, que tem por objetivo descobrir onde está a obstrução do fluxo de informação, recursos ou materiais;
- As tradicionais perguntas “O que ocorre se...” podem ser respondidas sem que mude o sistema real.

Sturrock (2009) apresenta passos que são importantes para se conduzir metodicamente um projeto de simulação, que consiste em fases como:

- Definir objetivo do trabalho, identificando os interessados no projeto e como eles avaliarão o desempenho deste;
- Entendimento do sistema, especificação funcional de onde se quer chegar avaliando o nível de detalhe a ser observado e construído;
- Administração do projeto, evitando pular fases, mas fazendo-o no tempo requerido, pois um projeto prático feito após a decisão tomada não terá muito valor;
- Coletar dados de entrada, que é uma fase essencial, pois dados de entrada errados ou inexistentes na simulação fará com que essa se torne obsoleta;
- Construir e verificar o modelo, que serão fases interativas, pois a cada verificação, cada erro detectado, deve-se atualizar o modelo para que retrate a realidade;
- Validação do modelo, que deve ser feita observando-se a proximidade à realidade, e a validade do modelo para resolução do problema. O autor lembra que a validação perfeita não existe, pois o modelo perfeito é apenas o modelo real;
- A última fase é a experimentação, análise e apresentação dos resultados obtidos. Nela se apresentam os diferentes cenários testados e análises de cada um, fazendo proposições sobre as variáveis e o modelo que será bem conhecido pelo simulador.

Dentre todos os passos que são importantes para se conduzir metodicamente um projeto de simulação, se apresenta o último que é a experimentação, análise e apresentação dos resultados obtidos. Nela que se busca atingir os objetivos colocados no início do projeto, realizando os testes necessários e comparando resultados.

No que diz respeito ao uso da SED no SP, Kulkarni e Venkateswaran (2016) chegam a conclusão em seu trabalho que a SBA tem vantagens em relação a SED no SP, devido à possibilidade da primeira trabalhar com aleatoriedades no sistema, tornando-o mais real. Essas situações inesperadas que acontecem podem acionar lógicas de SP, pois os agentes podem responder a estas aleatoriedades, como quebras de máquinas, por exemplo.

Esta resposta a eventos inesperados é adquirida por meio da capacidade de re-sequenciamento dos modelos de SBA, que por sua vez provem da capacidade de resposta dos agentes autônomos. Isso acontece porque os estados dos agentes dentro do diagrama de estados podem mudar por diversos mecanismos, entre eles os eventos inesperados. Na SED as lógicas ficam fixas em determinados pontos do modelo, perdendo essa capacidade de re-sequenciamento, que segundo Frantzén *et al.* (2011) é uma necessidade dos modelos modernos que trabalham com SP, tornando o mesmo mais robusto.

Como exemplo do uso da SED no SP, Vinod e Sridharan (2011) usam a mesma testando diferentes técnicas de atribuição de datas devidas para os pedidos dos clientes, além de várias lógicas de sequenciamento diferentes. Apesar de atingir seus objetivos no trabalho, os autores afirmam que existe a necessidade de testar diferentes lógicas para o SP em modelos com padrões variados de chegadas de pedidos e interrupções no sistema, como paradas de máquinas.

Para contextualizar melhor o uso dos agentes na simulação, o item 3.2 apresentará em maiores detalhes a SBA, que posteriormente será integrada a SED, a fim de se obter avanços de robustez em termos de SP, o que é a intenção desta tese.

3.2. Simulação baseada em agentes

Este item apresenta uma breve fundamentação teórica do principal assunto a ser abordado nesse projeto, a SBA.

Partindo de uma abordagem puramente computacional, Braun e Rossac (2005) explicam que, no passado, com a evolução dos sistemas computacionais e da comunicação em rede, surge a necessidade de gerenciar tais sistemas de forma automática, uma vez que o controle puramente manual já não se mostrava eficiente. A fim de simplificar esse gerenciamento, novos conceitos de arquiteturas de sistemas nasceram gradativamente de acordo com as novas necessidades, surgindo uma abordagem que contemplasse a construção de aplicações distribuídas. Assim, uma das principais motivações foi descentralizar tarefas.

Dentro desse contexto, surgem os agentes móveis. Proposto na década de 80, esse conceito aparece de forma paralela em estudos das áreas de inteligência artificial. Esses agentes móveis são programas que podem ser despachados de um computador de origem, viajar entre servidores em uma rede até que se tornem hábeis para completar sua tarefa (CHESS *et al.*, 1995). Esses autores constataram que um agente móvel é capaz de interagir com outros, e com o ambiente no qual está inserido, a

partir da troca de informações ou mensagens. Além de ser capaz de implementar características como autonomia, capacidade de aprendizagem, reatividade e cooperação.

É possível notar que estes primeiros agentes que surgiram são muito parecidos com os agentes utilizados nos modelos de SBA, tendo características muito semelhantes, procurando simplificar tarefas que se tornavam muito complexas até então, buscando sempre a descentralização. Esses agentes móveis cumpriram sua função e realmente simplificaram tarefas no caso de gerenciamento de redes. Analogamente, uma das funções dos agentes na simulação pode ser a de simplificar tarefas complexas, como por exemplo, a de programar a produção através da simulação.

Giroux *et al.* (1996) constatam que a importância da utilização de agentes móveis em simulações deve-se principalmente a três características:

- Sua capacidade de transportar informações de forma autônoma, o que possibilita maior interação entre locais distintos de um sistema;
- A execução de tarefas em paralelo, o que torna possível manter, ao mesmo tempo, agentes em estados diferentes;
- A atividade social, a interação entre agentes e deles com o meio.

A modelagem baseada em agentes, segundo Sichman *et al.* (1998), se caracteriza por ser uma técnica de ampla aplicação em tomadas de decisões em grandes negócios, indústrias e até na área de saúde. Macal e North (2014) chegam a afirmar que essa modelagem é a terceira forma de produzir conhecimento científico, ficando atrás apenas dos processos de indução e dedução. Drogoul *et al.* (2003) afirmam que os grandes avanços computacionais e tecnológicos dos últimos tempos, tem viabilizado o uso dessa técnica nos mais variados campos de aplicação.

Os modelos de SBA baseiam-se no agente, sendo que cada um deles possui propriedades variáveis que são parâmetros que serão inseridos em suas regras de interação. Mesmo que vários agentes possuam a mesma regra de interação, se seus parâmetros forem diferentes eles terão comportamentos distintos, semelhante ao comportamento de sistemas com regras diferentes. Sichman *et al.* (1998) atestam que o todo é maior que a soma das partes individuais. Isso se deve ao fato de que uma simulação de indivíduos de baixo nível, pode gerar um sistema capaz de entender detalhes de fenômenos complexos.

De acordo com Barros *et al.* (2011), a SBA difere-se da SED, pois nesta última programa-se o processo que vai atuar sobre objetos, como peças, pessoas e documentos. Já na SBA programam-se as atividades das peças, pessoas e documentos, para que estes adaptem-se às situações que vão encontrando na busca de seus objetivos e “decidam” sobre seus destinos,

cada vez que entram em contato com outros agentes e com o meio ambiente (sistema) em que se encontram.

Segundo Macal e North (2014), a SBA não é apenas ligada à compreensão e concepção de agentes, também está relacionada a necessidade de representar comportamentos, a interação social, colaboração, o comportamento do grupo, e seu possível surgimento.

A SBA tem como característica ser uma abordagem flexível e poderosa para modelar os comportamentos dinâmicos e detalhados de sistemas complexos, isso será de extrema importância para esta pesquisa, pois o sequenciamento é feito em sistemas complexos, dinâmicos e com situações cotidianas inusitadas. Há diversos relatos de sucesso da aplicação da Simulação Baseada em Agentes no contexto industrial (MORTAZAVI *et al.*, 2015). Existem diversos trabalhos recentes como o de Sena *et al.* (2017) e Oliveira *et al.* (2017) exemplificando a implementação da SH no contexto industrial, utilizando agentes dentro de modelos discretos para resolução de problemas mais complexos.

De acordo com Macal *et al.* (2013), a SBA é composta por quatro elementos fundamentais, mostrados na equação (1):

$$\mathbf{SBA} = \{A, I, E, T\} \quad (1)$$

Onde:

- A letra “A” representa um conjunto de agentes;
- A letra “I” simboliza o espaço de interação; é nesse espaço que ocorre a interação entre os agentes;
- A letra “E” equivale ao ambiente independente dos agentes; nesse ambiente, métodos de atualização do estado do ambiente são efetuados, além disso, os agentes recebem informações específicas do ambiente;
- A letra “T” reflete o mecanismo de avanço temporal, ou seja, o processo de avanço do tempo na simulação.

A SBA reúne princípios de inteligência artificial, teoria de redes sociais, pesquisa operacional e ciências cognitivas. Seu principal objetivo é expandir a teoria tradicional de simulação para incluir agentes cujo comportamento pode ser alterado ao longo do tempo, dependendo das circunstâncias do ambiente (SAMUELSON, 2005).

Essa nova abordagem da simulação se baseia no conceito de que agentes autônomos que são relativamente simples e cujas interações seguem padrões simples podem, quando se juntam, produzir fenômenos complexos de difícil modelagem com base na abordagem tradicional (SAMUELSON e MACAL, 2006).

A natureza dos agentes é o que difere a SBA da SED segundo Chan *et al.* (2010), uma vez que na SBA os agentes são proativos, autônomos e inteligentes, podendo iniciar ações, comunicar com outros agentes e tomar decisões por conta própria. Já na SED as entidades ou recursos são simples, reativos e com capacidades limitadas.

Ainda não existe uma definição unificada sobre o que seria a SBA, porém Macal (2016) oferece quatro definições diferentes sobre a mesma, que se baseiam nas propriedades dos agentes e estão classificadas em ordem crescente de complexidade, o principal personagem dessa nova abordagem da simulação. Essas definições são:

- Simulação Baseada em Agentes Individual: na SBA Individual, os agentes possuem diversas características, e sua representação no modelo é a de um indivíduo distinto, que tem a capacidade de agir de forma independente durante o curso da simulação, como, por exemplo, um comportamento agendado para determinado momento. Sendo este o agente primário e mais simples de todos;
- Simulação Baseada em Agentes Autônoma: na SBA Autônoma, os agentes agem autonomamente com base em seus comportamentos internos, possuindo a capacidade de detectar condições que acontecem em qualquer momento dentro do modelo. Os agentes se comportam de maneira reativa, tomando suas próprias decisões com base em sua lógica interna, sem a necessidade de ser guiado;
- Simulação Baseada em Agentes Interativa: na SBA Interativa, os agentes possuem autonomia e interagem uns com os outros. Os agentes também interagem com o ambiente onde estão inseridos. Esta simulação exige outro grau de complexidade computacional. Porém, lógicas simples de interação entre os agentes entre si e com o ambiente podem gerar modelos muito úteis e avançados em termos de complexidade;
- Simulação Baseada em Agentes Adaptativa: na SBA Adaptativa, que seria a mais avançada entre os modelos baseados em agentes, os agentes autônomos e interativos mudam seus comportamentos no decorrer da simulação, podendo aprender e se adaptar perante situações novas. No caso de uma população de agentes, ela possui a capacidade de se ajustar de modo a incluir outros agentes que se adaptam à situação. Essa adaptação pode ser vista como o resultado da lembrança das ações do passado desse agente, e quais resultados elas atingiram, buscando sempre a melhoria. Ou seja, ele aprende com a experiência.

O Quadro 3.1 sintetiza as características dos quatro tipos de modelos de agentes citados anteriormente.

Quadro 3.1 - Definições para SBA com base nas propriedades do agente

Definições	Individualidade	Comportamentos	Interações	Adaptabilidade
SBA Individual	Agentes heterogêneos individuais	Prescrito, roteirizado	Limitado	Nenhum
SBA Autônoma	Agentes heterogêneos individuais	Autônomo, dinâmico	Limitado	Nenhum
SBA Interativa	Agentes heterogêneos individuais	Autônomo, dinâmico	Entre outros agentes e o ambiente	Nenhum
SBA Adaptativa	Agentes heterogêneos individuais	Autônomo, dinâmico	Entre outros agentes e o ambiente	Agentes mudam seus comportamentos durante a simulação

Fonte: Macal (2016)

De acordo com Sakurada e Miyake (2009), os modelos baseados em agentes permitem que as entidades tenham a habilidade de detectar particularidades do ambiente, interagir com outros agentes, e escolher um curso de ação. Agentes podem ser empregados em sistemas de movimentação, transporte e manufatura, onde a interação e tomada de decisão local são extremamente importantes.

Existem algumas situações em que a SBA oferece vantagens em relação a SED e outras técnicas de modelagem. Axtell (2000) discute algumas dessas razões e procura critérios com os quais torna-se benéfico pensar em termos de agentes. Apenas alguns desses critérios sendo satisfeitos, já se torna viável o uso de agentes, esses critérios são:

- Quando o problema tem uma representação natural sendo composta de agentes;
- Quando as decisões e comportamentos podem ser bem definidos;
- Quando é importante que os agentes tenham comportamentos que refletem como os indivíduos realmente se comportam;
- Quando é importante que os agentes se adaptem e mudem seus comportamentos;
- Quando é importante que os agentes aprendam e se envolvam em interações estratégicas e dinâmicas;

- Quando é importante que os agentes tenham relações dinâmicas com outros agentes, onde os relacionamentos entre eles se formam, mudam e se deterioram;
- Quando é importante modelar os processos pelos quais os agentes formam as organizações, onde adaptação e aprendizagem são importantes a nível da organizacional;
- Quando é importante que os agentes tenham um componente espacial para seus comportamentos e interações;
- Quando o passado não prevê o futuro devido a dinâmica dos processos de crescimento e mudança;
- Quando o dimensionamento para níveis arbitrários é importante em termos do número de agentes, interações e estados dos agentes;
- Quando a mudança estrutural do processo precisa ser um resultado endógeno do modelo, em vez de uma entrada para o modelo.

3.2.1. Agentes

Macal e North (2014) explicam algumas propriedades e atributos dos agentes, com o propósito de facilitar a modelagem destes sistemas:

- **Autonomia:** o agente é autônomo e na medida em que pode funcionar independentemente do seu meio ambiente ou de sua interação com outros agentes. Ele tem sua lógica limitada às situações de interesse do programador. Chama-se de comportamento do agente, como as ações e decisões do agente perante sua sensibilidade às informações que são captadas do ambiente;
- **Modularidade:** um agente é um elemento discreto e identificável com uma série de características ou atributos, além da sua capacidade de efetuar decisões. A modularidade indica que o agente possui um limite. É por meio desse limite que se torna possível discernir o que do modelo faz parte ou não de um agente, ou se alguma característica é uma particularidade compartilhada entre os agentes de um modelo;
- **Sociabilidade:** indica sua interação com outros agentes. Existem protocolos e mecanismos que direcionam a forma como vão se envolver com outros agentes, exemplo: disputa por espaço, a prevenção de colisões, o reconhecimento do agente, a troca de informações, a influência e outros mecanismos específicos.

Além dessas características citadas, há outras propriedades que podem ou não ser consideradas como propriedades necessárias para os mesmos. As quais seriam:

- O agente pode viver em um ambiente ou não. Se viver, pode interagir com o mesmo. O comportamento do agente é baseado em seu atual estado e em suas interações com o ambiente e com outros agentes, ou seja, seu comportamento vai sempre depender da situação em que se encontra;
- O agente pode ter metas que determinam seu comportamento. Pode ainda medir as consequências de suas ações através da influência das mesmas nas metas a serem alcançadas, assim ele faz comparações para sempre modificar seu comportamento com a intenção de melhorar seus resultados;
- Sendo assim, o agente pode aprender e adaptar seu comportamento baseado em suas experiências. Isso requer que o agente tenha memória em forma de atributo dinâmico do agente;
- O agente frequentemente tem atributos que mostram seu estado atual de recursos.

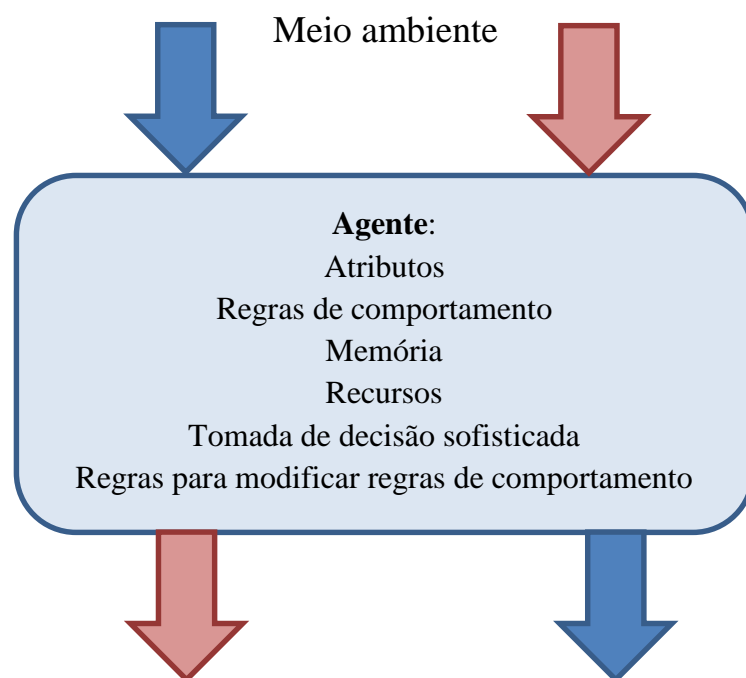


Figura 3.1 - Características do agente

Fonte: adaptado de Macal e North (2014)

A Figura 3.1 ilustra o comportamento do agente, que varia de acordo com o seu tipo ou sofisticação, o número de informações que são consideradas em uma tomada de decisão, suas lógicas ou modelos internos, o ambiente externo em que se situa, reações e comportamentos dos agentes que interagem com ele, além de suas memórias passadas que usará para guiar suas futuras decisões.

Tentando explicar quais características um objeto ou entidade precisa ter para ser chamado de agente, Grigoryev (2012) apresenta algumas propriedades necessárias para tanto:

- Em muitos modelos pode não existir o espaço físico, mesmo assim o agente pode existir, pois ele não precisa ocupar um espaço físico;
- Agentes não precisam ser pessoas, o que antes eram denominados como entidades, veículos, recursos (como máquinas) ou peças, agora podem ser chamados de agentes, pois adquiriram propriedades adicionais que estão sendo explicadas nesta tese;
- Um objeto passivo pode ser um agente mais simples;
- Um modelo pode ter somente um ou vários agentes em sua composição;
- Existem modelos de SBA em que os agentes não interagem.

3.3. Simulação Híbrida

No dicionário Aurélio encontra-se algumas definições para a palavra híbrido, dentre elas citam como sendo híbrido aquilo que tem elementos diferentes em sua composição. Logo, simulação híbrida possui este conceito, pois é a união de dois ou mais tipos de simulação. Pode ser uma união, por exemplo, da simulação dinâmica com a SED, formando um modelo discreto-contínuo. Ou pode ser, entre outros casos, uma união entre SBA e SED, o que é o caso desta pesquisa.

De acordo com Baines e Kay (2002) e Dubiel e Tsimhoni (2005), existem três limitações envolvendo a modelagem do fator humano através da SED, as quais são:

- As funções e os movimentos genéricos que as entidades necessitam realizar no ambiente virtual;
- A capacidade que a entidade possui para tomar decisões. Com a SED, muitos pontos de decisão precisariam ser modelados pelo analista, de modo que esses pontos estejam bem próximos um do outro;
- A determinação dos pontos de decisão do fator humano. Desse modo, as entidades do modelo não são autônomas.

Os autores comentam que, por meio de uma programação inteligente e das funções provenientes da própria SED, algumas dessas limitações podem ser superadas. Entretanto, esse fato não refuta o argumento de que a SED não é a melhor ferramenta para modelar o comportamento humano de modo mais detalhado.

A SED usa a programação voltada ao processo, ou seja, o processo age sob as entidades e recursos, os pontos de decisão que estas entidades podem escolher estão fixos em alguns pontos do processo. Na SBA, de acordo com Sakurada e Miyake (2009), os agentes podem, a todo momento, avaliar as condições do sistema e tomar novas decisões, independentemente de sua posição no mesmo, o que seria improvável de acontecer na SED.

As diferenças entre os resultados do modelo real ao computacional é um problema comum, mesmo trabalhando-se em cima dos possíveis defeitos do modelo, as vezes não se chega ao resultado esperado. Brailsford *et al.* (2012) comentam que isso levantou uma necessidade da criação dos agentes, pois muitas vezes esse problema pode ser resolvido com a SBA, que modela o comportamento humano.

O Quadro 3.2 faz uma síntese comparativa entre a SED e a SBA, no que diz respeito às características individuais dos recursos e entidades, bem como suas interações, transições de estado e movimentações no modelo, além de citar diferenças no fluxo de processos e filas.

Quadro 3.2 - Comparação entre SED e SBA.

