

UNIVERSIDADE NOVE DE JULHO - UNINOVE
PROGRAMA DE MESTRADO EM ENGENHARIA DE PRODUÇÃO

**ABORDAGEM PARA OTIMIZAÇÃO MULTIOBJETIVO DE REGRAS HEURÍSTICAS DE SEQUENCIAMENTO
EM SISTEMAS DE MANUFATURA *JOB SHOP* POR MEIO DE SIMULAÇÃO COMPUTACIONAL
ACOPLADA AO ALGORITMO GENÉTICO**

MARILDA FATIMA DE SOUZA DA SILVA

SÃO PAULO

22 DE MARÇO DE 2011

MARILDA FATIMA DE SOUZA DA SILVA

ABORDAGEM PARA OTIMIZAÇÃO MULTIOBJETIVO DE REGRAS HEURÍSTICAS DE SEQUENCIAMENTO

EM SISTEMAS DE MANUFATURA *JOB SHOP* POR MEIO DE SIMULAÇÃO COMPUTACIONAL

ACOPLADA AO ALGORITMO GENÉTICO

Dissertação de mestrado apresentada ao programa de Pós-Graduação em Engenharia de Produção da Universidade Nove de Julho - Uninove, como requisito parcial para a obtenção do grau de Mestre em Engenharia de Produção.

Orientador: Prof. Dr. Fabio Henrique Pereira

Co-Orientador: Prof. Dr. Felipe Araújo Calarge

SÃO PAULO

22 DE MARÇO DE 2011

Silva, Marilda Fátima de Souza.

Abordagem para otimização multiobjetivo de regras heurísticas de sequenciamento em sistemas de manufatura job shop por meio de simulação computacional acoplada ao algoritmo genético / Marilda Fátima de Souza Silva.

116 f.

Dissertação (mestrado) – Universidade Nove de Julho - UNINOVE, São Paulo, 2011.

Orientador (a): Prof. Dr. Fabio Henrique Pereira.

1. Sequenciamento. 2. Algoritmo genético. 3. Otimização da simulação.

I. Pereira, Fabio Henrique.

CDU 621

“A meta de uma discussão ou debate não
deveria ser a vitória, mas o progresso”.

Joseph Joubert.

AGRADECIMENTOS

Obrigado Senhor:

Porque sem Ti nada seria possível.

Por ter me permitido nascer num lar saudável e seguro tendo por pais João e Elza e por irmãos: Joel, Jair, Maisa e Ricardo.

Por ter colocado em meu caminho professores e amigos que de uma forma ou de outra, se não moldaram meu caráter, moldaram minha alma.

Especialmente, pelo professor Fabio Henrique Pereira, que acreditou e com paciência e dedicação me orientou no desenvolvimento deste trabalho.

Pelo professor Felipe Araújo Calarge, pelos comentários e sugestões na co-orientação.

Pelas minhas amigas, Aparecida de Fátima Castello Rosa e Maria Aparecida Pereira, que me apoiaram e incentivaram neste trabalho, sobretudo pelo carinho e confiança nas nossas conversas.

Pelas funcionárias da secretaria Michele, Carolina e Vanessa pela cordialidade.

Por todo o pessoal da Uninove que manteve nossos locais seguros, limpos, arejados e informatizados.

Pelo Programa de Mestrado da Universidade Nove de Julho que me proporcionou uma bolsa de estudos.

Por mais esta etapa vencida.

Dedico este trabalho a duas pessoas amadas
e maravilhosas: ao Pedro, meu marido, e à
Bárbara, minha filha.

SILVA, Marilda Fatima de Souza da. *Abordagem para otimização multiobjetivo de regras heurísticas de sequenciamento em sistemas de manufatura job shop por meio de simulação computacional acoplada ao algoritmo genético*. 2011. 118 p. Dissertação (Mestre em Engenharia de Produção). Universidade Nove de Julho - UNINOVE, São Paulo.

RESUMO

Desde o surgimento do Sistema Toyota de Produção (STP), conhecido por produção enxuta, e mais recentemente as exigências do mercado consumidor, as empresas de manufatura têm trabalhado para melhorar seus processos de produção, com objetivo de reduzir níveis de estoque, eliminar desperdícios, manter/aumentar a qualidade, a competitividade e o lucro. Neste contexto, o seqüenciamento de ordens de produção é um dos problemas mais complexos enfrentados pelas empresas e objeto de diversos estudos. Desta forma, este trabalho apresenta uma abordagem alternativa na resolução desse problema, através da utilização de um modelo de simulação como função objetivo do Algoritmo Genético ao invés de uma equação matemática analítica. Para o atendimento de novas situações, não será necessário mudar a equação e sim ajustar o modelo e efetuar uma nova simulação. Regras heurísticas de programação em ambientes *job shop* foram consideradas, com rotas, datas de entrega e tempos de operação gerados aleatoriamente, a fim de determinar a melhor técnica de programação quanto ao desempenho em relação ao tempo total de atravessamento, número de ordens atrasadas e tempo total de atraso. Resultados obtidos corroboram o método adotado. A abordagem de otimização multiobjetivo baseia-se no acoplamento do Algoritmo Genético com o modelo de simulação Arena por meio da linguagem de programação *Visual Basic for Application* e o controlador *ActiveX Automation*.

Palavras-chave: Sequenciamento. Algoritmo Genético. Otimização da Simulação.

SILVA, Marilda Fatima de Souza da. **Approach for optimization of heuristic rules multiobjetivo sequencing in job shop manufacturing systems through simulation coupled with genetic algorithm.** 2011. 101 p. Dissertation (Master in Production Engineers). Universidade Nove de Julho - UNINOVE, São Paulo.

ABSTRACT

Since the arising of the Toyota Production System (TPS), known as lean production, and more recently the consumer market demands, manufacturing companies have worked to improve their production processes in order to reduce inventory levels, to eliminate waste and to maintain quality, competitiveness and profit. In this context, the sequencing of production orders is one of the most complex problems facing businesses and it is object of several studies. Thus, this paper presents an alternative approach in resolving this problem, i.e.: to use a simulation model as the objective function in genetic algorithm instead analytical mathematical equation. So, for each situation, will not need to change the equation, but adjust the model and to make a new simulation. Heuristics sequencing rules in job shop environments were considered, with routes, due dates and times of operation generated randomly, in order to determine the best programming technique for performance in relation to the total time of crossing, the total tardiness and the number of tardy jobs. Results corroborate the method adopted. Multi-objective optimization approach is based on the coupling of Genetic Algorithm with an Arena simulation model through the Visual Basic for Application language and the *ActiveX Automation* controller.

Key words: Scheduling. Genetic Algorithm. Simulation Optimization.

SUMÁRIO

1	INTRODUÇÃO	15
1.1	CONSIDERAÇÕES INICIAIS	15
1.1.1	Escolha das medidas de desempenho	19
1.2	REVISÃO BIBLIOGRÁFICA	20
1.3	CONTRIBUIÇÃO DA DISSERTAÇÃO	24
1.4	OBJETIVO	25
1.4.1	Objetivo geral	25
1.4.2	Objetivos específicos	25
1.5	RELEVÂNCIA DO TRABALHO	26
1.6	MATERIAIS, MÉTODOS E TÉCNICAS	27
1.7	ESTRUTURA DO TRABALHO	29
2	FUNDAMENTAÇÃO TEÓRICA	30
2.1	SIMULAÇÃO COMPUTACIONAL	30
2.1.1	Vantagens e Desvantagens	34
2.1.2	Experimentação, verificação e validação do modelo	36
2.1.3	<i>Softwares</i> de Simulação	38
2.2	ANÁLISES ESTATÍSTICAS	40
2.2.1	Intervalo de confiança (IC)	41
2.3	SISTEMAS DE PRODUÇÃO	42
2.3.1	Sistema <i>Job shop</i>	46
2.3.2	Teoria das filas	47
2.3.3	Regras de sequenciamento da produção	48
2.3.4	Medidas de Desempenho	51

2.3.4.1 Indicadores de Desempenho	52
2.4. TÉCNICAS METAHEURÍSTICAS DE OTIMIZAÇÃO	54
2.4.1 Algoritmos Genéticos	55
2.4.1.1 Codificação, funcionamento e nomenclatura	58
2.4.1.2 Seleção	60
2.4.1.3 Operadores Genéticos	62
2.4.1.4 Função de Aptidão	65
2.4.1.5 Critérios de Parada	65
2.4.2. Otimização multiobjetivo	66
2.4.2.1 Frentes de Pareto	67
3 APLICAÇÃO DO MODELO	69
3.1 DESCRIÇÃO DO MODELO	70
3.1.1 Implementação do Modelo Arena	73
3.2 MODELO DE OTIMIZAÇÃO	78
3.3 ACOPLAMENTO ENTRE O MODELO DE SIMULAÇÃO E O AG	82
3.4 VALIDAÇÃO	86
4 ANÁLISE E DISCUSSÃO DOS RESULTADOS	92
5 CONCLUSÃO	99
5.1 SUGESTÕES PARA CONTINUIDADE DA PESQUISA	100
REFERÊNCIA BIBLIOGRÁFICA	101
ANEXO A- Código VBA	110
ANEXO B – Trabalhos oriundos desta dissertação	115

LISTA DE FIGURAS

FIGURA 2.1: A evolução dos <i>softwares</i> de simulação	31
FIGURA 2.2: Passos na formulação de um estudo envolvendo modelagem e simulação	34
FIGURA 2.3: Ilustração de intervalo de confiança	42
FIGURA 2.4: Esquema de Produção.....	42
FIGURA 2.5: Representação da formação de fila.....	48
FIGURA 2.6: Evolução das abordagens utilizadas na solução de problemas de seqüenciamento da programação	51
FIGURA 2.7: Ótimo global e ótimo local de uma função bidimensional	55
FIGURA 2.8: Estrutura de um Cromossomo	57
FIGURA 2.9: Representação Gráfica do AG	59
FIGURA 2.10: Seleção pelo método da Roleta.....	61
FIGURA 2.11: Exemplo de aplicação do operador de mutação	63
FIGURA 2.12: Exemplo de aplicação do operador de cruzamento em um ponto	64
FIGURA 2.13: Frente de Pareto para um problema com dois objetivos.....	68
FIGURA 3.1: AG com uma nova função de aptidão	70
FIGURA 3.2: Modelo de simulação desenvolvido no Arena.....	73
FIGURA 3.3: Detalhe de carregamento do módulo <i>Create</i> para a chegada de ordens de produção.....	74
FIGURA 3.4: Sequenciamento de ordens de produção no modelo. Destaque para a rota definida para a tarefa 1	75
FIGURA 3.5: Expressões para cálculo do tempo de processamento e do prazo de entrega....	75
FIGURA 3.6: Valores e variáveis atribuídas a cada uma das ordens.....	76
FIGURA 3.7 Implementação de Prioridade	77

FIGURA 3.8: Representação gráfica do Cromossomo	79
FIGURA 3.9: Definições de peso, função e cromossomo no AG	81
FIGURA 3.10: Definições de gene, população, mutação, cruzamento no AG	81
FIGURA 3.11: Indicação de gravação dos resultados da função objetivo e da melhor combinação no AG	82
FIGURA 3.12: Limpeza e posterior gravação do arquivo que será utilizado pelo AG.....	83
FIGURA 3.13: Atualização da prioridade no simulador para execução e posterior envio de resultado para o AG	83
FIGURA 3.14: Gravação do arquivo de saída, chamada da função aptidão, e gravação dos resultados para o simulador	84
FIGURA 3.15: Representação gráfica da função do ActiveX.....	85
FIGURA 3.16: Validação da Prioridade.....	87
FIGURA 4.1: Comparação dos resultados de acordo com o número de população e geração.	92
FIGURA 4.2: Desempenho do AG em função dos pesos de α , β e γ	94
FIGURA 4.3: Soluções obtidas pelo AG, candidatas à frente de Pareto.....	95
FIGURA 4.4: Tempo Total de Atraso	96
FIGURA 4.5: Número de Ordens Atrasadas	96
FIGURA 4.6: Tempo total de atravessamento	97
FIGURA 4.7: Sequenciamento de ordens por tipo de máquina	98

LISTA DE TABELAS

TABELA 1.1: Resumo de pesquisas envolvendo o uso de simulação.....	21
TABELA 2.2: Relação de <i>Softwares</i> de simulação atuais	39
TABELA 2.2: Definição dos principais termos utilizados nas técnicas de AGs	56
TABELA 3.1: Rota e tempo total para produção em <i>job shop</i>	72
TABELA 3.2: Parâmetros, definições e justificativas.....	78
TABELA 3.3: Pesos considerados para cada um dos objetivos.....	80
TABELA 3.4: Condição de processamento do modelo de validação	86
TABELA 3.5: Parâmetros de processamento para a validação do modelo.....	88
TABELA 3.6. Resultado do Simulador <i>versus</i> resultado do <i>Excel</i>	88
TABELA 3.7: Análise estatística dos resultados dos processamentos de acordo com os pesos dos parâmetros α , β e γ – MKP	89
TABELA 3.8: Análise estatística dos resultados dos processamentos de acordo com os pesos dos parâmetros α , β e γ – NOA	90
TABELA 3.9: Análise estatística dos resultados dos processamentos de acordo com os pesos dos parâmetros α , β e γ – TTA	90
TABELA 4.1: Resumo dos resultados obtidos nos testes	93

LISTA DE SIGLAS E SÍMBOLOS

ACS – *Ant Colony Systems* – Sistema de Colonia de Formigas

AG – Algoritmo Genético

ASPOL – Linguagem de programação orientada a processo

BS – *Bee Systems* – Sistema das abelhas

C++ – Linguagem de programação orientada a objeto

DELPHI – Linguagem de programação

EDD – *Earliest Due Date* – Menor data de entrega

F – Funcional

FIFO – *First in, First out* – Primeiro que entra, primeiro que sai

FORTRAN – Formula Translator – Linguagem de programação

GPSS – *General Purpose Simulation System* – Sistema de Simulação de Propósito Geral

IC – Intervalo de Confiança

JAVA – Linguagem de programação

JIT – *Just in Time* – Produção da quantidade certa, no momento certo

LIFO – *Last in, First out* – Último que entra, primeiro que sai

LIPT – *Longest Imminent Processing Time* – Maior tempo de Processamento Individual

LPT – *Longest Processing Time* – Maior tempo de Processamento

LS – *Least Slack* – Menor tempo entre tarefa

MATLAB – *MATrix LABoratory* – Linguagem de programação

MIT – *Massachusetts Institute of Technology* – Instituto de Tecnologia de Massachusets

MKP – Tempo Total de Atravessamento

NOA – Número de Ordens Atrasadas

PL – Linguagem de programação

PSO – *Particle Swarm Optimization* – Otimização por Enxames de Partículas

SA – *Simulated Annealing* – Recozimento Simulado

SIMAN – *SIMulation ANalisys* Linguagem para simulação especializada em sistemas de manufatura

SIMULA – *SIMULATION LAnguage* – Linguagem de simulação

SIMSCRIPT – Linguagem de programação para simulação

SIPT – *Shortest Imminent Processing Time* – Menor tempo de processamento Individual

SLAM – *Simulation Language for Alternative Modeling* – Linguagem de simulação para Modelagem alternativa

SMF – Sistema de Manufatura Flexível

SOL – Linguagem de simulação orientada

SPT – *Shortest Process Time* – Menor tempo de processo

TTA – Tempo Total de Atraso

TS – *Tabu Search*

WIP – *Work in Process* – Estoques intermediários

VBA – *Visual Basic for Application*

α – Objetivo MKP

β – Objetivo TTA

γ – Objetivo NOA

INTRODUÇÃO

Neste capítulo são apresentadas as considerações iniciais, uma revisão da literatura, a relevância, a contribuição, o objetivo, a metodologia e a estrutura utilizada para a construção desta dissertação.

1.1. CONSIDERAÇÕES INICIAIS

Para se manterem competitivas frente às novas demandas do mercado globalizado, como o crescente aumento na expectativa dos consumidores e a necessidade de produzir com qualidade uma maior variedade de produtos em pequenas e médias quantidades, as empresas têm sido forçadas a repensarem em suas estruturas de produção com a finalidade de atender a multiobjetivos, como: reduzir os custos, os prazos de entrega, melhorar a qualidade e aumentar a produtividade, entre outros.

Assim, torna-se imprescindível identificar e ajustar de forma adequada os parâmetros de controle para os diferentes cenários de produção de tal forma que seu processo produtivo torne-se eficiente, flexível e as movimentações e o número de operações sejam os menores possíveis.

Conforme Ghinato (2000), o conceito desenvolvido pelo Sistema Toyota de Produção, de “zero” defeito, fez surgir uma forma de trabalho destinada ao melhoramento contínuo, na qual as perdas e os desperdícios devem ser eliminados, a chamada “Produção Enxuta”. Notadamente, aquelas empresas de sistema flexível de manufatura, que possuem um ambiente *job shop* que é caracterizado por um conjunto de ordens ou tarefas que são processadas em uma ou várias máquinas com critérios pré-determinados, onde a produção possui mix elevado,

as quantidades são pequenas, chegando a pedidos unitários, não se conhece a demanda, apenas sabe-se que ela é aleatória e segue uma distribuição de probabilidade.

De acordo com Watanabe, Ida e Gen (2005) dentre os principais e mais difíceis problemas enfrentados pelas empresas de manufatura flexível, está o sequenciamento da produção (*scheduling*), também chamado de agendamento, ou seja: identificar qual ou quais são os melhores modos de ordenar o programa de produção nas máquinas de tal forma que possam ser satisfeitos vários objetivos simultaneamente, como exemplo de objetivos, pode-se citar o atendimento do prazo de entrega, o uso da máxima capacidade produtiva, o menor tempo de troca (*set up*) de especificações e de máquinas.

Portanto, neste aspecto, a simulação tem sido uma das ferramentas mais utilizadas na área manufatureira, uma vez que ela proporciona meios para identificar, analisar e melhorar os parâmetros de produção e de processo, tais como: análise de quantidade de maquinário e operadores; tempos de processo; avaliação de desempenho e de procedimentos operacionais, bem como avaliar rotas de atividades onde se incluem o *layout* (arranjo físico), o sequenciamento de produção e fluxo produtivos, além de proporcionar um conhecimento dos pontos em que o sistema permite uma maior flexibilidade de produção (COSTA e JUNGLES, 2006; PONTES, YAMADA e PORTO, 2007).

Existem atualmente no mercado vários *softwares* comerciais de simulação, chamados simuladores, dentre os quais se podem citar o AutoModTM, ExtendTM, ProModel[®], MatLab[®] e Arena[®].

Nesta dissertação foi utilizado o Arena[®] (lançado em 1993, pela *Systems Modelling Corporation*) devido à disponibilidade de acesso e a interface amigável do *software*. O Arena[®] é formado por um conjunto de painéis (*templates*) com módulos e comandos, com os quais é possível modelar uma situação real, além de possuir uma interface gráfica que facilita a construção do modelo (BANKS, 1998; FREITAS FILHO, 2008; TORGÀ, 2007).

Conforme Person *et al.* (2006), um simulador permite testar e analisar diversas situações sem interromper o sistema real, porém ele é exclusivamente uma ferramenta de avaliação e suporte que por si só não apresenta uma otimização de desempenho, ou seja, não oferece a geração nem a otimização de soluções.

Em consonância ao autor citado anteriormente, Chwif e Medina (2007, p.157-158, escrevem que: “Embora a simulação computacional apresente uma série de vantagens [...] ela é uma ferramenta *avaliadora* de soluções e não uma *geradora* de soluções”. Então, para a obtenção de melhores soluções, costuma-se aliá-la a uma técnica de otimização. Nesse contexto, entende-se por otimização a escolha da melhor solução, ou do melhor conjunto de soluções, chamada de solução ótima, dentro de um conjunto de alternativas disponíveis para um determinado problema.

A otimização pode ser feita com técnicas heurísticas ou metaheurísticas. Técnicas heurísticas são utilizadas para encontrar boas soluções, mas não garantem que a solução seja ótima ou próxima da ótima (WEISE, 2008), enquanto que a técnica metaheurística utiliza a heurística com alguns mecanismos que possibilitam escapar de soluções que ainda não são as melhores e são chamadas de ótimos locais.

Nesse contexto, várias técnicas de otimização podem ser utilizadas, e entre elas pode-se citar o Recozimento Simulado - *Simulated Annealing* (SA), baseada no processo de recozimento de metais; Otimização baseada em Enxame de Partículas - *Particle Swarm Optimization* (PSO), metáfora do comportamento social de um bando de pássaros voando, e seus meios de informações e formação de voo; *Tabu Search* (TS). Árvore de Decisão - *Branch and Bound* (Ramificar e Podar), semelhante ao sistema de poda em árvores; Sistema da Colônia de Formigas *Ant Colony Systems*- (ACS ou ACO), com base na marcação do caminho do formigueiro até a comida; Sistema das Abelhas *Bee Systems* – (BS), tomando por

base o comportamento das abelhas e a sua capacidade de organização: operárias, rainha; (BASIRI *et al.*, 2009; LIAN, JIAO e GU, 2006; SERAPIÃO, 2009; XIA e WU, 2005).

Optou-se nesta dissertação pela utilização da metaheurística Algoritmos Genéticos (AGs), baseada nos princípios da evolução natural de Charles Darwin (1859), na qual a sobrevivência decorre de um processo de seleção feito pelo meio ambiente e apenas os indivíduos mais adaptados possuem chances de sobreviver e, por conseguinte, reproduzir-se, e nos trabalhos desenvolvidos por Gregor Mendel (1865, 1869) sobre genética natural (MITCHELL, 1999). Essa escolha se deu em função de algumas vantagens oferecidas pelo AG como a simplificação na formulação e na solução dos problemas, além de ser indicado para problemas de otimização complexos que envolvam muitas variáveis em um espaço de busca de dimensões elevadas. (AKASAWA, 2007; SANTA CATARINA, 2009; DING, BENYOUCEF e XIE, 2006). Ademais, o método é altamente paralelizável, pois permite a avaliação independente de cada conjunto de parâmetros viáveis na solução de determinados problemas (MITCHELL, 1997).

Os AGs representam um modelo de otimização desenvolvido por Holland (1975), na Universidade de Michigan, e podem ser definidos como sendo algoritmos que buscam a otimização a partir de um elenco de soluções candidatas, ao contrário de outras técnicas tradicionais que iniciam sua pesquisa a partir de uma única solução. A cada iteração, uma seleção é realizada para eliminar as soluções inferiores de modo a alcançar uma solução global ótima (GONÇALVES, MENDES e RESENDE, 2005; PATNAIK, BISWAS e MAHAPATRA, 2007).

Essas soluções pontenciais são codificadas como uma sequência de caracteres de comprimento fixo, chamado um cromossomo. Assim, para cada interação (também chamada de geração) uma avaliação destes cromossomos é realizada usando uma função de aptidão ou *fitness*, para atribuir um valor para cada solução. O valor de aptidão deve representar uma

solução boa, que é normalmente definida com relação à população atual, e é usado para classificar os membros da população a fim de proporcionar uma espécie de probabilidade limitar para operação de reprodução. Se, por exemplo, o objetivo da otimização com o AG é maximizar uma função, a solução com valor relativamente elevado de *fitness* deve ter uma probabilidade maior de reprodução. Em seguida, depois de definir um limite de probabilidade, a população é atualizada usando operações genéticas para a seleção, cruzamento e mutação e o processo é repetido até que um critério de parada, que pode ser definido como atingir um limite de tempo predefinido, um determinado número de gerações ou convergência de população, seja satisfeito (MITCHELL, 1997).

O processo de otimização dos parâmetros foi realizado usando o ambiente de algoritmo genético desenvolvido por GALib, uma biblioteca em C++ de componentes do algoritmo genético (WALL, 1996).

1.1.1. Escolha das medidas de desempenho

Para este modelo de simulação e otimização foi considerado a minimização das seguintes medidas de desempenho: *Makespan* (MKP), Tempo Total de Atraso (TTA) e o Número de Ordens Atrasados (NOA). Mesquita *et al.* in Lustosa *et al.* (2008, p. 166) definem MKP como sendo “tempo necessário para conclusão de todas as ordens abertas, ou seja, intervalo de tempo entre a liberação da primeira operação da primeira ordem e conclusão da última operação da última ordem processada”. Entretanto, para efeito desta dissertação, o termo *Makespan* refere-se aos tempos médios para conclusão das ordens. Considera-se TTA como sendo o somatório dos tempos de atraso das ordens ao final do processo, e como NOA a quantidade de ordens em atraso no final do processo, caracterizados quando o término da

produção de uma ordem ocorre após a data de entrega planejada (ARENALES *et al.*, 2007; SANTORO e MESQUITA, 2008; MESQUITA, *et al.* in LUSTOSA *et al.*, 2008).

Estas medidas de desempenho, embora não sejam totalmente independentes possuem uma otimização separada o que gera um problema multiobjetivo, uma vez que o objetivo é reduzi-las simultaneamente.

Destaca-se que as siglas MKP, NOA e TTA, foram definidas no âmbito deste trabalho apenas para facilitar a notação.

Detalhes sobre o acoplamento entre o simulador e o AG são apresentados na seção 4.3.

1.2. REVISÃO BIBLIGRÁFICA

O estudo da aplicação dos conceitos de modelagem e simulação e da otimização em problemas de seqüenciamento/agendamento da produção tem atraídos diversos pesquisadores, visando desenvolver metodologias que possibilitem às empresas conduzir sua atividade com maior eficiência. Desta forma, apresenta-se, neste item, uma parte do que tem sido estudado nos últimos anos, sobre este assunto.

Gupta e Sttaford Jr (2006) apresentam uma revisão histórica sobre o estudo de sequenciamento de produção, desde o ano 1954 até o ano 2004. Identificam neste período a evolução na busca por soluções de problemas de agendamento de produção, desde buscas puramente teóricas com programações matemáticas e simulação de Monte Carlo, numa época em que ainda não existiam computadores com capacidade para grandes cálculos e simulações, incluindo as soluções de análises combinatórias considerando principalmente a otimização do *makespan*. Eles relatam que após o aparecimento da Teoria de NP - Completo (subconjunto de problemas que podem ser reduzidos há um tempo polinomial) passaram a considerar outras funções objetivos como, por exemplo, tempos de processamento estocásticos. Na mesma

velocidade em que se desenvolveu a computação expandiu-se também novos métodos para tratar do problema de agendamento como já citado anteriormente, tais como o AG, Busca Tabu, entre outros.

Nesse contexto, as técnicas de simulação e otimização tem sido exaustivamente utilizada em diversas situações. Resumido na Tabela 1.1 a seguir, basicamente, a utilização da simulação, enquanto que a otimização será tratada oportunamente.

TABELA 1.1: Resumo de pesquisas envolvendo uso de simulação

Objetivo do estudo	Autor (es)
Redimensionamento de arranjo físico (processos de manufatura e prestação de serviços)	Costa e Jungles, 2006; Eken e Orneck, 2008; Gorgulho Junior e Gonçalves Filho, 2007; Montevechi, <i>et al.</i> , 2007; Oliveira, 2008; Soares, 2006; Pitombeira Neto, 2007; Pontes, Yamada e Porto, 2007.
Balanceamento de linhas de produção	Almeida, <i>et al.</i> , 2006; Farnes e Pereira, 2007; Uddin, Soto e Lastra, 2010.
Avaliação do comportamento de ordens de produção no processo em SFM ou de movimento de clientes em algum tipo de serviço (bancos, clínicas, hospitais, hotéis, etc.)	Akasawa, 2007; Alba <i>et al.</i> , 2007; Araújo, Santos e Arroyo, 2009; Baptiste, Flamini e Sourd, 2008; Basiri <i>et al.</i> , 2009; Chan e Chan, 2004; Ding, Benyoucef e Xie, 2006; Framinan e Leisten, 2008; Gonçalves, Mendes e Resende, 2005, 2008; Grabowski e Pempera, 2005; Huegler e Vasko, 2007; Sakurada e Miyake, 2009; Xu e Zhou, 2009.
Estoque de materiais em processo	Pitombeira Neto, 2007; Santoro e Mesquita, 2008.

Entre os trabalhos citados na Tabela 1.1 merece destaque aquele desenvolvido por Santoro e Mesquita (2008), pois aborda o problema de sequenciamento de produção em ambiente onde as ordens seguem uma rota específica e pré-determinada, conhecido por

ambiente *job shop*. Nesse trabalho, a simulação foi utilizada para avaliar o efeito do estoque de materiais em processo em relação ao atraso total para o atendimento da produção e o número total de ordens atrasadas. Dois ambientes distintos foram combinados com quatro regras de seqüenciamento de produção e após a criação do modelo usando o *Visual Basic* para aplicativos em planilhas *Excel* (Microsoft *Excel*TM) e as execuções dos experimentos puderam concluir que é possível diminuir o atraso total e o número de ordens atrasadas e ainda manter estoques intermediários chamados de *work in process* (WIP) com um volume constante ao invés de falta ou excesso de estoque nas máquinas.