Elemento de Modelagem	Simulação a Eventos Discretos (SED)	Simulação Baseada em Agentes (SBA)
Individuação das entidades e recursos	Ignora a variação de desempenho das entidades e seus comportamentos proativos. Porém, possibilita definir grupos distintos de entidades com características em comum que seguem fluxos distintos no processo.	O comportamento de cada entidade é individualizado, definido através de estados, acionado por eventos, mensagens ou condições do sistema. Possibilita modelar grupos heterogêneos e suas interações, nas quais cada agente pode ter incentivos e motivações particulares.
Entidades e Recursos	Comunicação/ Interação entre entidades e recursos	A comunicação/interação pode ocorrer entre entidades e/ou recursos. A troca de informações pode alterar o comportamento dos elementos e influenciar as ações tomadas no sistema simulado.
	Movimentação das entidades e recursos no modelo	Possibilita livre movimentação das entidades no sistema simulado de acordo com os estímulos do ambiente que são recebidos e processados pelas entidades e/ou recursos. Não requer que as rotas que podem ser percorridas sejam previamente determinadas.
Processos	Definição do fluxo de Atividades / Esperas / Filas	Modelos descentralizados, ou seja, o comportamento das entidades é feito em nível micro (individual) e o comportamento global do sistema emerge das interações individuais (<i>bottom-up modeling</i>)
Reconfiguração do modelo na simulação	Transição de estados das entidades e recursos	A transição de estados possibilita modelagem de cenários flexíveis que podem ser reconfigurados durante a simulação. Possibilita programar comportamentos distintos (estados) acionados por alterações do ambiente de simulação.

Fonte: Sakurada e Miyake (2009).

Adicionando ao Quadro 3.2 que foi apresentado, Siebers *et al.* (2010) mostram outros atributos que definem a SED e a SBA no Quadro 3.3, além de permitir identificar em qual das duas abordagens o modelo se encaixa. Ao analisar o Quadro 3.3, os autores chegam a conclusão que, na área da Pesquisa Operacional, modelos puramente baseados em agentes são muito mais difíceis de serem encontrados do que modelos de Simulação Híbrida. Pois ao invés disso, existe uma combinação da SED a SBA, onde o fluxo do processo é representado como um modelo de SED e adiciona-se algumas entidades ativas ao invés das passivas da SED, que são agentes autônomos, podendo mostrar um comportamento proativo.

Quadro 3.3 - Outros atributos comparativos entre SED e SBA.

Modelos de Simulação a Eventos Discretos	Modelos de Simulação Baseada em Agentes
Orientado ao processo (abordagem de modelagem <i>top-down</i>); foco na modelagem do sistema em detalhe, não na modelagem das entidades	Orientado ao objeto; com base no indivíduo (abordagem de modelagem <i>bottom-up</i>); foca na modelagem das entidades e das interações entre elas
Controle por meio de processos (controle centralizado)	Cada agente possui seu próprio controle de processo (controle descentralizado)
Entidades passivas; a inteligência, como a tomada de decisão, é modelada como parte do sistema	Entidades ativas; a inteligência é representada individualmente dentro de cada entidade
As filas são um elementos-chaves	Não apresenta nenhum conceito de filas
O fluxo de entidades ocorre através de um sistema; o comportamento macro é modelado	Não há nenhum conceito de fluxos; o comportamento macro não é modelado, pois ele emerge das micro decisões dos agentes individuais
As distribuições de entrada são muitas vezes baseadas na coleta e medição de dados	As distribuições de entrada são muitas vezes baseadas em teorias ou dados subjetivos

Fonte: Siebers *et al.* (2010).

O Quadro 3.3 mostra, na primeira linha, a diferença entre a programação orientada ao objeto e a orientada ao processo. Essa é de fato a primeira grande diferença, uma vez que a programação orientada ao processo é usada através da modelagem *top down*, pois define primeiro o processo, e depois como ele agirá sob as entidades e recursos do sistema. Já na

programação orientada ao objeto, existe as definições das regras individuais, dessas regras surgem o comportamento macro do sistema, por isso é chamada de *bottom top*.

Outra grande diferença apresentada pelo Quadro 3.3 é em relação às entidades serem ativas ou passivas, o que as define como sendo agentes ou simples entidades que fazem parte de um processo.

Ao mostrar no Quadro 3.3 que, na SBA, não há o conceito de filas nem dos fluxos de produção, e os dados de entrada são subjetivos ou baseados em teorias, percebe-se porque os autores na pesquisa operacional não costumam utilizar sistemas puramente baseados em agentes, pois esses fatores são importantes para qualquer modelo de simulação de qualquer processo de fabricação.

3.4. Sequenciamento da Produção

Este item objetiva explicar alguns conceitos básicos sobre o SP, que se enquadram dentro da área de Planejamento e Controle da Produção. Estes conceitos são importantes para complementação da pesquisa.

Esta tese tratará de uma das tarefas relacionadas ao planejamento e controle da produção: o sequenciamento. Contudo, o PCP também engloba atividades de: programação, carregamento e controle da produção. A preocupação principal do trabalho será a de escolher em que ordem fazer, e não fazer o controle da produção verificando se está conforme o plano ou programando quando produzir, o que seriam atividades do controle e programação.

Lustosa *et al.* (2008) explicam que o sequenciamento se refere à sequência de produção das ordens nos recursos, respondendo à pergunta: em que ordem fazer? As prioridades são dadas perante a um conjunto de regras de sequenciamento. Esta atividade está sujeita a problemas como: quebras de máquinas, restrições físicas à escolha das sequências, interrupção da ordem em processamento, necessidades de mudanças na sequência de produção, etc.

Logo, este trabalho não deve ser confundido com a programação da produção, que responde à pergunta: quando fazer? Em que se gera um cronograma dizendo o começo e o fim de cada tarefa. Programas são declarações de volumes e horários, gerando um relatório, e devem ser feitos após o sequenciamento. A programação pode ser realizada para frente ou para trás. Sendo que para frente, a produção começa logo quando o primeiro pedido chega, entrando na fila para ser produzido. Já a programação para trás, começa-se no último momento possível, subtraindo o tempo de processo, assim como é feito nos sistemas controlados pelo Just in Time.

O modelo desta tese faz a programação para frente, visto que as ordens começam a ser produzidas assim que chegam, porém não é o foco desta pesquisa estudar a programação da produção.

Outra característica do modelo programado neste trabalho é que ele usará o *layout* funcional. Embora existam críticas em relação à produtividade deste *layout*, Biagio (2015) afirma que este arranjo físico é o mais utilizado no mundo, pois micro e pequenas empresas muitas vezes não possuem recursos para dedicarem a um produto específico. Além disso, elas produzem em pequenos volumes, não justificando uma produção em célula ou linha.

Segundo Slack *et al.* (2009) o sistema de produção que usa o *layout* funcional ou por processo normalmente tem sua produção empurrada e possui a vantagem de ter uma grande flexibilidade. Nesse sistema, o SP é uma área de grande relevância nos resultados. O SJSF é muito utilizado por empresas de usinagem que produzem seus lotes sob encomenda, variando completamente a forma da peça e quantidade produzida de um pedido para outro. Nesses sistemas, a sequência de produção influenciará completamente em resultados como: o percentual de utilização das máquinas e o *makespan*, que são os objetivos de produção principais desta pesquisa. Os mesmos serão priorizados de acordo com as regras de sequenciamento implementadas pelo agente.

A escolha do melhor *layout* tem relação com o tipo de processo que se pretende implementar na empresa, se o processo será por tarefa, se serão produzidos em pequenos ou grandes lotes, ou ainda se será uma produção em massa ou até contínua.

A melhor escolha para um processo depende do volume e personificação. Segundo Krajewski *et al.* (2016), o tipo de processo por tarefa ou *jobs* usa o *layout* funcional, que é quando trabalhadores e máquinas do mesmo tipo estão agrupadas em setores, também chamados de *shop floors*. Exemplos: armário sob medida, peças usinadas únicas, moldes, vestuário. O SJSF é útil tanto na produção de tarefas unitárias quanto na de pequenos lotes, também feitos sob encomenda.

Lustosa *et al.* (2008) afirmam que, dentre os problemas de SP, o sistema mais complexo é SJSF, visto que pode apresentar a maior variedade de roteiros de produção, onde cada ordem assume um roteiro diferente, e também deve-se respeitar a disponibilidade dos recursos. Ele afirma ainda, que simular o desempenho sob diversas regras de prioridade é uma ótima alternativa para resolver esses tipos de problemas de sequenciamento.

O autor cita que modelos matemáticos podem ser usados para problemas de sequenciamento, mas um grande defeito é que eles procuram simplificar a teoria da programação, demandando que: todas as ordens estejam disponíveis na data zero; os tempos de

processamento sejam constantes; não se considerem *setups*; não se considerem quebras de máquinas; não se considerem interrupções do lote; não se considerem cancelamento da ordem de produção e não aconteçam modificações durante a produção das ordens.

Em problemas reais, ao tentar contornar esses dilemas, o gerente poderia: renegociar prazos, fracionar ordens, usar horas extras, concentrar-se nos gargalos, balancear a produção, re-sequenciar as ordens perante alguma parada por quebra de máquina, priorizar clientes importantes, entre outras funções normais do cotidiano de fábrica.

De acordo com Corrêa e Corrêa (2004), a viabilidade do SP se dá no momento em que se percebe que analisar exaustivamente todas as possibilidades de produção duraria muito tempo, ou seria até impossível. Portanto, procura-se sequenciar a produção definindo a ordem a ser produzida com o intuito de atingir um objetivo.

Os mesmos autores afirmam que a programação da produção engloba a parte do sequenciamento, pois consiste em alocar no tempo as atividades obedecendo a sequência e as restrições do processo, como capacidade produtiva por exemplo. Ao usar a simulação para resolver um problema de sequenciamento, o próprio modelo acaba alocando as atividades nos recursos ao longo do tempo, obedecendo as restrições programadas no mesmo.

As regras de sequenciamento levam em conta como informações de entrada:

- Tempo de processamento da ordem de trabalho (já leva em consideração o tempo de *setup*);
- Data prometida para entrega da ordem;
- Momento de entrada na fábrica ou no centro de trabalho;
- Quais máquinas e a sequência delas a ser utilizada pela ordem.

Com o objetivo de medir o desempenho deste sistema são utilizados os indicadores de desempenho, que serão explicados no item a seguir.

3.4.1. Medidas de desempenho da produção

Segundo Hopp e Spearman (2000), os objetivos para o sequenciamento e programação da produção são: cumprir a entrega dos produtos na data prometida, minimizar o estoque em processo, reduzir o tempo de produção dos lotes (*makespan*) e maximizar a utilização dos recursos. Contudo, os mesmos são conflitantes entre si, uma vez que quando se busca a otimização de um deles, os outros ficam comprometidos. Logo, a meta da programação detalhada da produção pode ser bastante confusa.

De uma maneira geral, o objetivo sempre é aumentar a lucratividade das empresas contrabalançando os conflitos dos objetivos gerais do setor de produção de uma empresa, os quais são: aumentar a produtividade, diminuir estoques, reduzir custos e conseguir maior flexibilidade. Empresas estão sempre buscando ferramentas que possam auxiliá-las nesta difícil tarefa.

Para conseguir um bom SP, Lustosa *et al.* (2008) citam que se deve procurar equilibrar os principais objetivos da produção, como: cumprimentos de prazos, velocidade de fluxo e utilização da capacidade.

Ainda de acordo com os autores, algumas medidas de desempenho são importantes para a programação detalhada da produção:

- Tempo de processamento total (*makespan*): é o tempo necessário para conclusão de todas as ordens abertas, ou seja, intervalo de tempo entre a liberação da primeira ordem e conclusão da última operação da última ordem processada. Minimizar o *makespan* sustenta as prioridades de tempo, porém, com a redução da velocidade de entrega, consegue ter um efeito colateral de melhorar também a confiabilidade de entrega na data prometida;
- Tempo de atravessamento da tarefa: é o somatório dos tempos de processamento, espera, *setups*, movimentações entre as operações, atrasos provenientes de quebra de equipamentos, indisponibilidade de equipamentos, entre outros;
- Atraso: é o somatório de tempo que um grupo de tarefas atrasou em relação a sua data de entrega. Minimizar o atraso está relacionado às prioridades competitivas de custo, confiabilidade (serviço insatisfatório) e tempo (rapidez de entrega);
- Percentual de ordens finalizadas no prazo: razão entre as ordens finalizadas no prazo e todas as ordens recebidas;
- Estoque em processo: do ponto de vista do sequenciamento, pode ser determinado pelo número de ordens abertas que estão ao longo do horizonte de programação;
- Percentual de utilização: razão entre o tempo efetivamente utilizado e o tempo total disponível do recurso. É um fator completamente relacionado ao custo.

Nem todas as medidas precisam ser utilizadas em um mesmo trabalho. É muito comum os autores escolherem apenas uma ou duas medidas para avaliação de seus modelos. Nesta tese, foram escolhidas medidas muito utilizadas no SP: o *makespan* e o percentual de utilização das máquinas.

Todas essas medidas de desempenho citadas são importantes para o SP, porém eventos inesperados ocorrem no chão de fábrica: quebra de máquinas, falta de materiais, ordens inesperadas e reprocessamento das ordens de forma rápida e confiável. Com a finalidade de contornar alguns desses eventos, essa pesquisa propõe o uso de agentes autônomos que podem tomar decisões e até interagir com máquinas, detectando problemas e procurando reprogramar a produção de modo que os resultados sejam melhorados.

3.4.2. Regras heurísticas de sequenciamento

Algumas regras de sequenciamento são apresentadas por Tubino (2017) com a finalidade que uma ordem seja a primeira a ser atendida entre outras:

- Regra de *Johnson*:

Etapas:

- Listar o tempo de operação para cada tarefa em ambas as máquinas;
 - Selecionar a tarefa com o menor tempo de operação;
 - Se o menor tempo é o da primeira máquina, fazer esta tarefa primeiro; se o menor tempo é o da segunda máquina, fazer esta tarefa por último;
 - Repetir as etapas 2 e 3 para cada tarefa restante até a programação estar completa.
- *First in first out* (FIFO): as ordens são alocadas de acordo com sua chegada no centro de trabalho, a primeira que chega é a primeira a ser produzida.

Essa é uma regra de sequenciamento que é conhecida como sendo uma regra justa, muito usada para enfileirar pessoas. Também é importante para casos em que a validade do item expira rapidamente, reduzindo o tempo de espera.

Porém, do ponto de vista da produção, essa regra não busca reduzir atrasos, tempo de atravessamento e custo, não sendo vista como uma das melhores regras a serem usadas;

- *Last in first out*: as ordens são alocadas de acordo com sua chegada no centro de trabalho, a última que chega é a primeira a ser produzida.
É a regra de sequenciamento escolhida por razões práticas, como a saída de elevadores por exemplo. Essa regra é ainda muito pior que a primeira, e por isso pouco utilizada, pois é uma regra injusta, além de não buscar atingir nenhum objetivo de produção importante, como custo, rapidez ou confiabilidade;
- Data prometida: o trabalho é sequenciado de acordo com a data prometida de entrega.

Essa é uma regra que foca principalmente na minimização de atrasos, melhorando não só a confiabilidade da entrega.

Seu ponto negativo é nítido, uma vez que ela independe do tamanho do pedido ou da importância do consumidor. Essa regra também não prioriza a eficiência produtiva, o que pode aumentar os custos da empresa;

- **Maior tempo de processamento:** prioriza as ordens com tempos de operação maiores. É uma regra que visa a alta utilização da mão de obra e equipamentos, reduzindo, portanto, o custo de produção. Isso acontece porque ocupa os centros de trabalho por um longo tempo, além de reduzir percentual de tempo de *setup* de modo geral. Entretanto, essa regra não busca otimizar a rapidez, confiabilidade ou flexibilidade, podendo até comprometer esses resultados;

- **Menor tempo de processamento:** prioriza as ordens com tempos de operação menores, tende a reduzir o *makespan*.

É uma regra que acelera o faturamento e a disponibilidade de caixa. Tem um efeito colateral também, no desempenho de entregas, ou seja, no número de trabalhos completados. Contudo, é uma regra que pode afetar a produtividade e prejudicar consumidores maiores;

- **Menor fila adiante:** a ordem vai para o posto com a menor fila adiante.

É uma regra que busca não deixar os postos ociosos, balanceando a utilização. Tem, portanto, um efeito positivo nos custos;

- **Prioridade ao consumidor:** os clientes mais importantes ganham atendimento prioritário. Essa regra é comumente usada quando há muitos clientes pequenos e alguns clientes muito grandes, dando prioridade aos últimos.

É uma regra que pode baixar o desempenho e a produtividade da operação se o fluxo de trabalho for muito perturbado por conta de clientes importantes. Porém, aumenta a confiabilidade da empresa perante esses grandes clientes, cabendo a mesma decidir quais são suas prioridades.

Além do uso na indústria em geral, é muito usada por bancos, hotéis e hospitais no setor de urgência.

As três últimas regras são baseadas em cálculos de índices, normalmente empregadas em sistemas que utilizam o MRPII, que contem módulos que calculam esses índices, encarregados de gerar prioridades para as ordens liberadas pelo módulo MRP.

- ICR – Índice Crítico: os lotes serão processados de acordo com o menor valor do resultado de:

$$\frac{(data\ de\ entrega) - (data\ atual)}{(tempo\ de\ processamento)}$$

As regras ICR e IFO são baseadas no conceito de folga entre a data de entrega do lote e o tempo de processamento, sendo que a regra IFO considera não só a operação imediata, como todas as demais à frente.

Assim como na lógica da data prometida, as regras ICR e IFO privilegiam o atendimento ao cliente;

- IFO - Índice de Folga: os lotes serão processados de acordo com o menor valor do resultado de:

$$\frac{(menor\ valor\ de\ data\ de\ entrega) - (somatório\ dos\ tempos\ de\ processamento)}{(número\ de\ operações\ restantes)}$$

- IFA – Índice de Falta: os lotes serão processados de acordo com o menor valor do resultado de:

$$\frac{(quantidade\ em\ estoque)}{(taxa\ de\ demanda)}$$

O índice de falta relaciona os estoques atuais com a demanda, procurando evitar que os estoques se esgotem, o que causa prejuízo ao fluxo produtivo, sendo empregada para produtos acabados ou para os itens intermediários que compõem os mesmos. Essa regra é usada para produtos de venda contínua (pois considera a taxa de demanda), podendo usar, por exemplo, um sistema *make to stock*, o que não é o caso desta pesquisa.

É importante salientar que não existe nenhuma regra mágica. A escolha de cada regra depende do objetivo de produção a ser atingido. Normalmente esse objetivo é definido pelo que os clientes esperam da empresa, seja baixo custo, entrega rápida, ou talvez até aceitem uma entrega mais longa desde que o produto não atrase ou que seu custo seja baixo. É evidente que os clientes esperam todas essas qualidades, mas sempre há uma delas que é mais importante.

Sendo assim, os objetivos estratégicos das empresas são importantes de serem citados pois estão diretamente ligados às regras de sequenciamento. De acordo com a regra escolhida um objetivo estratégico será priorizado, visando melhorar o que realmente interessa à empresa tendo em vista a manufatura. O meio de medir estes objetivos são os indicadores mostrados no item 3.4.1.

Corrêa e Corrêa (2004) afirmam que o objetivo estratégico da empresa é entregar valor aos clientes de acordo com a estratégia definida, procurando sempre o resultado financeiro,

visando um determinado mercado a ser atingido. Na escolha da estratégia, é necessário se fazer um *trade off* na escolha dos objetivos, e que escolhas implicam em renúncias. Não se pode esperar um produto veloz e confiável na entrega, com alta qualidade, baixo custo e com uma entrega flexível. É preciso conhecer as principais necessidades dos clientes para decidir a renúncia nesses *trade off's*.

Os mesmos autores citam os cinco objetivos de desempenho a serem priorizados pela empresa, objetivando o ganho de pedidos na corrida com os concorrentes. Os quais são:

- Velocidade: pode ser vista como velocidade de atendimento, cotação e entrega. No caso de um setor de produção, seria mais relacionado a entrega do produto ao cliente;
- Confiabilidade: pode ser vista como pontualidade de entrega, integridade no sentido de cumprir as promessas, segurança do produto e robustez na manutenção do atendimento. Todos são relacionados ao chão de fábrica;
- Qualidade: é vista como conformidade às especificações, desempenho e recursos do produto, além da durabilidade e limpeza do mesmo. Todos esses fatores não estão diretamente relacionados ao sequenciamento, pois a sequência de produção não afeta estas características, não havendo uma regra de sequenciamento voltada à qualidade;
- Flexibilidade: pode ser em relação aos produtos, no sentido de introduzir ou modificar pedidos), em relação ao *mix* de produção, flexibilidade de entrega e volumes de produção flexíveis;
- Custo: simplesmente relacionado ao custo de se produzir algum produto, relacionado a utilização das máquinas e mão de obra no que diz respeito a eficiência produtiva e a um baixo nível de estoque em processo.