Como a simulação é exclusivamente uma ferramenta de avaliação e não oferece nem a geração e nem a otimização de seqüenciamento, seu uso aliado à otimização é também bastante difundido. Observa-se que diversas pesquisas estudam modos de atuar sobre o seqüenciamento de ordens na produção de tal forma que as principais exigências das empresas sejam alcançadas, entre elas estão: reduzir WIP, as filas, o atraso de atendimentos aos clientes, aumentar a produtividade, a capacidade produtiva.

A investigação realizada por Moreira (2007), sobre o controle de fluxo de materiais, utilizando simultaneamente quatro possibilidades de decisão sobre a produção, a saber: aceitação ou rejeição de uma ordem, a definição da data de entrega, bem como qual ordem deverá ser processada primeira, além de considerar a possibilidade de quebra de máquina foram simuladas. Os resultados apresentados mostram uma melhoria no desempenho das medidas de atraso médio e tempo total médio gasto no sistema, enquanto que as falhas de máquina impactam o atraso médio de diferentes formas de acordo com a regra adotada.

Baptiste, Flamine e Sourd (2008), atentos á importância dos estoques na técnica de *just in time* (JIT), que é caracterizada pela produção da quantidade certa, no momento certo, apresentaram um estudo sobre o seqüenciamento de ordens num ambiente *job shop*, criaram uma equação para custear e penalizar o processo tanto pelo adiantamento das ordens que

geram um estoque, quanto do retardamento das ordens que geram um atraso de entrega ao cliente. Os autores usaram um algoritmo *branch and bound*, para encontrarem os limites superiores e inferiores da equação, e concluíram pela superioridade dos seus valores, tanto para os limites superiores quanto para os inferiores. Estudo semelhante foi desenvolvido por Araujo, Santos e Arroyo (2009).

O modelo desenvolvido por Akasawa (2007) propõe que a simulação computacional possa ser utilizada como ferramenta auxiliar e complementar no planejamento e no controle da produção de um ambiente *job shop*, em empresas vinculadas aos âmbitos de defesa e aeroespacial. O problema estudado consistia em analisar - a carga de produção e a previsão de término de cada item e a otimização do fluxo produtivo nos setores de usinagem, acabamento superficial e controle de qualidade, com o propósito de reduzir as filas.

Kumar e Sridharan (2007) apresentaram o estudo de simulação computacional conduzido para investigar o compartilhamento de ferramentas de corte em um Sistema Flexível de Manufatura (SFM) multimáquinas sob o efeito das regras de sequenciamento de produção quando as ferramentas estão ocupadas numa determinada máquina e surge uma ou mais solicitações de uso da mesma ferramenta. Conduziram sua pesquisa considerando três diferentes cenários na busca pela melhor identificação do parâmetro de substituição de ferramentas de corte. A utilização da simulação computacional permitiu-lhes compreender a dinâmica do sistema de produção e a análise dos resultados ajudaram a definir melhores políticas operacionais.

Também merecem destaque as pesquisas nas quais o objetivo é a redução do tempo de travessamento da ordem no processo (*makespan*). A minimização do *makespan* foi abordada em trabalhos de Baykasoglu e Ozba^kır (2008); Framinan e Leisten (2008); Gonçalves, Mendes e Resende (2005, 2008); Grabowski e Pempera (2005); Hendel e Sourd (2006); Januário *et al.* (2009); Leite e Arroyo (2006); Lian, Jiao e Gu (2006); Manikas e Chang

(2009); Morandin Junior *et al.* (2007); Qian *et al.* (2009); Ruiz e Stutzle (2008); Scholz-Reiter e Gorges (2010); Shevasuthisilp e Intawong (2009); Varadharajan e Rajendran (2005); Wang, Zhang e Zheng (2006); Watanabe, Ida e Gen (2005); Xia e Wu (2005); Xu e Zhou (2009); Yamamoto, Arruda e Libert (2007).

Em resumo, observa-se que o número de tarefas *versus* o número de máquinas *versus* o número de regras de entrada das ordens no sistema de produção são os principais causadores dos problemas de aumento de tempo total de atravessamento, atraso de entrega de produtos aos clientes, formação de filas nos postos de trabalho, antecipação ou atraso da produção, entre outros e que esses efeitos estão sendo tratados por pesquisadores em seus trabalhos. Em geral, as pesquisas estão propondo soluções que embora possam ser utilizadas em outros contextos, na maioria das vezes são boas para situações específicas conforme podem ser vistas nas citações anteriores.

1.3. CONTRIBUIÇÃO DA DISSERTAÇÃO

As características dos problemas de otimização no planejamento e controle da produção em SFM, bem como a dimensão dos problemas de interesse prático, conferem complexidade ao problema aqui tratado. Consequentemente torna-se difícil identificar os parâmetros que possam afetar o MKP, o TTA e NOA e a interação entre estes fatores.

Neste contexto: agendamento de produção, simulação e otimização, este trabalho apresenta uma proposta de abordagem diversa daquelas até então estudadas. Normalmente, as funções de aptidão utilizadas na otimização dos parâmetros de produção são fórmulas matemáticas que devido a sua complexidade necessitam de várias simplificações (MESQUITA *et al.* in LUSTOSA *et al.*, 2008) e em seus modelos são incluídas restrições que terminam por não representarem convenientemente a realidade do sistema. Então a

substituição da função de aptidão do AG por um modelo de simulação, pode significar uma simplificação uma vez que o simulador possibilita a criação de um modelo muito próximo da realidade, no qual as restrições são contempladas automaticamente.

1.4. OBJETIVOS

Para uma melhor compreensão, os objetivos foram divididos em: objetivo geral e objetivo específico:

1.4.1 Objetivo geral

O objetivo geral desta pesquisa é propor, desenvolver e testar uma abordagem computacional baseada no acoplamento de um modelo de simulação e o AG para a solução do problema de sequenciamento da produção num ambiente de manufatura *job shop*, isto é: encontrar por meio da simulação e da otimização computacional uma ou várias soluções satisfatórias para o problema de seqüenciamento de produção de empresas que possuem uma linha de produção composta de várias máquinas, não necessariamente diferentes, que recebem ordens de produção com um mix elevado e que seguem um roteiro de produção pré-definido com uma ou mais operações realizadas sequencialmente.

1.4.2 Objetivos específicos

O primeiro objetivo específico deste trabalho foi construir um modelo de simulação e acoplá-lo ao AG.

O segundo objetivo foi verificar a viabilidade, a eficiência e a eficácia da abordagem p

proposta, na qual se aplica um modelo de simulação ao invés de uma modelagem matemática analítica como função de aptidão (*fitness*) nos AGs.

O terceiro objetivo específico foi criar um modelo de simulação de eventos discretos que possa ser utilizado diariamente como uma ferramenta de apoio ao SFM e que propicie meios rápidos e seguros de sequenciamento de tarefas na produção.

O quarto objetivo específico foi melhorar o desempenho do sistema analisado através da otimização multiobjetivo com vistas a reduzir o MKP, o TTA e o NOA e não somente contribuir como uma previsão de cenários futuros.

1.5. RELEVÂNCIA DO TRABALHO

A busca por melhoria contínua no processo de produção, especificamente no sequenciamento de tarefas, torna a contribuição deste trabalho de simulação numa proposta de abordagem de uso simples, que pode ser utilizada no planejamento da produção, pois oferece respostas eficientes em um ambiente de constantes mudanças e ao mesmo tempo poder auxiliar estrategicamente os tomadores de decisões no processo diário de uma empresa de manufatura.

O tema de sequenciamento de produção, segundo Mesquita *et al.* in Lustosa *et al.* (2008, p.182) ainda permanece relevante, pois escreve que:

... Apesar da grande quantidade de conhecimento científico desenvolvido pelos pesquisadores do mundo inteiro durante quase meio século de pesquisas, os problemas reais de programação ainda impõe desafios tanto à comunidade acadêmica, quanto aos profissionais de Engenharia de Produção e Sistemas de Informação (MESQUITA *et al.* in LUSTOSA *et al.*, 2008, p182).

1.6. MATERIAIS, MÉTODOS E TÉCNICAS

O método de pesquisa axiomática quantitativa é recomendado quando se usa a técnica de simulação, pois, conforme Bertrand e Fransoo (2002), embora esta técnica possa levar a resultados qualitativamente inferiores cientificamente, poderá proporcionar resultados superiores quando o problema ou o processo a ser estudado não pode ser justificado apenas com as análises da matemática pura. A simulação assume de forma axiomática que é possível construir um modelo objetivo, que possa explicar total ou parcialmente os processos operacionais da realidade bem como captar as nuances que envolvem a tomada de decisão.

A utilização da simulação se justifica neste caso, devido à necessidade de se resolver vários problemas de otimização combinatória, que são caracterizados pela busca de soluções ótimas em conjuntos finitos.

Banks e Gibson (2009) apontam que alguns pesquisadores sentem dificuldade em parar o refinamento tornando o modelo cada vez maior, enquanto que outros param muito cedo e ambos os procedimentos tornam estes modelos mais difíceis de serem verificados e validados. Portanto, desde que não é possível declarar um modelo de simulação ótimo para serem verificados e validados, muitos testes deve ser aplicados para conseguir a garantia de que o modelo possa ser usado como uma representação da realidade.

A etapa de verificação e validação é das mais importantes e difíceis na simulação, pois seus objetivos são: reduzir a desconfiança, aumentar a aceitação e a credibilidade por parte dos tomadores de decisão. Com a finalidade de verificação, as seguintes perguntas foram respondidas conforme pode ser visto na seção 3.4: O modelo esta implementado corretamente? Os parâmetros de entrada e a estrutura lógica estão apresentados corretamente? O número de amostras é suficiente?

Nesta dissertação foram testados alguns limites de processamento até encontrar o número de vezes adequado para responder as perguntas acima. Utilizando-se a Análise dos dados do *Excel* foi possível verificar se a quantidade de amostras utilizadas foi suficiente para representar a superfície de resposta para os três objetivos propostos anteriormente.

Enquanto que para validação comparou-se os resultados obtidos numa planilha *Excel*, com os valores retornados pelo Arena, observando uma situação de valores fixos no simulador. Ainda para a validação foi construído um modelo de menores proporções a fim de garantir que o sistema estava mesmo preparado para o processamento considerando as regras de seqüenciamento baseadas em prioridade conforme apresentado no capítulo 4.4.

Construiu-se um modelo com base nas características apresentadas no artigo publicado por Santoro e Mesquita (2008), considerando oito máquinas e dez rotas de produção, com no máximo 10 operações em cada rota. Dentre as várias bibliotecas existentes para a implementação do AG, optou-se pela GAlib, desenvolvida por Wall (1996) do Instituto de Tecnologia de Massachusets - *Massachusetts Institute of Technology* (MIT), pois se trata de uma solução gratuita e de código aberto baseada na linguagem C++, além de ser amplamente utilizada e, portanto, confiável.

Para o desenvolvimento do modelo de simulação e otimização proposto nesta dissertação, cuja finalidade foi determinar o melhor sequenciamento de produção, de forma a minimizar os tempos totais de atraso (TTA), o número de ordens atrasadas (NOA) e o tempo total de travessamento (MKP), foi feito um acoplamento entre o Arena e o AG. Neste acoplamento foi utilizado o controlador *ActiveX Automation*, que é uma ferramenta da Microsoft que permite a automação do modelo através de um controle externo, e a linguagem *Visual Basic for Application* (VBA), que possibilita o controle sobre o envio e o recebimento de arquivos textos entre os aplicativos e fornece acesso aos objetos e informações estatísticas do modelo de simulação no ARENA®.

Ambas as ferramentas gratuitas e disponíveis, respectivamente, nos endereços eletrônicos: <http://www.lancet.mit.edu/ga/> e <http://www.paragon.com.br/>.

Como o modelo de simulação proposto tem elementos aleatórios como, por exemplo, a chegada de ordens e o tempo de processamento, as respostas do modelo (*MKP*, *TTA* e *NOA*) também são variáveis aleatórias. Desta forma, executar modelos de simulação é o mesmo que conduzir um experimento estatístico, ou seja, deve-se testá-lo repetidas vezes e realizar uma análise estatística que inclua um intervalo de confiança ou um intervalo de precisão. Para a análise estatística dos resultados da simulação neste trabalho optou-se pelo uso do pacote *Microsoft Office Excel 2007*.

Esta dissertação cita apenas as regras mais utilizadas na literatura, mas não esgota o tema visto que Panwalkar e Iskandar (1977) conseguiram identificar e analisar aproximadamente 100 regras de sequenciamento de produção.

1.7. ESTRUTURA DO TRABALHO

A dissertação esta dividida em 5 capítulos, sendo que o primeiro capítulo aborda a introdução com as considerações iniciais, o objetivo, o estado da arte, a relevância, a contribuição, os materiais, técnicas e métodos. No segundo capítulo apresenta-se a revisão da literatura sobre modelagem e simulação, regras de sequenciamento, e otimização com AGs. O terceiro capítulo é dedicado à aplicação do modelo de simulação e otimização dos parâmetros, a análise e a discussão dos resultados obtidos estão no quarto capítulo e para finalizar o quinto capítulo trata das conclusões e recomendações para trabalhos futuros.

2. FUNDAMENTAÇÃO TEÓRICA

Apresentam-se neste capítulo os conceitos que são necessários para a compreensão do trabalho desenvolvido.

Na primeira seção são apresentadas a simulação computacional, as vantagens e desvantagens, validação e verificação do modelo e os principais *softwares* de simulação existentes atualmente no mercado.

Na segunda seção estão descritos os métodos de análises estatísticas, como se define um intervalo de confiança para os resultados obtidos, a escolha dos testes e a definição do número de replicações.

A terceira seção contém os principais conceitos relacionados a um sistema de produção como: formação de filas, regras de seqüenciamento, tipos de produção, medidas de desempenho, destacando-se o problema de sequenciamento da produção em ambientes *job shop*.

Na quarta seção discorre-se brevemente sobre a técnica metaheurística de otimização, e de forma suficientemente detalhada e com alguns exemplos apresenta-se o AG.

2.1. SIMULAÇÃO COMPUTACIONAL

A década de 1950 é citada como sendo o início da simulação computacional. Devido principalmente ao alto custo a simulação era voltada basicamente para a área militar e para algumas empresas de grande porte como indústrias de aço e aeroespacial. Existiam poucos especialistas que desenvolviam modelos de simulação complexos usando a linguagens de programação como o FORTRAN (FORmula TRANslator). Na proporção em que estas linguagens evoluem, tornam-se mais fáceis de serem compreendidas e utilizadas, sendo que

atualmente os *softwares* tendem a atender aplicações específicas como: manufatura, serviços, telecomunicações, entre outras (CHWIF e MEDINA, 2006; KELTON, SADOWSKI e SADOWSKI, 2000).

Um resumo desta evolução pode ser vista na Figura 2.1.

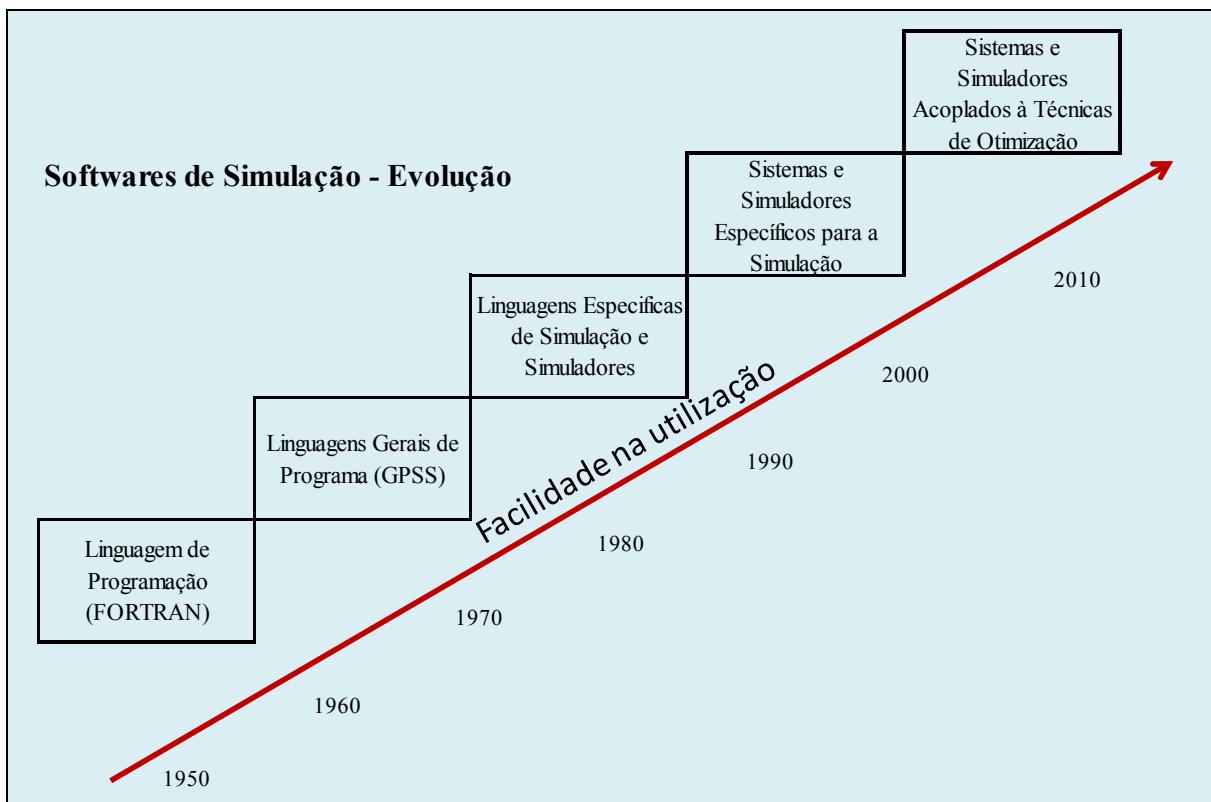


FIGURA 2.3: A evolução dos *softwares* de simulação
Fonte: Adaptado de Chwif e Medina (2006)

Embora a simulação computacional tenha recebido diversas definições ao longo do tempo, optou-se por utilizar, como exemplo, as duas definições a seguir:

- “Simulação é a aplicação de uma metodologia que permite descrever o comportamento de um sistema usando um modelo matemático ou um modelo simbólico” (SOKOLOWSKI, 2009).
- A simulação computacional refere-se aos métodos para estudar diversos modelos reais ou artificiais em sistemas de avaliação numérica utilizando-se de *softwares* projetados

para imitar um sistema de operações e/ou características normalmente por um período de tempo. De forma prática, significa que é um processo de criação de um modelo computadorizado tendo por objetivo a realização de experimentos numéricos de tal forma que proporcione uma compreensão deste sistema submetido a um determinado conjunto de condições (KELTON, SADOWSKI e SADOWSKI, 2000).

A partir da década de 1990, com o desenvolvimento de *softwares* de simulação com uma interface mais amigável e com capacidade de processamento gráfico, aliado a características específicas de simulação, onde se necessita avaliar sistemas complexos, modelar comportamento dinâmico e principalmente, quando não existe uma solução analítica, a simulação computacional torna-se uma alternativa segura, se confrontada com a simulação em sistemas reais, nos quais pode haver sérios riscos tanto pessoais quanto materiais, além do custo envolvido.

A simulação computacional é capaz de trabalhar como uma variedade maior de modelos científicos do que uma análise matemática. A simulação é utilizada em casos onde os modelos ou problemas são muitos complexos para uma análise matemática formal (BERTRAND e FRANSOO, 2002).

Os modelos de simulação podem ser contínuos ou discretos:

- a) Modelo de simulação contínua: proposto para solucionar problemas que variam continuamente ao longo do tempo. Emprega equações diferenciais para calcular as mudanças das variáveis de estado ao longo do período estudado. A definição para variáveis de estado é dada como um conjunto mínimo de variáveis capazes de descrever totalmente o comportamento do sistema em um determinado instante de tempo.
- b) Modelo de simulação discreta: proposto para solucionar problemas que mudam o seu estado em pontos discretos no tempo, em função da ocorrência de eventos de modo

especial em sistemas de manufatura.

Eventualmente pode ser necessária a construção de um modelo que inclua uma combinação dos modelos contínuos e discretos (CHWIF e MEDINA, 2006; KELTON, SADOWSKI, e SADOWSKI, 2000; ALTIOK e MELAMED, 2007).

A simulação computacional pode ser aplicada em diferentes áreas e sistemas, como por exemplo: na área de Transporte no sistema de alterações de procedimentos operacionais em companhias aéreas, portuárias, ferroviárias, etc. Na Computação no sistema de criação de web sites, na Administração, na Medicina, na Engenharia, na Produção nos sistemas de *layout* entre outras.

Conforme ressaltado por Chwif e Medina (2006), não se pode esperar que o uso da simulação computacional preveja o futuro, mas sim que ela possa prever com certa confiança e respeitado um conjunto de premissas, o comportamento de um sistema com base em dados de entradas específicos.

A simulação computacional não pode ser considerada como um modelo matemático, embora utilize fórmulas matemáticas na busca por soluções dos diversos sistemas. Não deve ser confundida com uma técnica de otimização, pois ela é uma ferramenta de análise de cenários. Porém, ela pode ser combinada com algoritmos de otimização para identificar melhores soluções.

A simulação computacional não pode substituir o ser humano no processo de tomada de decisão e também não deve ser considerada como o último recurso e a solução para todos os problemas.

Os passos utilizados na formulação de um estudo envolvendo modelagem e simulação estão apresentados na Figura 2.2.

FIGURA 2.4: Passos na formulação de um estudo envolvendo modelagem e simulação
Fonte: Freitas Filho, p.29 (2008)

Na impossibilidade de utilização de um sistema real para construir o modelo, pode-se buscá-lo na literatura. Ressalta-se que para a realização desta pesquisa, utilizaram-se as principais características do ambiente arquitetado por Santoro e Mesquita (2008), conforme pode ser visto na seção 3.1.

2.1.1. Vantagens e Desvantagens

Uma vez que a simulação é uma ferramenta que pode ser usada para estudar desde os estágios iniciais de um projeto até sólidos sistemas já existentes, vários autores, entre eles Altiok e Melamed (2007); Banks (1998); Freitas Filho (2008); Kelton, Sadowski e Sadowski (2000); Montgomery (2005) apontam vantagens e desvantagens na sua utilização.

Como vantagens têm-se, entre outras:

- Que o modelo desenvolvido pode ser utilizado várias vezes com a finalidade de avaliar novos projetos; mesmo que as informações de entrada sejam apenas esquemas,

rascunhos ou incompletas.

- Possibilita analisar e entender quais são as principais variáveis do sistema, como elas interagem entre si e com os outros elementos do sistema em relação às medidas de desempenho.
- Permite identificar os pontos em que há formação de filas no sistema.
- Admite a replicação exata dos vários experimentos, portanto podem-se testar diversas alternativas para o sistema.
- A simulação computacional permite a cobertura de longos períodos, com um tempo reduzido de simulação.
- Pode reproduzir certos fenômenos de forma mais lenta ou mais acelerada, dilatar ou comprimir o tempo, permite conhecer o sistema da forma como ele realmente ocorre e não como se pensa que ele ocorra.
- Permite exercer um controle sobre as condições experimentais de tal forma que se pode replicar o experimento fixando-se todos os valores desejados para todos os parâmetros.
- Sua aplicação, em geral é mais fácil que métodos analíticos, pois estes exigem um grande número de simplificações que acabam por limitar a análise a uma reduzida quantidade de medidas de desempenho.
- Avaliar novas políticas e procedimentos operacionais, regras de decisão, fluxo de informação, etc. podem ser feita sem perturbar o normal andamento do sistema real.
- Possibilita encontrar respostas para as perguntas “como”, “por que” e “o que aconteceria se” determinados fenômenos acontecessem possam ser testados e confirmados sem que o sistema real seja prejudicado.

Como principais desvantagens têm se que:

- Esta técnica não oferece respostas otimizadas, pois, apenas testa as proposições fornecidas.
- Os modelos devem ser executados diversas vezes a fim de se prever o desempenho do sistema.

Não chega a ser uma desvantagem, mas da mesma forma que qualquer outro tipo de estudo, é necessário um treinamento específico no sentido de que, o aprendizado se dá de acordo com a experiência. Exige-se um conhecimento profundo do sistema simulado para criar o modelo, experimentá-lo e interpretar corretamente os resultados apresentados pela simulação, pois estas fases consomem muito tempo e recurso. Deve-se cuidar para que uma eventual simplificação não gere resultados insatisfatórios.

2.1.2. Experimentação, verificação e validação do modelo

Para garantir que o modelo utilizado reflita de fato o sistema que se deseja estudar é importante que sejam realizados experimentos na simulação, da mesma forma como são feitos na pesquisa empírica (FREITAS FILHO, 2008; KELTON, SADOWSKI e SADOWSKI, 2000).

A experimentação é definida como um ou uma série de testes, nos quais se pode observar as modificações nas respostas do sistema e identificar as razões destas mudanças quando elas são feitas em alguma das variáveis de entrada. A experimentação oferece um modo sistemático, disciplinado e controlado para a avaliação das atividades desenvolvidas nas organizações, reduz o número de testes e otimiza o uso de recursos (MONTGOMERY, 2005).

Conforme Barros Neto, Scarminio e Bruns (2007), na realização de experimentos e na análise de seus resultados são necessários a utilização do bom senso, e cuidar para que o

chamado senso comum não leve a conclusões que ainda não são as melhores. Portanto alguns princípios básicos de planejamento devem ser seguidos de forma rigorosa para que os resultados obtidos possam ser verificados/confirmados.

O princípio básico da verificação é assegurar que o sistema real está sendo representado pelo modelo inicial. Enquanto que a validação tem como objetivo garantir que o modelo inicial represente com acuracidade o sistema real (PONTES, YAMADA e PORTO; 2007). Objetiva-se, desta forma, garantir que as informações sejam confiáveis e que os recursos disponíveis para a experimentação sejam bem utilizados.

A verificação é constituída por uma série de ações utilizadas para assegurar se a forma conceitual adotada na formulação do modelo foi transcrita de forma correta e consistente, de acordo com a linguagem de programação ou de simulação adotada. Aconselha-se analisar os dados de saída depois de várias execuções, com diferentes situações, comparando-os com valores encontrados no sistema real, ou na literatura existente. Se possuir animação, verificar se esta imita o sistema real. Rastrear o programa para verificar como os procedimentos estão sendo executados (BANKS *et al.*, 2009).

A validação consiste num conjunto de ações utilizadas para confirmar se um determinado modelo representa de forma fiel o sistema em estudo. Classifica-se em validação estatística e validação subjetiva (BERTRAND e FRANSOO, 2002).

Enquanto a validação estatística utiliza como ferramentas, entre outras, a análise de variância, determinação de intervalo de confiança, testes de hipótese, análise de regressão, ajustamento de curvas, a validação subjetiva é indicada quando inexistem condições apropriadas para um aprofundamento, então o modelo será considerado válido se os resultados encontrados tanto pelo sistema real quanto pelo modelo forem apresentados de forma idêntica a um grupo de especialistas no assunto e/ou conhecedores do sistema e estes não conseguirem identificar de forma consensual a origem dessas informações (ALTIOK e

MELAMED, 2007; FREITAS FILHO, 2008; KELTON, SADOWSKI e SADOWSKI, 2000).

Em resumo, o processo de verificação e validação do modelo consiste em comparar os resultados obtidos na simulação com os resultados apresentados nas observações realizadas sobre o sistema real. Quanto mais próximos são estes valores, mais confiável é o modelo. Na impossibilidade de se ter os resultados reais, podem ser utilizados resultados similares obtidos na literatura e testes de validação estatísticos, como a aderência que mede o quanto a freqüência observada está próxima da freqüência esperada, nível de confiança que representa a probabilidade de que o intervalo estimado contenha o parâmetro populacional. Mais detalhes relativos a análise estatística poderão ser vistos na seção 2.2.

Observa-se que muitos sistemas reais são afetados por variáveis aleatórias e que uma única observação da realidade não é suficiente para criar e nem validar um modelo, da mesma forma que somente uma replicação do experimento não fornece resultados consistentes.

Portanto, se neste momento ocorrer anomalias é necessário retornar aos passos anteriores de experimentação. Estes passos estão definidos, na Figura 2.2: Passos em um estudo envolvendo modelagem e simulação, na seção 2.1.

2.1.3. *Softwares de Simulação*

Com o aparecimento dos computadores digitais entre as décadas de 50 e 60, surgem também as linguagens de programação, entre elas, o FORTRAN utilizado para efetuar simulações complexas como o cálculo de estruturas, estatísticas de contabilidade e processamento de lista de materiais, entre outras. Estas simulações eram personalizadas e inflexíveis. No caso de alterações, normalmente era menos dispendioso e problemático iniciar uma nova simulação.

Atualmente, devido à competição entre as empresas produtoras de *softwares* de

simulação, são feitos lançamentos de produtos cada vez mais poderosos que oferecem novas e maiores facilidades tais como ferramentas de suporte ao processo de modelagem, recursos de análises estatísticas e interfaces gráficas mais amigáveis.