Segundo Slack *et al.* (2009), internamente, o custo está relacionado a uma alta produtividade, a rapidez a um baixo tempo de atravessamento, a qualidade ao processo livre de erros, a flexibilidade à habilidade para mudar e a confiabilidade a uma operação confiável.

Apesar da necessidade de se fazer um *trade off*, há aspectos de desempenho que não são conflitantes e até se ajudam, como um programa de eliminação de falhas que reduz defeitos, retrabalho e atrasos na produção, ou uma melhoria na rapidez que pode aumentar a confiabilidade de entregas e flexibilidade.

4. MÉTODO DE PESQUISA

Este capítulo contextualiza o método de pesquisa utilizado nessa tese: modelagem e simulação. Em seguida é definido o modelo de pesquisa. A pesquisa também é classificada segundo: sua natureza, seus objetivos, sua forma de abordar o problema e o método de pesquisa utilizado.

O segundo item descreverá o Protocolo ODD (*Overview, Design concepts, Details*) que foi a maneira utilizada para descrever o modelo feito nesta pesquisa.

4.1. Classificação da pesquisa

O método de pesquisa utilizado é a modelagem e simulação. Segundo Turrioni e Mello (2012) ela é empregada quando se deseja experimentar, através de um modelo, um sistema físico, determinando-se como este sistema responderá a modificações que lhe são propostas. Nesta tese foram feitos modelos de simulação para testar diferentes lógicas de sequenciamento sobre problemas cotidianos do chão de fábrica.

A presente tese segue a linha de pesquisa quantitativa. Segundo Bertrand e Fransoo (2002) a classificação quantitativa é dada às pesquisas onde modelos de relações causais entre variáveis de controle e variáveis de saída são desenvolvidos, e estas são analisadas e alteradas sobre um domínio específico. Esta pesquisa usa métodos estatísticos para comparar os resultados atingidos nas resoluções dos problemas.

A metodologia de pesquisa pode ser classificada como axiomática ou empírica, dentro da pesquisa quantitativa. As pesquisas axiomáticas produzem conhecimento matemático sobre o comportamento das variáveis dentro de um sistema. Já na pesquisa empírica, procura-se imitar um sistema real transmitindo-o ao computador. Neste tipo de pesquisa, os resultados teóricos podem ser aplicados em processos reais.

Ainda existe a classificação das pesquisas por serem descritivas ou normativas. As descritivas, como o próprio nome traduz, procura compreender e descrever um modelo, explicando e analisando as características do mesmo. Já a normativa, procura desenvolver novas maneiras, estratégias e ações para resolver algum problema, e assim o faz, comparando os resultados obtidos em seu trabalho com algum resultado gerado por um método convencional e usual.

Nessa tese foi utilizado o modelo de pesquisa empírico normativo, em que existe o interesse em desenvolver políticas, estratégias ou ações para melhorar um sistema real corrente.

Empírico porque usa um tipo de sistema de manufatura que foi transmitido ao computador por meio da simulação, e através dos sucessivos experimentos chega-se às conclusões desta pesquisa. É normativo porque procura desenvolver uma nova estratégia para resolver o problema de sequenciamento, propondo um sequenciamento mais robusto e eficiente.

De acordo com Turrioni e Mello (2012), esta pesquisa pode ser classificada da seguinte forma:

- Quanto à natureza: aplicada, visto que a presente pesquisa se caracteriza por seu interesse prático, os resultados poderão ser aplicados ou utilizados na solução dos problemas que ocorrem na realidade;
- Quanto ao objetivo: normativa, pois procura desenvolver novas maneiras, estratégias e ações para resolver algum problema, comparando os resultados alcançados com os resultados usuais para resolver determinado problema;
- Quanto à forma de abordar o problema: quantitativa, uma vez que considera que tudo pode ser quantificável, traduzindo em números, as opiniões e informações obtidas para classificá-las e analisá-las numericamente. Ou seja, é uma análise mais objetiva do que a abordagem qualitativa. Também requer o uso de recursos e de técnicas estatísticas;
- Quanto aos métodos: modelagem e simulação, onde este método deve ser usado quando se deseja prever o efeito de mudanças no sistema e avaliar seu desempenho ou comportamento, sendo utilizado principalmente quando o sistema não pode ser modificado na prática para testes, como uma linha de montagem por exemplo. O método é usado quando o sistema real é muito complexo para ser analisado satisfatoriamente por um modelo matemático.

4.2. Protocolo ODD

Os modelos de SBA possuem uma estrutura mais complexa e são mais difíceis de analisar, compreender e comunicar. Muitas vezes, descrições publicadas de modelos de SBA são difíceis de ler, sendo incompletas, ambíguas e, portanto, menos acessíveis (GRIMM *et al.* 2006).

A fim de combater estes problemas, Grimm e Railsback (2005) criaram um protocolo padrão que poderia orientar os leitores e autores de SBA, o Protocolo ODD (*Overview, Design concepts, Details*). O mesmo consiste em uma ferramenta que tenta padronizar as descrições dos modelos de SBA, incluindo detalhes suficientes para permitir a sua replicação. O Protocolo

ODD tem sido aperfeiçoado desde o seu nascimento de forma a fazer com que as descrições deste tipo de modelo sejam mais compreensíveis e completas (GRIMM *et al.* 2010).

O Protocolo ODD é uma ferramenta imprescindível na construção de um modelo de SBA. Este proporciona uma visão geral do modelo, descrevendo-o de forma rápida, completa, simples e organizada. Além de informações do modelo, traz informações muito importantes dos agentes que compõem o modelo.

O Quadro 4.1 traz sinteticamente a estrutura do Protocolo ODD proposta por Grimm *et al.* (2006), informando o passo a passo que deve ser seguido para condução de um projeto de SBA.

Quadro 4.1 - Estrutura do Protocolo ODD

Elementos do Protocolo ODD	
Visão Geral	Proposta
	Entidades, variáveis de estado e escalas
	Visão geral do processo e programação
Conceitos de projeto	Princípios básicos
	Emergência
	Adaptação
	Objetivos
	Aprendizado
	Predições
	Sensibilidade
	Interações
	Estocasticidade
	Coletividades
Observações	
Detalhes	Inicialização
	Dados de entrada
	Submodelos

Fonte: Grimm *et al.* (2006)

A seguir é apresentado uma explicação detalhada de cada tópico:

- Proposta:
 - É uma declaração clara e concisa da questão ou problema abordado pelo modelo, que envolve qual o sistema que está sendo modelado e o que está se tentando aprender sobre ele;

- Entidades, variáveis de estado e escalas:
 - Entidades são os objetos (coisas) representados no modelo (outros agentes, ambiente, etc.);
 - As entidades possuem variáveis de estado que os representam;
 - Há dois tipos de escalas: temporal e espacial;
- Visão geral do processo e programação:
 - Representa a estrutura de um modelo;
 - Define-se os processos que mudam as variáveis de estado das entidades do modelo;
 - Faz-se uma descrição sucinta de cada um dos processos executados pelas entidades do modelo;
 - Define-se, também, a programação do modelo (sequência de ações): quais processos as entidades executam e qual a ordem;
- Conceitos de projeto:
 - Descreve como implementar um conjunto de conceitos básicos que são importantes para o modelo;
 - Fornecem um modo padronizado de pensar sobre características muito importantes dos agentes;
 - São divididos em: princípios básicos, emergência, adaptação, objetivos, aprendizado, previsões, sensibilidade, interações, coletividades e observações;
- Inicialização:
 - São as condições iniciais;
 - Podem ser o número de agentes criados e/ou os valores iniciais das suas variáveis de estado;
 - Como os valores iniciais das variáveis do ambiente são estabelecidos;
- Dados de entrada:
 - Existe uma fonte externa de dados?
 - Modelos frequentemente incluem variáveis ambientais, tais como temperatura ou preço de mercado que mudam ao longo do tempo;
- Submodelos:
 - É apresentado um detalhamento dos processos;

- Quais são os parâmetros do modelo, as suas dimensões e valores de referência? Como foram projetados ou escolhidos? Como foram parametrizados e então testados?

4.3. Passo a passo para a condução da pesquisa

A seguir é apresentado um passo a passo sobre o que será feito para a condução desta pesquisa:

- Desenvolver o Protocolo ODD descrevendo o modelo em detalhes:
 - Apresentando uma visão geral dos processos produtivos, as propostas dos modelos e seus componentes;
 - Os detalhes do agente principal: o Gerente, incluindo seu diagrama de estado e suas programações internas;
 - A explicação da tabela de dados de entrada e as condições de inicialização;
 - Além de alguns detalhes importantes da criação dos modelos;
- Preencher a tabela de dados de entrada de acordo com as ordens de produção que se pretende atender;
- Desenvolver o modelo computacional programando as lógicas dos agentes;
- Executar o modelo e a coleta de seus resultados:
 - *Makespan*;
 - Percentual de utilização média de todos os setores;
- Validar estatisticamente as variáveis de resposta do modelo, comparando os resultados de cada resposta entre o modelo com o Gráfico de Gantt;
- Realizar um teste “*what if*” simulando a produção das mesmas ordens com e sem o re-sequenciamento durante a produção.

5. DESCRIÇÃO DO MODELO HÍBRIDO PARA SEQUENCIAMENTO DA PRODUÇÃO EM SISTEMAS *JOB SHOP* FLEXÍVEIS

Em função das dificuldades e limitações do sequenciamento relatadas anteriormente nesta tese, tais como: quebra de máquinas, tempos estocásticos de processo e necessidades de modificações na sequência durante a produção das ordens, foi proposto a criação de um modelo de SP em SJSF que resolva estes problemas. Ou seja, um modelo mais robusto perante a eventos inesperados e sempre buscando resultados mais abrangentes.

A criação do modelo foi para avaliar os resultados da ação do agente Gerente na resolução desses problemas. Este modelo evoluiu de um modelo inicial que usava apenas a sequência de chegada (FIFO), para um modelo intermediário que usava a regra MTP implementada pelo agente Gerente, melhorando os resultados obtidos em relação ao modelo inicial. Porém, para resolver as dificuldades do sequenciamento citadas no parágrafo anterior, foi necessário evoluir para um modelo final, em que o Gerente além de utilizar a mesma lógica de sequenciamento do modelo intermediário, irá se deparar com quebras de máquinas inesperadas e necessidades de balanceamento de filas, modificando em tempo real a sequência de produção, com a meta de evitar que algum dos setores fique ocioso por muito tempo. Para isso, o Gerente implementa outra lógica de sequenciamento a ser explicada neste capítulo.

5.1. O Protocolo ODD do modelo proposto

Conforme foi citado no item 3.2, os modelos que possuem agentes têm uma estrutura mais complexa e são mais difíceis de analisar, compreender e comunicar. Por esta razão, Grimm *et al.* (2006) criam o Protocolo ODD, buscando uma explicação padronizada deste tipo de modelo que possui suas particularidades. Assim, evita-se explicações incompletas e ambíguas sobre os modelos e seus agentes.

Outra característica já citada é a inclusão de detalhes suficientes para permitir a sua replicação. Por isso, o protocolo tem sido aperfeiçoado desde o seu nascimento de forma a fazer com que as descrições deste tipo de modelo sejam mais compreensíveis e completas, proporcionando uma visão geral do modelo. Além de informações do modelo, traz informações muito importantes dos agentes que compõem o modelo.

O Quadro 4.1 traz sinteticamente a estrutura do Protocolo ODD proposta por Grimm *et al.* (2006), informando o passo a passo que deve ser seguido para condução de um projeto de SBA. Ao preencher suas lacunas, torna-se mais difícil e improvável que faltem informações relevantes do modelo a ser entendido. Os itens a seguir apresentam a descrição prática dos modelos desta pesquisa, feitos de acordo com a metodologia criada pelo protocolo em questão. Os passos são seguidos à risca para evitar que informações sejam omitidas durante a descrição dos modelos.

5.1.1. Proposta

Esta tese propõe a criação de um modelo genérico para SP em sistemas JSF. Para isso, foram usados dados fictícios a fim de simular uma situação produtiva neste tipo de sistema. Porém, esses dados podem ser facilmente modificados por meio da planilha de interface, mostrada na Tabela 5.1 mais adiante. Logo, o modelo pode ser utilizado para um sistema real se a planilha for preenchida com dados reais, desde que sejam feitas as adaptações necessárias inerentes ao processo real.

A pesquisa testará o agente sequenciador dentro de um modelo feito em um sistema de produção JSF. Neste caso, o modelo abrange até seis setores, em que cada um reúne um grupo de máquinas iguais que realizam a mesma função. O número de máquinas de cada setor pode ser facilmente modificado na planilha de interface. Para esta pesquisa foram utilizados cinco setores com: 3, 4, 3, 2 e 4 máquinas em cada setor, respectivamente. A não utilização de todos os setores foi proposital com o sentido de demonstrar como o modelo deve ser utilizado para menos de seis setores. O número de setores (*shop floors*) que o sistema tem também pode ser facilmente modificado na planilha, bem como o roteiro de fabricação das peças, tornando o mesmo um modelo mais abrangente, que servirá para diferentes aplicações dentro de sistemas de produção do tipo JSF.

O sistema pressupõe uma estratégia de produção do tipo *make to order*, visando a situação em que empresas clientes mandam desenhos da peça a ser fabricada e o setor de engenharia de processos deve projetar o processo de fabricação para a peça. Este setor normalmente projeta os processos que serão necessários para a fabricação da peça em questão, a ordem desses processos e calcula o tempo de processo necessário para fabricação em cada máquina. O cálculo desses tempos utilizados é baseado, citando como exemplo a operação de torneamento, em informações como: número de passos, velocidade de corte e velocidade de avanço necessárias para fabricação da peça requerida. Essa estratégia de produção é muito

usada por empresas de usinagem que produzem por encomenda, em que cada peça tem formato único. Esse tipo de processo pede um sistema JSF, pois o mesmo traz mais flexibilidade de produção em termos de *mix* e volume, necessário para a produção por encomenda.

5.1.2. Entidades, variáveis de estado e escalas

O segundo item descreve mais algumas informações do modelo. As ideias para as criações desses tipos de agentes foram retiradas dos trabalhos mostrados no item 2.2 (Casos na literatura de SP em SJSF). Foi realizada uma reunião de ideias retiradas dos artigos do item citado. O agente principal é o agente administrador que irá implementar as regras de sequenciamento. Os agentes secundários são divididos entre os agentes recurso, que são as máquinas de cada setor, e os agentes tarefa, que são os agentes lote e produto.

O problema JSF necessita desses agentes secundários pois é preciso representar as máquinas e os produtos que irão percorrer os processos. Esses agentes trazem em si as necessidades de produção, os tempos, rotas e restrições de capacidade de processo.

- Agente principal: Gerente;
- Agentes secundários: máquinas de 1 a 6, lote e produto;
- O comportamento do agente principal é baseado em uma regra pré-estabelecida para o SP, que obedece ao algoritmo MTP, e a uma variação da lógica de Menor Fila Adiante. Lembrando que este modelo fará o sequenciamento de um conjunto de lotes até que se terminem todos, ou seja, não existe o risco do lote com maior tempo de processo ficar sempre na fila esperando. O comportamento dos agentes secundários é baseado na planilha de interface;
- A escala será temporal, contabilizada em minutos;
- Diagrama de estado do agente principal:

Conforme citado, o modelo final evoluiu em três passos principais:

- a) **Modelo inicial:** este não tinha interferência do agente Gerente, pois usava a lógica de chegada (FIFO). Ele testa a produção dos lotes preenchidos na ordem da planilha, mostrados na Tabela 5.1 mais adiante. Apesar de ser um modelo mais simples, ele já era um modelo híbrido pois usava os agentes tarefa e os agentes recurso, mesmo não utilizando o agente administrador, que é o que possui o diagrama de estado com as regras de SP. Seus resultados não são mostrados pois o mesmo utiliza uma lógica considerada ruim em relação aos objetivos de produção;

- b) **Modelo intermediário:** este modelo usava a lógica MTP implementada pelo Gerente, reforçando-o como um modelo de Simulação Híbrida, sendo um pouco mais completo. É importante ressaltar que a SED seria capaz de implementar essa lógica em um modelo. O que seria mais improvável é a implementação do re-sequenciamento aplicado pelo modelo final, que se trata de uma continuação desse modelo intermediário. Sendo assim, a presente tese não objetiva a comparação da SED e da SH, mas sim da evidência das vantagens do uso de agentes quando se procura um modelo mais robusto, capaz de re-sequenciar a produção em tempo real e buscando objetivos mais abrangentes.

A Figura 5.1 mostra o diagrama de estado do Gerente no modelo intermediário, que usa apenas uma lógica para sequenciar as ordens de produção. O agente começa em um estado definido como “esperando”, em que ele aguarda a chegada de novas peças para começar a sequenciar. Em seguida este agente é acionado a mudar de estado ao chegar a primeira ordem de produção, passando para o estado “sequenciando”. Obviamente, neste estado ele calculará a sequência a ser produzida de acordo com a lógica pré-definida, o MTP. Depois de sequenciar, ele volta ao estado “esperando”. A lógica programada (em linguagem Java) dentro do estado “sequenciando” está no item Programação deste capítulo;

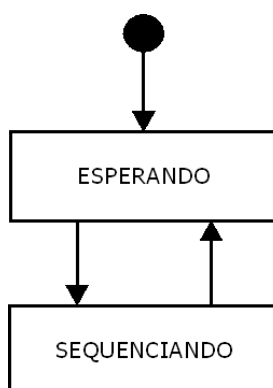


Figura 5.1 - Diagrama de estado do agente no modelo intermediário.

Fonte: Elaboração própria

- c) **Modelo final:** esse modelo também usa a lógica MTP inicialmente, mas após sequenciar procurará re-sequenciar a produção com uma variação da lógica Menor Fila Adiante, com o intuito de balancear as filas dos setores e, conseqüentemente, a produção nos mesmos, visto que o objetivo desta última lógica é evitar a ociosidade nos processos subsequentes. Este modelo final evoluiu o diagrama de estado e a ação do agente Gerente.

O modelo final usa diagrama de estado que é mostrado na Figura 5.2, em que o agente começa no estado “esperando lotes” e em seguida sequencia seguindo a lógica MTP, conforme o primeiro diagrama. Após sequenciar ele vai para o estado “esperando”. Depois disso, ele pode ir para o estado “balanceando filas” quando existir um fator que ative o re-sequenciamento, sendo que agora ele terá uma nova lógica que consiste em uma variação da lógica de Menor Fila Adiante. Porém, para aplicação desta lógica, o agente deverá dividir as filas de produtos esperando para serem produzidos em cada um dos setores pelo número de máquinas dos mesmos, dessa forma o agente obtém uma razão para cada setor. A menor delas é o setor onde o próximo lote a ser produzido deve ser enviado, a fim de se balancear as filas considerando o número de máquinas, evitando a ociosidade. Para fazer isso, o agente vai na fila de lotes que já está sequenciada segundo a lógica MTP e busca o primeiro lote da fila que vai para o setor com a menor razão calculada entre os setores, colocando-o para ser produzido no chão de fábrica. A consideração do número de máquinas em cada setor é muito importante, pois se, por exemplo, as filas de dois setores são iguais, a fila que tende a esvaziar primeiro é a fila cujo setor tem o maior número de máquinas. Logo, uma lógica que considere a razão do número de produtos na fila dividido pelo número de máquinas tende a ser mais eficiente que uma lógica que considere apenas o número de produtos na fila. As lógicas programadas dos estados “sequenciando” e “balanceando filas” também estão no item programação deste capítulo, lembrando que o estado “sequenciando” do diagrama de estado do modelo final é o mesmo do modelo intermediário, porque usa a mesma lógica inicialmente, antes de partir para o re-sequenciamento.

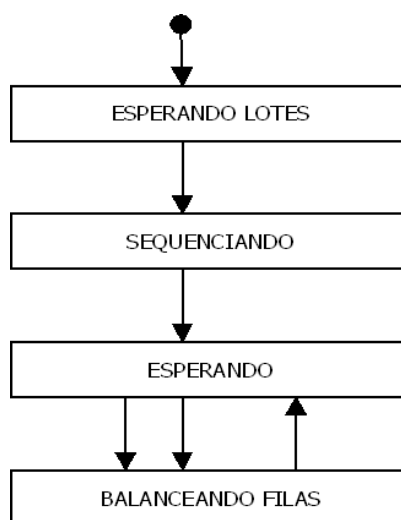


Figura 5.2 - Diagrama de estado do agente no modelo final.

Fonte: Elaboração própria

5.1.3. Visão geral do processo

O mapeamento foi feito seguindo a técnica de mapeamento IDEF-SIM, proposta por Montevechi *et al.* (2010). Essa técnica consiste em uma reunião de símbolos usados em outras técnicas IDEF e fluxogramas, a qual foi feita exatamente para uso do mapeamento voltado para a simulação. A Figura 5.3 mostra o mapeamento do sistema JSF mencionado, cujos parâmetros podem ser facilmente modificados na planilha de interface com o usuário. Caso haja dúvidas no entendimento do mapeamento, o Anexo A desta tese, mostra a simbologia usada na modelagem IDEF-SIM. É importante ressaltar que algumas figuras estão em inglês pois os nomes mostrados serão referidos na programação em Java, a serem apresentadas nos quadros mais adiante.

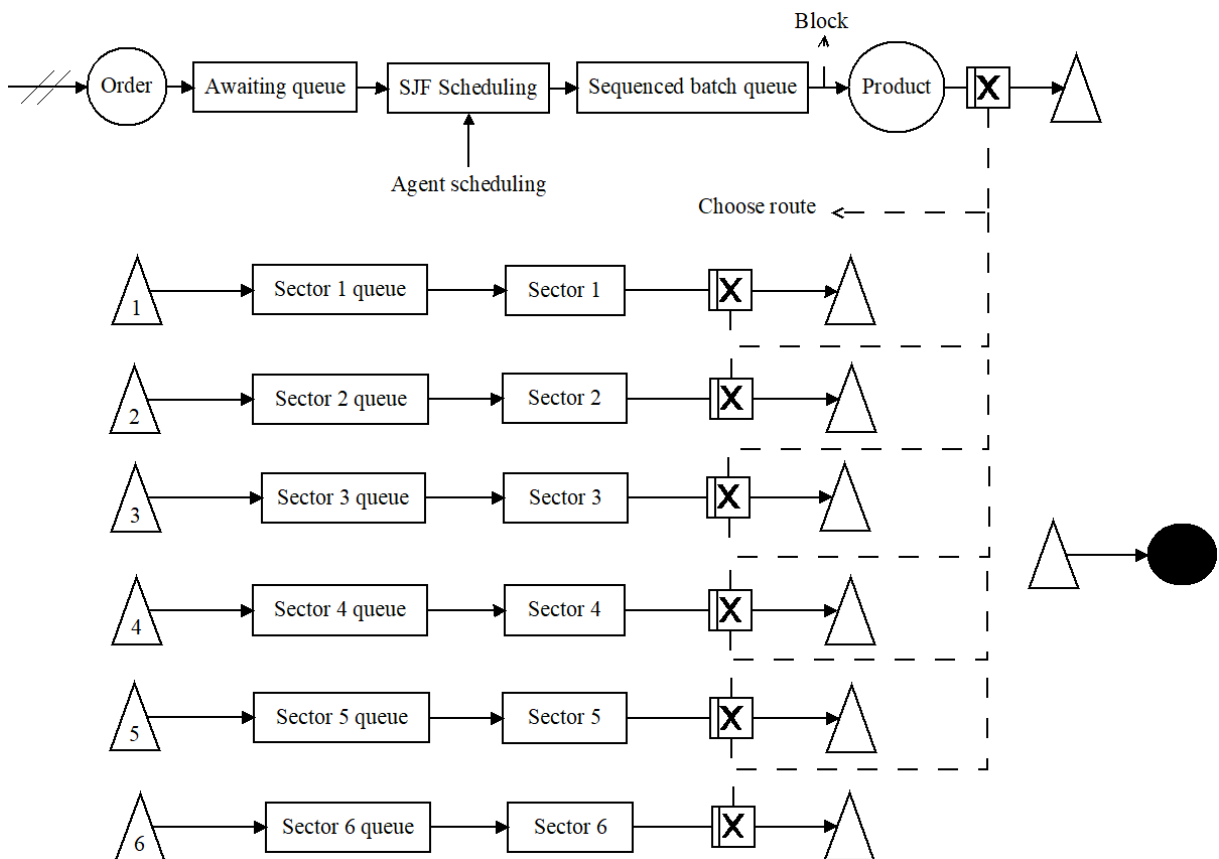


Figura 5.3 - Mapeamento do SJSF do modelo final.

Fonte: Elaboração própria.

Um evento denominado de “*OrdersEntrance*” corresponde aos pedidos dos clientes. Uma vez realizados os pedidos, os lotes a serem fabricados entram na fila de acordo com a ordem de chegada. As linhas tracejadas ligando ao “Choose route” indicam a informação de que os símbolos “OU” são o que proporcionam flexibilidade de rotas ao modelo. A lógica do processo produtivo mostrada na Figura 5.3 é explicada a seguir:

- O evento de entrada corresponde aos pedidos dos clientes, que entram no processo em forma de lotes a serem produzidos;
- Os lotes, então, formam uma fila de lotes sequenciados pela ordem de chegada;
- Os mesmos saem da primeira fila e são sequenciados pelo agente Gerente, indo para a segunda fila;
- Em seguida eles saem da segunda fila já desmembrados em produtos, de acordo com a quantidade encomendada em cada lote;
- O produto “pergunta” na saída da fila, qual o primeiro processo que ele percorrerá, entrando, assim, no primeiro processo através das entradas dos processos;
- Logo depois, ele “pergunta” na saída do primeiro processo para qual processo irá a seguir, ou se irá para o fim, na entrada final;
- Caso não vá para o fim, ele vai para a segunda máquina requerida, é processado, e pergunta novamente qual é o terceiro processo necessário, ou se vai para o fim. O produto realiza esse ciclo até passar por todos os processos requeridos na planilha;
- Quando chega no último setor, o produto é processado na última máquina, e depois vai para o fim do sistema.

O produto poderá assumir qualquer roteiro dentro do chão de fábrica, pois o modelo traz essa flexibilidade. De acordo com as informações inseridas em uma planilha a ser explicada mais adiante, a peça pode ser programada para percorrer qualquer ordem dentro dos setores. Há várias opções de roteamento: a peça pode percorrer de um a seis processos de fabricação disponíveis, em qualquer ordem que seja necessária. Percebe-se que a combinação de roteiros possíveis que uma peça pode percorrer é bastante considerável, tornando o problema mais complexo, o que torna a simulação a ferramenta adequada, visto que a mesma tem a capacidade de simplificar problemas que seriam intratáveis para uma resolução analítica.

Voltando ao mapeamento da Figura 5.3, é possível notar um símbolo novo na modelagem IDEF-SIM: o portão de bloqueio, que foi criado no trabalho de dissertação de Gabriel (2018). Ele consiste na adição desse símbolo aos símbolos da modelagem IDEF-SIM criados anteriormente. No caso do modelo desta tese, o portão indica um bloqueio após a fila já sequenciada, sendo que a liberação acontece quando qualquer uma das filas dos setores zera. Após a passagem de um lote o bloqueio fecha novamente, só abrindo quando uma das filas zerarem de novo, ou seja, passa apenas um lote por vez. O motivo para esse bloqueio é melhor explicado no final deste capítulo.

O modelo evolui dentro do mesmo sistema produtivo, ou seja, o mapeamento do processo produtivo mantém-se constante durante a evolução. Conclui-se que a parte discreta do modelo sempre foi a mesma, mudando apenas a ação do agente.

O modelo final testa a produção dos lotes preenchidos na planilha, mostrados na Tabela 5.1, produzidos na ordem calculada pela lógica MTP. Porém, ele possui uma nova atuação do agente Gerente, que usará uma variação da lógica de Menor Fila Adiante para balancear a produção de duas em duas horas, ou a partir do momento em que houver uma quebra de máquina.

Essas quebras serão aleatórias, com tempos entre quebras e tempos de reparo obedecendo a distribuição exponencial, baseado no trabalho de Mokhtari (2015) que assume que as taxas de quebra de máquinas e tempos de reparo assumem distribuições exponenciais. Os valores dessas medidas também podem ser facilmente inseridos no modelo, de acordo com o que acontece no processo.

Já a ideia do re-sequenciamento periódico foi inspirada em trabalhos como o de Ning *et al.* (2016), que usa a janela de tempo para re-sequenciamento de tempos em tempos para modificações locais nas sequências.

Conforme foi citado anteriormente, cada lógica de sequenciamento busca atingir um objetivo, e que muitas vezes são conflitantes entre si. Os objetivos estão diretamente relacionados aos fatores de competitividade: rapidez, confiabilidade, custo e flexibilidade, ficando de fora o fator qualidade do produto, que não é influenciado pela sequência de produção. Estes objetivos, por sua vez, são genéricos demais para serem medidos, logo, precisam de medidores quantitativos que os traduzem. A fim de medir alguns destes objetivos, nesta pesquisa serão usados: o percentual de utilização das máquinas em cada setor e o *makespan*.

A seguir são explicadas, em detalhes, as funções das lógicas de sequenciamento utilizadas durante a evolução do modelo:

- a) **FIFO**: utilizada no modelo inicial, tem a função de diminuir a variância do tempo de permanência de cada peça na fila, sendo considerada uma regra justa e por isso muito utilizada para filas de pessoas. Porém, em relação aos objetivos de produção, ela tem pouca influência. Logo, tende a ter os piores resultados destes indicadores de produção. Por esta razão foi buscada uma nova lógica de SP inicial que fosse mais eficiente;
- b) **MTP**: tende a melhorar a rapidez e, colateralmente, a confiabilidade, pois maior rapidez de produção tende a diminuir os atrasos. Ela diminui o *makespan*, mas tem um efeito

negativo no nível de utilização das máquinas, pois gera mais janelas de tempo ocioso durante a produção dos lotes. Portanto, deixam a desejar no quesito custo (eficiência produtiva), o que aliás é um grande problema em empresas com sistemas JSF, devido a confusão do fluxo produtivo.

O que melhoraria a utilização das máquinas seria a lógica contrária: maior tempo de processamento, a qual ocuparia as máquinas por mais tempo pois produziria as maiores ordens primeiro. Porém, esta lógica é ruim nos quesitos rapidez e confiabilidade. Conforme pode-se notar, é difícil sequenciar otimizando mais objetivos de produção ao mesmo tempo. Senso assim, o modelo final é também uma tentativa de sequenciamento com múltiplos objetivos, além de ser mais robusto;

- c) **MTP e Menor Fila adiante modificada:** buscando uma maior evolução, o modelo final tem a intenção de melhorar este problema crítico de eficiência produtiva sem perder a rapidez. Para isso, o agente entra mais uma vez em ação, porém agora durante o tempo real de simulação. A cada duas horas ou a cada quebra de máquina, o agente é acionado para balancear esta produção, re-sequenciando as ordens em tempo real. Para isso usará uma variação da lógica de Menor Fila Adiante, explicada anteriormente. Em detalhes, o agente medirá o número de peças na fila de cada setor e dividirá pelo número de máquinas do mesmo, gerando uma razão para cada. A menor razão valor indica o setor que tende a ficar ocioso primeiro. Com essa razão calculada, o agente detectará a menor razão entre os setores e buscará na fila (já sequenciada com o MTP) o primeiro lote cujo primeiro destino é o setor com a menor razão. Este lote assumirá a ponta da fila, balanceando as filas adiante e minimizando a ociosidade. Com isso, o grande problema da lógica MTP será minimizado, uma vez que a lógica Menor Fila Adiante tem a missão de diminuir a ociosidade de processo adiante.

A ideia de utilizar a variação dessa lógica, também para casos de quebras de máquinas, vem da situação em que uma máquina do setor quebra e o próximo lote a ser produzido iria justamente para este setor. Isso acarretaria um maior desbalanceamento e consequente atraso ainda maior na produção, pois o número de máquinas do setor estaria reduzido. E para o caso de ausência de quebras, a lógica é acionada a cada duas horas, visto que existe a necessidade de equilibrar a mão de obra entre os setores, que seria também uma forma de balanceamento da produção para sistemas deste tipo, onde o tradicional balanceamento de linha não ocorre.

5.1.4. Programação

Este item tem a intenção de mostrar os detalhes da programação em Java das lógicas implementadas pelo agente Gerente. Tais lógicas se situam dentro dos estados mostrados nos diagramas das Figuras 5.1 e 5.2. Este não é um item essencial para o entendimento desta tese, porém é fundamental para aqueles que pretendem replicar o modelo ou mesmo entender mais a fundo como ocorre o funcionamento do agente.

A lógica contida dentro do agente funciona como se fosse seu “cérebro”, tudo que o agente pode “pensar” está contido ali. Estes estados podem ser acionados:

- Pela chegada de produtos a serem fabricados, como é o caso da lógica MTP localizada no estado “sequenciando” mostrados nas Figuras 5.1 e 5.2;
- Por um tempo pré-determinado;
- Por algum evento aleatório que ocorra no processo, como uma quebra de máquina, que aliás é um dos mecanismos de acionamento da lógica Menor Fila Adiante localizada no estado “balanceando filas”, mostrado na Figura 5.2.

Para melhor entendimento dos termos utilizados em linguagem Java, mostra-se na Figura 5.4, o *print screen* da tela principal do modelo mostrada na Figura 5.3, que corresponde a modelagem em IDEF-SIM da lógica do processo, explicada no item anterior. É importante frisar que esta tela principal da lógica do processo é a mesma durante a evolução do modelo. O que muda, na verdade, é a ação programada dentro do Gerente. Alguns nomes: “*Sectors*” que são os processos de fabricação; filas como “*EntranceQueue*” e “*BatchQueue*”; entradas como “*InitialEntrance*” e “*RouteEntrance*”; além de saídas como “*Exits*” dos setores e “*ChooseRoute*”, são mostrados nessa figura.

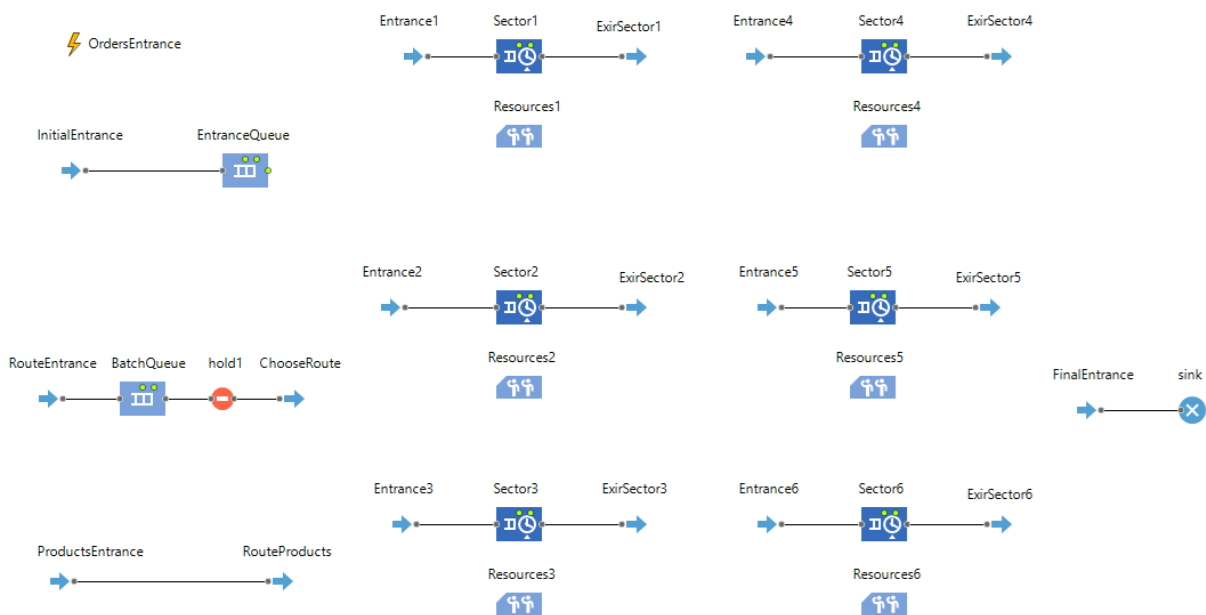


Figura 5.4 - Modelo computacional do SJSF (parte discreta).

Fonte: Elaboração própria

Em síntese ao que foi explicado no item anterior, a fila “*EntranceQueue*” corresponde aos pedidos aguardando para o primeiro sequenciamento, usando o MTP. A fila “*BatchQueue*” corresponde aos lotes já sequenciados, esse é o local onde o agente re-sequenciará a produção esporadicamente usando a variação da lógica Menor Fila Adiante. Em seguida os lotes são desmembrados em produtos e lançados ao processo produtivo através da saída “*ChooseRoute*”. As entradas e saídas dos setores correspondem aos locais onde as lógicas de roteamento dos produtos são executadas, de acordo com o que foi definido na planilha de interface do modelo. Essas lógicas são o que proporcionam flexibilidade de rotas aos produtos. O número de máquinas em cada setor é definido também nessa planilha, em que os agentes recurso transferem esses valores para os *resource pools* de cada setor. Sendo assim, esses números também podem ser facilmente alterados, dando mais flexibilidade ao modelo.

Em toda a explicação das duas lógicas de sequenciamento, quando se expressar a palavra agente, corresponderá ao agente principal do modelo: o Gerente. Usando os termos mostrados na Figura 5.4, o Quadro 5.1 mostra a programação em Java da lógica MTP.

Quadro 5.1 - Lógica MTP.

```
Batch Smaller;
int InitialQuantity = get_Main().EntranceQueue.size();
for (int i = 0; i < QuantityInicial; i++) {
    Smaller = get_Main().EntranceQueue.get(0);
    for (int j = 1; j < get_Main().EntranceQueue.size(); j++) {
        if (get_Main().EntranceQueue.get(j).SumTimes < Smaller.SumTimes) {
            Smaller = get_Main().EntranceQueue.get(j);
        }
    }
    System.out.println("Batch " + (Smaller.line - 2) + " with " + (i+1) + " smaller time with number of batches
remaining: " + get_Main().EntranceQueue.size());
    get_Main().RouteEntrance.take(get_Main().EntranceQueue.remove(Smaller));
}
```

Fonte: Elaboração própria

Esta lógica funciona da seguinte maneira:

- Inicialmente, o agente cria uma variável local chamada de “*smaller*” para o lote;
- O agente, então, verifica na fila “*EntranceQueue*” o número de pedidos (lotes) aguardando para serem produzidos. Essa informação é necessária para saber o número de ciclos a serem realizados;
- Em seguida é feito um ciclo, em que o lote definido como “*smaller*” é o primeiro da fila. Assim, define-se como o lote com MTP como sendo o primeiro;
- Dentro deste ciclo é feito outro ciclo, que compara a tempo total para fabricação de cada lote com o lote definido como “*smaller*”;
- Então, a variável “*smaller*” assume o valor do lote com menor tempo de fabricação, dentro deste segundo ciclo;
- Depois o agente registra o menor lote em ordem crescente, pois esta impressão está dentro do primeiro ciclo. Isso é importante para verificar se o agente está identificando os lotes com menores tempos de forma correta, o que de fato acontece, validando esta parte da ação do agente;
- Em seguida, o agente manda a entrada “*RouteEntrance*” pegar o lote com MTP da fila “*EntranceQueue*”. Assim, o lote irá para uma segunda fila de lotes já sequenciados, a “*BatchQueue*”;
- Finalmente, o ciclo inicial é executado novamente, identificando o próximo menor lote e colocando-o na “*BatchQueue*”, até que todos os lotes tenham acabado.

A próxima lógica Menor Fila Adiante modificada é usada somente no modelo final e fica situada no último estado “balanceando filas”, mostrada no Quadro 5.2.