Os *softwares* de simulação funcionam basicamente apoiados em interfaces gráficas, onde o usuário pode utilizá-lo de forma intuitiva através de menus, gráficos e caixas de diálogos. A possibilidade de construir modelos executados com animações pode facilitar o entendimento do sistema, pois permitem a inclusão de movimentos que alteram a sua dinâmica.

Ao lado dos *softwares* de simulação estão as linguagens de simulação, que formam um conjunto de um ou mais programas destinados a aplicações mais particulares. Sua principal vantagem é proporcionar abertura para a geração dos mais diversos tipos de sistema, porém exige que seus usuários tenham um profundo conhecimento deste tipo de linguagem para poderem criar sistemas mais complexos. Exemplos destas linguagens são: GPSS, SIMAN, SLAM, SIMSCRIPT, MATLAB, JAVA, PL, C++, VBA, SIMULA, DELPHI, ASPOL, SOL.

A Tabela 2.1 apresenta uma relação de alguns *softwares* de simulação atualmente no mercado. A lista não é exaustiva e a omissão de outros *softwares* não é proposital.

TABELA 2.3: Relação de *Softwares* de simulação atuais

PRODUTO	EMPRESA	Homepage
<i>Arena</i>	<i>Rockwell Automation</i>	www.rockwellautomation.com
<i>Automod</i>	<i>Applied Materials</i>	www.appliedmaterials.com
<i>Extend</i>	<i>Imagine That Inc</i>	www.extendsim.com
<i>GPSS/H</i>	<i>Wolverine Software Corporation</i>	www.wolverinesoftware.com
<i>ProModel</i>	<i>ProModel Corporation</i>	www.promodel.com
<i>Simscript III</i>	<i>CACI Products Company</i>	www.caciasl.com
<i>Simprocess</i>	<i>CACI Products Company</i>	www.caciasl.com
<i>Witness</i>	<i>Lanner Group</i>	www.lanner.com
<i>Sim Factory</i>	<i>Sim Factory LCC</i>	www.thesimfactory.com
<i>processModel</i>	<i>Processmodel</i>	www.processmodel.com

Fonte: Internet - Compilado pela autora

Neste trabalho foi utilizado o *software* de simulação Arena em sua versão acadêmica, em destaque na Tabela 2.1, principalmente em função da sua disponibilidade e facilidade de uso, e também por possuir funcionalidades que permitem a integração com linguagens genéricas de programação, como *C++*, *Visual Basic (VB)* e *Java*, e com aplicativos externos através do controlador *ActiveX/OLE 3.0 Automation* e da linguagem *Visual Basic for Application (VBA)*.

2.2. ANÁLISES ESTATÍSTICAS

No contexto da simulação computacional, conhecer e saber usar métodos estatísticos pode ser a diferença entre apresentar uma boa solução para determinado problema ou apresentar qualquer solução (BANKS, 2009).

A importância da análise estatística reside na necessidade de saber qual a probabilidade da casualidade ser a responsável pelo resultado encontrado na pesquisa ou no experimento. As análises estatísticas podem ser descritivas quando apenas expõe os resultados, ou inferencial quando analisa as informações.

A estatística descritiva procura somente relatar e avaliar uma determinada amostra sem tirar quaisquer conclusões ou inferências sobre a população. Utiliza-se de gráficos, tabelas e medidas de sínteses como porcentagens, índices e médias.

Em contrapartida, a estatística inferencial permite que sejam feitas conclusões, estimativas e generalizações sobre um conjunto de informações de tal forma que se possa inferir sobre o comportamento da população.

Na estatística inferencial utilizam-se dois tipos de tratamento: estimativa dos resultados, no qual os resultados são apresentados com intervalos de confiança, e teste de hipóteses no qual os valores são apresentados como valores de probabilidades por meio de

testes estatísticos (MONTGOMERY, 2005).

Nesta dissertação optou-se pela utilização do intervalo de confiança para validar a quantidade de replicações do simulador.

2.2.1. Intervalo de confiança (IC)

Confiança estatística é a idéia do quanto se acredita que o valor encontrado esteja dentro de um determinado intervalo, isto é, quanto mais perto este resultado estiver da realidade observada, maior é a probabilidade de que ele seja um bom resultado. (CHWIF e MEDINA, 2006).

Define-se como intervalo de confiança um intervalo estimado de um parâmetro estatístico, ou seja: ao invés de estimar o parâmetro por apenas um valor, é dado um intervalo de estimativas prováveis. Saber quanto prováveis são essas estimativas depende do grau de confiança adotado. A definição do grau de confiança é utilizada para descrever uma probabilidade com um valor em percentual denominado de α . Quanto maior é o valor de α maior é a probabilidade de que o parâmetro encontrado representa a população, porém devem-se evitar os casos extremos de intervalo de confiança, pois como o intervalo é dado por $1 - \alpha$, se α for igual a zero o intervalo é de 100% e obriga-se a buscar todos os pontos e o inverso também é verdadeiro, ou seja, se α é igual a 100, o intervalo se reduz a zero. (MARTINS, 2008, p.68).

A Figura 2.3 apresenta uma expressão gráfica na qual $1 - \alpha$ é o intervalo de confiança e $\frac{\alpha}{2}$ é o erro estimado, para uma distribuição Normal, a qual é caracterizada pela posição da média no centro da distribuição e pelo desvio-padrão equidistante do ponto médio. (MONTGOMERY, 2005).

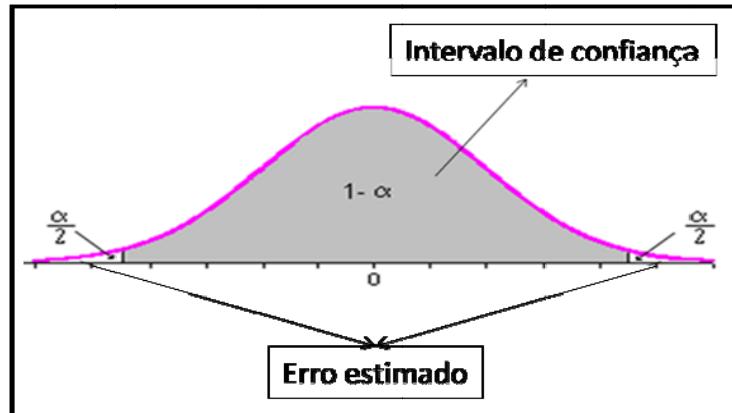


FIGURA 2.3: Ilustração de intervalo de confiança
Fonte: Adaptação de Montgomery (2005)

2.3. SISTEMAS DE PRODUÇÃO

A definição e a classificação dos sistemas de produção são temas consolidados na literatura especializada e são apresentadas de modo semelhantes pelos diferentes autores pesquisados, entre eles Costa *et al.* In Lustosa *et al.* (2008), Ghinato (2000), Barco e Villela (2008). Essa definição e classificação são apresentadas abaixo com base nesses autores.

Entende-se por sistemas de produção a existência de transformação de recursos de entrada (*input*) em saídas (*output*) sob a forma de bens ou serviços. Uma ilustração do sistema pode ser vista na Figura 2.4.

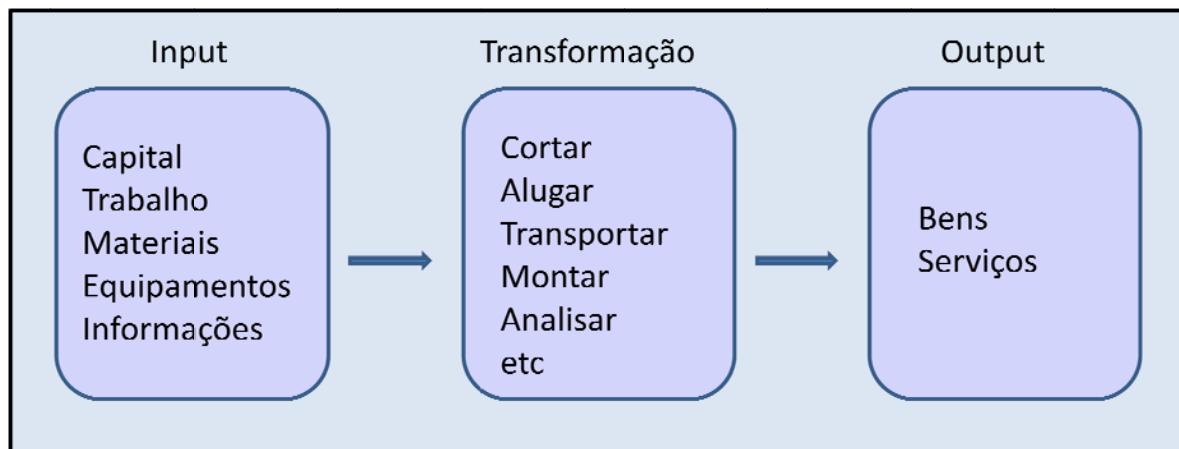


FIGURA 2.4: Esquema de Produção
Fonte: Costa *et al.* in Lustosa *et al.* (2008)

Os sistemas de produção, segundo Costa *et al.* in Lustosa *et al.* (2008), podem ser classificados de acordo com:

- Nível de padronização dos produtos, cujas características são:
- Produtos padronizados: bens ou serviços que apresentam um alto grau de uniformidade; são produzidos em grande escala. Normalmente, devido a sua organização, normalmente também há a padronização na alocação dos recursos, nos métodos de trabalho e nos controles de produção. Exemplos: eletrodomésticos, roupas, alimentos industrializados.
- Produtos sob medida ou personalizado: bens ou serviços que são desenvolvidos para atender um cliente específico, o que ocasiona uma ociosidade nos equipamentos e onde não são possíveis a padronização, os métodos e os recursos. Exemplos: construção civil, estaleiro, alta costura.

Tipos de operação, que se caracterizam por diversos tipos de processos que podem ser contínuos ou discretos. São considerados contínuos quando não se pode identificar individualmente e apresentam alta uniformidade de produção, são passíveis de automação por possuírem produtos e processos interdependentes, com pouco ou nenhuma flexibilidade. Exemplos: Energia elétrica, combustível. São discretos quando podem ser isolados em unidades ou em lotes, a saber:

- Repetitivo em massa: produção em grande escala, demandas estáveis, estrutura especializada e pouco flexível. Exemplos: Indústrias automotivas, de eletrodomésticos, serviços de transporte aéreo.
- Repetitivo em lote: produção em média e pequena escala, com sequência de operações programadas assim que as operações anteriores forem realizadas; estrutura relativamente flexível, pouca especialização de equipamentos e mão de obra polivalente que visa atender diferentes pedidos de clientes e flutuações de demanda.

Exemplos: Oficinas de reparo para automóveis, e aparelhos eletrônicos, laboratório de análises clínicas, restaurantes.

- Por projeto: atende uma necessidade específica do cliente, com o qual possui ligação estreita, alta flexibilidade de recursos produtivos. Exemplos: fabricação de navios, aviões, usinas, propaganda, arquitetura.
- Ambiente de produção que é caracterizado pelo posicionamento do estoque no processo e por informar a complexidade do fluxo de materiais:
 - *Make-to-stock*: produzir para estoque, baseados em previsão de demanda, produtos padronizados para atendimento rápido ao cliente, alto custo do estoque. Exemplos: produtos de prateleira e de consumo geral
 - *Assembler-to-order*: montagem sob encomenda, produtos pré-fabricados ou modulares, que depois são montados de acordo com o cliente, atendimento em médio prazo, custo razoável do estoque. Exemplos: computadores pessoais, celulares, móveis.
 - *Make-to-order*: produzir sob encomenda, a produção só inicia após a formalização do pedido pelo cliente, prazo de atendimento é alto e o estoque está concentrado no início do processo. Exemplos: pintura, a maioria dos serviços.
 - *Engineer-to-order*: engenharia sob encomenda, produção da qual o cliente participa desde o início, antes mesmo do pedido formal, não possui estoque, prazo de atendimento é muito longo, possui alta complexidade no fluxo de materiais. Exemplos: Construção de uma usina, de uma ponte.

Fluxo do processo que possui características de processo em linha, ou em lote, ou por projeto.

- Os processos em linha distinguem-se por terem uma sequência de operações muito bem definidas e por apresentarem operações de precedência e operações subseqüentes.

Nestas operações, os produtos devem estar padronizados e fluírem de uma operação para outra em uma sequência pré-estabelecida a fim de que uma não retarde a outra. São classificadas por tipo de produção em massa como em linhas de montagem, ou produção contínua como a indústria de eletricidade.

- Os processos em lote são caracterizados por criar uma grande variedade de produto com baixo volume de produção que possuem uma sequência própria de tarefas, um fluxo intermitente, alta flexibilidade, dificuldade de controle devido ao fluxo desordenado.
- A principal característica de um processo por projeto é a produção de um único produto, como a construção de um prédio ou de um navio, cuja sequência de tarefas é organizada em função desta própria característica.

Quanto à natureza dos produtos os sistemas de produção podem ser classificados como bem ou serviço, ou seja, quando um sistema de produção gera algo tangível é chamado de manufatura de bens e quando gera algo intangível é chamado de prestador de serviços.

Os sistemas de produção no que tange ao ordenamento da produção podem ser organizados de duas formas: produção empurrada e produção puxada.

- Produção empurrada: É um sistema que inicia seu processo enviando materiais para a primeira etapa de acordo com uma previsão de demanda e em seguida, o produto semi-acabado é lançado para a próxima etapa sem que haja uma requisição por parte desta, isto é, os materiais são empurrados ao longo do processo produtivo
- Produção puxada: É um sistema que só inicia seu processo no momento exato e na quantidade exata da sua necessidade de acordo com o pedido do cliente, ou seja, ela ajusta a sua produção à demanda pelo produto.

Neste aspecto, um dos mais conhecidos métodos para controle de produção puxada é o sistema *Kanban*. Sua principal função é regular os níveis de estoque, mantendo-os nos

menores níveis possíveis, sem afetar a produção. Ele exerce seu controle através do uso de cartões entre postos de trabalho consecutivos, permitindo que a produção se inicie no posto anterior somente quando o estoque intermediário do posto posterior está baixo.

Este método surgiu no Japão pós-guerra, como parte de um novo conceito de gerenciamento de produção que procura otimizar a organização de tal forma que o cliente possa ser atendido no menor prazo, com maior qualidade e com o menor baixo custo e que ficou conhecido como “Sistema de Produção Enxuta” ou Sistema Toyota de Produção.

Visando confirmar a eficiência do sistema puxado sobre o sistema empurrado, Costa e Jungles (2006) compararam esses dois processos de produção simulando uma linha de montagem e concluíram que a produção puxada aproxima-se mais da produção desejada pelos clientes em função do atendimento das ordens no prazo certo, da redução do estoque, o que torna o fluxo da informação mais objetiva, aumenta a produtividade e ainda mantém o local de trabalho mais ordenado.

2.3.1. Sistema *job shop*

Segundo MacCarthy e Liu (1993) apud Morais, Menegardi e Cantieri (2009) as principais causas dos problemas de produção podem ser devida ao tipo de rota das ordens nas máquinas ou ainda de acordo com o tipo de processo.

Existem vários tipos de rotas e processos, sendo que os mais usuais podem ser definidos como:

- a) máquina única: utilização de uma máquina e um estágio de produção;
- b) máquinas paralelas: uso de máquinas semelhantes para um estágio de produção, onde cada ordem precisa de apenas uma destas máquinas;
- c) *flow shop*: todas as ordens possuem a mesma rota de processamento, com uso de

apenas uma máquina por vez;

- d) *job shop*: cada ordem tem sua própria rota de processamento, utilizando apenas uma máquina em cada estagio de produção e
- e) *job shop* com múltiplas máquinas: em cada estágio de produção possui um conjunto de máquinas paralelas, em pelo menos um estágio, porém cada ordem é executada em apenas uma máquina em cada estágio.

O modelo clássico de *Job Shop* tratado na literatura apresenta as seguintes características: um conjunto de n ordens $\{O_1, O_2, O_3 \dots O_n\}$ que deve ser processado em m máquinas $\{M_1, M_2, M_3 \dots M_m\}$ de acordo com p processos $\{P_1, P_2, P_3 \dots p_n\}$ e algumas restrições tais que: existe uma sequência de processo, cada máquina processa uma ordem por vez, esta ordem deve ser processada do começo ao fim, os tempos de processamento podem ser fixos ou variados, os prazos de entregas podem ser diferentes. Então, um dos maiores problemas é combinar estas restrições e ao mesmo tempo atender objetivos conflitantes: minimizar o tempo de travessamento da ordem, minimizar o atraso na entrega do produto ao cliente, minimizar o número de ordens atrasadas, minimizar o tempo ou o custo de trocas de produção, maximizar a utilização do equipamento, maximizar a utilização da mão de obra, entre outros.

Às vezes estas ordens de produção podem formar filas em frente às máquinas aguardando o momento de serem processadas.

2.3.2. Teoria das filas

Quando os recursos são limitados, eles devem ser compartilhados. As filas são normalmente formadas quando se quer fazer uso de um determinado recurso e este é insuficiente, ou está indisponível, seja porque está ocupado, quebrado. Pode-se citar como

exemplo o número de recursos ou servidores (máquinas, atendentes, caixas de banco, posto de pedágio, etc.) são insuficientes para atenderem os clientes (ordens de produção, clientes do banco, carros, etc.), conforme esquematizado na Figura 2.5. Deve-se notar que a fila inicia-se no instante i em que o cliente entra no sistema e termina no instante f quando o cliente é atendido pelo servidor.

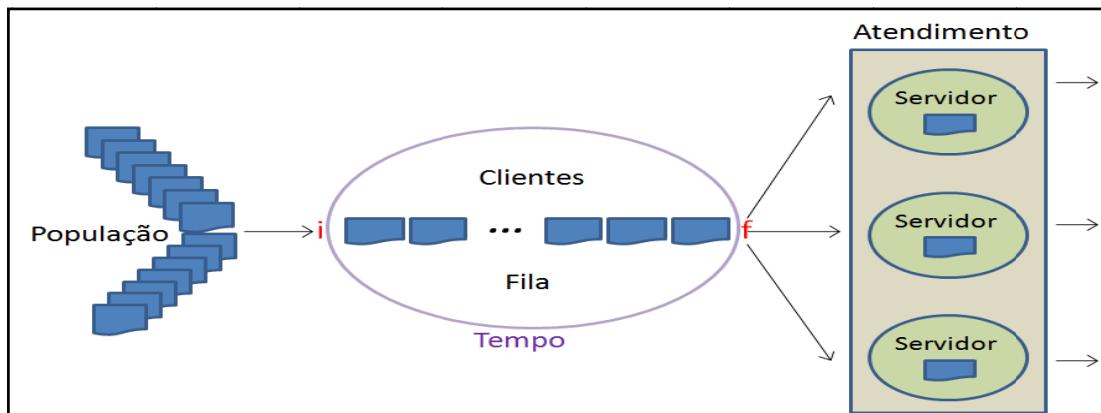


FIGURA 2.5: Representação da formação de fila

Nas empresas de manufatura a geração de filas implica na formação de *WIP*, portanto definir quais regras de sequenciamento será utilizado, permitirá que o sistema faça a opção do melhor caminho para atender a produção.

Para realizar uma boa programação é necessário procurar equilibrar os principais objetivos da programação, inclusive os recursos produtivos. Então, nesta situação, uma regra de prioridade com um bom desempenho pode ser bem proveitosa, mas será inútil caso a capacidade de produção seja insuficiente.

2.3.3. Regras heurísticas de sequenciamento da produção

Entende-se por regras de sequenciamento de produção, que também podem ser chamadas de agendamento da produção e de regras de expedição, o ato de lançamento de

ordens, peça ou tarefas no sistema. Estas denominações são equivalentes e cada pesquisador escolhe aquela que melhor lhe convém.

As regras de sequenciamento são heurísticas usadas para selecionar a partir de informações sobre as ordens de produção e/ou sobre o sistema produtivo, qual das ordens esperando na fila de um grupo de recursos terá prioridade de processamento. Normalmente, as principais informações são: Tempo de processamento, data de entrega e quanto custa a realização desta ordem. O seqüenciamento da produção tem sido objeto de estudo de diversos pesquisadores, mas os problemas ainda permanecem.

Como descrito por Chan e Chan (2004, p.429) “uma regra de expedição é usada para selecionar a próxima peça a ser transformada a partir de um conjunto de peças à espera de serviço, de acordo com algumas prioridades”.

As principais regras de expedição adaptadas de Mesquita *et al.* in Lustosa *et al.* (2008), Chan e Chan, (2004) e de Suresh e Sridharan (2007), podem ser definidas como sendo:

- FIFO – (*First in, first out*) A prioridade é dada a primeira peça que entra, deve ser a primeira a sair. Pode ser considerada como uma ordem de chegada na máquina, na fábrica. Esta regra é procura minimizar o tempo de permanência na máquina ou na fábrica.
- LIFO – (*Last in, first out*) A prioridade é dada pela ultima peça que entra deve ser a primeira a sair. Por ser adversa e negativa no que tange a confiabilidade e rapidez de entrega, e não ter uma sequência baseada em qualidade, flexibilidade ou custo esta regra é pouco, utilizada.
- SPT – (*Shortest processing time*) A prioridade é dada pelo menor tempo de processamento total. É classificada em ordem crescente de tempo. Sua utilização visa reduzir o tamanho das filas e o aumento do fluxo.

- LPT – (*Longest Processing Time*) A prioridade é dada pelo maior tempo de processamento total. Contrário da regra SPT. Sua utilização visa a redução de troca de máquinas.
- EDD – (*Earliest Due Date*) – A prioridade é dada pela execução das ordens mais urgentes em termos de prazo de entrega. A finalidade é reduzir atrasos.
- LS – (*Least Slack*) A prioridade é dada pelo menor folga entre a data de entrega e o tempo total de processamento entre as tarefas que estão à espera. Classificada por prazo de entrega e visa reduzir atrasos.
- SIPT – (*Shortest Imminent Processing time*) A prioridade é dada pelo menor tempo de processamento individual. Semelhante a SPT
- LIPT – (*Longest Imminent Processing Time*) A prioridade é dada pelo maior tempo de processamento individual. Semelhante a LPT.

Importante ressaltar que normalmente as empresas aplicam uma combinação destas regras a fim de conseguirem um melhor desempenho, uma vez que a utilização individual nem sempre apresenta a melhor solução.

Além das regras de sequenciamento acima, podem ser aplicadas também outras regras heurísticas tais como a regra de Johnson que minimiza o tempo total de processamento, ou seja, fornece um tempo mínimo de processamento para o sequenciamento de n ordens, desde que estes tempos de processamento sejam conhecidos e constantes, tenham rotas de produção independentes e que sigam na mesma direção, não possuam prioridades e as ordens são enviadas para o próximo estágio apenas quando completadas. Embora cada uma dessas regras possua suas vantagens e desvantagens, para os problemas de *job shop* é quase impossível determinar de antemão qual é a melhor regra. Portanto, apenas a simulação de cada regra poderia indicar a melhor opção para um dado sistema.

A Figura 2.6 mostra uma evolução das abordagens utilizadas na solução de problemas

de agendamento/seqüenciamento e programação de produção a partir do ano de 1920. Antes desta data, no final do século XIX, início do século XX, a produção industrial teve dois momentos marcados por importantes mudanças, a saber: em 1896 tem inicio a administração científica idealizada por Taylor e em 1908 a produção em massa desenvolvida por Ford.

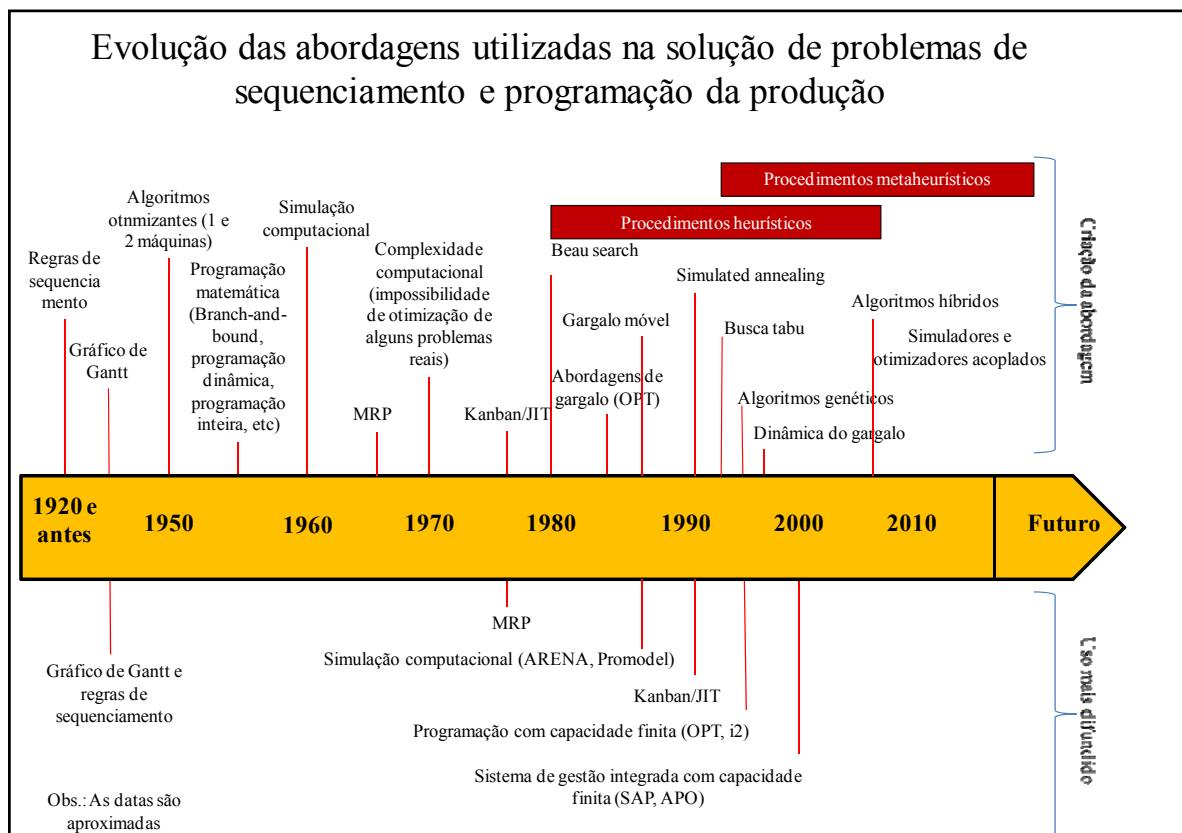


FIGURA 2.6: Evolução das abordagens utilizadas na solução de problemas de seqüenciamento da programação
Fonte: Adaptado de www.veraxc.com

Ao longo do tempo, a evolução das abordagens utilizadas na solução de problemas de sequenciamento e programação da produção são equivalentes a evolução dos *softwares* de simulação apresentados anteriormente na Figura 2.1, na seção 2.1.

2.3.4 Medidas de Desempenho

Medidas de desempenho são indicadores utilizados para medir o comportamento do Sistema em determinadas atividades com a finalidade de agir sobre os desvios em relação aos

aos objetivos propostos.

Esta medição somente poderá ser feita caso seja compreendida e aceita por todos, permita a reprodução e possibilite a tomada de ação.

Para gerenciar o comportamento do sistema de produção em qualquer empresa quer seja de manufatura quer seja de serviços, existe uma série de indicadores de desempenho, como por exemplo: número de ordens atrasadas, tempo total de produção, pontualidade, tempo total de atraso, *WIP*. Os principais objetivos desta medição estão ligados basicamente a três classes: cumprimentos de prazos, velocidade de fluxo e utilização (FREITAS E MESQUITA In LUSTOSA *et al.*, 2008) sendo que a primeira esta relacionando ao atendimento ao cliente o que implica na sobrevivência e lucratividade da empresa.

2.3.4.1 Indicadores de Desempenho

Cada objetivo traçado pela empresa está atrelado a uma medida de desempenho que por sua vez esta vinculada a um ou mais indicadores de desempenho. Por exemplo, a velocidade do fluxo de produção está relacionada ao tempo de travessamento das ordens pela fábrica.

O significado de algumas variáveis e parâmetros relacionados às ordens de produção se faz necessário em função da apresentação dos cálculos dos indicadores de desempenho, conforme definição de Freitas e Mesquita In Lustosa *et al.*(2008).

- Data de disponibilidade: liberação da ordem para entrada no processo de produção.
- Data prometida ou data de entrega: data de conclusão desejada para a ordem.
- Tempo de processo: tempo que a ordem esta em processo nas máquinas. Neste tempo é incluído o tempo de preparo.
- Data de término: data de conclusão efetiva da ordem.

- Tempo de fluxo da ordem: Data de término (-) data de liberação.
- Pontualidade: Data de término (-) data prometida (=) Zero. Se menor que Zero tem se um adiantamento da ordem (*earliness*); se maior que Zero tem se um atraso da ordem (*tardiness*).