Quadro 5.2 - Lógica Menor Fila Adiante modificada.

```

double RatioSector1 = get_Main().Sector1.queueSize() / NResources1;
double RatioSector2 = get_Main().Sector2.queueSize() / NResources2;
double RatioSector3 = get_Main().Sector3.queueSize() / NResources3;
double RatioSector4 = get_Main().Sector4.queueSize() / NResources4;
double RatioSector5 = get_Main().Sector5.queueSize() / NResources5;
double RatioSector6 = get_Main().Sector6.queueSize() / NResources6;

if (((RatioSector1 < RatioSector2) && (RatioSector1 < RatioSector3) && (RatioSector1 < RatioSector4) &&
(RatioSector1 < RatioSector5) && (RatioSector1 < RatioSector6)) || (get_Main().Sector1.queueSize() == 0)){
    for (int j = 0; j < get_Main().BatchQueue.size(); j++) {
        if (get_Main().BatchQueue.get(j).FirstMachine == 1) {
            Batch chosen = get_Main().BatchQueue.get(j);
            for (int k = 0; k < chosen.Quantity; k++) {
                get_Main().ProductsEntrance.take(new Product (chosen.TimeSector1,
                    chosen.TimeSector2, chosen.TimeSector3, chosen.line ,chosen.Sequence,
                    chosen.TimeSector4, chosen.TimeSector5, chosen.TimeSector6));
            }
            get_Main().BatchQueue.remove(j);
        }
        break;
    }
}
} else if (((RatioSector2 < RatioSector1) && (RatioSector2 < RatioSector3) && (RatioSector2 < RatioSector4) &&
(RatioSector2 < RatioSector5) && (RatioSector2 < RatioSector6)) || (get_Main().Sector2.queueSize() == 0)){
    for (int j = 0; j < get_Main().BatchQueue.size(); j++) {
        if (get_Main().BatchQueue.get(j).FirstMachine == 2) {
            Batch chosen = get_Main().BatchQueue.get(j);
            for (int k = 0; k < chosen.Quantity; k++) {
                get_Main().ProductsEntrance.take(new Product (chosen.TimeSector1,
                    chosen.TimeSector2, chosen.TimeSector3, chosen.line, chosen.Sequence,
                    chosen.TimeSector4, chosen.TimeSector5, chosen.TimeSector6));
            }
            get_Main().BatchQueue.remove(j);
        }
        break;
    }
}
} else if (((RatioSector3 < RatioSector1) && (RatioSector3 < RatioSector2) && (RatioSector3 < RatioSector4) &&
(RatioSector3 < RatioSector5) && (RatioSector3 < RatioSector6)) || (get_Main().Sector3.queueSize() == 0)){
    for (int j = 0; j < get_Main().BatchQueue.size(); j++) {
        if (get_Main().BatchQueue.get(j).FirstMachine == 3) {
            Batch chosen = get_Main().BatchQueue.get(j);

            for (int k = 0; k < chosen.Quantity; k++) {
                get_Main().ProductsEntrance.take (new Product (chosen.TimeSector1,
                    chosen.TimeSector2, chosen.TimeSector3, chosen.line, chosen.Sequence,
                    chosen.TimeSector4, chosen.TimeSector5, chosen.TimeSector6));
            }
            get_Main().BatchQueue.remove(j);
        }
        break;
    }
}
} else if (((RatioSector4 < RatioSector1) && (RatioSector4 < RatioSector2) && (RatioSector4 < RatioSector3) &&
(RatioSector4 < RatioSector5) && (RatioSector4 < RatioSector6)) || (get_Main().Sector4.queueSize() == 0)){
    for (int j = 0; j < get_Main().BatchQueue.size(); j++) {
        if (get_Main().BatchQueue.get(j).FirstMachine == 4) {
            Batch chosen = get_Main().BatchQueue.get(j);
            for (int k = 0; k < chosen.Quantity; k++) {

```

```

        get_Main().ProductsEntrance.take (new Product (chosen.TimeSector1,
        chosen.TimeSector2, chosen.TimeSector3, chosen.line, chosen.Sequence,
        chosen.TimeSector4, chosen.TimeSector5, chosen.TimeSector6));
    }
    get_Main().BatchQueue.remove(j);
    break;
}
}
} else if (((RatioSector5 < RatioSector1) && (RatioSector5 < RatioSector2) && (RatioSector5 < RatioSector3) &&
(RatioSector5 < RatioSector4) && (RatioSector5 < RatioSector6)) || (get_Main().Sector4.queueSize() == 0)){
    for (int j = 0; j < get_Main().BatchQueue.size(); j++) {
        if (get_Main().BatchQueue.get(j).FirstMachine == 5) {
            Batch chosen = get_Main().BatchQueue.get(j);
            for (int k = 0; k < chosen.Quantity; k++) {
                get_Main().ProductsEntrance.take (new Product (chosen.TimeSector1,
                chosen.TimeSector2, chosen.TimeSector3, chosen.line, chosen.Sequence,
                chosen.TimeSector4, chosen.TimeSector5, chosen.TimeSector6));
            }
            get_Main().BatchQueue.remove(j);
        }
        break;
    }
} else {
    for (int j = 0; j < get_Main().BatchQueue.size(); j++) {
        if (get_Main().BatchQueue.get(j).FirstMachine == 6) {
            Batch chosen = get_Main().BatchQueue.get(j);
            for (int k = 0; k < chosen.Quantity; k++) {
                get_Main().ProductsEntrance.take (new Product (chosen.TimeSector1,
                chosen.TimeSector2, chosen.TimeSector3, chosen.line, chosen.Sequence,
                chosen.TimeSector4, chosen.TimeSector5, chosen.TimeSector6));
            }
            get_Main().BatchQueue.remove(j);
        }
        break;
    }
}
}

```

Fonte: Elaboração própria

Essa lógica funciona da seguinte maneira:

- Primeiramente, o agente calcula as razões de cada setor, dividindo o número de produtos na fila do primeiro setor pelo número de máquinas do mesmo. Ele faz o mesmo processo para todos os outros setores calculando todas as razões;
- Se a razão da fila do primeiro setor for menor que as outras, ele entra no primeiro bloco;
 - Assim o agente entra em um ciclo, verificando na “*BatchQueue*”, que é a fila de lotes já sequenciada pela lógica MTP, qual o primeiro lote dessa fila em que o parâmetro “*FirstMachine*” é igual a 1. O número 1 no modelo corresponde ao primeiro setor de produção no modelo;
 - Em seguida o agente pega esse lote e o remove da “*BatchQueue*”, colocando-o no chão de fábrica;

- Esse lote, então, é desmembrado em produtos, na quantidade definida pelo usuário. Essa transformação ocorre dentro de outro ciclo, que cria um agente “*Product*” com os mesmos parâmetros do agente “*Batch*”, tais como: tempos de processamento em cada máquina e a sequência de máquinas que percorrerá no processo produtivo;
- O agente então termina a lógica interrompendo o restante da mesma com o comando *break*;
- Caso a menor razão entre os setores seja a da fila do segundo setor, o agente entra no segundo bloco, que é parecido com o primeiro. A principal diferença consiste em achar o primeiro lote na “*BatchQueue*” que tem o valor 2 para o parâmetro “*FirstMachine*”, que corresponde ao segundo setor de produção;
- O agente realiza este mesmo mecanismo para todos ou outros setores: calculando as razões, procurando a menor entre elas e achando na “*BatchQueue*” o lote que vai para o setor com a menor razão;
- Feito isso, o lote também é desmembrando em produtos e colocado no processo produtivo, sendo que o comando “*break*” terminará a lógica Menor Fila Adiante modificada em todos os ciclos.

Este item mostrou, em detalhes a programação utilizada pelo agente Gerente nas implementações das lógicas de sequenciamento utilizadas nesta pesquisa. A seguir serão mostrados alguns detalhes do projeto do modelo.

5.1.5. Projeto

A fase de projeto do Protocolo ODD descreve um conjunto de conceitos básicos que são importantes para o modelo, e é composta por:

- Princípios básicos: o Gerente sequenciará a produção obedecendo a lógica programada, além de re-sequenciar em alguns momentos no modelo final;
- Objetivos: o sucesso do Gerente será medido em relação a redução no *makespan* e no aumento no percentual de utilização das máquinas;
- Interação: o agente Gerente poderá receber a mensagem de quebra de máquina, originada dos agentes máquina;
- Estocasticidade: os agentes máquina transformarão os tempos determinísticos em tempos estocásticos;

- Observação: os tempos de processo de cada peça na fila são definidos pelos usuários do modelo antes de começar a produção, normalmente calculados pelo setor de engenharia de processos.

5.1.6. Inicialização

O modelo terá sempre a característica inicial que for programada na planilha da Tabela 5.1: chegará um determinado número de lotes a serem produzidos, definidos na planilha. Cada lote corresponde a um pedido em que a quantidade de peças a serem produzidas é também colocada nesta planilha. Os tempos de fabricação em cada máquina também devem estar inseridos na mesma, bem como o roteiro de produção necessário a fabricação de cada peça.

5.1.7. Dados de entrada

Para considerar estas informações, o modelo possui uma interação com esta planilha do Excel, onde o programador inserirá informações como: em quais processos a peça passará, em que ordem, e o tempo que permanecerá em cada um deles para que a operação seja realizada. A quantidade requerida em cada pedido também é inserida nesta planilha, que é mostrada na Tabela 5.1.

Estes dados são propositalmente simples, com poucas ordens, tempos inteiros e poucos produtos por lote, pois eles serão usados na primeira parte da análise de resultados que é a validação. Para isso, estas mesmas ordens serão programadas manualmente por meio do gráfico de Gantt e seus resultados serão comparados com os do modelo. Sendo assim, dados com muitas ordens, muitos produtos por ordem e tempos decimais inviabilizariam a construção do gráfico. Porém, um problema bem mais complexo será apresentado na segunda parte da análise dos resultados.

Tabela 5.1 - Dados de entrada do modelo.

Ordens	Tempo (minutos)						Sequência						Número de produtos	Soma tempos	Setor	Número de máquinas
	Setor 1	Setor 2	Setor 3	Setor 4	Setor 5	Setor 6	1°	2°	3°	4°	5°	6°				
1	10	15	0	0	0		1	2					5	125	1	3
2	12	9	13	0	0		2	1	3				3	102	2	4
3	0	0	0	0	12		5						7	84	3	3
4	8	0	0	12	10		5	4	1				5	150	4	2
5	0	17	11	0	0		2	3					4	112	5	4
6	8	5	12	11	5		4	2	1	3	5		2	82	6	0
7	9	0	0	0	0		1						10	90		
8	0	8	4	13	8		3	5	4	2			3	99		

Fonte: Elaboração própria

A primeira coluna representa os pedidos de empresas clientes na ordem que chegaram. As seis colunas seguintes representam os tempos que cada produto daquele lote gastará em cada máquina. Quando o tempo for zero significa que a peça não necessitará passar por aquele processo. Depois dos tempos, as seis colunas de sequência, indicam a sequência de processo que a peça passará, pois ela pode passar pelos mesmos processos, mas em ordens diferentes devidos a necessidades diferentes de cada uma. A peça também pode passar por apenas um, dois, três, quatro ou cinco processos, bastando deixar as colunas de sequência e os tempos sem serem preenchidas. A seguir vem uma coluna com o número de unidades requeridas por pedido. E finalmente, uma coluna que multiplica o tempo total para fabricar cada peça com o número de unidades requeridas, que será usada em uma das lógicas de sequenciamento utilizadas pelo agente Gerente.

5.1.8. Submodelos

Conforme foi citado anteriormente, os submodelos correspondem a uma apresentação dos detalhamentos dos processos. Procura-se mostrar a manipulação dos parâmetros dos agentes dentro do processo. No caso desta pesquisa, mostra-se como os agentes “*Product*” decidem sobre seus destinos dentro do processo. São lógicas simples mas eficazes, que acabam por reproduzir um comportamento complexo de ser modelado sem a aplicação dessas lógicas.

Muitos dos parâmetros utilizados pelas peças são retirados da planilha mostrada na Tabela 5.1 do item anterior. Todos os quatro parâmetros de tempo (“*TimeSector1*” ao “*TimeSector6*”), os parâmetros “*Quantity*” e “*line*”, além do parâmetro “*FirstMachine*” utilizado na lógica Menor Fila Adiante do Gerente, são inseridos no modelo através de um evento chamado de “*OrdersEntrance*” mostrado na Figura 5.4, a lógica em Java dessa inserção de parâmetros é mostrada no Quadro 5.3.

Quadro 5.3 - Evento de entrada do lote no modelo

```

for (int i = 2; i < (int) Orders.getLastRowNum(1); i++) {
    int line = i+1 ;
    double TimeSector1 = Orders.getCellNumericValue (1, line, 2);
    double TimeSector2 = Orders.getCellNumericValue (1, line, 3);
    double TimeSector3 = Orders.getCellNumericValue (1, line, 4);
    double TimeSector4 = Orders.getCellNumericValue (1, line, 5);
    double TimeSector5 = Orders.getCellNumericValue (1, line, 6);
    double TimeSector6 = Orders.getCellNumericValue (1, line, 7);
    int Quantity = (int) Orders.getCellNumericValue (1, line, 14);
    double SumTime = Orders.getCellNumericValue (1, line, 15);
    int FirstMachine = (int) Orders.getCellNumericValue (1, line, 8);
    InitialEntrance.take (new Batch(TimeSector1, TimeSector2, TimeSector3, TimeSector4, TimeSector5,
        TimeSector6, line, 1, SumTime, Quantity));
}

```

Fonte: Elaboração própria

O parâmetro “*line*” é um parâmetro de localização dentro desta própria lógica do Quadro 5.3, ela identifica em que linha da Tabela 5.1 estão localizados os parâmetros. Os parâmetros de tempo serão utilizados pelos agentes máquinas no momento da fabricação. Os parâmetros “*SumTime*” e “*FirstMachine*” são utilizados nas lógicas de sequenciamento e re-sequenciamento do agente Gerente. Já o parâmetro “*Quantity*” é usado na lógica de quebra do lote em produtos na saída “*ChooseRoute*”, mostrada no Quadro 5.4.

Quadro 5.4 - Quebra do lote em produtos.

```

for (int k = 0; k < agent.Quantity; k++) {
    ProductsEntrance.take (newProduct (agent.TimeSector1, agent.TimeSector2, agent.TimeSector3,
        agent.TimeSector4, agent.TimeSector5, agent.TimeSector6, agent.line, agent.Sequence));
}

```

Fonte: Elaboração própria

Além disso, mostra-se neste item, a lógica de movimentação dos agentes “*Product*” no processo. Tal lógica é o que traz a flexibilidade de movimentação desse agente dentro do modelo, que faz possível que o roteiro de fabricação já seja programado dentro da própria

planilha de interface. Essa lógica é mostrada no Quadro 5.5 e está localizada dentro das saídas dos setores.

Quadro 5.5 - Lógica de movimentação dos produtos no modelo.

```
agent.Sequence++;
if (!(Orders.cellExists(1, agent.line, (agent.Sequence + 7))) || (agent.Sequence == 7)) {
    FinalEntrance.take(agent);
} else if (Orders.getCellNumericValue(1, agent.line, (agent.Sequence + 7)) == 1) {
    Entrance1.take(agent);
} else if (Orders.getCellNumericValue(1, agent.line, (agent.Sequence + 7)) == 2) {
    Entrance2.take(agent);
} else if (Orders.getCellNumericValue(1, agent.line, (agent.Sequence + 7)) == 3) {
    Entrance3.take(agent);
} else if (Orders.getCellNumericValue(1, agent.line, (agent.Sequence + 7)) == 4) {
    Entrance4.take(agent);
} else if (Orders.getCellNumericValue(1, agent.line, (agent.Sequence + 7)) == 5) {
    Entrance5.take(agent);
} else if (Orders.getCellNumericValue(1, agent.line, (agent.Sequence + 7)) == 6) {
    Entrance6.take(agent);
} else {
    FinalEntrance.take(agent);
}
```

Fonte: Elaboração própria

O parâmetro “*Sequence*” foi gerado na lógica de criação dos produtos do Quadro 5.4. Inicialmente, ele assume o valor 1 do agente “Batch”, gerado no Quadro 5.3. Ele é um dos parâmetros que localizam as células na planilha de interface. A localização em Java é feita no modo (aba, linha, coluna). Os outros três parâmetros finais do agente “Batch” não serão usados pelo agente “*Product*”.

A lógica funciona da seguinte maneira:

- O parâmetro “*Sequence*” é incrementado no início porque esta lógica fica nas saídas de cada setor, depois de ter passado pelo processo. Logo, ele serve para adiantar uma coluna na planilha de interface;
- O primeiro “*if*” indica que, se essa célula da planilha está vazia ou se o valor do parâmetro “*sequencia*” for igual a 7, o agente “*Product*” vai para o fim do modelo, no “*FinalEntrance*”;
- Os outros seis “*else if*” indicam que, se a célula tem valor: 1, 2, 3, 4, 5 ou 6, ela vai para o setor correspondente. Assim, como o “*Sequence*” é incrementado em 1 no começo da lógica e o valor de “*line*” só varia de lote para lote, a lógica sempre manterá a linha constante (quando os produtos pertencerem ao mesmo lote), e pegará os valores das colunas: 9, 10, 11, 12 ou 13 da Tabela 5.1.

Há uma programação similar ao Quadro 5.5 na saída “*RouteProducts*”. A diferença é que a primeira linha da programação não existe, e a coluna é sempre a 8 em vez de “*Sequence + 7*”.

Existe, também, um bloqueio antes da entrada dos lotes no chão de fábrica. Neste local, o agente libera um lote por vez para o chão de fábrica, sempre que uma das filas de um dos setores zera. Isso é necessário porque se todas as ordens fossem para as filas dos processos de uma só vez, o primeiro lote sendo produzido iria para o fim da fila do próximo setor, que estaria lotada com todas as ordens a serem produzidas. Assim, os resultados do modelo seriam sempre os mesmos, não importando a lógica de sequenciamento utilizada. Além disso, não haveria sentido em balancear as filas nos setores do chão de fábrica, pois todas estariam lotadas com as todas ordens a serem produzidas.

Finalmente, com este item de submodelos, termina-se a descrição do modelo por meio do Protocolo ODD. Dessa forma, todas as lógicas do modelo foram explicadas e a replicação do mesmo torna-se viável para leitores da tese, caso queiram seguir essa pesquisa adiante após o término da mesma.

O capítulo 4, explicou em detalhes o modelo segundo o Protocolo ODD, que teve a intenção mostrar como foi construído o mesmo e sua evolução durante o processo de pesquisa. O item a seguir trata da validação deste modelo, a fim de constatar se ele realmente traz as respostas esperadas, de acordo com as ordens a serem produzidas.

6. VALIDAÇÃO DO MODELO COMPUTACIONAL

Apesar de se tratar de um modelo genérico, que tem a intenção de ser usado para diversos sistemas de produção que se inserem dentro sistema JSF, o modelo precisa passar por algum tipo de validação. Essa validação consistirá de uma produção fictícia de oito lotes, já mostrados na Tabela 5.1. Funciona como se as ordens a serem produzidas fossem calculadas com esses tempos e inseridas na planilha. Conforme foi citado, o setor de projeto do processo deve se incumbir de determinar a rota entre os setores e os tempos a serem gastos de acordo com o desenho da peça.

Os resultados gerados pelo modelo (*makespan* e utilizações dos cinco setores) devem ser comparados estatisticamente aos resultados calculados por meio do gráfico de *Gantt*, a ser mostrado na Figura 6.1. Sendo assim, é mostrado um exemplo que utiliza cinco setores com poucas máquinas, usando tempos inteiros de processo. Isso é feito para facilitar a construção do gráfico de *Gantt* pois com um número de ordens alto, muitas máquinas e tempos decimais de processo, seria inviável a construção do gráfico. A utilização deste gráfico para problemas muito complexos é impraticável, exaltando a importância de uma plataforma genérica de SP para viabilizar esta atividade.

A Figura 6.1 mostra o gráfico de *Gantt*, em que cada cor corresponde a um pedido de compra. Os pedidos estão classificados com a mesma lógica do modelo: MTP. Foi possível constatar a importância do modelo computacional, devido à simplificação de um problema complexo trazido pela simulação, pois para um caso simples de apenas algumas ordens e máquinas encontraram-se dificuldades consideráveis para preencher o gráfico. Ao programar estas ordens manualmente abre-se margens de erro mesmo com um problema simples, pois qualquer erro na sequência, soma de tempos ou negligência de qualquer produto é suficiente para alterar o gráfico completamente, devendo este ser reiniciado.

Logicamente, foram usados os mesmos números de setores e máquinas do modelo computacional, para fins comparativos. Ou seja, cinco setores com: 3, 4, 3, 2 e 4 máquinas em cada setor, respectivamente.

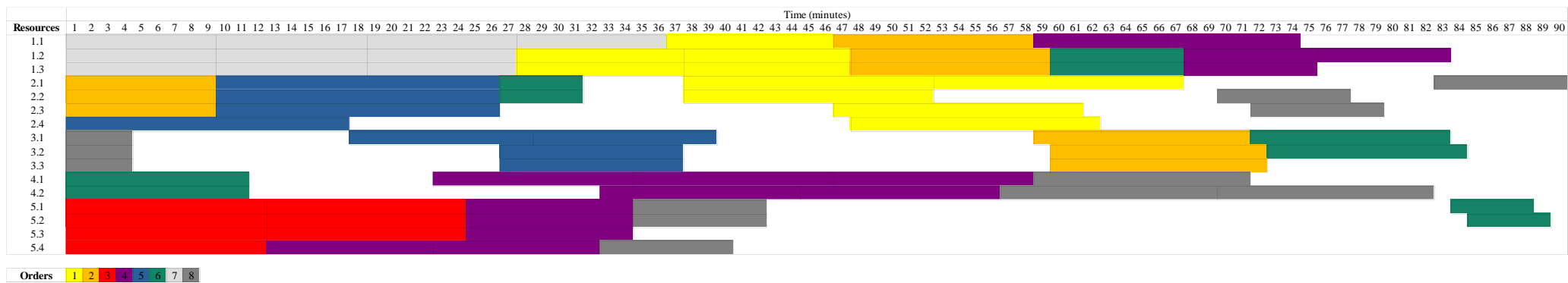


Figura 6.1 - Gráfico de *Gantt* das oito ordens produzidas

Fonte: Elaboração própria

O gráfico mostra que o *makespan* foi de 90 minutos, pois nesse tempo todas as ordens foram finalizadas.

O percentual de utilização de cada setor foi calculado dividindo-se o tempo em que as máquinas estão sendo utilizadas pela capacidade de produção de cada setor, que é o número de máquinas do setor vezes 92 minutos (tempo definido para parar a simulação que é necessário para terminar todas as ordens em todas réplicas na simulação). Para fins comparativos entre *Gantt* e o modelo, esse valor precisa ser o mesmo.

Em seguida, esses mesmos pedidos foram inseridos na planilha de interface para simular os resultados médios no modelo (após 10 réplicas). A variação causada pela distribuição triangular nas réplicas destina-se a tornar o modelo mais real. Possivelmente, essa variação não será a mesma do processo real, mas se evitará valores determinísticos, o que desviaria ainda mais o modelo da realidade.