Com a finalidade de medir a eficácia de uma decisão, utilizam-se indicadores de desempenho, os mais comuns são:

- Tempo médio de fluxo ou de atravessamento: Média entre os tempos de fluxos das ordens.
- Atraso Médio ou atraso máximo: Média dos atrasos ou maior atraso entre as ordens consideradas.
- Tempo total de trabalho: Intervalo de tempo entre a liberação da primeira operação, da primeira ordem e a conclusão da ultimo processo da última ordem.
- Média do estoque em processo: Média da quantidade de ordens abertas e ainda não finalizadas.
- Número de ordens atrasadas: quantidade de ordens que deixaram de ser entregues ao cliente no prazo estipulado:
- Tempo total de atraso: é o somatório de todos os tempos de todas as ordens que ficaram atrasadas.
- Utilização: Diferença entre o tempo efetivamente utilizado e o tempo total disponível do recurso. Se o tempo utilizado for menor que o disponível, então o recurso pode estar ocioso.

Note-se que, conforme mencionado anteriormente, estes indicadores estão atrelados aos objetivos: cumprimento de prazo, velocidade de fluxo e utilização da capacidade.

2.4. TÉCNICAS METAHEURÍSTICAS DE OTIMIZAÇÃO

Otimização implica, basicamente, em encontrar soluções de boa qualidade para um determinado problema, através da utilização de técnicas computacionais. As técnicas metaheurística são utilizadas para a solução de problemas de otimização combinatória chamados de NP - Difíceis (não polinomiais).

Estes problemas envolvem a combinação de vários fatores e buscam o valor máximo ou mínimo de uma função, cujas variáveis devem obedecer a um conjunto de restrições, em um determinado domínio (BLUM e ROLI, 2008). Portanto são problemas que não podem ser resolvidos por técnicas tradicionais como: Programação Linear, Programação Dinâmica, Programação Não-Linear, principalmente porque a verificação de todas as soluções possíveis é muito lenta e dispendiosa. Observa-se, porém, que muitos deles podem ser modelados e solucionados.

Conforme Weise (2008), um ótimo local é a melhor solução encontrada num subespaço de busca onde se localizam um ponto de mínimo ou de máximo, enquanto que um ótimo global é a melhor solução encontrada em todo o espaço de busca.

Cada ponto do espaço de busca é uma solução candidata para a resolução do problema.

A Figura 2.7 apresenta um espaço de busca de todas as soluções candidatas para um determinado problema enquanto identifica a localização de pontos máximo e mínimo global e de pontos máximo e mínimo locais.

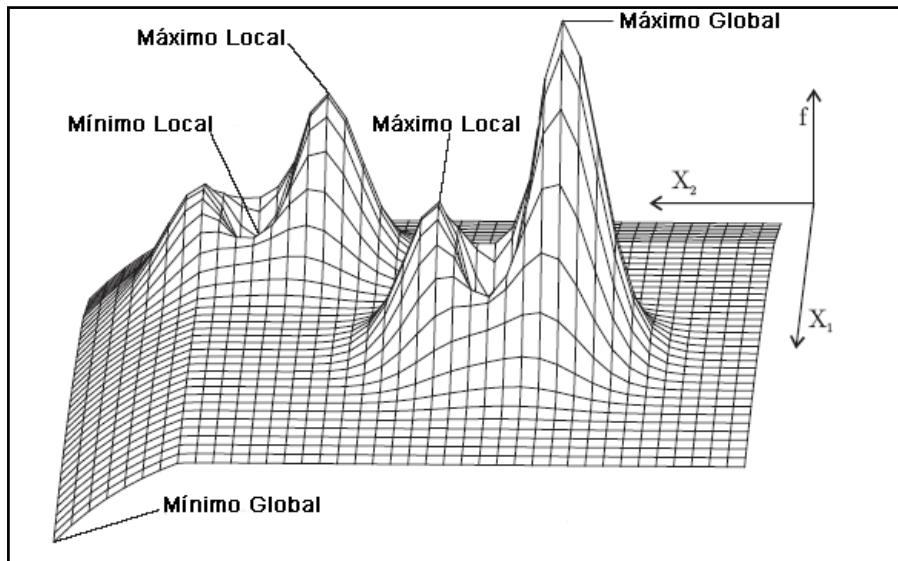


FIGURA 2.7: Ótimo global e ótimo local de uma função bidimensional
Fonte: Adaptado de Weise (2008)

As pesquisas nestes espaços podem ser *exaustivas*: quando verifica o valor de cada ponto do espaço; *aleatórias* quando verificam alguns pontos do espaço e guarda o melhor valor encontrado e *metaheurística* quando utiliza a escolha aleatória aliada ao conhecimento de resultados anteriores para serem guiados neste espaço.

Normalmente as técnicas metaheurística possuem procedimentos específicos, com capacidade de investigar o espaço de busca e ainda evitar que ocorra uma convergência prematura para o primeiro ótimo local encontrado, além de impedir que sejam desconsideradas outras soluções que poderiam ser melhores (CHAVES, *et al.*, 2007).

Dentre as diversas técnicas existentes para a otimização, optou-se pelo AG que será abordado no próximo item.

2.4.1 Algoritmos Genéticos (AGs)

Antes de iniciar a apresentação das funcionalidades dos AGs, faz-se necessário definir ou relembrar algumas das nomenclaturas utilizadas e que estão reunidas na Tabela 2.2., onde também é possível identificar a analogia com a Biologia.

TABELA 2.2: Definição dos principais termos utilizados nas técnicas de AGs

Nomenclatura	Definição
Cromossomo	Indivíduo que representa uma das possíveis soluções no espaço de busca. Normalmente um vetor formado por um conjunto de bits que pode ter uma representação binária ou real dentre outras. Cada bit representa uma característica do gene.
População	Conjunto de cromossomos ou indivíduos.
Fenótipo	Representa o cromossomo decodificado que é o conjunto de parâmetros do problema.
Genótipo	Representa a estrutura do cromossomo codificado.
<i>Fitness</i> ou Aptidão	Função de avaliação para classificação das soluções em termos de aptidão.
Indivíduo	É um representante da população que significa uma possível solução.
Gene	Características do indivíduo e é indissociável. Controla a herança genética e possui um local fixo no cromossomo.
Alelo	É o valor de cada característica existente no gene.
<i>Locus</i>	Posição fixa do gene no cromossomo.
Operadores genéticos	Técnicas que recombinam ou transformam os indivíduos da população atual.
Pai	Indivíduos da população aptos a reproduzir-se, representam as melhores soluções para os problemas
Filho	Indivíduo gerado através do cruzamento
Seleção	Operador genético que seleciona os indivíduos para a próxima população.
<i>Crossover</i> ou Cruzamento	Operador genético que recombina os indivíduos selecionados.
Mutação	Operador genético que modifica aleatoriamente os indivíduos já cruzados, usado para preservar a diversidade genética.

Fonte: produzida pela autora

Os AGs fazem parte do conjunto de métodos computacionais inspirados na evolução das espécies, que comumente são chamados de Algoritmos Evolucionários. Estes algoritmos buscam a evolução de uma população de possíveis soluções de um problema com o uso de processos conceituais semelhantes ao da natureza, tais como cruzamento, seleção, recombinação e mutação.

Os AGs tem por base o princípio da sobrevivência “do mais forte” ou melhor adaptado

conforme definição darwinista dos fenômenos naturais sobre a origem das espécies e da herança genética na luta pela sobrevivência (MITCHEL, 1997; MICHALEWICZ, 1996; SANTA CATARINA, 2009). Ressalta-se que o AG é um método estocástico, não determinístico de busca e otimização.

Alguns algoritmos iniciam a otimização a partir de uma única solução, enquanto que o AG já trabalha com uma população de soluções, onde cada solução é um vetor hiperespacial que representa uma potencial solução para determinado problema de otimização (BASIRI *et al.*, 2009). Isto significa que o AG evolui através de uma população cujo tamanho determina o número de soluções do espaço de busca que são considerados.

Cada vetor que representa uma solução no espaço de busca é considerado como um cromossomo na população. O número de genes em cada cromossomo depende do número de parâmetros do problema e representam a maneira de otimizá-lo, e o valor atribuído a cada gene é chamado de alelo (BASIRI *et al.*, 2009). Sua estrutura pode ser observada na Figura 2.8, onde esta representação gráfica traz um cromossomo composto por 4 genes para um problema de otimização com 4 dimensões. Cada gene representa uma variável do problema e é indissociável. Cada alelo é uma característica específica de cada um dos genes.

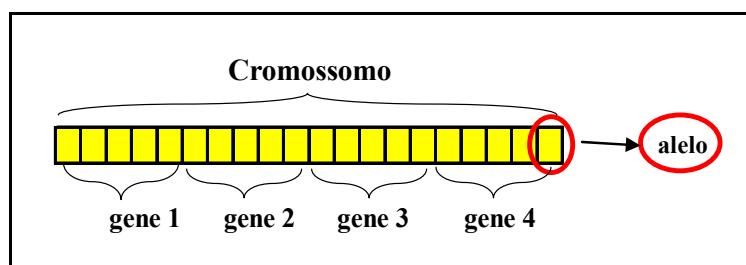


FIGURA 2.8: Estrutura de um Cromossomo

Observa-se que similarmente à natureza o processo de transmissão e recombinação de material genético ocorre no nível das informações da estrutura do cromossomo (genótipo) usadas para formar o conjunto de parâmetros (fenótipo).

Durante o processo de evolução, a população é avaliada e seus indivíduos estão sujeitos aos processos de seleção e reprodução. No AG, esses processos são chamados de operadores e são eles: seleção, combinação ou cruzamento (*crossover*) e mutação.

O procedimento para a criação da população é iniciado, na maioria das vezes, por meio da geração de números aleatórios. Se for possível inicializá-la com cromossomo já conhecido a evolução poderá ser mais rápida, caso contrário o melhoramento da população é feito através de sucessivas execuções promovendo uma combinação entre a população atual e as melhores soluções encontradas no processamento anterior (PACHECO, 1999).

Os parâmetros mais importantes a serem definidos no AG são o tamanho da população e o número de gerações. A escolha destes parâmetros é definida de acordo com a complexidade do problema, deve ser observado que estes são os valores que determinam o espaço de busca. A determinação de um conjunto de valores otimizado para estes parâmetros dependerá da realização de um grande número de experimentos e testes até que se atinjam valores considerados adequados para a solução daquele determinado problema.

2.4.1.1. Codificação, funcionamento e nomenclatura

A codificação do cromossomo pode ser feita de forma binária, real ou com alguma outra codificação específica.

- Forma Binária: Codificação clássica utilizada por Holland (1992) que continua sendo utilizada até hoje. Nela, os elementos do cromossomo pertencem ao conjunto $\{0,1\}$, ou seja, no menor conjunto possível de elementos que podem ser codificados, por conseguinte os genes terão um comprimento maior. A principal vantagem desta codificação é a facilidade em transformá-la em número inteiro ou real e a desvantagem é o tamanho, por exemplo, na precisão decimal, pois quanto maior o cromossomo,

maior é o tempo necessário para calcular o equivalente decimal deste cromossomo (PACHECO, 1999).

- Forma Real ou ponto flutuante: É uma maneira direta de representar variáveis numéricas contínuas através do seu próprio valor. Os cromossomos possuem um comprimento menor assim necessitam de menos memória no computador além de aumentar a velocidade de processamento no cálculo aritmético (MICHALEWICZ, 1996).

Em resumo, os processos evolutivos dos AGs funcionam de acordo com as seguintes etapas: seleção, operadores genéticos, população, função de aptidão, critérios de parada, cujas definições podem ser vistas na Tabela 2.2 da seção 2.4.1. Estas etapas podem ser vistas de forma sucinta no modelo gráfico apresentado na Figura 2.9. As etapas de funcionamento são descritas considerando basicamente os autores Gen e Chang (2000); Michalewicz (1996); Mitchell (1997); Santa Catarina (2009) e Sean (2010).

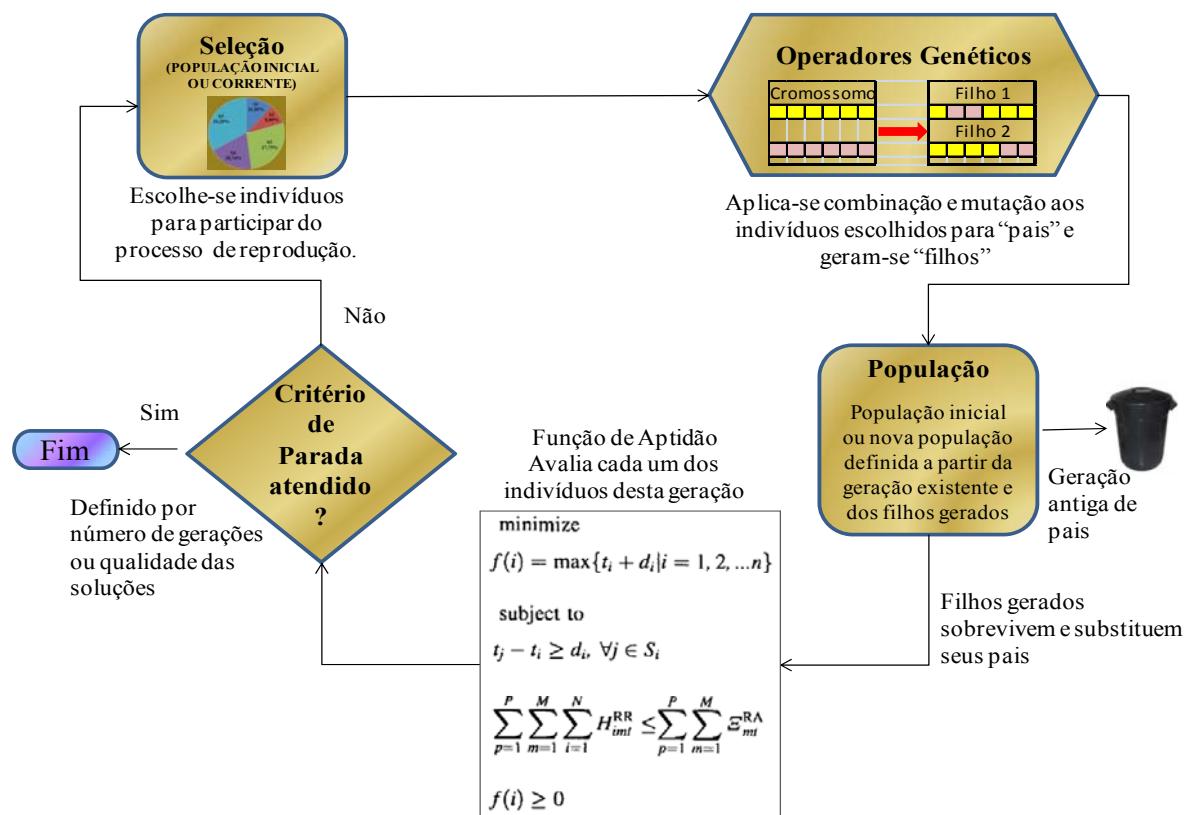


FIGURA 2.9: Representação Gráfica do AG
Fonte: Adaptado de Linden (2006) e Kim (2009)

2.4.1.2 Seleção

Seleção é um processo em que os indivíduos são selecionados com uma probabilidade proporcional à sua aptidão para ser pais, ou seja: ela é composta por indivíduos que representam possíveis soluções para o problema.

A população inicial pode ser gerada de modo aleatório, ou formada por soluções que tenham sido encontradas por outros métodos, tais como informações reais em históricos.

A população corrente ou atual é aquela gerada através da aplicação dos operadores genéticos.

O processo de seleção em AG escolhe os indivíduos para a reprodução. Esta seleção baseia-se na aptidão, ou seja, existe maior probabilidade de escolha dos indivíduos mais aptos.

Existem diversos métodos que podem ser utilizados para a seleção entre eles pode-se citar o elitismo, que consiste em preservar para a geração seguinte os melhores cromossomos da geração atual evitando que eles sejam destruídos pelos outros métodos. O elitismo melhora significativamente o desempenho do AG (MITCHELL, 1999).

O Torneio (*tournament selection*) é um método de escolha apoiado na competição entre os indivíduos e aquele que tiver maior aptidão sobrevive para a próxima geração.

A Roleta (*Roulette Wheel*), conforme Santa Catarina (2009) é o mecanismo de seleção mais utilizado, donde a probabilidade de seleção dos indivíduos é diretamente proporcional ao seu valor da função de *fitness*. Desta forma, os mais aptos possuem uma parcela maior na roleta enquanto que os mais fracos possuem uma parcela menor na roleta. Por fim, roda-se a roleta um determinado número de vezes de acordo com a população e seguem para a próxima geração os indivíduos que foram sorteados.

Com a finalidade de ilustrar um destes métodos optou-se pela roleta que pode ser vista

na Figura 2.10. Essa figura possui uma tabela que apresenta a seleção de 5 indivíduos S_1, S_2, \dots, S_5 com seu respectivo valor de aptidão individual. Calcula-se a o valor de aptidão relativo e inclui esses valores de forma proporcional numa roleta. Observe que o S_5 ocupa uma área maior na roleta, visto que ele possui o maior valor relativo, enquanto que o S_2 ocupa a menor área, pois possui o menor valor relativo. Então, a roleta será girada algumas vezes e espera-se que pare mais vezes na área maior, elegendo desta forma o individuo que irá compor a próxima população

Seleção (Si)	Indivíduo (Si)	Aptidão f(Si)	Aptidão Relativa
S_1	100001	2,46	11,36%
S_2	100011	1,92	8,86%
S_3	001100	6,01	27,75%
S_4	101010	4,06	18,74%
S_5	001001	7,21	33,29%

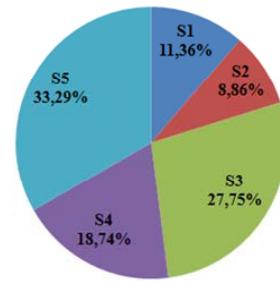


FIGURA 2.10: Seleção pelo método da Roleta
Fonte: Adaptado de Santa Catarina, 2009

Sean (2010) aconselha cuidado na escolha uma vez que, o tipo de seleção pode causar uma convergência prematura para ótimos locais distantes de um ótimo global, ou o contrário convergir muito lentamente. Entende-se por convergência prematura quando a população é rapidamente dominada por indivíduos que possuem uma elevada aptidão, porém sem corresponder a solução ótima, e que acaba levando a um ótimo local de baixa qualidade (MITCHELL, 1999). Por outro lado, a convergência lenta ou até mesmo a não convergência é causada pela dificuldade em alcançar ótimos locais ou ótimo global de boa qualidade, devido à insuficiência de diversificação na população, que pode impedir o AG de continuar sua evolução rumo á solução ótima (MITCHELL, 1999).

2.4.1.3. Operadores Genéticos

Conforme Holland (1992), os operadores genéticos tornam possível reproduzir filhos diferentes dos pais proporcionando as modificações necessárias nos genótipos de tal forma que os fenótipos mantenham muitas das características herdadas dos pais, mas não sejam idênticos. Portanto sua função básica é converter a população através de sucessivas gerações, até encontrar um resultado satisfatório. Os operadores genéticos são imprescindíveis para que a população se diversifique e conserve características de adaptação obtidas pelas gerações anteriores.

Aplicam-se os operadores de cruzamento, de mutação nesta população de tal forma que a geração antiga dos indivíduos, chamada de pais é eliminada e a nova geração de indivíduos, chamada de filhos passa por uma avaliação individual.

Foram desenvolvidos vários operadores de cruzamento e de mutação, sendo que alguns deles são apropriados a um tipo particular de codificação e outros com propósitos mais genéricos. Aqui serão citados apenas os mais utilizados.

- a) Mutação: processo pelo qual o gene sofre alterações em sua estrutura através da realização de trocas aleatórias de alguns valores dos indivíduos. É utilizada para manter a diversidade genética da população, ou seja: ela possibilita o aparecimento de novas características, evitando que a busca fique apenas numa região de um ótimo local.

Na Figura 2.11, pode ser visto um exemplo de aplicação do operador de mutação em dois pontos aleatórios, os números 1 das terceira e sétima posições são alterados para 0.

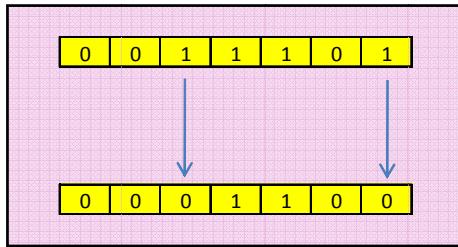


FIGURA 2.11: Exemplo de aplicação do operador de mutação

Segundo Basiri *et al.* (2009), mutação é a alteração dos *bits* (do inglês, simplificação do dígito binário - *Binary Digit*) de um indivíduo com uma pequena probabilidade pré-definida, às vezes conhecida como coeficiente de mutação que é utilizada com o propósito de garantir que uma determinada posição não fique estagnada e que a probabilidade de se chegar a qualquer ponto do espaço de busca nunca seja igual a zero.

Vale ressaltar que se este coeficiente de mutação for muito alto, a busca pode se tornar aleatória, além de possibilitar que uma boa solução seja aniquilada.

Estes novos indivíduos alterados compõem a nova população

b) Cruzamento: processo no qual dois pais diferentes são iterativamente selecionados a partir do conjunto de p pais para trocar informações entre si com a finalidade de gerarem dois novos indivíduos (filhos). Isto é feito escolhendo aleatoriamente um ponto de ruptura para os pais e trocando-se posições entre eles. Então uma mutação é aplicada a todos os descendentes.

Seu objetivo é tentar obter melhores indivíduos á partir daqueles já selecionados, proporcionando um conhecimento prévio do espaço de busca já percorrido. Da mesma forma que o operador de mutação, o cruzamento esta atrelado a um taxa de aplicação.

A Figura 2.12 apresenta um exemplo deste operador, no caso, o cruzamento é executado em um ponto aleatório da cadeia de genes. O filho 1 é formado pelas partes do primeiro pai que estão fora do ponto de corte e pelas partes dentro do ponto de corte do segundo pai, enquanto que o filho 2 tem sua formação de modo inverso.

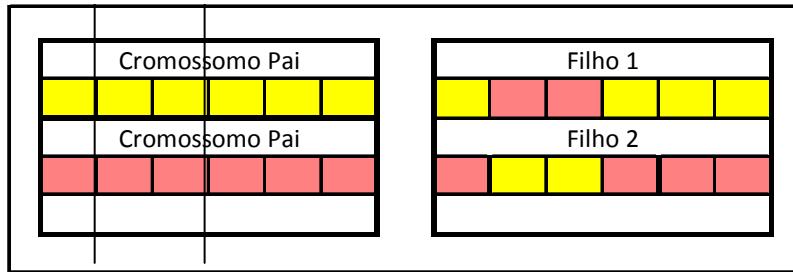


FIGURA 2.12: Exemplo de aplicação do operador de cruzamento em um ponto

Outros métodos de cruzamento como os multipontos, dividem o cromossomo em vários pontos e os recombina para formar os filhos, assemelha-se mais ao processo que acontece na vida real. Sua principal vantagem é garantir uma maior variedade genética. Estes operadores são inadequados para AGs com codificação real, uma vez que apenas trocam os valores dos genes.

Deste modo, recomenda-se neste caso a utilização dos operadores de cruzamento aritmético, a saber: média aritmética, média geométrica e heurística. Os cruzamentos: média aritmética e média geométrica geram um novo cromossomo através da média aritmética ou geométrica dos respectivos pais. Para o cruzamento heurístico utiliza-se uma interpolação linear dos pais para gerar os filhos.

Como visto, uma grande variedade de alternativas podem ser utilizadas na implementação dos AGs, porém há uma discussão sobre como configurar estes parâmetros, uma vez que eles não interagem de forma linear, desta forma, não podem ser otimizados individualmente (MITCHELL, 1999), então uma possibilidade é ajustá-los empiricamente ou baseá-los em trabalhos semelhantes.

Conforme Santa Catarina e Bach (2003), mesmo uma opção inadequada na implementação do AG não o impedirá de convergir para uma solução otimizada, apenas poderá refletir no tempo para a sua convergência.

2.4.1.4 Função de Aptidão

A fim de evitar confusão é necessário reforçar a definição dos termos função objetivo e função de aptidão. A função objetivo fornece uma medida de qualidade em relação a um determinado problema em particular e esta limitado a um indivíduo, isto é relaciona o indivíduo a solução do problema. Por outro lado a função de aptidão ou de *fitness* é o processo de avaliar cada indivíduo da população de acordo com a função objetivo com vistas a selecioná-los conforme sua habilidade em sobreviver e reproduzir-se, fundamental para o funcionamento dos AGs.

A análise do desempenho da população com a finalidade de verificar se o objetivo foi atingido (isto é: se houve convergência) pode ser feito através de vários fatores, entre outros, tem-se os valores máximo, mínimo e médio da função de aptidão.

Em algumas situações o valor da aptidão pode ser fornecido pela própria função objetivo. Porém, deve-se considerar que isto nem sempre é possível, pois caso haja muita proximidade entre os valores oferecidos nesta função o processo evolutivo pode se tornar aleatório, dependendo do critério de seleção adotado.

O AG é regulado por sua população, então se houver na população inicial um super indivíduo, que represente a solução do problema, ainda assim o AG continuará seu processo de busca até atender o critério de parada (item 2.4.1.6), quando isto ocorre, costuma-se memorizar o indivíduo mais apto, mesmo que ele não faça mais parte da população atual. Desta forma fica garantido que, ao final, este será o resultado esperado.

2.4.1.5. Critérios de Parada

Os AGs necessitam possuir um critério de parada pré-definido, uma vez que ele não

tem meios de parar espontaneamente, nem mesmo quando encontra a solução ótima. São quatro os principais critérios de parada:

- O critério de parada mais comum e simples é definir um número fixo de gerações, neste caso, corre-se o risco da melhor solução ainda não ter sido encontrada.
- Outra possibilidade é interromper o processamento quando se encontra uma solução que possa ser considerada boa, e então é necessário conhecer pelo menos uma solução nesta condição para que se possa fazer a comparação.
- Um terceiro critério pode ser a convergência da população quando se decide por interromper o processo quando, por exemplo, 90% dos indivíduos da população apresentam 90% de suas posições com o mesmo valor.
- Outro critério pode ser um tempo fixo para execução, uma vez que um tempo de processamento muito longo pode ser inviável em termos de custo.

2.4.2. Otimização multiobjetivo

De acordo com Gen e Chang (2000), desde 1960, há um interesse crescente de pesquisadores por problemas de otimização multiobjetivo considerando diversos ambientes. Os problemas de otimização multiobjetivo são aqueles que possuem várias funções-objetivo que precisam ser otimizadas simultaneamente e que na maioria das vezes podem ser conflitantes como, por exemplo, reduzir o número de ordens e reduzir o número de trocas.

Na prática, um dos métodos mais simples de otimização multiobjetivo consiste em reduzir o problema a um único objetivo por meio da combinação de diferentes objetivos em uma soma ponderada. Um dos métodos utilizados com esta finalidade é baseado em pesos ou método de pesos, onde são aplicados pesos a cada uma das diferentes funções objetivos (ARENALES, 2007).

No caso de vários objetivos, não necessariamente existe uma solução que é a melhor em relação a todos os objetivos devido à dificuldade de avaliar e medir o conflito entre os objetivos. Uma solução pode ser a melhor em um objetivo, mas a pior para outro. Portanto, normalmente existe um conjunto de soluções que leva a um bom resultado desde que não se compare os objetivos entre si, ou seja, não é possível melhorar uma função sem prejudicar a outra. Tais soluções são chamadas de soluções não dominadas (*non-dominated*) ou soluções ideais de Pareto.

Conforme Alba *et al.* (2007), cabe ao decisor optar pela solução mais adequada ao seu problema de acordo com estas soluções idéias que também são chamadas de Frente de Pareto.

2.4.2.1. Frentes de Pareto

Numa situação em que é quase impossível determinar apenas uma solução que otimize todas as funções objetivo simultaneamente, busca-se construir um conjunto de diversas soluções que possam, de alguma forma, apresentar os melhores e mais adequados valores para as funções objetivos avaliadas.

Para fazer parte deste conjunto é necessário atender a condição de ótimo de Pareto que é determinada pela impossibilidade de se conseguir um segundo conjunto de variáveis do sistema que tenha a capacidade de melhorar todas as funções objetivos, sem causar o efeito contrário em pelo menos uma delas, que conforme já citado recebem o nome de soluções não dominadas. O conjunto destas soluções não-dominadas formam a *frente de Pareto* para o problema de otimização tratado (COELLO, LAMONT e VAN VELDHUIZEN, 2007).

Conforme Sean (2010), as frentes de Pareto definem as fronteiras dos objetivos. Por exemplo, em uma situação de dois objetivos a demarcação pode ser feita através de uma curva delimitando a fronteira externa. No caso de três objetivos, pode ser criada uma superfície

demarcada. Quando a quantidade de objetivos cresce, é necessário aumentar o número de amostras para que a precisão permaneça.