Os resultados calculados em *Gantt* para utilização dos setores e o *makespan* estão na Tabela 6.1, assim como os resultados gerados pelo modelo.

Tabela 6.1 - Resultados calculados por *Gantt* e os obtidos no modelo

	Número de máquinas	Capacidade disponível	Tempo utilizado	Utilização e <i>makespan</i> em <i>Gantt</i>	Utilização e <i>makespan</i> no modelo	Desvio padrão no modelo
Setor 1	3	276	229	84,06	84,06	0,76
Setor 2	4	368	204	55,43	55,58	0,55
Setor 3	3	276	113	43,12	42,99	0,46
Setor 4	2	184	121	65,76	65,22	0,85
Setor 5	4	368	168	45,65	45,62	0,53
<i>Makespan</i>	-	-	-	90,00	90,42	0,74

Fonte: Elaboração própria

Percebe-se que os valores médios gerados pelo modelo estão muito próximos aos calculados pelo gráfico de *Gantt*. Contudo, para validação do modelo é necessária a comparação estatística de fato. Porém, antes da comparação estatística foram feitos os testes de normalidade para os seis conjuntos de dados gerados pelo modelo, mostrados na Figura 6.2.

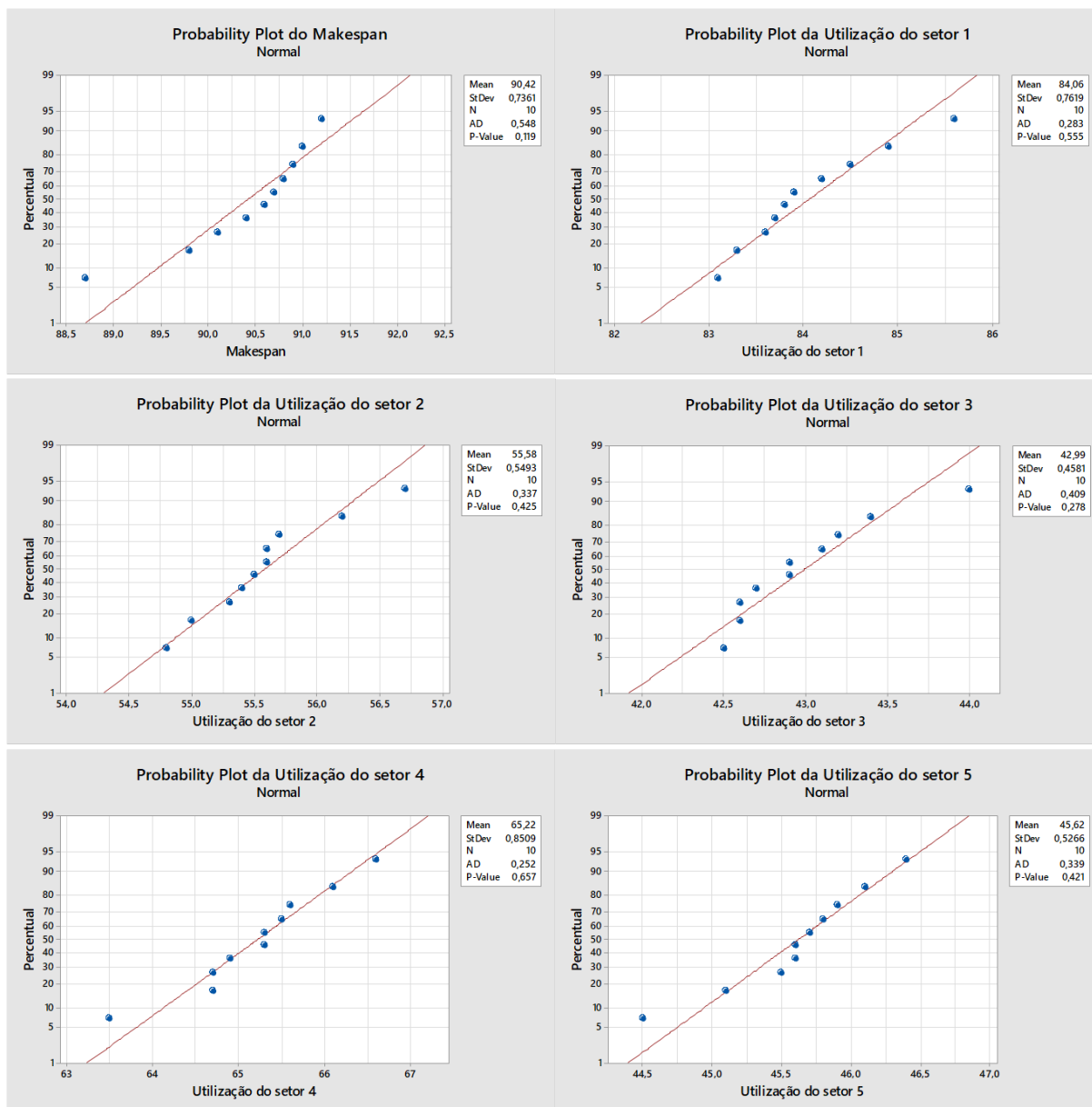


Figura 6.2 - Testes de normalidade dos dados gerados pelo modelo

Fonte: Elaboração própria

Através dos *P-values* pôde-se comprovar a normalidade dos seis conjuntos de dados de resposta gerados pelo modelo. Em seguida, são mostradas as comparações estatísticas dos dados estocásticos gerados pelo modelo e os dados determinísticos calculados pelo gráfico de *Gantt* na tabela 6.2.

Tabela 6.2 - Comparações estatísticas entre resultados do modelo e gráfico de *Gantt*

Respostas	Teste de Hipóteses	Média	Desvio padrão	Intervalo de 95% de confiança para a média	T-Value	P-Value
Makespan (minutos)	Ho: $\mu = 90$ H ₁ : $\mu \neq 90$	90,42	0,736	(89,893; 90,947)	1,80	0,105
Utilização setor 1 (%)	Ho: $\mu = 84,06$ H ₁ : $\mu \neq 84,06$	84,06	0,762	(83,515; 84,605)	0,00	1,000
Utilização setor 2 (%)	Ho: $\mu = 55,43$ H ₁ : $\mu \neq 55,43$	55,58	0,549	(55,187; 55,973)	0,86	0,410
Utilização setor 3 (%)	Ho: $\mu = 43,12$ H ₁ : $\mu \neq 43,12$	42,99	0,458	(42,662; 43,318)	-0,90	0,393
Utilização setor 4 (%)	Ho: $\mu = 65,76$ H ₁ : $\mu \neq 65,76$	65,22	0,851	(64,611; 65,829)	-2,01	0,076
Utilização setor 5 (%)	Ho: $\mu = 45,65$ H ₁ : $\mu \neq 45,65$	45,62	0,527	(45,243; 45,997)	-0,18	0,861

Fonte: Elaboração própria

Realizando o teste *One Sample-T*, que de acordo com Montgomery (2005) é usado para comparar uma série a um valor determinístico, é possível constatar através dos *P-values* mostrados na Tabela 6.2, que não foi possível rejeitar a hipótese de igualdade dos indicadores, pois os *P-values* são consideravelmente maiores do que 0,05.

O segmento de reta embaixo dos *Boxplot* são os intervalos de 95% de confiança. Percebe-se, que todos os valores determinísticos calculados para os indicadores se encontram dentro dos intervalos de confiança. Logo, o modelo comprova que reproduz o resultado esperado e está validado.

A Figura 6.3 ilustra as comparações *One Sample-T* através dos gráficos *Boxplots*, demonstrando graficamente as comparações estatísticas de todos os resultados.

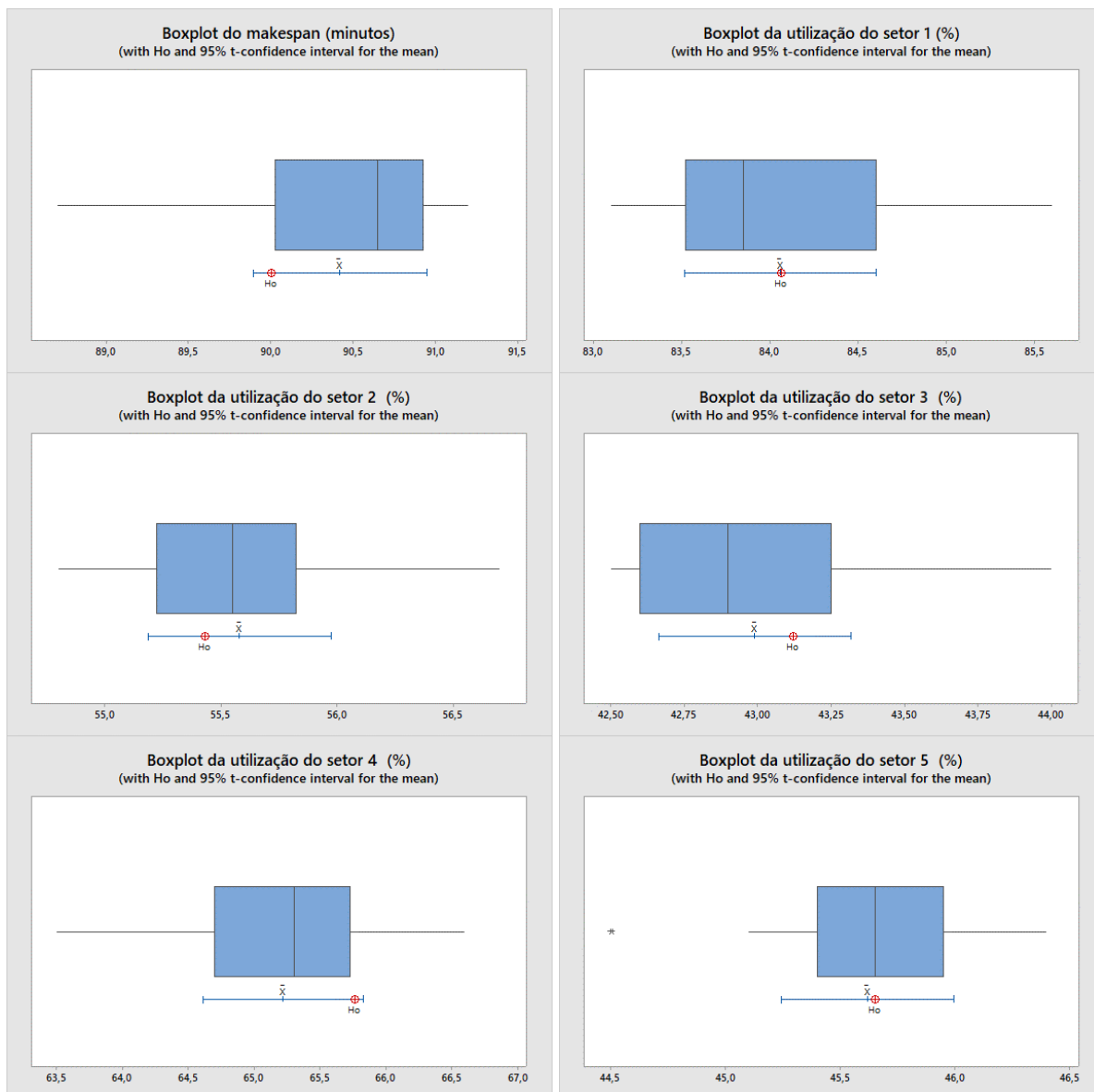


Figura 6.3 - Boxplot dos testes *One Sample-T*.

Fonte: Elaboração própria

Considerando os resultados, é possível notar que o agente foi eficiente implementando a lógica de SP em um modelo simulado. Isso porque os testes *One Sample T* entre os resultados de *Gantt* e os gerados pelo modelo falharam em rejeitar a hipótese nula de que os valores são iguais.

Depois de validado, outras lógicas podem ser incorporadas a este modelo, apenas programando-as dentro do diagrama de estados do agente Gerente. Isto pode ser considerado como um teste “*what if*”, muito popular na área da simulação.

Este capítulo demonstrou que o modelo foi feito e executado com sucesso, e pode ser utilizado para produções de ordens diferentes das que foram utilizadas nesta tese, além de poder

utilizar números diferentes de máquinas e de setores também. Isto gera flexibilidade ao modelo computacional perante a diferentes sistemas dentro do sistema produtivo JSF.

A criação deste modelo foi feita com o intuito de responder às dificuldades e limitações apontadas na literatura. Este capítulo trouxe uma validação estatística dos resultados do modelo feito nesta pesquisa.

É importante ressaltar que esta validação usou um número menor de máquinas por setores e de ordens de produção, além de um número baixo de quantidade em cada ordem e tempos de processos inteiros. Isto foi feito para viabilizar a construção do gráfico de *Gantt* a fim de validar o modelo, pois um número elevado nesses parâmetros geraria um gráfico gigantesco e com muita margem para erros no mesmo. O gráfico traz uma visão do que acontece no processo real, visando mostrar que o modelo gera os resultados reais. Também deve-se frisar que o gráfico de *Gantt* é incapaz de demonstrar o tempo de reparo e a quebra de máquinas aleatória que o modelo final reproduz. Além do fato de o *makespan* ter um valor baixo, sendo que uma quebra de máquina já superaria este tempo e inviabilizaria a validação. Por estas razões, a validação foi feita com o modelo sem quebras de máquinas.

7. ANÁLISE DOS RESULTADOS: TESTES DO MODELO FINAL

Para demonstrar e comprovar as vantagens do re-sequenciamento com relação a robustez e a busca de múltiplos objetivos, serão mostrados os resultados vantajosos do re-sequenciamento no modelo final por meio de um teste *what if* com e sem essa característica.

Este capítulo mostra a comparação estatística através do teste “*Two sample T*”, que de acordo com Montgomery (2005), compara duas séries de valores da mesma medida. O teste foi feito escolhendo-se a opção “*greater than*”, que comprova se uma série é, de fato, maior ou não que outra. Em outras palavras, foram feitas dez replicações do modelo com e sem re-sequenciamento e foram comparados o *makespan* e os percentuais de utilização. Isso foi feito considerando que os tempos de processo calculados, que são determinísticos, foram transformados em tempos estocásticos no modelo por meio da distribuição triangular, que normalmente é usada quando existe uma ideia subjetiva da população, por meio dos seus extremos e da sua moda.

Foram utilizadas 10 réplicas porque a partir desse valor a média tende a se estabilizar, mesmo utilizando a resposta que mais varia no modelo, o *makespan*. A Figura 7.1 mostra a evolução do valor médio dessa resposta recalculado a cada réplica.

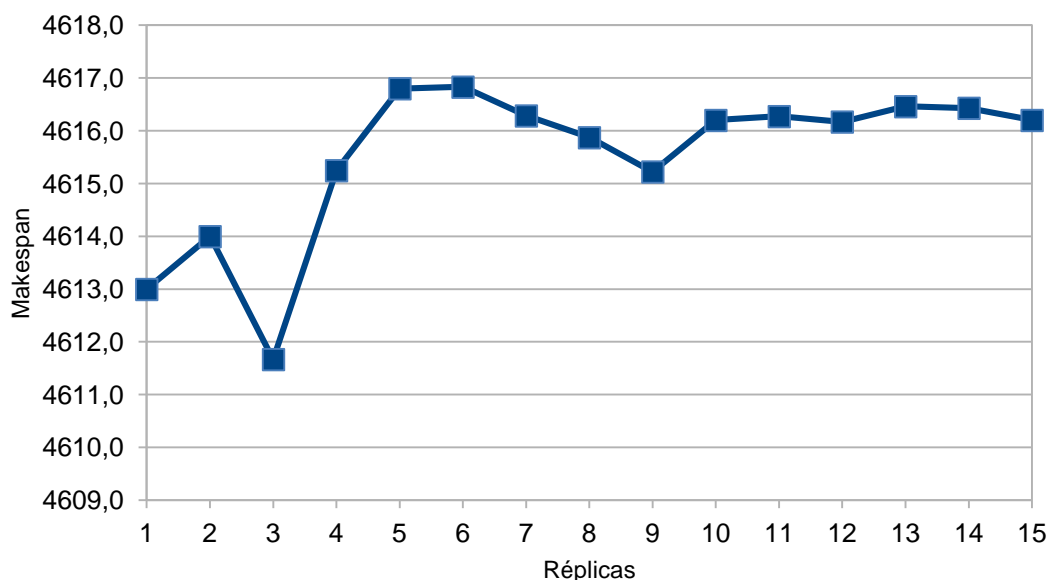


Figura 7.1 - Valor do *makespan* (minutos) médio durante as réplicas.

Fonte: Elaboração própria

Nesta pesquisa, usou-se um percentual de mais ou menos 10 % como valores mínimo e máximo. Este percentual pode ser facilmente modificado no modelo de acordo com o que o gerente real acha pertinente ao seu processo. A moda é o valor que o modelo retira da planilha de interface. Isso significa dizer que se um tempo foi calculado como 20 minutos, a distribuição o consideraria como uma triangular com valores mínimo e máximo 18 e 22, respectivamente. Isso é importante pois, na realidade, os tempos nunca são exatamente como calculados e o sequenciamento deve ser eficiente perante essa incerteza. Isso garante que cada peça, de um mesmo lote, tenha um tempo diferente ao passar pelas máquinas, mesmo que o valor da moda seja o mesmo.

As comparações foram feitas utilizando-se um número de ordens maior, com maiores quantidades, tempos de processo decimais e aleatórios. Ou seja, um problema bem mais complexo que o da validação, que teve que ser simples para viabilizar a construção do gráfico de *Gantt*. Como o modelo tem um *makespan* bem mais longo, foram adicionadas quebras de máquinas aleatórias na simulação com e sem re-sequenciamento.

Seguindo as recomendações de Mokhtari (2015), a quebra assumiu valores de tempo de reparo e tempo entre quebras obedecendo a distribuição exponencial. Além do autor citado, o próprio manual de usuário do *software* de simulação *Anylogic* explica essa distribuição como sendo muito utilizada para tempos de manutenção, apesar dessa referência não ser científica. É importante salientar que não foi utilizado o tempo entre quebras e sim tempo inicial de quebra, pois definiu-se que cada máquina poderia quebrar apenas uma vez durante a produção dos lotes, o que seria mais realista. A distribuição exponencial usa um termo denominado de “lambda” (λ) que suaviza ou atenua a curva exponencial, deixando os valores mais aleatórios ou mais precisos. O valor *default* é $\lambda = 1$, porém quando reduzido deixa os valores gerados mais aleatórios. Este trabalho usou $\lambda = 0,5$ para tempo das quebras iniciais para distribuir mais estes valores gerados e 1 para tempos de reparo, que poderiam ser mais concentrados. Os valores dos tempos de reparo para cada tipo de máquina e os tempos de quebras iniciais utilizados no modelo estão mostrados na Tabela 7.1.

Tabela 7.1 - Tempos de reparo e tempos para quebras iniciais (minutos).

	Setor 1	Setor 2	Setor 3
Tempo inicial de quebra	Exponencial (0.5, 600)	Exponencial (0.5, 900)	Exponencial (0.5, 1300)
Tempo médio de reparo	Exponencial (1, 90)	Exponencial (1, 60)	Exponencial (1, 40)

Fonte: Elaboração própria

A planilha de dados de entrada com as informações das ordens utilizadas para essas comparações estatísticas dos resultados com e sem re-sequenciamento é mostrada na Tabela 7.2. É importante ressaltar que esta é uma planilha de dados fictícios, visando simular um sistema JSF. Contudo, estes dados podem ser modificados e inseridos dados de um sistema real, como tempos, rotas e quantidades reais, sendo este o motivo da criação desta planilha de interface.

Tabela 7.2 - Ordens a serem produzidas.

Pedido	Tempo (min)			Sequência			Número de unidades	Soma Tempos (minutos)	Setores	Número de máquinas
	Setor 1	Setor 2	Setor 3	1º	2º	3º				
1	10,1	15,3	0	1	2		66	1676,4	1	4
2	12,4	8,8	30,6	2	3	1	42	2175,6	2	5
3	0	14,5	5,4	3	2		210	4179	3	3
4	0	20,7	0	2			51	1055,7	4	
5	18	17,9	12,3	1	2	3	60	2892	5	
6	12,6	0	0	1			63	793,8	6	
7	55,9	22,1	40,7	2	1	3	3	356,1		
8	12	0	16,1	3	1		30	843		
9	14,3	17,2	16,8	2	3	1	36	1738,8		
10	18,7	16,4	11,5	1	2	3	42	1957,2		
11	0	20,9	0	2			150	3135		
12	15,1	11	7,9	1	2	3	30	1020		
13	0	10,3	25,6	3	2		99	3554,1		
14	23,9	10,8	15,6	2	3	1	66	3319,8		
15	30,1	0	0	1			45	1354,5		
16	8	0	22,3	3	1		18	545,4		
17	0	10,5	23,3	2	3		123	4157,4		
18	14,8	0	0	1			90	1332		
19	16,6	9,7	13,2	3	1	2	60	2370		
20	6,2	8,9	9,7	1	2	3	30	744		

Fonte: Elaboração própria

Os testes “*Two Sample T*” foram reunidos na Tabela 7.4 e mostram que as hipóteses de igualdade foram rejeitadas, concluindo que os percentuais de utilizações de todos os setores do modelo com re-sequenciamento são maiores do que sem, além do *makespan* que reduziu. O que prova isso são os *P-values*, que são menores que 0,05 usando a opção “*greater than*”.

Os testes foram feitos se considerando variâncias iguais, pois os testes “*Two Variances*” foram feitos para todos os conjuntos de dados e as hipóteses nulas de igualdade não foram

rejeitadas, conforme os *P-values* mostrados na Tabela 7.3. Logo, foi checada a opção “*assume equal variances*” dentro dos testes “*Two Sample T*”.

Tabela 7.3 - Comparações estatísticas das variâncias das respostas com e sem re-sequenciamento.

Teste de hipótese "Two Var"	P-value
Variância da utilização do setor 1 com e sem re-sequenciamento	0,555
Variância da utilização do setor 2 com e sem re-sequenciamento	0,816
Variância da utilização do setor 3 com e sem re-sequenciamento	0,515
Variância do <i>makespan</i> com e sem re-sequenciamento	0,776

Fonte: Elaboração própria

O *makespan* e as utilizações mostraram uma melhoria significativa, que são mostradas pelas estimativas de diferença. A diminuição do *makespan* é um fator positivo e mostra uma evolução no objetivo rapidez. O aumento da utilização é, logicamente, um fator positivo e mostra uma evolução no objetivo eficiência produtiva, ou seja, custo.

Tabela 7.4 - Comparações estatísticas do modelo com e sem re-sequenciamento.

Respostas	Teste de Hipóteses	Média	Desvio padrão	Estimativa de diferença	T-Value	P-Value (greater than)
Makespan (minutos)	H ₀ : $\mu_1 = \mu_2$	$\mu_1 = 4616,2$	$desv_1 = 6,46$	237,5	85,42	0,000
	H ₁ : $\mu_1 > \mu_2$	$\mu_2 = 4378,7$	$desv_2 = 5,96$			
Utilização setor 1 (%)	H ₀ : $\mu_1 = \mu_2$	$\mu_1 = 60,34$	$desv_1 = 0,158$	-3,36	-52,83	0,000
	H ₁ : $\mu_2 > \mu_1$	$\mu_2 = 63,7$	$desv_2 = 0,125$			
Utilização setor 2 (%)	H ₀ : $\mu_1 = \mu_2$	$\mu_1 = 67,03$	$desv_1 = 0,116$	-3,68	-67,19	0,000
	H ₁ : $\mu_2 > \mu_1$	$\mu_2 = 70,71$	$desv_2 = 0,129$			
Utilização setor 3 (%)	H ₀ : $\mu_1 = \mu_2$	$\mu_1 = 94,72$	$desv_1 = 0,0632$	-5,13	-197,5	0,000
	H ₁ : $\mu_2 > \mu_1$	$\mu_2 = 99,85$	$desv_2 = 0,0527$			

Fonte: Elaboração própria

A superioridade estatística é comprovada pelos *P-values* que são muito menores que 0,05, pois mesmo quando se usa três casas decimais os *P-values* têm valor zero. Tendo, assim, evidências suficientes para rejeitar a hipótese nula de igualdade e ir para a hipótese alternativa de superioridade.

Conforme explicado, o modelo sem re-sequenciamento usa a lógica do MTP implementada pelo Gerente em sua atividade de sequenciamento. Essa lógica tende a reduzir o *makespan*, mas não se preocupa com as utilizações.

No modelo com re-sequenciamento o agente também usa a lógica MTP, mas tende a balancear as filas dos setores de produção usando a lógica de Menor Fila Adiante modificada mostrando, assim, resultados melhores. Percebe-se que esta atividade praticamente eliminou as janelas de ociosidade no setor gargalo (setor 3). Isso confirma a teoria de que esse comportamento do agente pode minimizar o grande fator negativo do uso da lógica MTP: a baixa eficiência produtiva. Isso traz um resultado positivo no fator competitivo de rapidez, atingindo uma produtividade razoável, e não deixando tanto a desejar no fator de competitividade custo.

A Figura 7.2 mostra a redução do *makespan* e os aumentos nas utilizações com o uso de re-sequenciamento.

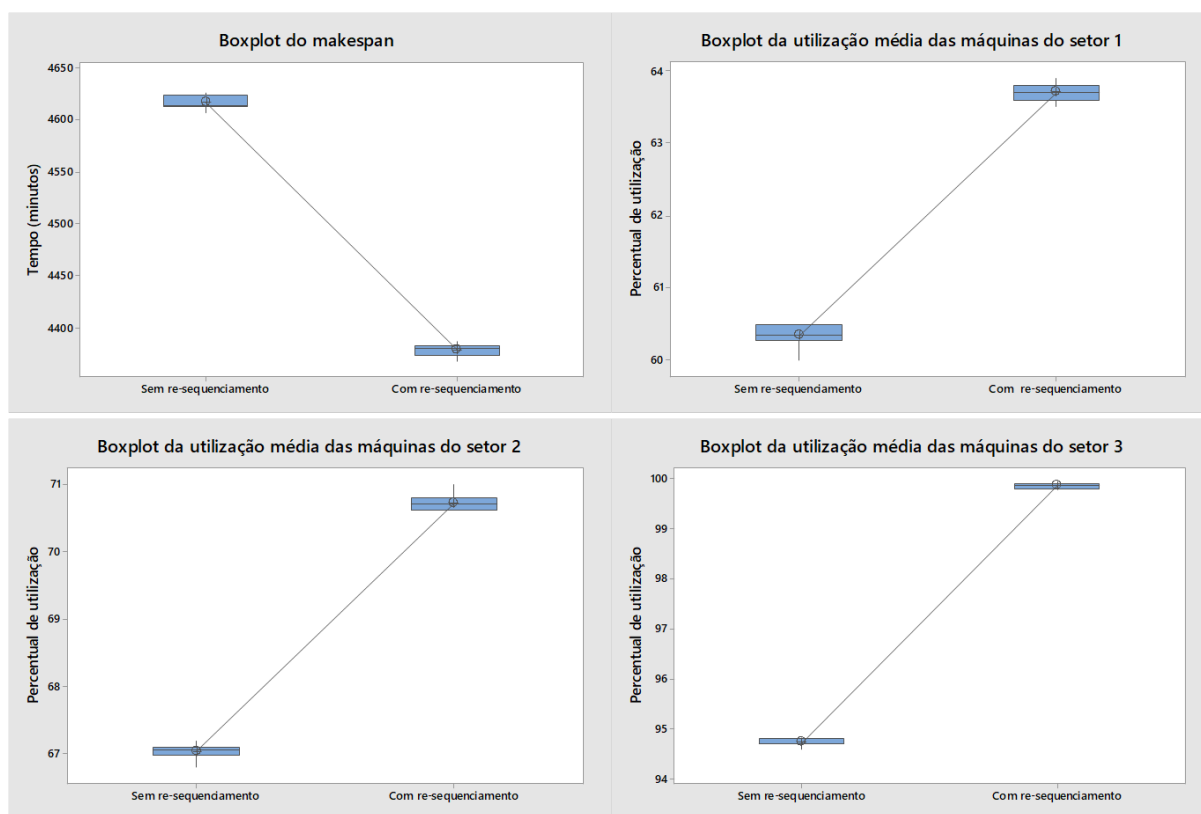


Figura 7.2 - Boxplot do *makespan* e utilizações dos setores com e sem re-sequenciamento.

Fonte: Elaboração própria

Com os gráficos de *boxplot* é possível notar, visualmente, uma redução no *makespan* e um aumento nas utilizações com o re-sequenciamento. Ficando comprovado, estatisticamente, a evolução em todos os resultados produtivos.

Macedo (2012) constatou que, mesmo conhecendo-se com exatidão o percurso que as ordens de produção seguem e os tempos de processamento em cada seção, é muito difícil prever o estado de ocupação de cada setor. Sendo essa uma das maiores preocupações do problema de

planeamento e controle de operações em SJSF. O modelo gerado aqui tem como uma de suas características, a de prever o percentual de utilização de cada setor baseado nas ordens a serem produzidas.

7.1. Passo a passo para execução do sequenciamento

Este item traz uma descrição do passo a passo para se utilizar o modelo pronto de sequenciamento. Ele deve gerar os resultados na simulação que o sistema real geraria quando a produção de fato ocorrer, se o gerente do sistema real utilizar a mesma política de sequenciamento do agente Gerente. É importante lembrar que um *job* se trata de uma ordem de produção customizada de um lote de peças iguais entre si. As etapas a serem seguidas para utilização do modelo de sequenciamento são mostradas a seguir:

- Definir o número de setores do SJSF;
- Definir o número de máquinas em cada setor;
- Preencher esse número de máquinas e setores na planilha de interface;
- Preencher o número de *jobs* a serem fabricados e o número de peças em cada *job*;
- Baseado no desenho da peça, projetar o processo de fabricação, avaliando em quais setores cada *job* deverá passar, e em que ordem (Rota entre os processos);
- Também baseado no desenho da peça, calcular o tempo em que cada peça do *job* gastará em cada máquina;
- Preencher, na planilha de interface, as informações dos dois passos anteriores;
- Simular o modelo e analisar o *makespan* e os percentuais de utilizações dos setores;
- Tomar alguma ação corretiva se os resultados não forem agradáveis, como remanejamento de operadores entre os setores, por exemplo.

8. CONCLUSÕES

8.1. Verificação dos objetivos e solução da questão de pesquisa

Conforme citado na introdução desta tese, a simulação é uma importante ferramenta para tratar de problemas de sequenciamento, tendo em vista uma de suas principais virtudes, a característica de trabalhar bem com problemas complexos.

O objetivo principal desta pesquisa foi resolver problemas tais como: falta de robustez perante problemas cotidianos, utilização de tempos determinísticos e foco em apenas um objetivo de produção. Estes objetivos foram atingidos, pois conseguiu-se superar todas essas limitações e dificuldades devido a ação do agente Gerente, mostrando como o mesmo pode ser eficaz na resolução deste tipo de problema. O modelo, então, atingiu um sequenciamento mais robusto às quebras de máquinas e usou tempos estocásticos de processo, tornando o modelo mais próximo ao que acontece na realidade. A robustez também é comprovada na melhoria dos resultados quando é utilizado o re-sequenciamento. Além disso, o modelo balanceou as filas dos setores durante o processo de produção, atingindo, assim, dois objetivos de produção ao mesmo tempo. Por fim, utilizou-se informações do sistema produtivo em uma das regras de sequenciamento (número de máquinas dos setores), o que também é uma das exigências da literatura.

O agente conseguiu realizar um sequenciamento visando dois objetivos simultaneamente, pois aumentou a rapidez do processo e também a utilização. Para isso, teve que extrair mais informações do modelo enquanto o mesmo rodava, neste caso, as filas em cada setor de processo. Isso permitiu também que ele usasse a lógica de balancear filas em caso de quebra de máquinas, pois caso uma delas quebrasse, o agente Máquina interagia com o agente Gerente mandando a informação de quebra. Nesse momento, o Gerente acionaria também a lógica de balanceamento, não deixando que a fila de peças na máquina que quebrou aumentasse em demasia. Essas características fizeram o modelo se tornar mais robusto, pois o sequenciamento não se tornava insignificante perante a eventos inesperados, como quebra de máquinas.

O fato de utilizar o número de máquinas de cada setor modificando uma das lógicas de sequenciamento, também está inserido nas demandas da literatura da área. Pois, conforme já citado, aconselha-se que se usem informações do sistema produtivo nas lógicas, ao invés de apenas informações oriundas exclusivamente da planilha de ordens.

O modelo também considerou tempos estocásticos de processo, o que é uma das demandas atuais quando se trabalha em sequenciamento, visto que os tempos utilizados nessa área, muitas vezes, são determinísticos. Isso se deve ao fato dos tempos serem calculados em função do formato da peça, e não cronometrados para ter uma série estocástica. Isso acontece porque os produtos ainda não foram para o processo e como na maioria das vezes são produtos únicos (visto que este sistema produz por encomenda), não foram produzidos antes para se ter informações dos tempos cronometrados.

Também vale ressaltar, que SJSF são muito difíceis de se controlar devido ao fluxo complexo e, frequentemente, não se consegue nem mesmo analisar onde estão os gargalos do sistema, o que seria fácil em uma produção celular ou em linha por exemplo, em que um simples estoque intermediário em demasia é suficiente para identificar gargalos. Muitas vezes, os gestores desses sistemas são “enganados” pelos operadores, que tendem a diminuir o ritmo perante a ociosidade, ainda que isto não tenha comprovação científica. Mas com o uso do modelo proposto nesta tese, o gerente de produção real conseguiria antecipar quais setores se ocupariam mais e quais menos, além do tempo em que essa produção demoraria para terminar, bastando apenas analisar as respostas do modelo. Nessa situação, talvez tomasse alguma providência em posse dessas informações, como por exemplo o deslocamento de um operador de um setor ocioso para um mais ocupado (desde que o operador tenha essa flexibilidade).

Respondendo ao problema de pesquisa, o agente Gerente se provou eficiente no SP, logo, a Simulação Híbrida também. Muitas programações feitas neste modelo seriam difíceis de serem programadas utilizando somente algoritmos complexos, visto que o agente teve vital importância nos resultados atingidos, simplificando a programação. Seus parâmetros simples ajudam na modificação de dados de entrada e também na programação. Seu diagrama de estados ajuda na visualização das lógicas de SP e nos seus acionamentos. Além disso, a sua lógica interna pode ser acionada a qualquer momento ou evento inesperado que ocorra no modelo, gerando a capacidade de re-sequenciamento.

O primeiro objetivo específico foi atingido, que era de avaliar a inserção dos agentes na simulação, usando sua lógica interna que pode ser acionada diante da ocorrência de qualquer evento. A avaliação foi positiva, pois o agente pôde acionar sua lógica interna não somente na chegada das ordens para fazer o primeiro sequenciamento, mas também acionou sua outra lógica nos eventos “quebra de máquina” que aconteciam aleatoriamente, bem como em alguns outros momentos pré-determinados da simulação. Os resultados dessa característica do agente foram positivos e demonstrados no Capítulo 7 desta tese.

O segundo objetivo específico também foi atingido, que era de construir uma ferramenta de interação entre modelo e usuário. Foi desenvolvido, com sucesso, um modelo genérico para sequenciamento em SJSF, em que o programador pode usar qualquer sequência de operações para realizar os testes no modelo, bastando alterar uma planilha de interface simples antes de rodar o mesmo. Isso fez com que aumentasse a interação entre usuário e modelo. Esta característica, aliás, faz parte do conceito dos modernos modelos *Digital Twin*, usados na Indústria 4.0. Com isso, qualquer peça que possa ser produzida naqueles processos de fabricação existentes na empresa pode ser facilmente programada na planilha de interface.

Esta tese conclui, assim como Rajabinasab e Mansour (2011), que o método baseado em agentes é mais eficiente e robusto em relação as heurísticas usadas isoladamente. Confirmando o disseram Lou *et al.* (2010), os agentes se mostraram úteis para atravessar problemas inesperados do chão de fábrica. Em sintonia com os trabalhos de Lou *et al.* (2012) e Burdett *et al.* (2019), o modelo gerado nesta tese é uma plataforma genérica para o SP em sistemas JSF, ainda que possa precisar de ajustes para um sistema real. Assim como Ning e Jim (2018), esse trabalho frisa a importância de um sequenciamento multi-objetivo para satisfazer as necessidades atuais. E confirmando o que disseram Xiong e Fu (2018), o uso de agentes traz uma facilidade maior de se mudar dados, usando parâmetros simples e reduzindo a complexidade computacional.

8.2. Contribuições da tese

Foi criado um meio de modelagem que realiza o sequenciamento de maneira mais robusta aos problemas cotidianos, neste caso a quebra de máquinas e mudanças na sequência de produção, buscando um tipo de balanceamento de carga de operações entre os setores e evitando maiores transtornos em caso de quebras. O modelo também transforma os tempos calculados que são determinísticos em estocásticos, tornando o modelo mais real. Além disso, existe uma união de lógicas buscando um sequenciamento multi-objetivo. Todos estes fatores são requerimentos da literatura apresentados nesta tese, que são atendidos pelo modelo final.

8.3. Sugestões para trabalhos futuros

Para estudos futuros, propõe-se utilizar mais fatores de re-sequenciamento, que são eventos inesperados durante a execução do modelo, como por exemplo, ordens prioritárias vindas de consumidores importantes ou até mesmo cancelamento de ordens, além de programar novas respostas por parte do agente gerenciador. Também pode-se usar outra lógica de sequenciamento principal, e tentar melhorar seus pontos negativos através das ações de respostas autônomas dos agentes, utilizando novas lógicas de SP como resposta do gerente. Além disso, pode-se inserir critérios de decisão nos agentes baseando-se na aprendizagem de máquinas, em que os agentes podem aprender com base nas consequências de suas atitudes e tomar novas medidas em uma próxima oportunidade. Esta seria a última evolução dos agentes, fazendo dos mesmos agentes adaptativos, conforme definido por Macal (2016).

Referências bibliográficas

- AMIRI, F.; SHIRAZI, B.; TAJDIN, A. Multi-objective simulation optimization for uncertain resource assignment and job sequence in automated flexible job shop. **Applied Soft Computing Journal**, v.75, p.190-202, 2019.
- ASADZADEH, L. A local search genetic algorithm for the job shop scheduling problem with intelligent agents. **Computers & Industrial Engineering**, v.85, p.376-383, 2015.
- ATIGHEHCHIAN, A.; SEPEHRI, M. M. An environment-driven, function-based approach to dynamic single-machine scheduling. **European Journal of Industrial Engineering**, v.7, n.1, p.100-118, 2013.
- AXTELL, R. Why agents? On the varied motivations for agent computing in the social sciences. In: PROCEEDINGS OF THE WORKSHOP ON AGENT SIMULATION: APPLICATIONS, MODELS AND TOOLS, 17., **Anais eletrônicos...** Argonne National Laboratory, IL, 2000. Disponível em <http://www2.econ.iastate.edu/tesfatsi/WhyAgents.RAxtell2000.pdf>.
- BAINES, T. S.; KAY, J. M. Human performance modelling as an aid in the process of manufacturing system design: a pilot study. **International Journal of Production Research**, v.40, n.10, p. 2321-2334, 2002.
- BAINES, T.; MASON, S.; SIEBERS, P.; LADBROOK, J. Humans: the missing link in manufacturing simulation? **Simulation Modelling Practice and Theory**, n.12, p.515-526, 2004.
- BANKS, J. **Handbook of simulation: principles, methodology, advances, applications and practice**. New York: John Wiley & Sons, 1998.
- BANKS, J.; CARSON, J. S.; NELSON, B. L.; NICOL, D. M. **Discrete Event System Simulation**. 5.ed. Upper Saddle River, NJ: Prentice Hall, 2009.
- BANKS, J.; GIBSON, R. R. Don't simulate when: 10 rules for determining when simulation is not appropriate. **IIE Solutions**, v.29, n.9, p.30-32, 1997.
- BARLAS, P.; HEAVEY, C. Automation of input data to discrete event simulation for manufacturing: A review. **International Journal of Modeling, Simulation, and Scientific Computing**, v.7, n.1, 2016.
- BARROS, J. P. S., BALDAM, R. L., JUNIOR, T. D. P. C., LEAL, E. D. A. S., SOUZA, M. A. V. F. Simulação baseada em agentes. In: XXXI ENEGEP, **Anais eletrônicos...** Belo Horizonte: Minascentro, 2011. Disponível em http://www.abepro.org.br/biblioteca/enegep2011_tn_sto_140_888_18230.pdf
- BERTRAND, J. W. M., FRANSOO, J. C. Operations management research methodologies using quantitative modeling. **International Journal of Operations & Production Management**, v. 22, n. 2, p.241-264, 2002.

BIAGIO, L. A. **Como administrar a produção**. São Paulo: Ed. Manole, 2015.

BOUAZZA, W.; SALLEZ, Y.; AISSANI, N.; BELDJILALI, B. A Model for Manufacturing Scheduling Optimization Through Learning Intelligent Products. **Studies in Computational Intelligence**, v.594, p.233-241, 2015.

BRAILSFORD, S. C.; HARPER, P. R.; SYKES, J. Incorporating human behavior in simulation models of screening for breast cancer. **European Journal of Operational Research**, v.219, p.491-507, 2012.

BRAUN, P.; ROSSAK, W. **Mobile agents: basic concepts, mobility models, and the Tracy toolkit**. New York: Elsevier, 2005.

BURDETT, R. L.; CORRY, P.; YARLAGADDA, P. K. D. V.; EUSTACE, C.; SMITH, S. A flexible job shop scheduling approach with operators for coal export terminals. **Computers and Operations Research**, v.104, p.15-36, 2019.