Pode-se observar a Figura 2.13 que apresenta a frente de Pareto para um problema com dois objetivos em que ambos devem ser minimizados. O espaço entre os dois eixos das funções f_1 e f_2 compreende a localização de todas as soluções para o problema proposto e é chamada de região viável.

A convergência para a frente de Pareto, ocorre, no caso representada pelos pontos na cor cinza, que variam entre os dois extremos de acordo com os pesos dados as funções. Qualquer outro ponto acima dessa curva apresenta uma solução que já está dominada, como por exemplo, o ponto na cor vermelha.

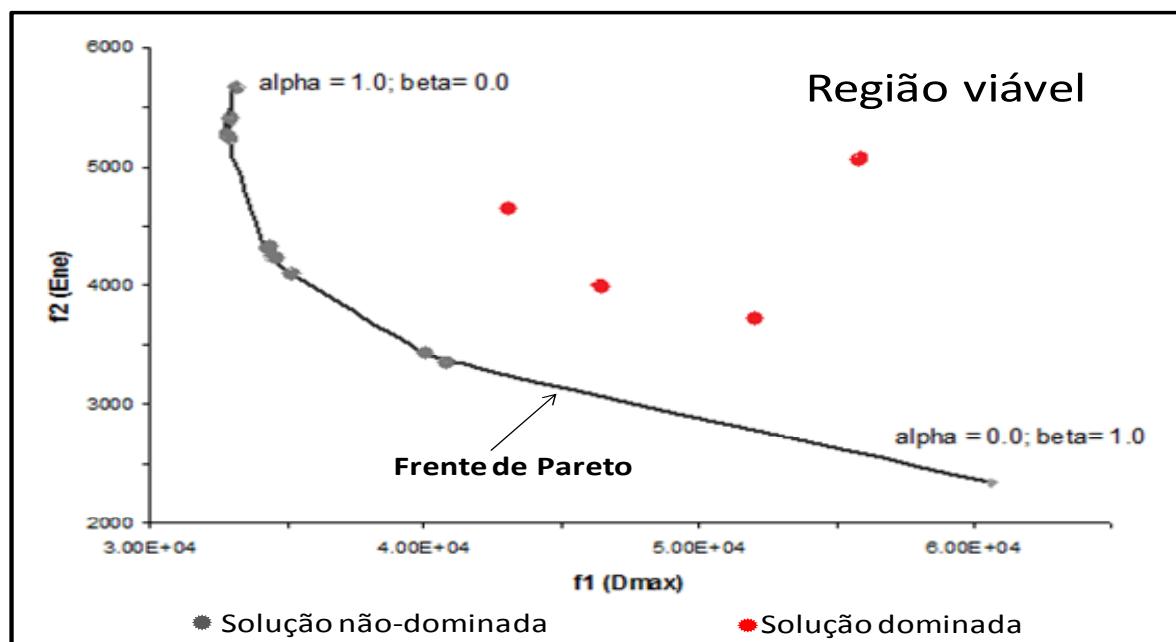


FIGURA 2.13: Frente de Pareto para um problema com dois objetivos
Fonte: adaptada de Pereira, Pires e Nabeta (2011)

3. APLICAÇÃO DO MODELO

Neste capítulo estão descritos os aspectos gerais de como o modelo de simulação e de otimização foram construídos, como foi feita o acoplamento entre o modelo Arena e o AG, e quais testes de validação foram realizados.

Porém, antes de discorrer sobre a parte prática da aplicação do modelo, é conveniente que se faça um breve esclarecimento do que significa substituir a função de aptidão de uma equação matemática analítica por um modelo de simulação no AG.

O AG otimiza os resultados conforme os critérios e definições apresentadas na seção 2.4.1. A partir da população inicial, em geral iniciada de forma aleatória, segue-se o seguinte critério: avalia-se a aptidão dos indivíduos da população e seleciona-se um grupo desses indivíduos para participar das operações genéticas de cruzamento mutação, com vistas a formar uma nova população. Todo o procedimento se reinicia até que o critério de parada tenha sido atingido.

Nessa técnica, a avaliação da aptidão dos indivíduos da população é um ponto fundamental para o funcionamento do método. A função objetivo fornece uma medida de qualidade individual em relação ao problema tratado, enquanto a função de aptidão ou de *fitness* expõe e avalia cada indivíduo da população de acordo com a função objetivo.

Nesta dissertação, todas as informações relativas à função de aptidão, todos os critérios e restrições foram inseridos no modelo de simulação que exercerá o papel de função de aptidão no AG.

A Figura 3.1 apresenta os mesmos passos da Figura 2.9, porém substituiu-se a função de aptidão de uma equação de matemática analítica por um modelo de simulação, segundo abordagem proposta neste trabalho.

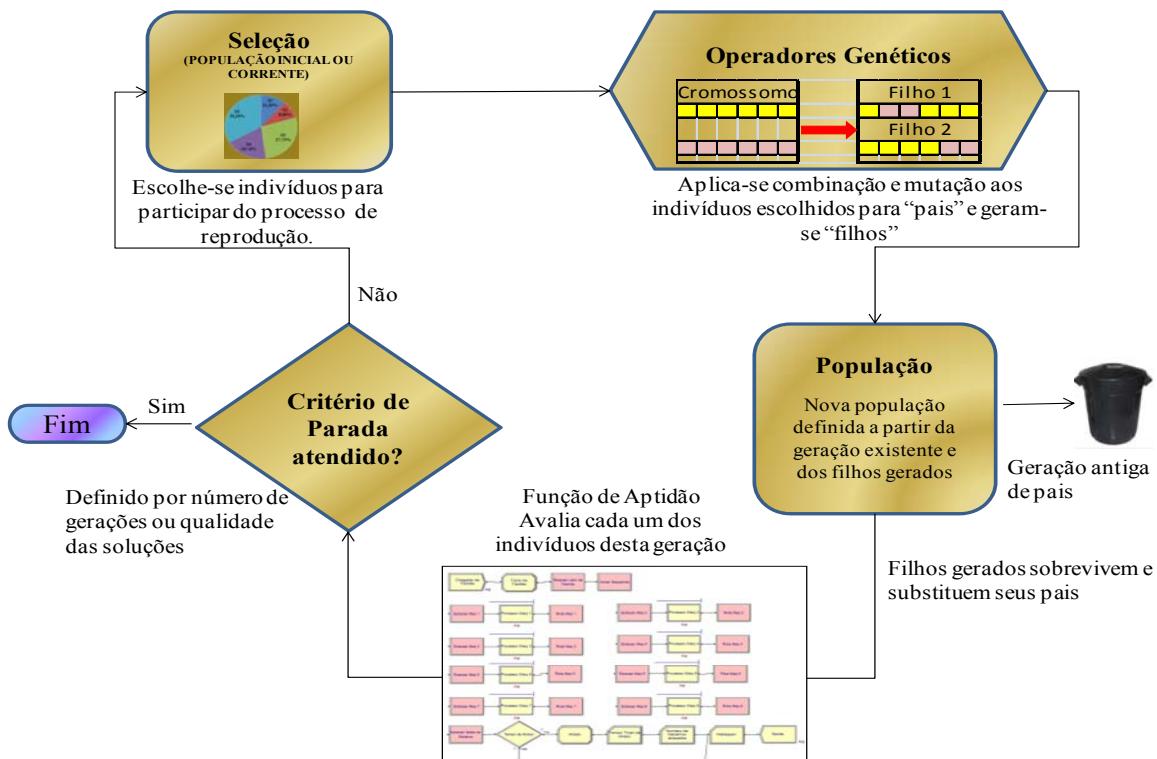


FIGURA 3.1: AG com uma nova função de aptidão
Fonte: Adaptado de LINDEN (2006)

Na seção 3.3 pode ser visto os detalhes sobre o acoplamento da AG com o modelo de simulação Arena.

3.1. DESCRIÇÃO DO MODELO

As condições iniciais para o modelo de simulação proposto, além daquelas descritas nos próximos parágrafos, foram:

- Cada máquina está sempre disponível, ou seja, não foram incluídas indisponibilidades temporárias causadas por quebras, manutenções ou *set ups*, ou por qualquer outro motivo;
- uma vez iniciado o processo de produção de uma ordem em uma máquina ele não poderá ser interrompido;

- c) finalizada a execução de uma tarefa em uma máquina, ela é enviada automaticamente para a próxima sem considerar eventual tempo de transporte;
- d) não há sobreposição de operações;
- e) as ordens são executadas independentemente umas das outras;
- f) apenas com a finalidade de manter-se uma proporcionalidade, definiu-se também que a probabilidade de chegada de cada um dos 10 tipos de ordem é de 10%.

Optou-se por simular o tipo de produção de um ambiente *Job Shop*, pois neste ambiente tem-se um processo que permite que as ordens de produção se movam de um posto de trabalho para qualquer outro posto, de acordo com a sequência de produção pré-determinada.

O modelo para a simulação foi construído com base nos cenários de produção configurados por Santoro e Mesquita (2008), que possui a quantidade de máquinas igual a oito (8), a quantidade máxima de dez (10) operações em cada ordem, sendo que estas operações indicam a rota a ser percorrida por cada uma das ordens.

Utilizou-se também, os mesmos parâmetros para a geração da data de entrega, dada pela distribuição Normal com média $(1+k) * t_0$ e desvio-padrão $0,1 * (1+k)*t_0$, onde t_0 é o tempo total de produção, e k é um fator de segurança em relação a este tempo. Para essa situação, foi utilizado um valor fixo de 30% para o fator de segurança. Como exemplo de formula carregada no Arena[®], temos: Normal (1.30*16.9,0.1*1.30*16.9), onde 16,9 é o tempo total da primeira ordem segundo a rota pré-definida. Todas estas informações podem ser vistas na Tabela 3.1.

Enfatiza-se que o cenário-base utilizado considerava que todas as ordens estavam disponíveis para entrada em produção no sistema no instante *Zero (0)*, porém no modelo para esta dissertação, os intervalos de tempos entre as chegadas das ordens foram aleatórias e, conforme Kelton (2000), a distribuição exponencial pode ser uma boa escolha para

representar esta aleatoriedade. Optou-se, então, por utilizar o intervalo de chegadas com média de 11 minutos, utilizando a Tabela 3 de geração de números aleatórios do Arena[®], ou seja, EXPO (11.0,3.0).

TABELA 3. 1: Rota e tempo total para produção em *job shop*

Tipos de ordem	Rota	Tempo total de produção
1	1, 2, 3, 6	16,9
2	1, 3, 7, 8	16,4
3	1, 5, 8	11,6
4	1, 2, 3, 5, 6, 7, 8	27,3
5	3, 4, 6, 7	16,3
6	1, 2, 4, 3, 1, 2, 4, 5, 7	36,0
7	2, 6, 5, 6, 8	20,7
8	1, 2, 4, 5, 7	19,7
9	2, 3, 4, 6, 7	21,0
10	1, 2, 4, 5, 8	20,6

Fonte: Santoro e Mesquita, 2008, p.81

Inicialmente, as regras utilizadas para o sequenciamento das ordens de produção foram: FIFO, LIFO, já disponíveis no simulador. Estas regras tinham por finalidade validar o modelo, o que será demonstrado oportunamente no item 4.4. Para a simulação do sistema após a validação, consideraram-se as regras de seqüenciamento FIFO, LIFO e Prioridade.

Foram consideradas 20 replicações com 10000 minutos cada uma. O número de replicações foi validado pela análise de dados com uma confiança de 99% para cada uma das rodadas. A quantidade de minutos utilizada é suficiente para contemplar o tempo que a simulação precisa para entrar no período de estabilidade. Diz-se que o sistema está estável quando o processo de produção entra em regime permanente, e eventuais variações causadas pelo *start-up* já foram eliminadas.

3.1.1. Implementação do modelo no Arena®

Nesta seção são apresentadas as principais telas de implementação do modelo no simulador Arena®.

A Figura 3.2, mostra o modelo de simulação desenvolvido, onde pode ser identificado os módulos Processo Maq i os quais representam cada uma das 8 máquinas do ambiente estudado, seguidas pelo módulo que informa a rota de cada ordem após passagem pela maquina. A primeira rota esta identificada no módulo Estação Lanc de Tarefas.

Após a Estação Saída do sistema, estão os módulos relacionados ao desempenho de cada uma das regras de sequenciamento das ordens de produção, bem como os módulos que documentam e armazenam estes resultados.

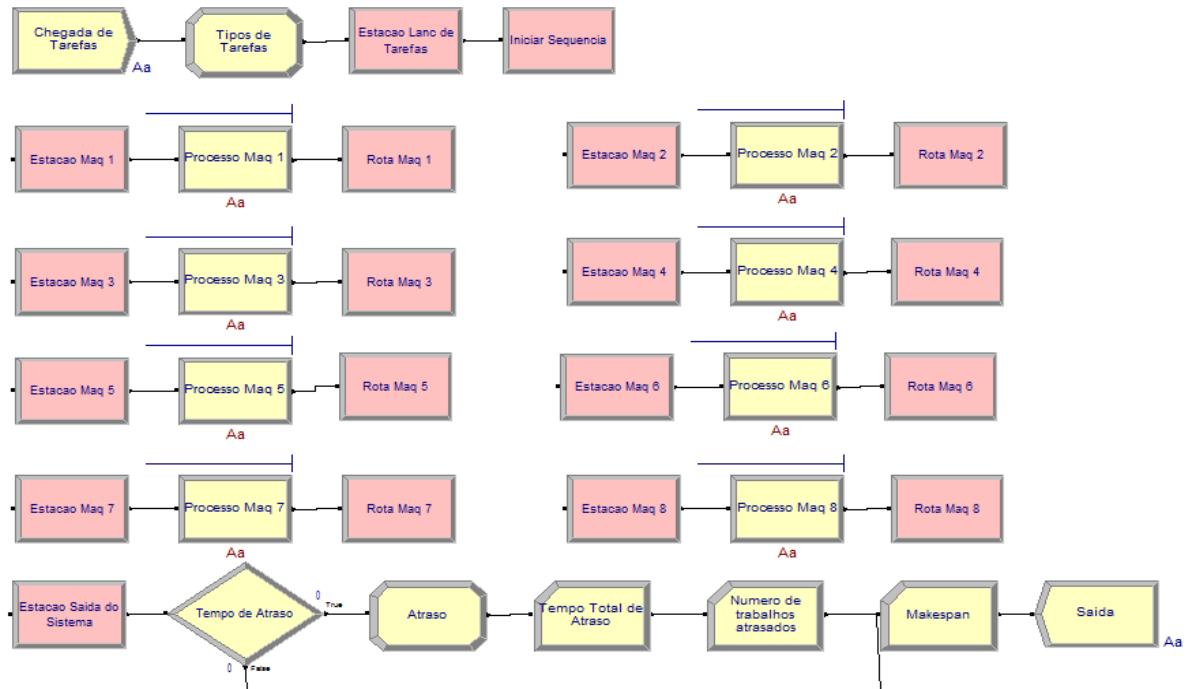


FIGURA 3.2: Modelo de simulação desenvolvido no Arena

Para identificar a chegada de ordens de produção no sistema é necessário inserir as informações no módulo *Create* que, neste caso, recebeu o nome de Chegada de Tarefas. As

informações contidas nesse módulo são relevantes, pois é nele que está determinado qual o comportamento de chegadas das ordens que, conforme já definido na seção 3.1, é descrito neste trabalho por uma distribuição de probabilidades para o intervalo de tempo entre chegadas. Também estão determinadas quantas ordens chega por vez, qual a quantidade máxima de ordens previstas e qual a unidade básica de tempo utilizada. Estas informações estão contidas na Figura 3.3.

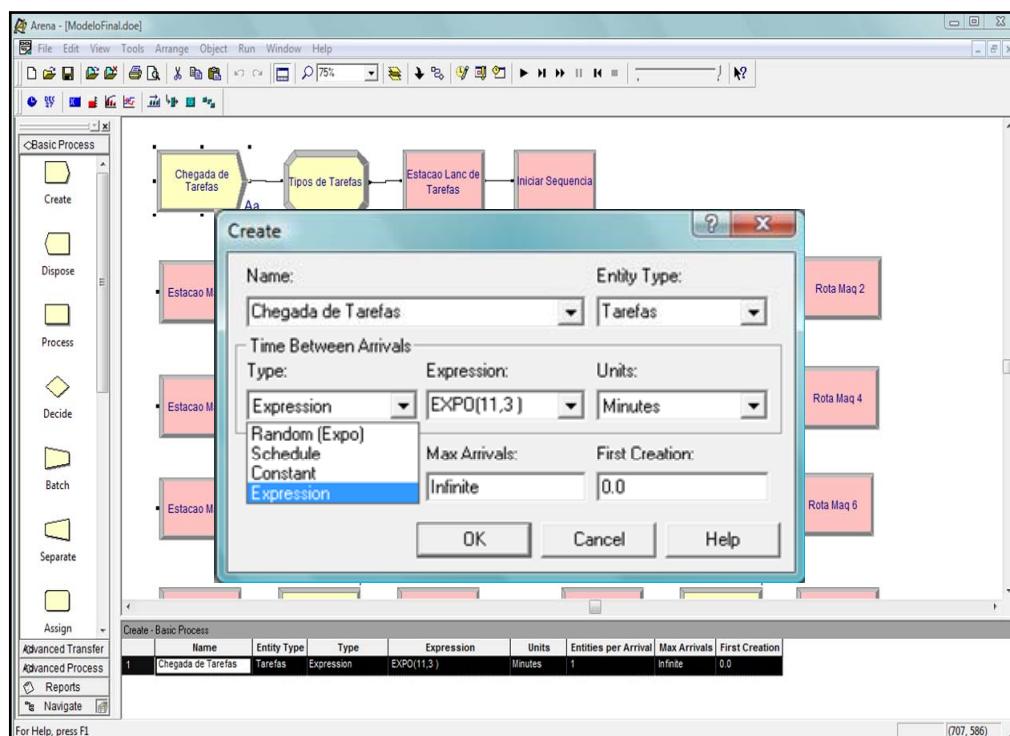


FIGURA 3.3: Detalhe de carregamento do módulo *Create* para a chegada de ordens de produção

Doravante, foram retiradas as imagens de fundo da tela com todos os *templates* uma vez que o modelo completo já foi mostrado nas Figuras 3.3 e 3.4. Serão evidenciadas apenas as informações inseridas nos módulos.

A Figura 3.4 mostra as rotas de cada uma das ordens que foram inseridas no módulo *Route* e completada com suas respectivas sequências no módulo *Sequence*.

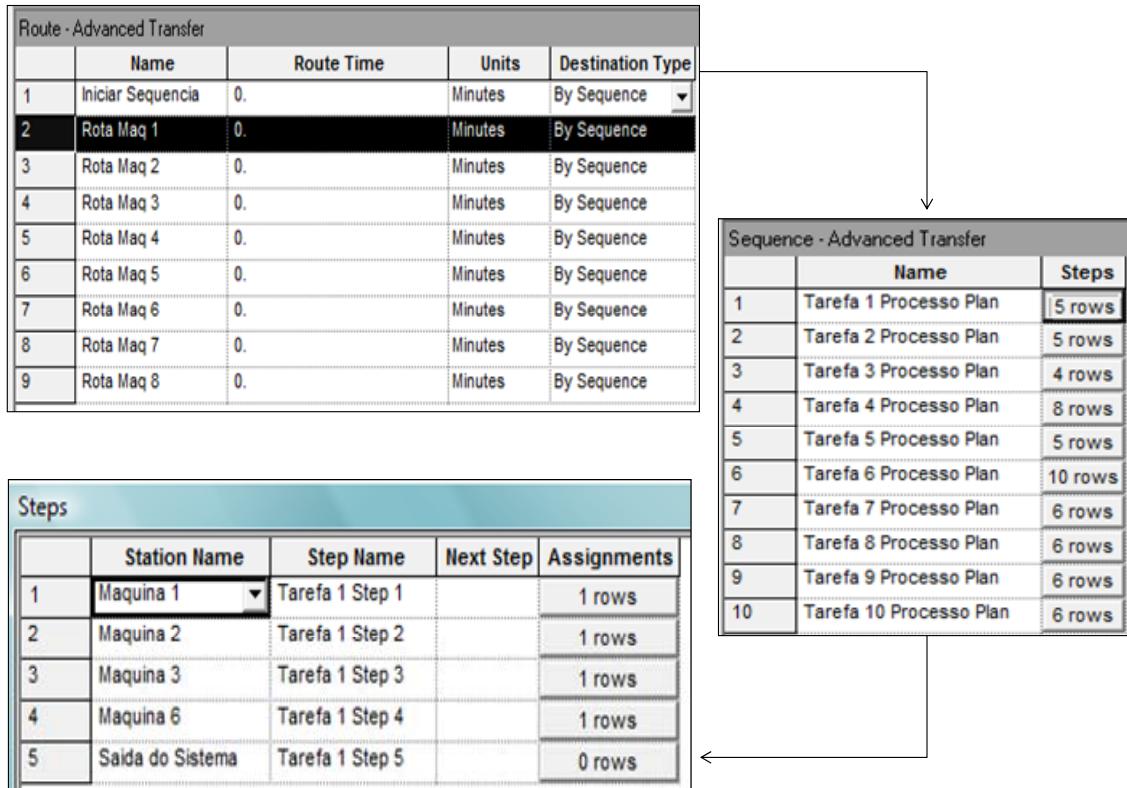


FIGURA 3.4: Sequenciamento de Ordens de produção no modelo. Destaque para a rota definida para tarefa 1

As equações utilizadas para o cálculo do tempo de processamento e para o prazo de entrega de cada ordem podem ser vistas na Figura 3.5.

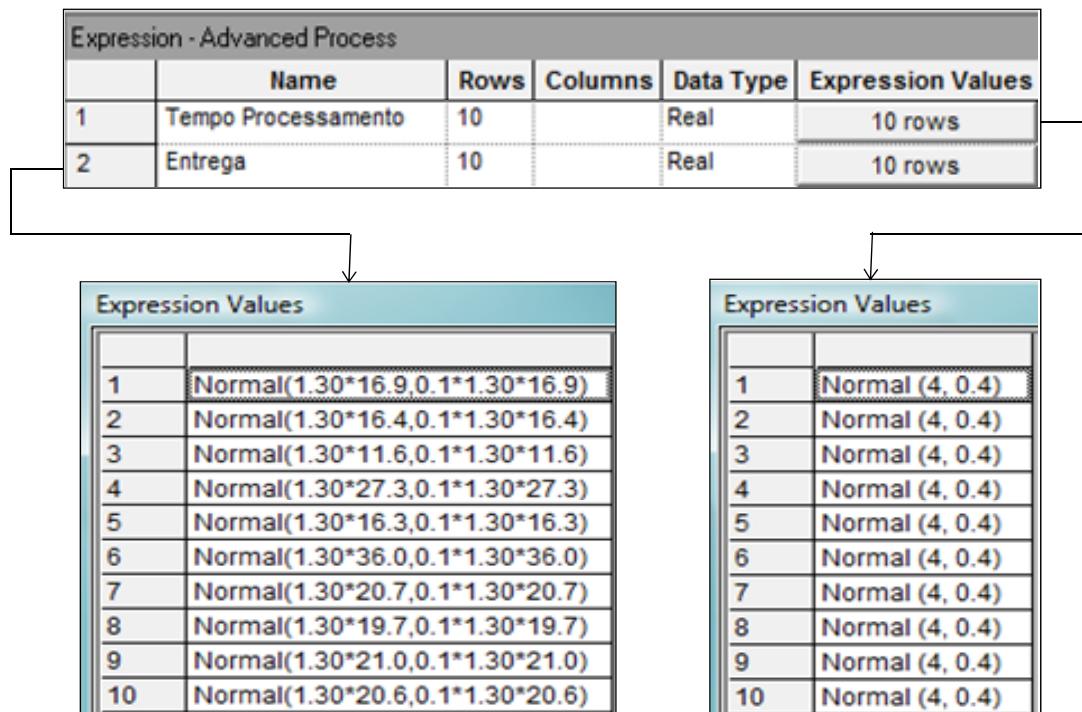
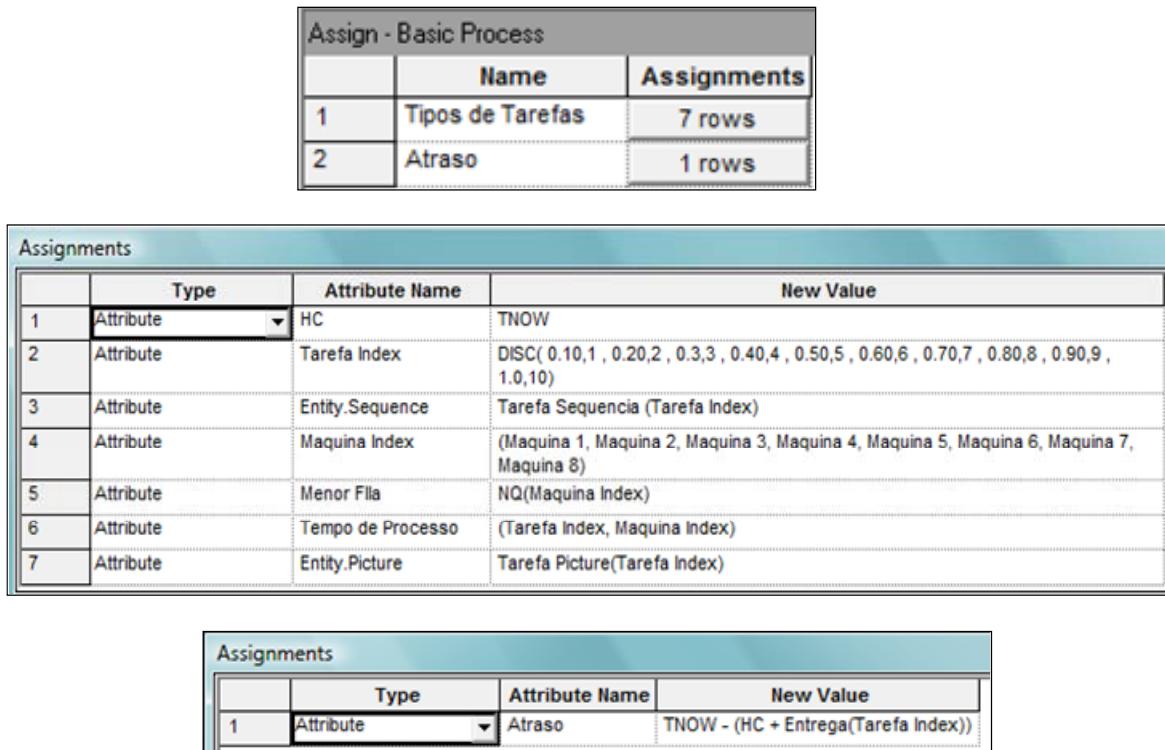


FIGURA 3.5: Expressões para cálculo de tempo de processamento e prazo de entrega

Como pode ser observado na Figura 3.6, as definições dos atributos das ordens de produção, como o tipo da ordem e seu tempo de processamento, são inseridos no modelo através do módulo *Assign* (módulo Tipos de Tarefas no modelo). Caso seja necessário alterar ou incluir uma nova definição, pode-se adicionar um novo módulo, como foi feito nesta simulação, para atribuir uma nova condição válida para as ordens que estivessem em atraso.



Assign - Basic Process

	Name	Assignments
1	Tipos de Tarefas	7 rows
2	Atraso	1 rows

Assignments

	Type	Attribute Name	New Value
1	Attribute	HC	TNOW
2	Attribute	Tarefa Index	DISC(0.10,1 , 0.20,2 , 0.3,3 , 0.40,4 , 0.50,5 , 0.60,6 , 0.70,7 , 0.80,8 , 0.90,9 , 1,0,10)
3	Attribute	Entity.Sequence	Tarefa Sequencia (Tarefa Index)
4	Attribute	Maquina Index	(Maquina 1, Maquina 2, Maquina 3, Maquina 4, Maquina 5, Maquina 6, Maquina 7, Maquina 8)
5	Attribute	Menor Fila	NQ(Maquina Index)
6	Attribute	Tempo de Processo	(Tarefa Index, Maquina Index)
7	Attribute	Entity.Picture	Tarefa Picture(Tarefa Index)

Assignments

	Type	Attribute Name	New Value
1	Attribute	Atraso	TNOW - (HC + Entrega(Tarefa Index))

FIGURA 3.6: Valores e variáveis atribuídas a cada uma das ordens

Intencionalmente, deixada para o final, a Figura 3.7 trás a tela onde são lançadas as informações necessárias para que o simulador identifique quais regras de sequenciamento devem ser seguidas pelas ordens que chegam, para ser processadas, em cada uma das máquinas, de acordo com a rota já definida. Consonante a informações anteriores, neste modelo a regra de sequenciamento adotada é Prioridade, condicionada a quanto menor o valor atribuído para a ordem, maior é a sua prioridade de atendimento. Cumpre ressaltar que uma mesma ordem pode possuir prioridades diferentes nas diversas máquinas definidas na sua rota.

The diagram illustrates the data flow for priority implementation. It starts with the 'Queue - Basic Process' table, which lists eight entries for 'Processo Maq' queues. An arrow points from this table to the 'Sequence - Advanced Transfer' table, which contains ten entries for 'Tarefa' process plans. Another arrow points from this table to the 'Steps' table, which details five steps for 'Tarefa 1'. A final arrow points from the 'Steps' table to the 'Assignments' table, which shows a single assignment for 'Prioridade' with a value of 4.78039.

Queue - Basic Process					
	Name	Type	Attribute Name	Shared	Report Statistics
1	Processo Maq 1.Queue	Lowest Attribute Value	Prioridade	<input type="checkbox"/>	<input checked="" type="checkbox"/>
2	Processo Maq 2.Queue	Lowest Attribute Value	Prioridade	<input type="checkbox"/>	<input checked="" type="checkbox"/>
3	Processo Maq 3.Queue	Lowest Attribute Value	Prioridade	<input type="checkbox"/>	<input checked="" type="checkbox"/>
4	Processo Maq 4.Queue	Lowest Attribute Value	Prioridade	<input type="checkbox"/>	<input checked="" type="checkbox"/>
5	Processo Maq 5.Queue	Lowest Attribute Value	Prioridade	<input type="checkbox"/>	<input checked="" type="checkbox"/>
6	Processo Maq 6.Queue	Lowest Attribute Value	Prioridade	<input type="checkbox"/>	<input checked="" type="checkbox"/>
7	Processo Maq 7.Queue	Lowest Attribute Value	Prioridade	<input type="checkbox"/>	<input checked="" type="checkbox"/>
8	Processo Maq 8.Queue	Lowest Attribute Value	Prioridade	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Sequence - Advanced Transfer				
	Name	Steps		
1	Tarefa 1 Processo Plan	5 rows		
2	Tarefa 2 Processo Plan	5 rows		
3	Tarefa 3 Processo Plan	4 rows		
4	Tarefa 4 Processo Plan	8 rows		
5	Tarefa 5 Processo Plan	5 rows		
6	Tarefa 6 Processo Plan	10 rows		
7	Tarefa 7 Processo Plan	6 rows		
8	Tarefa 8 Processo Plan	6 rows		
9	Tarefa 9 Processo Plan	6 rows		
10	Tarefa 10 Processo Plan	6 rows		

Steps				
	Station Name	Step Name	Next Step	Assignments
1	Maquina 1	Tarefa 1 Step 1		1 rows
2	Maquina 2	Tarefa 1 Step 2		1 rows
3	Maquina 3	Tarefa 1 Step 3		1 rows
4	Maquina 6	Tarefa 1 Step 4		1 rows
5	Saida do Sistema	Tarefa 1 Step 5		0 rows

Assignments			
	Assignment Type	Attribute Name	Value
1	Attribute	Prioridade	4.78039

FIGURA 3.7: Implementação de Prioridade

Os valores de prioridade atribuídos a estas ordens normalmente estão atrelados a informações de atributos como tempo, custo ou uma combinação de ambos e podem ser impostados manualmente ou de acordo com algum outro critério.

O critério utilizado para a atribuição de valores de prioridade neste módulo foi relacionado ao tipo de ordem e não ao tipo de máquina, pois eventualmente, poderão ser recebidas ordens que são processadas em uma máquina, vão para a máquina seguinte, são processadas e retornam para a máquina anterior e a sua prioridade deve permanecer. Ressalta-se que no caso em que duas ou mais ordens possuem a mesma prioridade, a regra de sequenciamento FIFO é ativada.

Oportunamente, a seção 3.3 detalhará como será feita a escolha e programação dos valores de prioridade na tela que esta representada na Figura 3.7.

3.2. MODELO DE OTIMIZAÇÃO

A otimização que será realizada pelo AG neste trabalho é relativa as regras de sequenciamento adotada no modelo de simulação, as quais são definidas por meio dos valores de prioridades dadas às ordens de produção.

Para que se possa realizar essa otimização, é necessário providenciar a customização do AG, ou seja, identificar na biblioteca GAlib os métodos que fornecem o suporte adequado à implementação do AG e adicionar as informações relevantes para o seu processamento.

Para esta dissertação, foram incluídos no AG os seguintes parâmetros e definições resumidas e justificativas na Tabela 3.2.

TABELA 3.2: Parâmetros, definições e justificativa

Parâmetro	Definição	Justificativa
Codificação do cromossomo	Forma Binária	Facilidade de transformação em números reais ou inteiro.
Reprodução e seleção natural	Roleta	Literatura: Mitchell (1997); Santa Catarina (2009) e Sean (2010).
Tamanho do cromossomo	47 genes	Número de parâmetros do problema - Figura 3.8.
Tamanho da população	30 indivíduos	Testes – Figura 4.1 e Tabela 4.1.
Taxa de substituição da população	80%	Literatura: Mitchell (1997); Santa Catarina (2009) e Sean (2010).
Taxa de mutação	0,05%	Literatura: Mitchell (1997); Santa Catarina (2009) e Sean (2010).
Taxa de cruzamento	80%	Literatura: Mitchell (1997); Santa Catarina (2009) e Sean (2010).
Número de gerações	100	Testes – Figura 4.1 e Tabela 4.1.
Critério de parada	Número de gerações	Literatura: Mitchell (1997); Santa Catarina (2009) e Sean (2010).
Parâmetro	Definição	Justificativa
Função de aptidão	Modelo de simulação	Abordagem proposta.
Função multiobjetivo	(1)	Equação de otimização das medidas de desempenho.

Cada cromossomo no AG deve representar os valores de prioridades de atendimento de cada ordem em cada uma das máquinas presentes na sua rota. Assim, considerando o

cenário em estudo com as rotas definidas na Tabela 3.1, trata-se de um problema de otimização com 47 genes ou parâmetros, o que indica sua complexidade.

Com vistas a um melhor entendimento do que significa um cromossomo binário com um comprimento de 47 genes, a Figura 3.8 traz uma representação gráfica, que deve ser lido como um cromossomo que possui 376 bits, dividido em 47 genes de 8 bits cada um. Então se pode traduzir esta figura da seguinte forma: a prioridade de atendimento da ordem do tipo 1 na Máquina *A*, por exemplo, representa um gene que é codificado na forma binária como uma sequência de 8 bits. Assim, um cromossomo é formado pelas prioridades de todas as ordens de produção nas máquinas em que essas ordens são processadas.

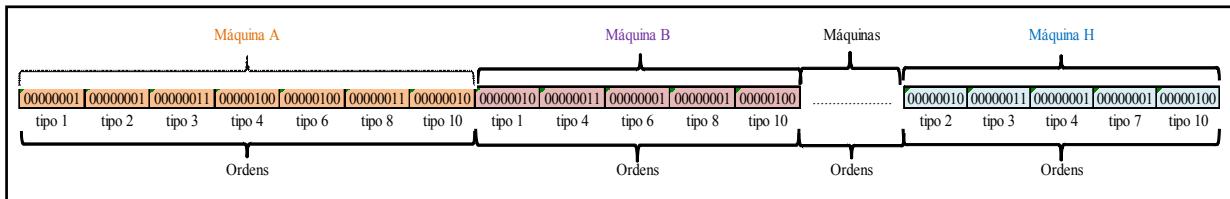


FIGURA 3.8: Representação gráfica do Cromossomo

Neste trabalho, os objetivos de minimização do MKP, do TTA e NOA foram combinados em um único funcional *F*, conforme equação (1).

Esse funcional multiobjetivo considerou os critérios de desempenho de MKP, TTA e NOA relacionados aos pesos representados por cada um dos parâmetros, α , β , γ .

$$F(\alpha, MKP, \beta, TTA, \gamma, NOA) = \alpha \sum_{i=1}^{20} \frac{MKP_i}{20} + \beta \sum_{i=1}^{20} \frac{TTA_i}{20} + \gamma \sum_{i=1}^{20} \frac{NOA_i}{20} \quad (1)$$

Foram executadas simulações considerando diferentes pesos para cada um dos objetivos conforme apresentado na Tabela 3.3. O somatório dos pesos deve ser sempre igual a 1, pois representa a proporção dada a cada um dos objetivos.

Optou-se por executar os processamentos com uma probabilidade inicial de igualdade para cada um dos objetivos e depois múltiplos de 25%.

TABELA 3.3: Pesos considerados para cada um dos objetivos

$\alpha = MKP$	$\beta = TTA$	$\gamma = NOA$
0,33	0,33	0,33
0	0	1
0,25	0,25	0,5
0,5	0,5	0
0,75	0	0,25
1	0	0
0	0,75	0,25
0	1	0
0,25	0,75	0
0	0,25	0,75
0,25	0	0,75

As demais informações constantes na Tabela 3.3: Parâmetros, definições e justificativas podem ser vistas nos recortes realizados no AG que estão na Figura 3.9 e Figura 3.10 e 3.11 cujas indicações são por si só explicativas.

```

// Define as funções
float f1 = 0.0;
float f2 = 0.0;
float f3 = 0.0;
// Define os pesos das funções
double alpha = 0.25;
double beta = 0.25;
double gama = 0.5;

for(i = 0; i < 20; i++)
{
    f1 -= Makespan[i];
    f2 -= TempoAtraso[i];
    f3 -= NunAtraso[i];
}

// Valores médios (20 replicacoes)
f1 = f1/20;
f2 = f2/20;
f3 = f3/20;
fitness = alpha * f1 + beta * f2 + gama * f3;
return fitness;

unsigned int i, j;
for(i=0; i<47; i++) → Definição do cromossomo
    fileOut << genome.phenotype(i) << endl;

fileOut.close();

```

Definição de funções e pesos

Definição da função de aptidão

Definição do cromossomo

FIGURA 3.9: Definições de peso, função e cromossomo no AG

```

GABin2DecPhenotype map;
for(i=0; i<n; i++)
    map.add(0, min[0], max[0]); } → Número de bits por gene

int popsize  = 30;
int ngen     = 100;
float pmut   = 0.05;
float pcross = 0.8; } → Número de população
                      Geração, mutação e
                      cruzamento

```

FIGURA 3.10: Definições de gene, população, mutação, cruzamento no AG

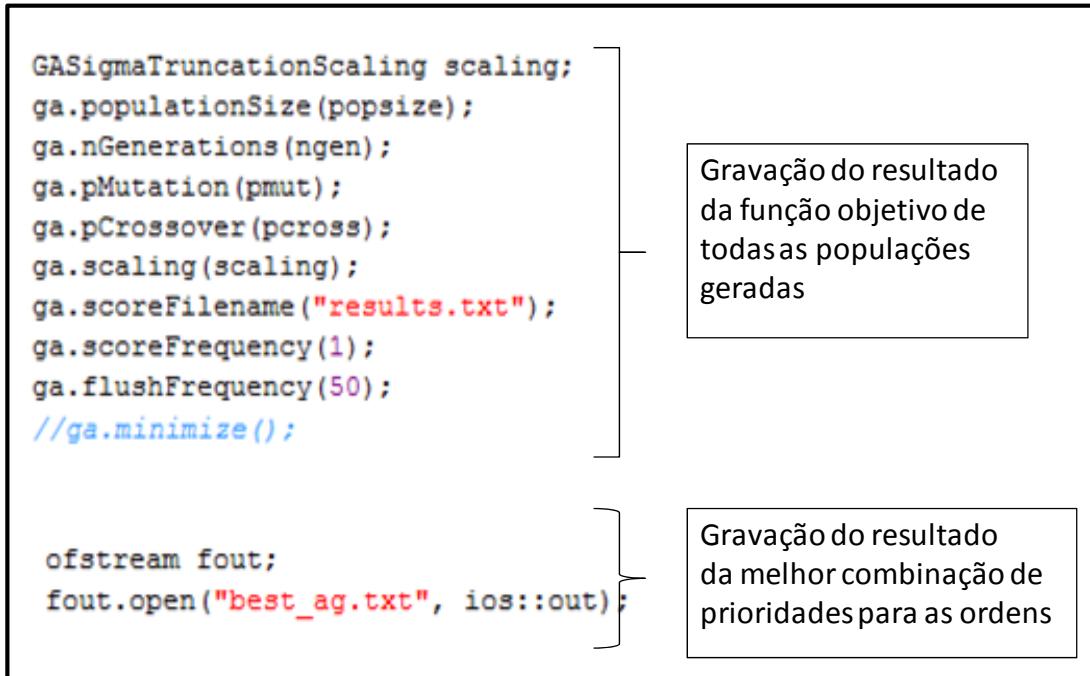


FIGURA 3.11: Indicação de gravação dos resultados da função objetivo e da melhor combinação no AG

3.3 ACOPLAGAMENTO ENTRE O MODELO DE SIMULAÇÃO E O AG

O acoplamento entre o modelo de simulação criado no Arena e o AG foi feita utilizando a linguagem VBA, cuja código completo poderá ser visto no Anexo A – Código VBA.

A seguir são mostradas partes do programa em VBA onde é feita a ligação com o AG.

O código VBA inicia com a limpeza do arquivo onde serão armazenados os resultados finais de MKP, TTA e NOA para cada uma das 20 replicações realizadas para o cromossomo recebido da população de “prioridades” gerada pelo AG. A seguir, faz a leitura de cada uma destas informações, conforme pode ser visto na Figura 3.11.

Após a leitura do arquivo recebido do AG, o simulador faz a gravação no modelo (Figura 3.7) de cada uma das prioridades em cada uma das máquinas relacionadas e prepara o arquivo para receber os resultados da simulação. Esta função esta ilustrada na Figura 3.12.

```

'Clear the Output File
Dim sOutputFile As String
sOutputFile = "C:/Users/Marilda/DissertacaoExe/samples_c/outputArena.txt"
iFileNum = FreeFile()
Open sOutputFile For Output As iFileNum
'close the file
Close iFileNum

'Faz a leitura do arquivo
sFileName = "C:/Users/Marilda/DissertacaoExe/samples_c/outputAG.txt"

```

FIGURA 3.12: Limpeza e posterior gravação do arquivo que será utilizado pelo AG

Logo que executa a simulação, realizando sua função de avaliação da aptidão do indivíduo, o simulador grava as novas informações dos critérios de desempenho para o MKP, TTA e NOA. Essas informações de desempenho são então lidas pelo AG que novamente fará seu processamento para otimizar estes parâmetros. Esse processo é repetido para todos os indivíduos da população nas diferentes gerações do AG.

```

'Aponta o modelo atual
Set oModel = ThisDocument.Model
'Muda Prioridades ordem 1
oModuleIndex = oModel.Modules.Find(smFindTag, "object.367")
Set oModule = oModel.Modules.Item(oModuleIndex)
oModule.Data("Value(1,1)") = StaticArray(1) 'Maq 1
oModule.Data("Value(1,2)") = StaticArray(8) 'Maq 2
oModule.Data("Value(1,3)") = StaticArray(15) 'Maq 3
oModule.Data("Value(1,4)") = StaticArray(32) 'Maq 6
.
'Create (or open) output file
sFileName = "C:/Users/Marilda/DissertacaoExe/samples_c/outputArena.txt"

```

FIGURA 3.13: Atualização da prioridade no simulador para execução e posterior envio de resultado para o AG

A informação das prioridades apresentadas na Figura 3.13 é incluída automaticamente nos módulos Sequence \Rightarrow Steps \Rightarrow Assignments conforme ilustração da Figura 3.7.

Partes das rotinas de execução do AG estão ilustradas na Figura 3.14, que apresenta o trecho do código que indica onde o AG deve gravar seu arquivo de saída para ser lido pelo simulador, chama a execução do simulador, aguarda sua execução e depois lê os arquivos

recebidos do simulador com as informações de desempenho do MKT, TTA e NOA, que serão usadas para o cálculo da aptidão no processo de otimização.

```

// Salva os valores do genoma (sequenciamento) em arquivo
ofstream fileOut;
fileOut.open("C:/Users/Marilda/DissertacaoExe/samples_c/outputAG.txt", ios::out)

// Chama Metodo que executa o modelo no Arena
printf("Running Arena Sample One...\n");
ArenaSample1();

// Le arquivo com resultados do Arena
ifstream fileIn;
fileIn.open("C:/Users/Marilda/DissertacaoExe/samples_c/OutputArena.txt");

```

FIGURA 3.14: Gravação do arquivo de saída, chamada da função de aptidão e gravação dos resultados para o simulador

Esta situação persistirá até que o critério de parada definido para o número de 100 gerações seja atendido. Ao final do processo, são feitas as gravações finais da solução conforme mostradas na Figura 3.10.

Para garantir o sincronismo nesse processo de comunicação entre o modelo de simulação e o AG, é necessária a existência de algum mecanismo que possa controlar a alternância da execução do processamento entre estados ativos e passivos, ou seja, a existência de um distribuidor de atividades planejadas e controladas entre o simulador e o AG uma vez que estes não podem trabalhar em simultâneo. Esta ação de sincronizar as atividades é realizada pelo ActiveX, e é ilustrada esquematicamente na Figura 3.15. (KELTON, SADOWSKI e SADOWSKI, 2000).

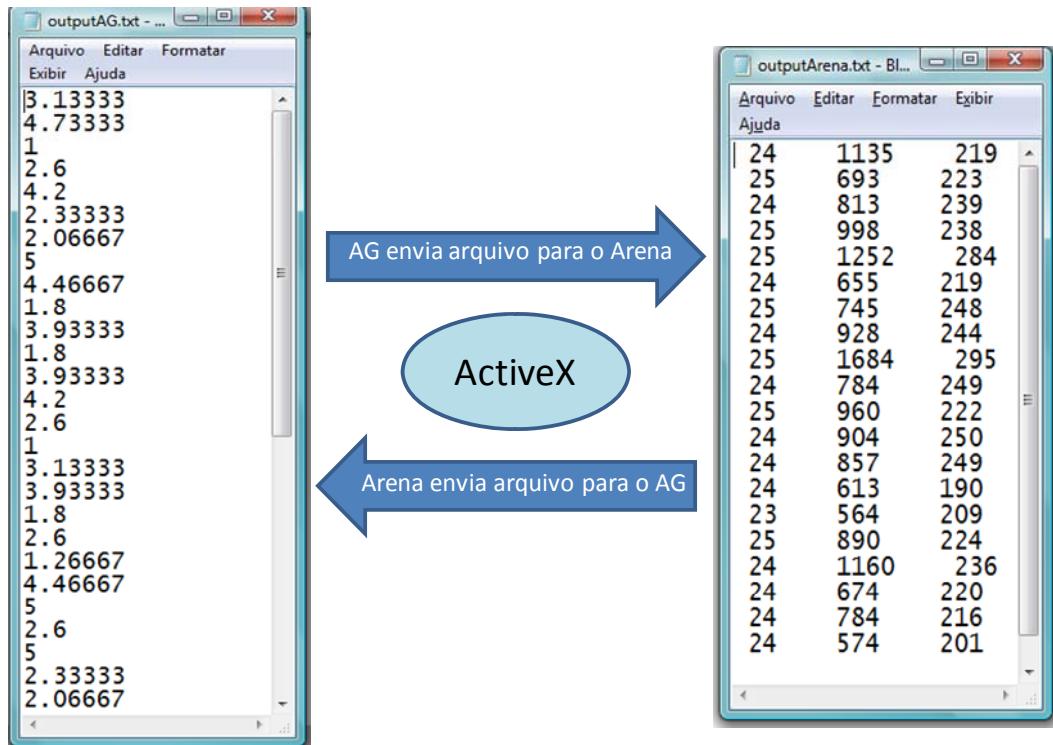


FIGURA 3.15: Representação gráfica da função do ActiveX

Essa sincronização é primordial para o funcionamento do acoplamento. Em resumo, após realizar a simulação e enviar ao AG as informações sobre os valores de MKP, TTA e NOA nas 20 replicações, o ARENA deve permanecer em espera até que o AG leia essas informações, calcule a aptidão do indivíduo avaliado e, então, escreva em arquivo os valores de prioridade referente ao próximo indivíduo da população para que o processo se reinicie com a realização de uma nova simulação pelo ARENA, considerando novos valores de prioridade.

Finalizado todo o processamento, a informação sobre a melhor combinação de prioridades é novamente inserida no modelo para que seja então realizado um último processamento de tal forma que o melhor resultado encontrado para as medidas de desempenho MKT, TTA e NOA sejam conhecidas.

3.4 VALIDAÇÃO

Para validar o modelo de simulação, relacionado à prioridade, foi construído um modelo de menor proporção, composto de três máquinas, dois tipos diferentes de ordens (ou tarefas), três operações e tempos de produção com média de 4 minutos e desvio padrão de 10% normalmente distribuídos. As rotas de cada tarefa e as condições de prioridade para esse sistema estão descritas na Tabela 3.4.

TABELA 3.4: Condição de processamento do modelo de validação

Rota	Prioridade
Tarefa 1/Máquina 1	1
Tarefa 1/Máquina 2	2
Tarefa 1/Máquina 3	3
Tarefa 2/Máquina 1	3
Tarefa 2/Máquina 2	1
Tarefa 2/Máquina 3	2

Especificamente, o simulador utilizado permite que se execute a simulação passo a passo, de forma lenta e com animação, possibilitando uma visão detalhada do que está acontecendo.

O primeiro teste de simulação foi verificar se o modelo atende a prioridade definida para cada ordem.

A Figura 3.16 apresenta dois momentos da simulação bem próximos um do outro, de tal forma que se pode ver no Momento 1, uma Tarefa do tipo 2 e duas tarefas do tipo 1, identificadas como entidades #6, #8 e #7, respectivamente. Em seguida, no Momento 2, pode se visto que a Tarefa do tipo 2 desapareceu e aparece uma nova Tarefa do tipo 1, identificada

como entidade #9, portanto pode se concluir que a prioridade esta sendo obedecida pelo modelo.

A necessidade desta verificação é importante para o momento do acoplamento com o AG, pois a prioridade das ordens deve permanecer na otimização.

FIGURA 3.16: Validação da Prioridade

Em outra validação, foram incluídas tanto no modelo de simulação quanto numa planilha do Excel todas as informações exatamente iguais conforme Tabela 3.5. Para o simulador deveria ser criada uma condição de que todas as ordens atrasassem, mas não provocasse fila, caso contrário os resultados poderiam ser distorcidos, conforme comprovado em outros testes realizados.

TABELA 3.5: Parâmetros de processamento para a validação do modelo

Condições de processamento	Excel	Simulador
Chegada de tarefas: constante	15 min.	15 min.
Tempo de processamento: constante	9 min.	9 min.
Prazo de entrega: constante após o recebimento da ordem	2 min.	2 min.
Tempo simulado: contínuo	10000 min.	10000 min.

Nesta simulação, independentemente de quantas replicações sejam realizadas, os resultados são os mesmos, uma vez que, nesse caso, todos os dados de entrada foram determinísticos (Tabela 3.6). Além disso, os resultados para os critérios de desempenho MKP, do TTA e do NOA, são exatamente os mesmos produzidos pela planilha *Excel*, conforme esperado. Esses resultados são apresentados na Tabela 3.6.

TABELA 3.6: Resultado do Simulador *versus* resultado do *Excel*

Resultado	Simulador	Excel
Tempo de simulação	10005 min.	10000 min.
MKP	9 min.	9 min.
TTA	4669 min.	4669 min.
NOA	667 ordens	667 ordens

Procedeu-se também a verificação de confiabilidade do número de replicações no processamento do modelo de simulação utilizando-se a Análise de Dados disponível no *Excel*.

Considerou-se como nível de confiabilidade para a média o índice de 99%, que representa a confiabilidade ao usar os valores encontrados como uma estimativa para a média da população que, neste caso, são todas as ordens recebidas.

As estatísticas descritivas para os critérios de desempenho MKP, TTA e NOA estão apresentadas, respectivamente, nas Tabelas 3.7, 3.8, e 3.9. Ressalta-se que para os propósitos

da avaliação realizada neste trabalho, apenas os valores de média, erro-padrão, nível de confiança e precisão foram considerados.

Vale relembrar que os pesos α , β e γ , de cada um dos objetivos estão atrelados respectivamente aos critérios de desempenho acima citados.

Como pode ser verificado na Tabela 3.7, para 20 replicações, com um nível de confiança de 99% tem-se um uma precisão entre 0,4% e 0,6% em relação tempo médio total de atravessamento das ordens. A precisão é definida como a razão entre os valores de erro padrão e média para cada uma das combinações testadas.

TABELA 3.7: Análise estatística dos resultados dos processamentos de acordo com os pesos dos parâmetros α , β e γ - MKP

	FIFO	LIFO	$\alpha=0,33$ $\beta=0,33$ $\gamma=0,33$	$\alpha=0$ $\beta=0$ $\gamma=1$	$\alpha=0,5$ $\beta=0,5$ $\gamma=0$	$\alpha=0,75$ $\beta=0$ $\gamma=0,25$	$\alpha=1$ $\beta=0$ $\gamma=0$	$\alpha=0$ $\beta=0,75$ $\gamma=0,25$	$\alpha=0$ $\beta=1$ $\gamma=0$	$\alpha=0,25$ $\beta=0,75$ $\gamma=0$	$\alpha=0$ $\beta=0,25$ $\gamma=0,75$	$\alpha=0,25$ $\beta=0$ $\gamma=0,75$
Média	24,250	24,250	24,300	24,200	24,300	24,400	24,500	24,200	24,250	24,350	24,400	24,350
Erro padrão	0,123	0,099	0,128	0,138	0,105	0,112	0,115	0,117	0,123	0,109	0,134	0,131
Mediana	24	24	24	24	24	24	24,5	24	24	24	24	24
Modo	24	24	24	24	24	24	24	24	24	24	24	24
Desvio padrão	0,550	0,444	0,571	0,616	0,470	0,503	0,513	0,523	0,550	0,489	0,598	0,587
Variância da amostra	0,303	0,197	0,326	0,379	0,221	0,253	0,263	0,274	0,303	0,239	0,358	0,345
Assimetria	0,132	1,251	-0,038	-0,120	0,945	0,442	0,000	0,294	0,132	0,681	1,245	-0,212
Intervalo	2	1	2	2	1	1	1	2	2	1	2	2
Mínimo	23	24	23	23	24	24	24	23	23	24	24	23
Máximo	25	25	25	25	25	25	25	25	25	25	26	25
Soma	485	485	486	484	486	488	490	484	485	487	488	487
Contagem	20	20	20	20	20	20	20	20	20	20	20	20
Nível de confiança(99,0%)	0,352	0,284	0,365	0,394	0,301	0,322	0,328	0,335	0,352	0,313	0,383	0,376
Precisão	0,005	0,004	0,005	0,006	0,004	0,005	0,005	0,005	0,005	0,004	0,005	0,005

Na Tabela 3.8 também pode ser verificado que mantido o número de 20 replicações, considerando $\alpha = 0,33$, $\beta = 0,33$ e $\gamma = 0,33$, com um nível de confiança de 99%, tem-se um erro padrão de 5,678 num total de 233 ordens, o que equivale a um nível de precisão de 2,4%.

TABELA 3.8: Análise estatística dos resultados dos processamentos de acordo com os pesos dos parâmetros α , β e γ - NOA

	FIFO	LIFO	$\alpha=0,33$ $\beta=0,33$ $\gamma=0,33$	$\alpha=0$ $\beta=0$ $\gamma=1$	$\alpha=0,5$ $\beta=0,5$ $\gamma=0$	$\alpha=0,75$ $\beta=0$ $\gamma=0,25$	$\alpha=1$ $\beta=0$ $\gamma=0$	$\alpha=0$ $\beta=0,75$ $\gamma=0,25$	$\alpha=0$ $\beta=1$ $\gamma=0$	$\alpha=0,25$ $\beta=0,75$ $\gamma=0$	$\alpha=0$ $\beta=0,25$ $\gamma=0,75$	$\alpha=0,25$ $\beta=0$ $\gamma=0,75$
Média	236,000	220,250	233,750	232,000	221,800	232,850	235,750	231,000	224,950	232,050	231,900	229,850
Erro padrão	6,285	4,863	5,678	6,148	5,701	5,509	5,263	5,292	4,826	5,668	4,627	5,913
Mediana	241,500	218,500	230,000	228,000	219,500	230,500	231,000	228,500	217,500	228,500	231,500	226,500
Modo	#N/D	202	219	228	231	211	268	232	229	229	218	213
Desvio padrão	28,107	21,747	25,393	27,494	25,494	24,637	23,537	23,669	21,583	25,349	20,693	26,444
Variância da amostra	790,000	472,934	644,829	755,895	649,958	606,976	553,987	560,211	465,839	642,576	428,200	699,292
Assimetria	0,259	1,022	0,785	1,188	0,548	1,229	0,978	0,931	0,797	0,978	0,022	1,114
Intervalo	106	86	105	105	107	90	100	106	77	114	76	110
Mínimo	188	187	190	194	170	203	198	191	196	188	193	189
Máximo	294	273	295	299	277	293	298	297	273	302	269	299
Soma	4720	4405	4675	4640	4436	4657	4715	4620	4499	4641	4638	4597
Contagem	20	20	20	20	20	20	20	20	20	20	20	20
Nível de confiança(99,0%)	17,981	13,912	16,245	17,588	16,309	15,761	15,057	15,141	13,807	16,216	13,238	16,917
Precisão	0,027	0,022	0,024	0,026	0,026	0,024	0,022	0,023	0,021	0,024	0,020	0,026

Nas mesmas condições de processamento, para o tempo total de atraso, verifica-se que o maior erro padrão encontrado foi de 69,91 minutos em 905 minutos, o que produz um nível e precisão de 7,7%, em que os pesos considerados foram de $\alpha = 0$, $\beta = 0$ e $\gamma = 1$ (Tabela 3.9).