CHAN, W. K. V.; SON, Y. J.; MACAL, C. M., Agent-Based Simulation Tutorial -Simulation of Emergent Behavior and Differences Between Agent-Based Simulation and Discrete-Event Simulation. In: PROCEEDINGS OF THE WINTER SIMULATION CONFERENCE, 2010, Baltimore, MD, USA. **Anais...** Baltimore, p.135-150, 2010.

CHESS, D. GROSOF, B.; HARRISON, C.; *et al.* Itinerant Agents for Mobile Computing. **IEEE Personal Communications**, v.2, n.5, p.34-49, 1995.

CHWIF, L.; MEDINA, A. C. **Modelagem e Simulação de Eventos Discretos: Teoria e Aplicações**. 3.ed. São Paulo: Editora dos Autores, 2010.

CORRÊA, H. L.; CORRÊA, C. A. **Administração de Produção e Operações: Manufatura e Serviços: Uma abordagem Estratégica**. São Paulo: Ed. Atlas, 2004.

DROGOUL, A.; VANBERGUE, D.; MEURISSE, T. Multi agent-based simulation: Where are the agents? In: SIMÃO, S. J.; BOUSQUET, F.; DAVIDSSON, P. Multi-agent-based simulation II. Berlin: Springer, p.1-15, 2003.

DUBIEL, B.; TSIMHONI, O. Integrating Agent Based Modeling into a Discrete Event Simulation. In: PROCEEDINGS OF WINTER SIMULATION CONFERENCE, 2005, Orlando, FL, USA. **Anais...** Orlando, p.1029-1037, 2005.

FRANCESCHINI, F.; MAISANO, D.; MASTROGIACOMO, L. Scientific journal publishers and omitted citations in bibliometric databases: Any relationship? **Journal of Informetrics**, v.8, n.3, p.751-765, 2014.

FRANTZÉN, M.; AMOS, H. C.; MOORE, P. A. Simulation-based scheduling system for real-time optimization and decision making support. **Robotics and Computer-Integrated Manufacturing**, v.27, p.696-705, 2011.

GABRIEL, G. T. Documentação da lógica de modelos de simulação por meio do uso da técnica de modelagem IDEF-SIM. 2018. 123p. Dissertação (Mestrado em Engenharia de Produção) – Instituto de Engenharia de Produção e Gestão, Universidade Federal de Itajubá, Itajubá, 2018.

- GIROUX, S.; MARCENAC, P.; CALDERONI, S.; GROSSER, D.; GRASSO, J. R. **A Report of a Case Study with Agents in Simulation**. London, UK: Practical Application Company, 1996.
- GOLDSMAN, D.; NANCE, R. E.; WILSON, J. R. A brief history of simulation. In: PROCEEDINGS OF THE WINTER SIMULATION CONFERENCE, 2009, Austin, TX, USA. **Anais...** Austin, p.310-313, 2009.
- GONG, X.; DENG, Q.; GONG, G.; LIU, W.; REN, Q. A memetic algorithm for multi-objective flexible job-shop problem with worker flexibility. **International Journal of Production Research**, v.56, n.7, p.2506-2522, 2018.
- GRIGORYEV, I. **Anylogic 6 in three days: a quick course in simulation modeling**. Hampton, NJ: Anylogic North America, 2012.
- GRIMM, V.; BERGER, U.; BASTIANSEN, F.; *et al.* A standard protocol for describing individual-based and agent-based models. **Ecological Modelling**, v.198, n.1–2, p.115-126, 2006.
- GRIMM, V.; BERGER, U.; DEANGELIS, D.; POLHILL, J.; GISKE J.; RAILSBACK, S. The ODD protocol: A review and first update. **Ecological Modelling**, v.221, n.23, p.2760-2768, 2010.
- GRIMM, V.; RAILSBACK, S. **Individual-Based Modeling and Ecology**. Princeton, NJ: Princeton University Press, 2005.
- HARREL, C.; GHOSH, B. K.; BOWDEN, R. O. **Simulation Using Promodel**. 2.ed. New York: McGraw-Hill, 2004.
- HOPP, W. J.; SPEARMAN, M. L. **Factory Physics: foundations of manufacturing management**. 2.ed. Chicago: Irwin/McGraw Hill, 2000.
- JACOBS, F. R.; R. B. CHASE. **Operations and Supply Chain Management**. 15.ed. Boston: McGrawHill/Irwin, 2018.
- JAHANGIRIAN, M.; ELDABI, T.; NASEER, A.; STERGIOULAS, L. K.; YOUNG, T. Simulation in manufacturing and business: A review. **European Journal of Operational Research**, v.203, n.1, p.1-13, 2010.
- KELTON, D. W.; SADOWSKI, R. P.; STURROCK, D. T. **Simulation with Arena**. 4.ed. New York: McGraw-Hill, 2007.
- KIRAN, A. S. Simulation and scheduling. In: BANKS, J. Handbook of Simulation: Principles, Methodology, Advances, Applications, and Practice. New York: John Wiley and Sons, p.677-717, 1998.
- KLEMMT, A.; WEIGERT, G.; WERNER, S. Optimisation approaches for batch scheduling in semiconductor manufacturing. **European Journal of Industrial Engineering**, v.5, n.3, p. 338-359, 2011.
- KRAJEWSKI, L; RITZMAN, L.; MALHOTRA, M. **Administração de Produção e Operações**. 11.ed. São Paulo: Editora Pearson, 2016.

- KULKARNI, P.; VENKATESWARAN, J. Simulation and Optimization Based Approach for Job Shop Scheduling Problems. In: IEEE INTERNATIONAL CONFERENCE ON INDUSTRIAL ENGINEERING AND ENGINEERING MANAGEMENT, 2016, Bali, Indonesia. **Anais...** Bali, p.360-364, 2016.
- LAW, A. M. How to build valid and credible simulation models. In: PROCEEDINGS OF THE WINTER SIMULATION CONFERENCE, 2006, Monterey, CA, USA. **Anais...** Monterey, p.58-66, 2006
- LAW, A. M.; KELTON, W. D. **Simulation modeling and analysis**. 4.ed. New York: McGraw-Hill, 2007.
- LEAL, F.; ALMEIDA, D. A. de; MONTEVECHI, J. A. B. Uma proposta de técnica de modelagem conceitual para a simulação através de elementos do IDEF. In: XL SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL, 2008, João Pessoa, BR. **Anais...** João Pessoa, p.2503-2514, 2008.
- LOU P.; LIU Q.; ZHOU, Z.; WANG, H.; SUN, S. X. Multi-agent-based proactive–reactive scheduling for a job shop. **International Journal of Advanced Manufacturing Technology**, v.59, p.311–324, 2012.
- LOU, P.; ONG, S. K.; NEE, A. Y. C. Agent-based distributed scheduling for virtual job shops. **International Journal of Production Research**, v.48, n.13, p.3889-3910, 2010.
- LUSTOSA, L.; MESQUITA, M.; OLIVEIRA, R.; QUELHAS, O. **Planejamento e controle da produção**. RJ: Elsevier, 2008.
- MACAL, C. M. Everything you need to know about agent-based modelling and simulation. **Journal of Simulation**, v.10, n.02, p.144-156, 2016.
- MACAL, C. M.; NORTH, M. J.; SAMUELSON, D. A. Elements of Agent-Based Simulation. In: GASS, S. I.; FU, M. C. (Ed). *Encyclopedia of Operations Research and Management Science*. 3.ed. New York: Springer, cap. Agent-Based Simulation, p.8-16, 2013.
- MACAL, C.; NORTH, M. Introductory Tutorial: Agent-Based Modeling and Simulation. In: PROCEEDINGS OF THE WINTER SIMULATION CONFERENCE, 2014, Savannah, GA, USA. **Anais...** Savannah, p.6-20, 2014.
- MACEDO, J. A. S. V. Melhorias no setor das bancadas - Simoldes Aços. 2012. 84p. Dissertação (Mestrado Integrado em Engenharia Mecânica) - Faculdade de Engenharia da Universidade do Porto, Porto, 2012.
- METAXIOTIS, K. S.; PSARRAS J. E.; ERGAZAKIS K. A. Production scheduling in ERP systems – An AI-based approach to face the gap. **Business Process Management Journal**, v.9, n.2, p.221–247, 2003.
- MOKHTARI, H.; DADGAR, M. Scheduling optimization of a stochastic flexible job-shop system with time-varying machine failure rate. **Computers and Operations Research**, v.61, p.31-45, 2015.
- MONTEVECHI, J. A. B. LEAL, F. PINHO, A. F. COSTA, R. F. S. OLIVEIRA, M. L. M. e SILVA, A. L. F. Conceptual modeling in simulation projects by mean adapted IDEF: an application in a Brazilian tech company. In: PROCEEDINGS OF THE WINTER

- SIMULATION CONFERENCE, 2010, Baltimore, MD, USA. **Anais...** Baltimore, p.1624-1635, 2010.
- MONTGOMERY, D. C. **Design and Analysis of Experiments**. 6.ed. New York: Wiley, 2005.
- MORTAZAVI, A.; KHAMSEH, A. A.; AZIMI, P. Designing of an intelligent self-adaptive model for supply chain ordering management system. **Engineering Applications of Artificial Intelligence**, v.37, p.207-220, 2015.
- MUDA, M. S.; HENDRY, L. C. The SHEN model for MTO SMEs: a performance improvement tool. **International Journal of Operations Production Management**, v.23, p.470-486, 2003.
- NANCE, R. E.; SARGENT, R. Perspectives on the evolution of simulation. **Operations Research**, v.50, n.1, p.161-172, 2002.
- NING, T.; GUO, C.; CHEN, R.; JIN, H. A novel hybrid method for solving flexible job-shop scheduling problem. **Open Cybernetics and Systemics Journal**, v.10, p.13-19, 2016.
- NING, T.; HUANG, M.; LIANG, X.; JIN, H. A novel dynamic scheduling strategy for solving flexible job-shop problems. **Journal of Ambient Intelligence and Humanized Computing**, v.7, n.5, p.721-729, 2016.
- NING, T.; JIN, H. A cloud based improved method for multi-objective flexible job-shop scheduling problem. **Journal of Intelligent and Fuzzy Systems**, v.35, n.1, p.823-829, 2018.
- O'KANE, J. F.; SPENCELEY, J. R.; TAYLOR, R. Simulation as an essential tool for advanced manufacturing technology problems. **Journal of Materials Processing Technology**, v.107, p.412-424, 2000.
- OLIVEIRA, M. L. M.; MONTEVECHI, J. A. B.; PINHO, A. F.; MIRANDA, R. C. Using hybrid simulation to represent the human factor in production systems. **International Journal of Simulation Modelling**, v. 16, p.263-274, 2017.
- PARTHANADEE, P.; BUDDHAKULSOMSIRI, J. Simulation modeling and analysis for production scheduling using real-time dispatching rules: A case study in canned fruit industry. **Computers and Electronics in Agriculture**, v.70, p.245-255, 2010.
- RAJABINASAB, A.; MANSOUR, S. Dynamic flexible job shop scheduling with alternative process plans: an agent-based approach. **International Journal of Advanced Manufacturing Technology**, v.54, p.1091-1107, 2011.
- REISI-NAFCHI, M.; MOSLEHI, G. Two-agent order acceptance and scheduling to maximise total revenue.. **European Journal of Industrial Engineering**, v.9, n.5, p. 664-691, 2015.
- REY, G. Z.; BEKRAR, A.; TRENTESAUX, D.; ZHOU, B. H. Solving the flexible job-shop just-in-time scheduling problem with quadratic earliness and tardiness costs. **International Journal of Advanced Manufacturing Technology**, v.81, n.9-12, p.1871-1891, 2015.
- ROBINSON, S. Discrete-Event Simulation: From the Pioneers to the Present, What Next? **Journal of the Operational Research Society**, v.56, n.6, p.619-629, 2005.

RODIČ, B. Industry 4.0 and the New Simulation Modelling Paradigm. **Organizacija**, v.50, n.3, p.193-207, 2017.

SAKURADA, N.; MIYAKE, D. I. Simulação baseada em agentes para modelagem de sistemas de operações. In: SIMPÓSIO DE ADMINISTRAÇÃO DA PRODUÇÃO, LOGÍSTICA E OPERAÇÕES INTERNACIONAIS, 2009, São Paulo, SP. **Anais...** São Paulo, p.1-16, 2009.

SAMUELSON, D. A. Agents of Change: How agent-based modeling may transform social science. **OR/MS Today**, v.32, n.1, 2005.

SAMUELSON, D. A.; MACAL, C. M. Agent-Based Simulation Comes of Age: Software opens up many new areas of application. **OR/MS Today**. v.33, n.4, 2006.

SARAIVA JÚNIOR, A. F.; TABOSA, C. D. M.; COSTA, R. P. Simulação de Monte Carlo aplicada à análise econômica de pedido. **Produção**, v.21, n.1, p.149-164, 2011.

SENA, D. C.; SILVA, E. M. M.; COSTA, A. P. R.; MONTEVECHI, J. A. B.; PINHO, A. F.; MIRANDA, R. C. Dynamic allocation of additional human resources using hybrid simulation. **International Journal of Simulation Modelling**, v.16, p.84-95, 2017.

SHARMA, P.; JAIN, A. Effect of routing flexibility and sequencing rules on performance of stochastic flexible job shop manufacturing system with setup times: Simulation approach. **Journal of Engineering Manufacture**, v.231, n.2, p.329-345, 2017.

SHEN, L.; DAUZÈRE-PÉRÈS, S.; NEUFELD, J. S. Solving the flexible job shop scheduling problem with sequence-dependent setup times. **European Journal of Operational Research**, v.265, n.2, p.503-516, 2018.

SHEN, W.; HAO, Q.; YOON, H. J.; NORRIE, D. H. Applications of agent based systems in intelligent manufacturing: an updated review. **Advanced Engineering Information**, v.20, p.415-431, 2006.

SICHMAN, J. S.; CONTE, R.; GILBERT, N. **Multi-agent Systems and agent-based simulation**. Springer, 1998.

SIEBERS, P. O.; MACAL, C. M.; GARNETT, J.; BUXTON, D.; PIDD, M. Discrete-event simulation is dead, long live agent-based simulation! **Journal of Simulation**, v.4, n.3, p.204-210, 2010.

SLACK, N.; CHAMBERS, S.; JOHNSTON, R. **Administração da Produção**. 3.ed. São Paulo: Atlas, 2009.

STURROCK, D. T. Tips for successful practice of simulation. In: PROCEEDINGS OF THE WINTER SIMULATION CONFERENCE, 2009, Austin, Texas, USA. **Anais...** Austin, p.34-39, 2009.

SUDO, Y.; MATSUDA, M. Agent based manufacturing simulation for efficient assembly operations. In: FORTY SIXTH CIRP CONFERENCE ON MANUFACTURING SYSTEMS, 2013, Setubal, Portugal. **Anais...** Setubal, p.437-442, 2013.

THÖRNBLAD, K.; STRÖMBERG, A. B.; PATRIKSSON, M.; ALMGREN, T. Scheduling optimisation of a real flexible job shop including fixture availability and preventive maintenance. **European Journal of Industrial Engineering**, v.9, n.1, p.126-145, 2015.

TUBINO, D. F. **Planejamento e Controle da Produção**. 3.ed. São Paulo: Atlas, 2017.

TURRIONI, J. B.; MELLO, C. H. P. **Metodologia de pesquisa em engenharia de produção**. Itajubá: Universidade Federal de Itajubá, 2012.

VINOD, V.; SRIDHARAN, R. Simulation modeling and analysis of due-date assignment methods and scheduling decision rules in a dynamic job shop production system. . **International Journal of Production Economics**, v.129, p.127–146, 2011.

WENG, M. X.; WU, Z.; QI, G.; ZHENG, L. Multi-agent-based workload control for make-to-order manufacturing. **International Journal of Production Research**, v.46, n.8, p.2197–2213, 2008.

XIONG, H.; FAN, H.; JIANG, G.; LI, G. A simulation-based study of dispatching rules in a dynamic job shop scheduling problem with batch release and extended technical precedence constraints. **European Journal of Operational Research**, v.257, n.1, p.13-24, 2017.

XIONG, W.; FU, D. A new immune multi-agent system for the flexible job shop scheduling problem. **Journal of Intelligent Manufacturing**, v.29, n.4, p.857-873, 2018.

YANG, X. P.; GAO, X. L. Optimization of dynamic and multi-objective flexible job-shop scheduling based on parallel hybrid algorithm. **International Journal of Simulation Modeling**, v.17, n.4, p.724-733, 2018.




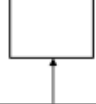
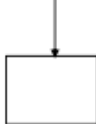




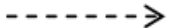



YE, J.; MA, H. Multiobjective joint optimization of production scheduling and maintenance planning in the flexible job-shop problem. **Mathematical Problems in Engineering**, v.2015, n.725460, 2015.

YU, J. M.; LEE, D. H. Solution algorithms to minimise the total family tardiness for job shop scheduling with job families. **European Journal of Industrial Engineering**, 12(1), pp. 1-23, 2018.

ZHANG, Q.; MANIER, H.; MANIER, M. A.; Bloch, C. Heuristics for predictive hoist scheduling problems. **European Journal of Industrial Engineering**, v.8, n.5, p.695-715, 2014.

ANEXO A – Simbologia usada no IDEF-SIM

Tabela A - Simbologia utilizada na técnica IDEF-SIM

Elementos	Simbologia	Técnica de origem
Entidade		IDEF3 (modo descrição das transições)
Funções		IDEF0
Fluxo da entidade		IDEF0 e IDEF3
Recursos		IDEF0
Controles		IDEF0
Regras para fluxos paralelos e/ou alternativos	 &	Regra E
	 x	Regra OU
	 o	Regra E/OU
Movimentação		Fluxograma
Informação explicativa		IDEF0 e IDEF3
Fluxo de entrada no sistema modelado		
Ponto final do sistema		
Conexão com outra figura		

Fonte: Leal, Almeida e Montevechi (2008)

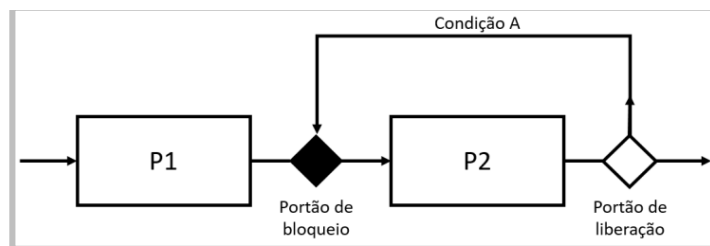


Figura A - Símbolo adicional utilizado no mapeamento

Fonte: Gabriel (2018)

APÊNDICE A - Artigos publicados

RODRIGUES, R. P.; PINHO, A. F.; SENA, D. C. **Application of hybrid simulation in production scheduling in job shop systems.** Simulation, 2019.

<https://doi.org/10.1177/0037549719861724>

Journal Impact factor = 1.455

RODRIGUES, R. P.; Pinho, A. F.; NUNES, D. L.; SENA, D. C.; BARBIERI, J. P. **Sequenciamento por meio da lógica Johnson utilizando a simulação baseada em agentes.** In: XLVIII Simpósio Brasileiro de Pesquisa Operacional (SBPO), 2016, Vitória, ES. XLVIII SBPO, 2016.

NUNES, D. L.; RODRIGUES, R. P.; Pinho, A. F.; BARBIERI, J. P.; TAMAKI, D. M. C. **Uso do ritmo circadiano e ritmo de trabalho em um processo produtivo através da simulação baseada em agentes.** In: XLVIII Simpósio Brasileiro de Pesquisa Operacional (SBPO), 2016, Vitória, ES. XLVIII SBPO, 2016.

SENA, D. C.; RODRIGUES, R. P.; Pinho, A. F.; Montevechi, J. A. B.; FERREIRA, R. C.; COSTA, A. P. R. **Análise do impacto de fatores de atendimento em uma unidade de pronto-atendimento utilizando simulação híbrida.** In: XLVIII Simpósio Brasileiro de Pesquisa Operacional (SBPO), 2016, Vitória, ES. XLVIII SBPO, 2016.

BARBIERI, J. P.; RODRIGUES, R. P.; NUNES, D. L.; Pinho, A. F. **Application of hybrid simulation in the representation of human behavior in a manufacturing system.** In: XLVIII Simpósio Brasileiro de Pesquisa Operacional (SBPO), 2016, Vitória, ES. XLVIII SBPO, 2016.

RODRIGUES, R. P.; MONTEVECHI, J. A. B.; GOMES, J. H. F.; QUEIRÓZ, J. A.; LEAL, F. **Otimização de simulação a eventos discretos utilizando a metodologia de superfície de resposta.** Simpósio Brasileiro de Pesquisa Operacional, Anais... Porto de Galinhas, PE, Brasil, 2015.

RODRIGUES, R. P.; PINHO, A. F.; GOMES, J. H. F.; QUEIRÓZ, J. A. **Uso da técnica de superfície de resposta na otimização da simulação computacional do processo.** Simpósio Brasileiro de Pesquisa Operacional, Anais... Salvador, BA, Brasil, 2014.