Nesta Tabela 3.9, identifica-se que a prioridade dada ao peso de redução do tempo de atraso é a que apresenta o melhor resultado, ou seja: 706,55 minutos, com um nível de precisão 7,2%.

TABELA 3.9: Análise estatística dos resultados dos processamentos de acordo com os pesos dos parâmetros α , β e γ - TTA

	FIFO	LIFO	$\alpha=0,33$ $\beta=0,33$ $\gamma=0,33$	$\alpha=0$ $\beta=0$ $\gamma=1$	$\alpha=0,5$ $\beta=0,5$ $\gamma=0$	$\alpha=0,75$ $\beta=0$ $\gamma=0,25$	$\alpha=1$ $\beta=0$ $\gamma=0$	$\alpha=0$ $\beta=0,75$ $\gamma=0,25$	$\alpha=0$ $\beta=1$ $\gamma=0$	$\alpha=0,25$ $\beta=0,75$ $\gamma=0$	$\alpha=0$ $\beta=0,25$ $\gamma=0,75$	$\alpha=0,25$ $\beta=0$ $\gamma=0,75$
Média	760,700	926,550	883,350	905,850	861,000	929,450	964,850	854,050	706,550	835,400	925,450	891,500
Erro padrão	56,846	61,079	60,389	69,910	53,248	68,659	47,774	69,303	51,165	62,359	63,440	60,756
Mediana	734,000	928,500	835,000	806,500	876,500	813,500	965,000	762,500	687,500	797,500	844,500	797,500
Modo	#N/D	#N/D	784	#N/D	#N/D	810	855	623	#N/D	1391	#N/D	679
Desvio padrão	254,225	273,155	270,067	312,645	238,134	307,051	213,650	309,934	228,817	278,876	283,710	271,711
Variância da amostra	64630,22	74613,42	72936,34	97746,98	56707,79	94280,47	45646,34	96059,00	52357,21	77771,83	80491,52	73827,00
Assimetria	0,518	1,156	1,448	0,996	0,281	0,837	0,229	1,989	1,142	0,775	1,450	0,763
Intervalo	1003	1121	1120	1173	892	1023	778	1236	919	915	1299	960
Mínimo	386	551	564	426	434	573	615	584	421	476	448	535
Máximo	1389	1672	1684	1599	1326	1596	1393	1820	1340	1391	1747	1495
Soma	15214	18531	17667	18117	17220	18589	19297	17081	14131	16708	18509	17830
Contagem	20	20	20	20	20	20	20	20	20	20	20	20
Nível de confiança(99,0%)	162,634	174,744	172,769	200,007	152,340	196,428	136,677	198,272	146,380	178,404	181,496	173,820
Precisão	0,075	0,066	0,068	0,077	0,062	0,074	0,050	0,081	0,072	0,075	0,069	0,068

Como teste, foi efetuado um incremento de 50% no número de amostras, porém o nível de precisão não apresentou variação suficiente grande que justificasse a alteração. Com efeito, pode-se concluir com base na literatura de que o valor estimado na amostra representa de fato o parâmetro populacional (MARTINS, 2008).

4 ANÁLISE E DISCUSSÃO OS RESULTADOS

A comparação entre os resultados da função de otimização, em que se considera o mesmo peso para os objetivos e a variação nos parâmetros de geração e população, obtidas nos testes realizados, conforme parágrafo anterior aponta para uma redução conforme pode ser visto na Figura 4.1. Na Figura tem-se a convergência do AG, com o uso da função objetivo com pesos iguais para α , β e γ , após 30, 50, 60, 80 e 100 gerações.

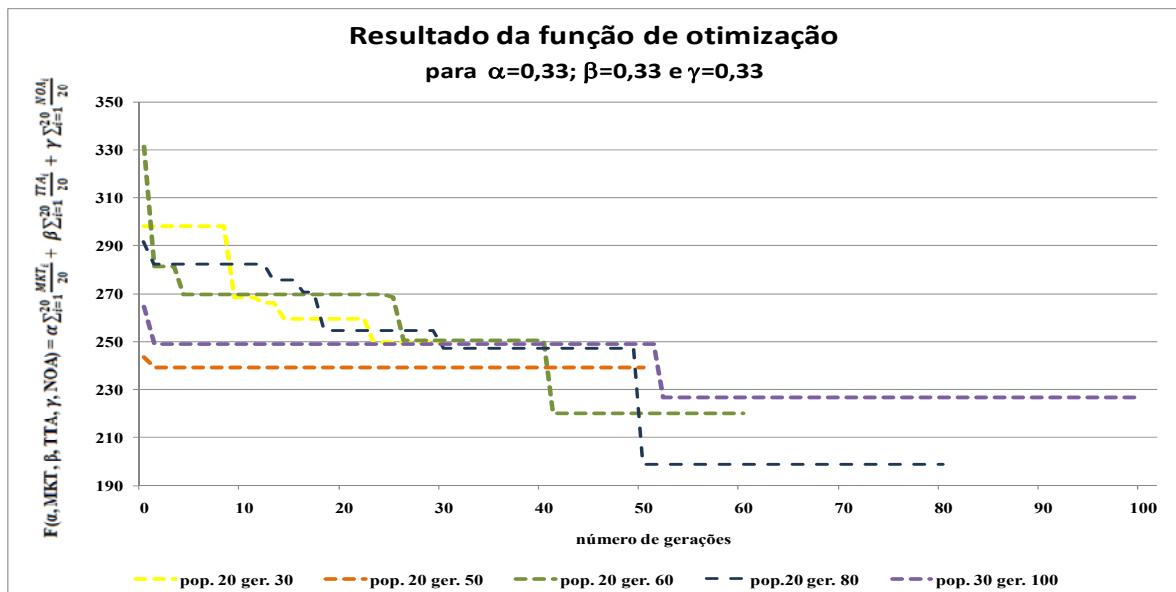


FIGURA 4.1: Comparação dos resultados de acordo com o número de população e geração

Embora o resultado da função objetivo tenha apresentado como melhor resultado a situação em que se considerou a população de 20 indivíduos e 80 gerações, optou-se por trabalhar com um valor maior de geração e população em função dos resultados individuais como poderá ser visto na Tabela 4.1, onde foram executadas variações em alguns pesos. Como exemplo, pode-se ver que para os pesos de $\alpha = 0,5$; $\beta = 0,5$ e $\gamma = 0$, o resultado para MKP e NOA são melhores para a população maior. Quando foram considerados os pesos de $\alpha=0$ $\beta=0,75$ $\gamma=0,25$ e $\alpha=0,75$ $\beta=0,25$ $\gamma=0$, os melhores resultados para todas as medidas de desempenho são superiores para a população igual a 30.

Portanto, depois dos testes onde foram avaliadas diversas alternativas com relação ao número de população e o número de gerações, optou-se por população igual a 30 indivíduos e a geração igual a 100.

TABELA 4.1: Resumo dos resultados nos testes

Pesos		$\alpha=0,33$	$\alpha=0,5$	$\alpha=0$	$\alpha=0$
		$\beta=0,33$	$\beta=0,5$	$\beta=0,75$	$\beta=1$
		$\gamma=0,33$	$\gamma=0$	$\gamma=0,25$	$\gamma=0$
(A)	MKP (min.)	24,300	24,300	24,200	24,250
pop. = 30	NOA (ordens)	233,750	221,800	231,000	224,950
ger. = 100	TTA (min.)	883,400	861,000	854,100	706,600
<hr/>					
(B)	MKP (min.)	24,200	24,350	24,400	24,400
pop. = 20	NOA (ordens)	226,950	231,900	231,850	231,600
ger. = 80	TTA (min.)	887,550	832,650	877,150	926,600
<hr/>					
(A) – (B)	MKP (min.)	0,100	-0,050	-0,200	-0,150
	NOA (ordens)	6,800	-10,100	-0,850	-6,650
	TTA (min.)	-4,200	28,350	-23,100	-220,050

Diferentemente da comparação efetuada para a Figura 4.1, onde a finalidade era mostrar a convergência relativa ao número de gerações e da população, na Figura 4.2 a comparação é realizada entre os resultados da função de otimização em função dos diferentes pesos dados a cada um dos objetivos. Nota-se que a convergência também ocorre, isto é, há uma redução dos valores iniciais conforme se aumenta o número de gerações.

Ainda em relação a este Figura, observe-se que para $\alpha=0,25 \beta=0,75 \gamma=0$, o valor inicial e o valor final da convergência é de aproximadamente 30% ou seja, reduz de 530 para 360.

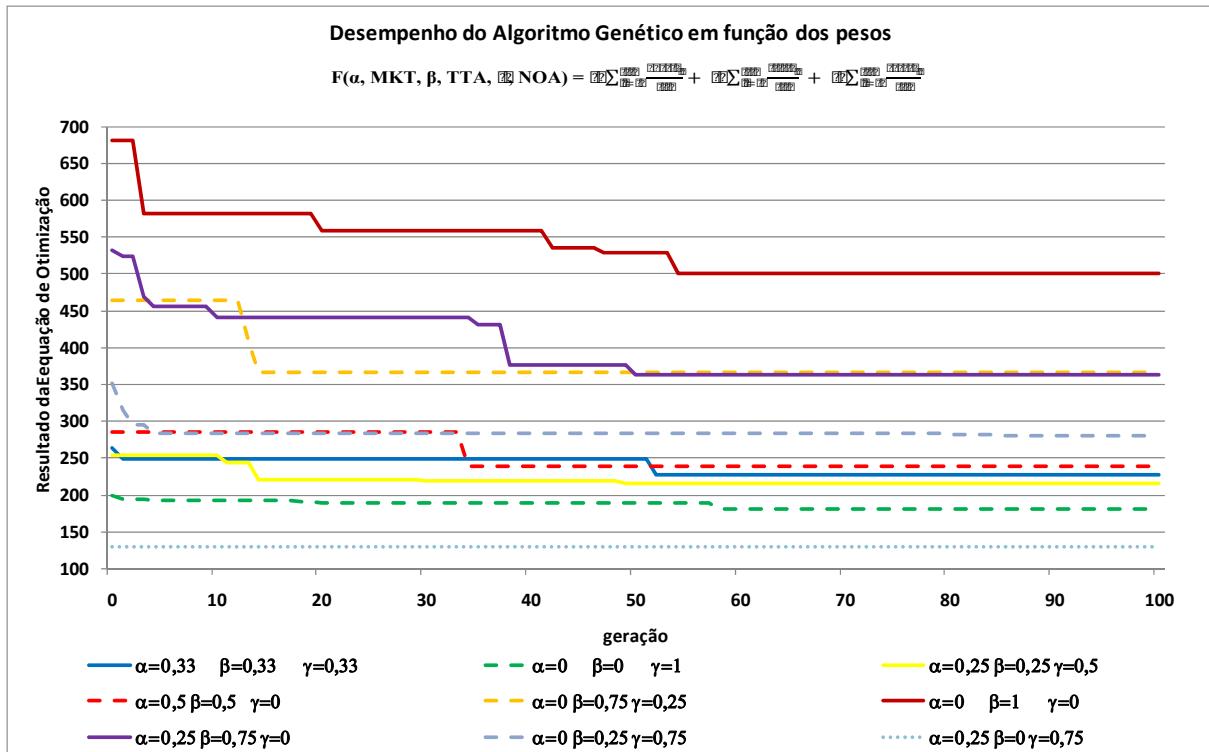


FIGURA 4.2: Desempenho do AG em função dos pesos de α , β e γ

Cada um dos processamentos pode ser analisado separadamente, ou em conjunto. É possível notar nas melhores soluções encontradas que, em geral, a melhora em um objetivo pode ocasionar uma piora no resultado em outro objetivo. Esse comportamento foi observado, por exemplo, na situação em que se prioriza o objetivo NOA e, como consequência, encontra-se o melhor resultado para o MKP ($\alpha=0$, $\beta=0$ e $\gamma=1$ na Figura 4.6).

Nesse mesmo caso, por outro lado, resulta um dos piores valores para o TTA enquanto que próprio NOA fica aproximadamente 5% maior que o seu melhor resultado quando o peso maior é dado aos objetivos MKP e TTA.

As melhores soluções obtidas pelo AG são candidatas a formarem a Frente de Pareto, (Figura 4.3) que é definida pelas soluções chamadas não dominadas, tais que qualquer melhoria em um objetivo só pode ocorrer com o agravamento do outro objetivo (SEAN, 2010).

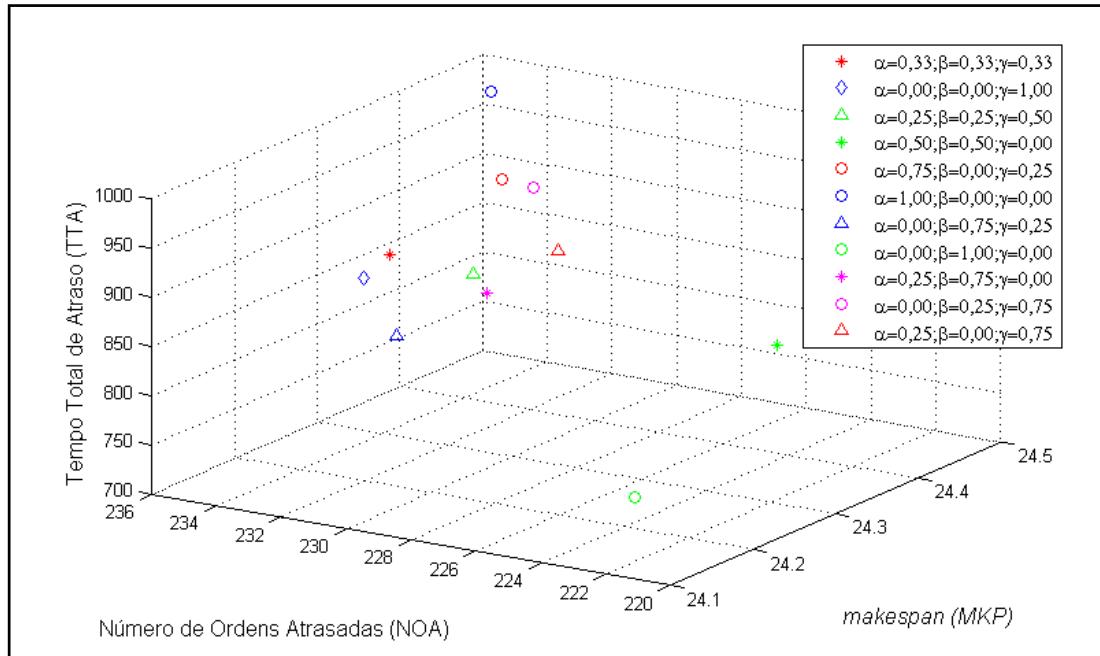


FIGURA 4.3: Soluções obtidas pelo AG, candidatas à Frente de Pareto

A Figura 4.4 apresenta uma comparação entre os valores médios das 20 replicações do modelo, considerando a solução ótima de sequenciamento obtida pelo AG em cada caso. Na Figura a análise é realizada sobre o tempo total das ordens atrasadas e pode-se notar que o melhor resultado foi alcançado quando o maior valor foi dado para o peso β que representa o objetivo TTA. Os valores: mínimo e máximo representam os extremos do intervalo de confiança de 99% para a média populacional.

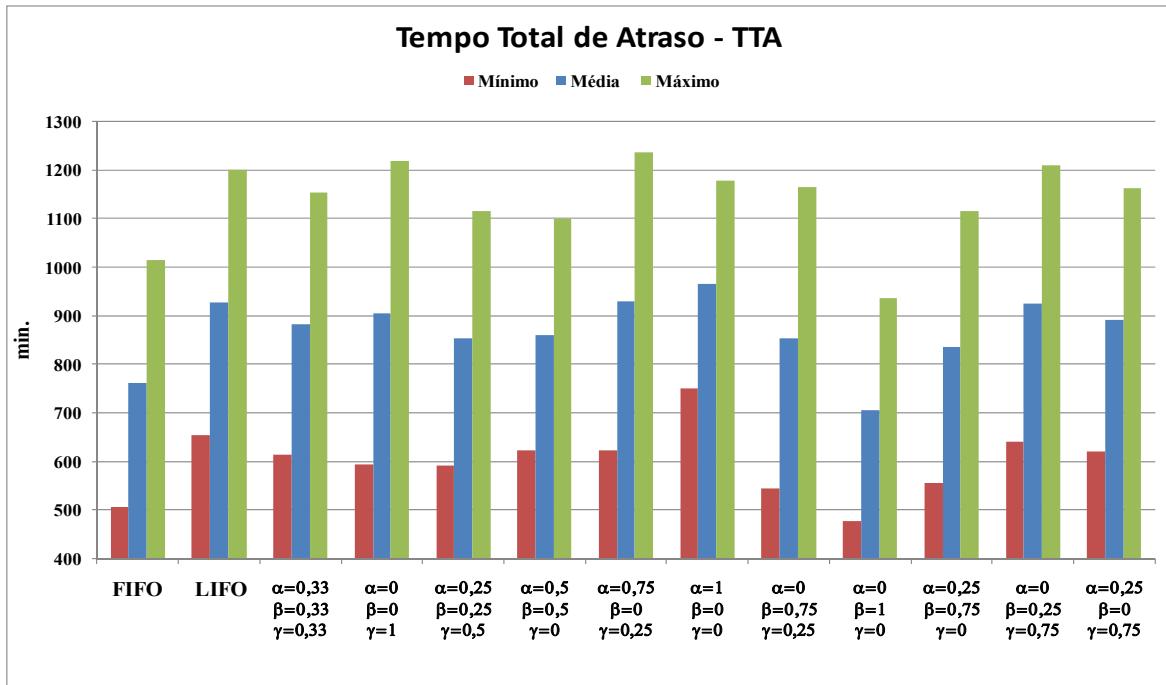


FIGURA 4.4: Tempo Total de Atraso - TTA

A Figura 4.5 traz os resultados que considera o objetivo de reduzir o número de ordens atrasadas. Como pode ser visto o melhor valor encontrado reside na igualdade de priorização dos pesos α e β , e não na priorização identificada por γ .

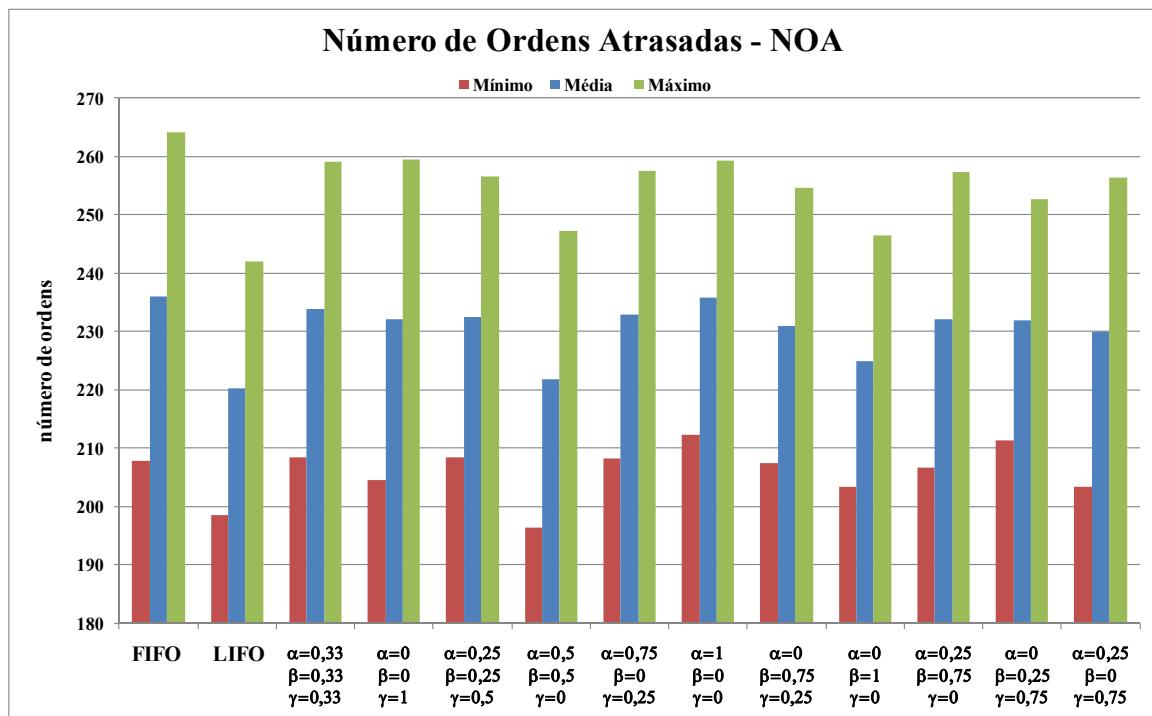


FIGURA 4.5: Número de Ordens Atrasadas – NOA

Porém, este valor ainda é superior se comparado com a opção LIFO, executada pelo próprio simulador. Ressalta-se que mesmo alterando os parâmetros de população para 30 indivíduos e 150 gerações, não foi possível encontrar o melhor resultado para o número de ordens atrasadas.

Uma possível justificativa deste resultado pode ser baseada nas condições de funcionamento do sistema, em que quase inexistem filas e, consequentemente, a regra FIFO é adotada. Testes adicionais devem ser realizados aumentando o ritmo de chegadas de ordem no sistema para comprovar essa hipótese.

Nos resultados da Figura 4.6, ocorreu um fenômeno em que os melhores valores encontrados para o tempo de travessamento são aqueles onde se prioriza o NOA, com α e β iguais a zero e γ igual a 1. Esse comportamento pode ser atribuído à dependência parcial entre os objetivos. Esta situação foi comprovada aumentando o número de gerações e mantendo o número de indivíduos. Situação semelhante foi observada para α igual a zero, β igual a 0,75 e γ igual a 0,25.

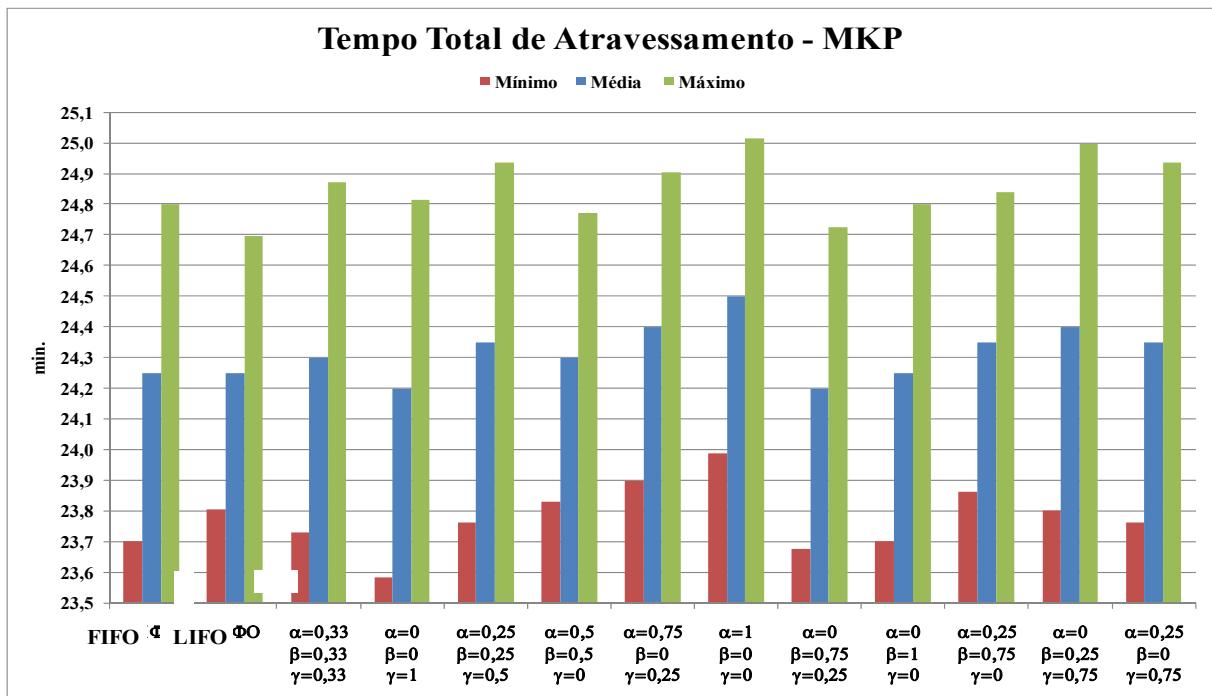


FIGURA 4.6: Tempo Total de Atravessamento - MKP

Para finalizar esta seção, apresenta-se na Figura 4.7, uma solução de sequenciamento otimizada pelo AG. Observe que cada ordem possui um ordenamento diferente em cada máquina por onde ela deve passar. As ordens que passam duas vezes pela mesma máquina de acordo com a rota de produção definida na Tabela 3.1 não é repetida, pois ela possui sempre a mesma prioridade em uma mesma máquina.

Também vale ressaltar que conforme definido anteriormente quando a mesma prioridade for atribuída a ordens diferentes na mesma máquina tem preferência de execução aquela que chegou primeiro (FIFO), neste resultado tem-se esta situação nas máquinas A e F com as ordens respectivamente dos tipos 2, 8, e 1, 7 e G, para as ordens dos tipos 5 e 9.

Máquina A	1,26667	1,8	2,06667	2,33333	2,33333	2,86667	3,93333
	tipo 6	tipo 3	tipo 10	tipo 2	tipo 8	tipo 4	tipo 1
Máquina B	2,06667	3,66667	3,66667	3,93333	4,46667	4,46667	5
	tipo 4	tipo 6	tipo 8	tipo 1	tipo 7	tipo 10	tipo 9
Máquina C	1,26667	1,53333	2,33333	4,2	4,73333	4,73333	
	tipo 4	tipo 5	tipo 6	tipo 1	tipo 2	tipo 9	
Máquina D	1,26667	3,66667	4,46667	4,73333			5
	tipo 5	tipo 9	tipo 6	tipo 10	tipo 8		
Máquina E	1	2,33333	2,6	3,66667	4,46667	4,73333	
	tipo 10	tipo 4	tipo 6	tipo 8	tipo 7	tipo 3	
Máquina F	1	1	1,8	2,6	4,46667		
	tipo 1	tipo 7	tipo 9	tipo 4	tipo 5		
Máquina G	1,8	1,8	2,33333	2,6	3,13333	4,46667	
	tipo 5	tipo 9	tipo 2	tipo 8	tipo 4	tipo 6	
Máquina H	2,33333	2,6	3,4	3,93333	4,46667		
	tipo 3	tipo 10	tipo 2	tipo 7	tipo 4		

FIGURA 4.7: Sequenciamento de ordens por tipo de máquina

5 CONCLUSÃO

Neste capítulo estão apresentadas as conclusões obtidas no desenvolvimento desta dissertação, pela aplicação da abordagem proposta. Apresentam-se também as contribuições advindas e as sugestões para trabalhos futuros.

A função de aptidão com a utilização de um modelo de simulação em substituição a um modelo matemático analítico é viável, e nesta dissertação pode ser feita com a utilização de uma linguagem de programação apropriada, que neste caso foi VBA.

Atingiu-se o objetivo de validar um modelo de simulação como função de aptidão em AGs em substituição a uma equação de matemática analítica, obtendo maior flexibilidade.

Uma vez que todas as observações necessárias sobre o desempenho do sistema utilizando a animação já haviam sido feitas, optou-se por processar em *batch*, o que significa que a animação deixa de ser realizada. Desta forma, o tempo de processamento reduziu para aproximadamente 3 horas equivalente a 94% em relação ao tempo anterior. A substituição do simulador por uma linguagem similar a do AG, poderá tornar o tempo de processamento menor ainda, proporcionando então a agilidade necessária para um auxílio em tempo real aos gestores de sequenciamento da produção.

Pode-se concluir que a adaptação da ferramenta de otimização para ambientes de manufatura reais pode ser feita adaptando o modelo de simulação, sem a necessidade de trabalhar com formulações matemáticas complexas.

Os resultados permitem concluir que é possível melhorar todos os resultados para os critérios de desempenho adotados. A maior dificuldade esta em reduzir o NOA em função das condições do sistema, onde a inexistência de filas aciona a regra FIFO. Novos testes aumentando o ritmo de chegadas de ordens devem ser realizados para comprovar essas causas.

A utilização desta abordagem permite também que se verifique qual o melhor índice de recebimento de ordens por tipo e se faça um planejamento em longo prazo. Deve-se, apenas observar a ocorrência de alguma variação representativa nesta quantidade a fim de se ajustar o sistema com a adoção do novo índice. Ressalta-se que esse ajuste no modelo pode ser feita de uma forma muito simples, apenas alterando no modelo a distribuição de probabilidades, ou seus parâmetros, que descrevem o processo de chegada conforme apresentado na FIGURA 3.3.

5.1 SUGESTÕES PARA CONTINUIDADE DA PESQUISA

A continuidade da pesquisa pode ser feita em várias direções, como por exemplo:

- Realizar um acoplamento do ARENA com o AG totalmente de forma interna: atualmente o resultado da simulação do Arena está acoplado ao AG através de uma ligação parcialmente fora do processo (por arquivos de texto);
- Substituição do simulador Arena por outro simulador.
- Utilização desta mesma abordagem em otimizações híbridas, ACO, entre outros.
- Pode-se também aplicar este modelo na lógica fuzzy.

Apresenta-se no Anexo B, uma lista dos trabalhos publicados oriundos desta dissertação.

REFERÊNCIA BIBLIOGRÁFICA

AKASAWA, A. **Aplicação da Simulação de Eventos Discretos como ferramenta integrada ao planejamento e programação da produção na manufatura ágil**. Itajubá, 2007, 118p. Dissertação (Mestrado em Ciências em Engenharia de Produção) Universidade Federal de Itajubá. Minas Gerais, 2007.

ALBA, E.; DORRONSORO, B.; LUNA, F., NEBRO, A. J.; BOUVRY, P.; HOGIE, L. A cellular multi-objective genetic algorithm for optimal broadcasting strategy in metropolitan MANETs. **Computer Communications**, v. 30, p. 685-697, 2007.

ALMEIDA, M. S.; COSTA, Y. P. J. S.; FRANCISCO, C. A. C.; GOMES, J. C. Utilização da simulação em ARENA 7.0 no auxílio ao balanceamento da célula de montagem de uma fábrica de calçados. In: **XXVI Encontro Nacional de Engenharia de Produção**. Fortaleza, CE, Brasil, 09 a 11 de outubro de 2006.

ALTIOK, T.; MELAMED, B. **Simulation Modeling and Analysis with Arena**. Oxford, Elsevier, 2007.

ARAUJO, R. P.; SANTOS, A. G.; ARROYO, J. E. C. Algoritmo Genético com busca local aplicado ao problema de just-in-time jobshop scheduling. In: **XLI SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL - SBPO**. Porto Seguro, BA, Brasil, 01 a 04 de setembro de 2009, p. 1872-1883.

ARENALES, M.; ARMENTANO, V.; MORABITO, R.; YANASSE, H. **Pesquisa Operacional**. Rio de Janeiro: Elsevier, 2007.

BANKS, J., CARSON II, J. S., NELSON, B. L., NICOL, D. M. **Discrete-Event System Simulation**. 5^a ed. Prentice Hall, 2009.

BANKS, J.; GIBSON, R. R. Seven sins of simulation practice - Practitioners be advised: avoid these mistakes or risk damage to your career. **Analytics**, p. 28-31 Summer 2009.

BANKS, J. **Handbook of Simulation: Principles, Methodology, Advances, Applications, and Practice**. New York, John Wiley and Sons, 1998.

BAPTISTE, P.; FLAMINI, M.; SOURD, F. Lagrangian bounds for just-in-time job-shop scheduling. **Computers & Operations Research**, v. 35, n. 3, p. 906-915, 2008.

BARCO, C. F.; VILLELA, F. B. Análise dos sistemas de programação e controle da produção. In: **XXVIII Encontro Nacional de Engenharia de Produção**. Rio de Janeiro, RJ, Brasil, 13 a 16 de outubro de 2008.

BARROS NETO, B.; SCARMINIO, I. S.; BRUNS, R. E. **Como fazer experimentos; Pesquisa e desenvolvimento na ciência e na indústria**, 3^a ed. Campinas, Unicamp, 2007.

BASIRI, M. E.; NEMATI, S.; AGHDAM, M. H.; GHASEM-AGHAEE, N. A novel ACO-GA hybrid algorithm for feature selection in protein function prediction. **Expert Systems with Applications**, v. 36, n.10, p. 12086-12094, 2009.

BAYKASOGLU, A.; OSBAKIR, L. Analysing the effect of flexibility on manufacturing systems performance. **Journal of Manufacturing Technology Management**, v.18, n.2, p. 172-193, 2008.

BERTRAND, J. W. M.; FRANSOO, J. C. Modelling and simulation Operations Management research methodologies quantitative modeling. **International Journal of Operations and Production Management**, v. 22, n.2, p. 241-261, 2002.

CHAN, F. T. S.; CHAN, H. K. Analysis of dynamic control strategies of an FMS under different scenarios. **Robotics and Computer-Integrated Manufacturing**, n. 20, p. 423–437, 2004.

CHAVES, A. A.; BIAJOLI, F. L.; MINE, O. M.; SOUZA, M. J. F. Metaheurísticas híbridas para resolução do problema do caixeiro viajante com coleta de prêmios. **Produção**, v. 17, n. 2, p. 263-272, 2007.

CHWIF, L.; MEDINA, A.C. **Modelagem e Simulação de Eventos Discretos: Teoria e Aplicações**. 2^a ed. São Paulo, Gráfica Palas Athena, 2007.

COELLO, C. A. C.; LAMONT, G. B.; VAN VELDHUIZEN, D. A. **Evolutionary Algorithms For Solving Multi-Objective Problems**. 2^a ed. Springer, New York, 800, 2007.

COSTA, A. C. F.; JUNGLES, A. E. O Mapeamento do Fluxo de Valor Aplicado a uma Fábrica de Montagem de Canetas Simulada. In: **XXVI Encontro Nacional de Engenharia de Produção**. Fortaleza, CE, Brasil, 09 a 11 de outubro de 2006.

COSTA, H. G.; NANCI, L. S. QUELHAS, O. L. G.; CARVALHO, R. A.; GUTIERREZ, R. H.; SIMÃO, V. G. Sistema de Produção. In: LUSTOSA, L. J.; MESQUISTA, M. A.;

QUELHAS, O.; OLIVEIRA, R. **Planejamento e Controle da Produção**. Rio de Janeiro: Elsevier, 2008.

DING, H.; BENYOUCEF, L.; XIE, X. A simulation-based multi-objective genetic algorithm approach for networked enterprises optimization, **Engineering Applications of Artificial Intelligence**, v. 19, p. 609-623, 2006.

EKREN, B. Y.; ORNEK, A. M. A simulation based experimental design to analyze factors affecting production flow time. **Simulation Modeling Practice and Theory**, n.16, p. 278-293, 2008.

FARNES, V. C. F.; PEREIRA, N. A. Balanceamento de linha de montagem com o uso de heurística e simulação: estudo de caso na Linha Branca. **GEPROS - Gestão da Produção, Operações e Sistemas**, ano 2, v. 2, p. 125-136, 2007.

FRAMINAN, J. M.; LEISTEN, R. A multi-objective iterated greedy search for flowshop scheduling with makespan and flowtime criteria. **Operation Research Spectrum**, v. 30, p. 787-804, 2008.

FREITAS FILHO, P. J. **Introdução à modelagem e simulação de sistemas**. 2^a ed. P 372, Florianópolis, Visual Books, 2008.

GAlib: Matthew's Genetic Algorithms Library. Disponível em <http://lancet.mit.edu/ga/>. Acesso em 10/11/2009.

GHINATO, P. Fundamentos do Sistema Toyota de Produção. In: ALMEIDA, A.D.; SOUZA, F. M. C. (Org.) **Produção e Competitividade: Aplicações e inovações**. UFPE, Recife, 2000.

GONÇALVES, J. F.; MENDES, J. J. M.; REZENDE, M. G. C. A genetic algorithm for the resource constrained multi-project scheduling problem. **European Journal of Operational Research**, n. 189, p. 1171-1190, 2008.

GONÇALVES, J. F.; MENDES, J. J.; REZENDE, M. G. A hybrid genetic algorithm for the job shop scheduling problem. **European Journal of Operational Research**, n. 167, p. 77-95, 2005.

GORGULHO JUNIOR, J. H. C.; GONÇALVES FILHO, E. V. Análise do desempenho dos arranjos físicos distribuídos operando sob roteamento de peças com flexibilidade de seqüenciamento. **Revista Gestão Industrial**, n. 01, v. 3, p. 01-12, 2007.

GRABOWSKI, J.; PEMPERA, J. Some local search algorithms for no-wait flow-shop problem with makespan criterion. **Computers & Operations Research**, n.32, p. 2197-2212, 2005.

GUPTA, J. N. D.; STAFFORD JR., E. F. Flowshop scheduling research after five decades. **European Journal of Operational Research**, n. 169, p. 699-711, 2006.

HENDEL, Y.; SOURD, F. Efficient neighborhood search for the one-machine earliness-tardiness scheduling problem. **European Journal of Operational Research**, n. 19; p. 173-108, 2006.

HOLLAND, J. H. **Adaptation in Natural and Artificial Systems**. The University of Michigan Press, Ann Arbor, 1975.

HOLLAND, J. H. Genetic Algorithms. **Scientific American**, p. 66-72, 1992

HUEGLER, P. A.; VASKO, F. J. Metaheuristics for meltshop scheduling in the steel industry. **Journal of the Operational Research Society**, n. 58, p. 791-796, 2007.

JANUÁRIO, T. O.; MOREIRA, M. C. O.; ARROYO, J. E. C.; SANTOS, A. G. Um algoritmo genético híbrido para o problema no-wait flowshop com makespan e atraso máximo. In: **XLI SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL - SBPO**. Porto Seguro, BA, Brasil, 01 a 04 de setembro de 2009, p. 2109-2120.

KELTON, W. D.; SADOWSKI, R., P.; SADOWSKI, D. A. **Simulation With ARENA**, 2^a ed. McGraw Hill, 2000.

KIM, J. L. Improved genetic algorithm for resource constrained scheduling of large projects. **Canadian Journal of Civil Engineering**. v. 36, p. 1016-1027, 2009.

KUMAR. N. S.; SRIDHARAN, R. Simulation modeling and analysis of tool sharing and part Scheduling decisions in single-stage multemachine flexible manufacturing systems. **Robotics and Computer-Integrated Manufacturing**, n. 23, p. 361-370, 2007.

LEITE, M.; ARROYO, J. E. C. Algoritmo busca tabu para a minimização do tempo de processamento e atrasos de entrega em sistemas de produção *flow shop* permutacional. In: **XXVI Encontro Nacional de Engenharia de Produção**. Fortaleza, CE, Brasil, 09 a 11 de outubro de 2006.

LIAN, Z.; JIAO, B.; GU, X. A similar particle swarm optimization algorithm for job-shop scheduling to minimize makespan. **Applied Mathematics and Computation**, v.183, p.1008-1017, 2006.

LINDEN, R. **Algoritmos Genéticos: Uma importante ferramenta da Inteligência Computacional**. 2^a ed. Brasport, 428 p. Rio de Janeiro, 2008.

MANIKAS, A.; CHANG, Y. Multi-criteria sequence-dependent job shop scheduling using genetic algorithms. **Computers & Industrial Engineering**, n. 56, p. 179-185, 2009.

MARTINS, M. E. G. **Introdução a Inferência Estatística**. Versão atualizada em fevereiro de 2008. Disponível em: <http://www.alea.pt/Html/statofic/html/dossier/html/introInfEstat.html>. Acessado em: 09/09/2009.

MESQUITA, M.; COSTA, H. G.; LUSTOSA, L.; SILVA, A. S. Programação detalhada da produção. In: LUSTOSA, L. J.; MESQUISTA, M. A.; QUELHAS, O.; OLIVEIRA, R. **Planejamento e Controle da Produção**. Rio de Janeiro: Elsevier, 2008.

MICHALEWICZ, Z. **Genetic Algorithms + Data Structures = Evolution Programs**. 3rd revised and extended. Berlim, Springer-Verlag, 1996.

MITCHELL, M. **An introduction to genetic algorithms**. A Bradford Book The MIT Press Cambridge, Massachusetts. London, England Fifth printing, 1999.

MITCHELL, T.M. **Machine Learning**. 16^a ed. New York: McGraw-Hill, 1997.

MONTEVECHI, J. A. B.; COSTA, R. F. S.; LEAL, F.; BATISTA, F. D.; MIRANDA, R. C.; SANTOS, A. C. O.; SANTOS, M. J. Modelagem de uma célula de manufatura complexa através de IDEFO e simulação de eventos discretos. In: **Simpósio de Pesquisa Operacional e Logística da Marinha**. Rio de Janeiro, RJ, Brasil, 07 e 08 de novembro de 2007.

MONTGOMERY, D. C. **Design and Analysis of Experiments**. 6^a ed. New York: J. Wiley, 2005.

MORANDIN JUNIOR, O.; KATO, E. R. R.; DERIZ, A. C.; SANCHES, D. S.. Uma modelagem para programação da produção de sistemas de manufatura com recursos compartilhados utilizando Algoritmos Genéticos. In: **XXVII Encontro Nacional de Engenharia de Produção**. Foz do Iguaçu, PR, Brasil, 09 a 11 de outubro de 2007.

MOREIRA, M. R. Controlo input-output em job-shops: Como as regras de decisões interagem e reagem a falhas nas máquinas. In: **XVII Jornadas Hispano-Lusas de Gestión Científica - Conocimiento, Innovación y Emprendedores: Caminho al Futuro**, p. 2257-2569, 2007.

OLIVEIRA, C. S. **Metodologia para utilização de simulação em projetos de manufatura enxuta**. Belo Horizonte, 2008, 187 p. Dissertação (Mestrado em Engenharia de Produção) Escola de Engenharia da Universidade Federal de Minas Gerais, 2008.

PACHECO, M. A. C. **Algoritmos Genéticos: Princípios e Aplicações**. 1999. ICA: Laboratório de Inteligência Computacional Aplicada - Disponível em www.ica.ele.puc-rio.br. Acessado em 15/09/2010.

PALOMINO, R. C. **Uma Abordagem para a Modelagem, Análise e Controle de Sistemas de Produção Utilizando Redes de Petri**. Florianópolis. Dissertação (Mestrado em Engenharia de Produção) Universidade Federal de Santa Catarina, 1995. Disponível em: http://www.eps.ufsc.br/disserta/palomino/capit_2/cp2a_pal.htm, Acesso em 10/11/2009

PANWALKAR, S. S. e ISKANDER, W. A survey of scheduling rules. **Operation Research**, v.1, n.25, p.45-61, 1977.

PARAGON disponível em www.paragon.com.br, acessado em 10/11/2009.

PATNAIK, A; BISWAS, S.; MAHAPATRA, S. S. An evolutionary approach to parameter optimization of submerged arc welding in the hard facing process, **Int. J. Manufacturing Research**, v. 2, n. 4, p. 462-483, 2007.

PEREIRA, F. H.; PIRES, C. L. NABETA, S. I. Optimal theoretical placement of rectifier substations on DC traction systems. **IET Electrical Systems in Transportation**. Submetido em agosto de 2010.

PERSON, A; GRIMM, H.; NG, A.; LEZAMA, T.; EKBERG, J.; FALK, S.; STABLUM, P. Simulated-based Multi-objective optimization of a real-world scheduling problem. In: **Proceedings of the 2006 Winter Simulation Conference**. Monterey, California, p. 1757-1764, December 03-06, 2006.

PITOMBEIRA NETO, A. R. Procedimento para a geração de arranjos físicos distribuídos baseados em otimização da simulação e algoritmos genéticos. In: **XXVII Encontro Nacional de Engenharia de Produção**. Foz do Iguaçu, PR, Brasil, 09 a 11 de outubro de 2007.

PONTES, H. L. J. ; YAMADA, M. C.; PORTO, A. J. V. Análise do arranjo físico de uma linha de montagem em uma empresa do setor de componentes automotivos utilizando simulação. In: 8º Congresso Ibero Americano de Engenharia Mecânica. Cuzco, Peru, 23 a 25 de outubro de 2007.

QIAN, B.; WANG, L.; HUB, R.; HUANG, D.X.; WANG, X. A DE-based approach to no-wait flow-shop scheduling. **Computers & Industrial Engineering**. n. 57, p. 787-805, 2009.

RUSSEL, S.; NORVIG, P. **Inteligência Artificial**. Rio de Janeiro. Campus. 2004.

RUIZ, R.; STUTZLE, T. An iterated greedy heuristic for the sequence dependent setup times flowshop problem with makespan and weighted tardiness objectives. **European Journal of Operational Research**, n. 187, p. 1143-1159, 2008.

SAKURADA, N.; MIYAKE, D. I. Aplicação de simuladores de eventos discretos no processo de modelagem de sistemas de operações de serviços. **Gestão e Produção**, v. 16, n. 1, p. 25-43, 2009.

SANTA CATARINA, A. **SAHGA- Um algoritmo genético híbrido com representação explícita de relacionamentos espaciais para análise de dados geoespaciais**. São José dos Campos, 2009, 112p. Tese (Doutorado do Curso de Pós-Graduação em Computação Aplicada) Instituto Nacional de Pesquisas Espaciais - INPE, 2009.

SANTA CATARINA, A; BACH, S. L. Estudo do efeito dos parâmetros genéticos sobre a solução otimizada e sobre o tempo de convergência em algoritmos genéticos com codificações binárias e reais. **Acta Scientiarum. Technology**, v. 25, n. 2, p. 147-152, Jul./Dez. 2003.

SANTORO, M. C.; MESQUITA, M. A. The effect of the workload on due-date performance in job shop scheduling. **Brazilian Journal of Operations and Production Management**, v. 5, p. 75-88, 2008.

SCHOLZ-REITER, B.; REKERSBRINK, H.; GORGES, M. Dynamic flexible flow shop problems—Scheduling heuristics vs. autonomous Control. **CIRP Annals - Manufacturing Technology**, n. 59, p.465-468, 2010.

SEAN, S. **Essentials of Metaheuristics**. Zeroth Edition Online Version 0.8 March, 2010. Disponível em <http://cs.gmu.edu/~sean/book/metaheuristics/>. Acessado em 18/03/2010.

SERAPIÃO, A., B., S. Fundamentos de otimização por inteligência de enxames: uma visão geral. **Sba Controle & Automação**, v. 20, n.3, p. 271-304, 2009.

SHEVASUTHISILP, S.; INTAWONG, S. Improvement of Scheduling Efficiency Using Heuristic Methods and Simulation Technique in Electronic Assemblies Industry Production. In: **Proceedings of the International MultiConference of Engenners and Computer Scientists**. Hong Kong, March 18-20, 2009.

SOARES, R. **Otimização de Layout Industriais Utilizando Heurística SDPI no treinamento de Redes Neurais MLP**. Belo Horizonte, 2006, 92p. Dissertação (Mestrado em Modelagem Matemática e Computacional) Centro Federal de Educação Tecnológica de Minas Gerais, 2006.

SOKOLOWSKI, J. A.; BANKS, C. M. **Principles of modeling and simulation: a multidisciplinary approach**. Ed. John Wiley and Sons, 2009.

SURESH, K.N.; SRIDHARAN, R. Simulation modeling and analysis of tool sharing and part Scheduling decisions in single-stage multimachine flexible manufacturing systems. **Robotics and Computer-Integrated Manufacturing**, n. 23, p. 361-370, 2007.

TORGA, B. L. M. **Modelagem, Simulação e Otimização em Sistemas Puxados de Manufatura**. Itajubá, 2007, 152p. Dissertação (Mestrado em Engenharia de Produção) Universidade Federal de Itajubá. Minas Gerais, 2007.

UDDIN, M. K.; SOTO, M. C.; LASTRA, J. L. M. An integrated approach to mixed-model assembly line balancing and sequencing. **Assembly Automation**, v. 30, n. 2, p. 164-172, 2010.

VARADHARAJAN, T. K.; RAJENDRAN, C. A multi-objective simulated-annealing algorithm for scheduling in flowshops to minimize the makespan and total flowtime of jobs. **European Journal of Operational Research**, n.167, p. 772-795, 2005.

VERAXC: Disponível em <http://www.veraxc.com>. Acesso em 10/11/2010.

WALL, M. **GALIB: A C++ library of genetic algorithm components**. Mechanical Engineering Departament, Massachusetts Institute of Technology, 1996.

WANG, L.; ZHANG, L.; ZHENG, D.Z. An effective hybrid genetic algorithm for flow shop scheduling with limited buffers. **Computers & Operations Research**, v. 33, p. 2960-2971, 2006.

WATANABE, M.; IDA, K.; GEN, M. A genetic algorithm with modified crossover operator and search area adaptation for the job-shop scheduling problem. **Computers & Industrial Engineering**, n. 48 p. 743-752, 2005.

XIA, W.; WU, Z. An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problems. **Computers & Industrial Engineering**, v. 48, p. 409-425, 2005.

WEISE, T. Global optimization algorithms: theory and application. Kassel: University of Kassel, 2008. 820 p. Disponível em: www.it-weise.de/projects/book.pdf. Acesso em: 20 dez. 2010.

XU, J.; ZHOU, X. A class of multi-objective expected value decision-making model with birandom coefficients and its application to flow shop scheduling problem. **Information Sciences**, n. 179, p. 2997-3017, 2009.

YAMAMOTO, L.; ARRUDA, L. V. R.; LIBERT, N. Sequenciamento de bateladas em scheduling usando otimização multiobjetivo. In:**4º Planejamento e Desenvolvimento em Petróleo e Gás - PDPETRO**. Campinas, SP, p.1-9, 21-24 de Outubro de 2007.

ANEXO A - Código VBA

Conforme antecipado, tem-se na sequência o código utilizado para acoplar o simulador e o AG. No primeiro bloco tem-se a inicialização do simulador. No segundo bloco estão a limpeza e a preparação para leitura e gravação de arquivos. No terceiro bloco tem-se a gravação dos novos valores definidos pelo AG para as prioridades. Para finalizar no quarto bloco apresenta a gravação do arquivo do simulador para o AG.

```

VERSION 1.0 CLASS
BEGIN
    MultiUse = -1 'True
END
Attribute VB_Name = "ThisDocument"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = True
Private Sub ModelLogic_RunBeginSimulation()
    Dim oModel As Arena.Model
    Dim oModule As Arena.Module
    Dim oModuleIndex As Long
    'Variables for file
    Dim sFileName As String
    Dim iFileNum As Integer
    Dim i As Integer
    Dim sBuf As String
    *****

```

Faz a limpeza do arquivo de desempenho enviado para o AG e leitura do arquivo de prioridade recebido do AG.

```

'Clear the Output File
Dim sOutputFile As String
sOutputFile = "C:/Users/Marilda/DissertacaoExe/samples_c/outputArena.txt"
iFileNum = FreeFile()
Open sOutputFile For Output As iFileNum
'close the file
Close iFileNum
*****
'Faz a leitura do arquivo
sFileName = "C:/Users/Marilda/DissertacaoExe/samples_c/outputAG.txt"
' does the file exist? simpleminded test:

```

```

If Len(Dir$(sFileName)) = 0 Then
    Exit Sub
End If
iFileNum = FreeFile()
Open sFileName For Input As iFileNum

Dim StaticArray(1 To 47) As String 'Armazena os valores lidos
i = 1
Do While Not EOF(iFileNum)

    Line Input #iFileNum, sBuf
    ' now you have the next line of the file in sBuf
    ' do something useful:
    StaticArray(i) = sBuf
    i = i + 1
Loop
' close the file
Close iFileNum
'

```

Efetua a gravação das mudanças de prioridades para cada ordem em cada máquina de acordo com arquivo recebido do AG.

```

'Aponta o modelo atual
Set oModel = ThisDocument.Model
'Muda Prioridades ordem 1
oModuleIndex = oModel.Modules.Find(smFindTag, "object.367")
Set oModule = oModel.Modules.Item(oModuleIndex)
oModule.Data("Value(1,1)") = StaticArray(1) 'Maq 1
oModule.Data("Value(1,2)") = StaticArray(8) 'Maq 2
oModule.Data("Value(1,3)") = StaticArray(15) 'Maq 3
oModule.Data("Value(1,4)") = StaticArray(32) 'Maq 6
'
'Muda Prioridades ordem 2
oModuleIndex = oModel.Modules.Find(smFindTag, "object.368")
Set oModule = oModel.Modules.Item(oModuleIndex)
oModule.Data("Value(1,1)") = StaticArray(2) 'Maq 1
oModule.Data("Value(1,2)") = StaticArray(16) 'Maq 3
oModule.Data("Value(1,3)") = StaticArray(37) 'Maq 7
oModule.Data("Value(1,4)") = StaticArray(43) 'Maq 8
'
'Muda Prioridades ordem 3
oModuleIndex = oModel.Modules.Find(smFindTag, "object.369")
Set oModule = oModel.Modules.Item(oModuleIndex)
oModule.Data("Value(1,1)") = StaticArray(3) 'Maq 1
oModule.Data("Value(1,2)") = StaticArray(26) 'Maq 5
oModule.Data("Value(1,3)") = StaticArray(44) 'Maq 8
'

```

```

'Muda Prioridades ordem 4
oModuleIndex = oModel.Modules.Find(smFindTag, "object.370")
Set oModule = oModel.Modules.Item(oModuleIndex)
oModule.Data("Value(1,1)") = StaticArray(4) 'Maq 1
oModule.Data("Value(1,2)") = StaticArray(9) 'Maq 2
oModule.Data("Value(1,3)") = StaticArray(17) 'Maq 3
oModule.Data("Value(1,4)") = StaticArray(27) 'Maq 5
oModule.Data("Value(1,5)") = StaticArray(33) 'Maq 6
oModule.Data("Value(1,6)") = StaticArray(38) 'Maq 7
oModule.Data("Value(1,7)") = StaticArray(45) 'Maq 8
'

'Muda Prioridades ordem 5
oModuleIndex = oModel.Modules.Find(smFindTag, "object.371")
Set oModule = oModel.Modules.Item(oModuleIndex)
oModule.Data("Value(1,1)") = StaticArray(18) 'Maq 3
oModule.Data("Value(1,2)") = StaticArray(21) 'Maq 4
oModule.Data("Value(1,3)") = StaticArray(34) 'Maq 6
oModule.Data("Value(1,4)") = StaticArray(39) 'Maq 7
'

'Muda Prioridades ordem 6
oModuleIndex = oModel.Modules.Find(smFindTag, "object.372")
Set oModule = oModel.Modules.Item(oModuleIndex)
oModule.Data("Value(1,1)") = StaticArray(5) 'Maq 1
oModule.Data("Value(1,2)") = StaticArray(10) 'Maq 2
oModule.Data("Value(1,3)") = StaticArray(19) 'Maq 4
oModule.Data("Value(1,4)") = StaticArray(22) 'Maq 3
oModule.Data("Value(1,5)") = StaticArray(5) 'Maq 1
oModule.Data("Value(1,6)") = StaticArray(10) 'Maq 2
oModule.Data("Value(1,7)") = StaticArray(19) 'Maq 4
oModule.Data("Value(1,8)") = StaticArray(28) 'Maq 5
oModule.Data("Value(1,9)") = StaticArray(40) 'Maq 7
'

'Muda Prioridades ordem 7
oModuleIndex = oModel.Modules.Find(smFindTag, "object.373")
Set oModule = oModel.Modules.Item(oModuleIndex)
oModule.Data("Value(1,1)") = StaticArray(11) 'Maq 2
oModule.Data("Value(1,2)") = StaticArray(35) 'Maq 6
oModule.Data("Value(1,3)") = StaticArray(29) 'Maq 5
oModule.Data("Value(1,4)") = StaticArray(35) 'Maq 6
oModule.Data("Value(1,5)") = StaticArray(46) 'Maq 8
'Muda Prioridades ordem 8
oModuleIndex = oModel.Modules.Find(smFindTag, "object.374")
Set oModule = oModel.Modules.Item(oModuleIndex)
oModule.Data("Value(1,1)") = StaticArray(6) 'Maq 1
oModule.Data("Value(1,2)") = StaticArray(12) 'Maq 2
oModule.Data("Value(1,3)") = StaticArray(23) 'Maq 4
oModule.Data("Value(1,4)") = StaticArray(30) 'Maq 5
oModule.Data("Value(1,5)") = StaticArray(41) 'Maq 7

```

```

'Muda Prioridades ordem 9
oModuleIndex = oModel.Modules.Find(smFindTag, "object.375")
Set oModule = oModel.Modules.Item(oModuleIndex)
oModule.Data("Value(1,1)") = StaticArray(13) 'Maq 2
oModule.Data("Value(1,2)") = StaticArray(20) 'Maq 3
oModule.Data("Value(1,3)") = StaticArray(24) 'Maq 4
oModule.Data("Value(1,4)") = StaticArray(36) 'Maq 6
oModule.Data("Value(1,5)") = StaticArray(42) 'Maq 7
'Muda Prioridades ordem 10
oModuleIndex = oModel.Modules.Find(smFindTag, "object.376")
Set oModule = oModel.Modules.Item(oModuleIndex)
oModule.Data("Value(1,1)") = StaticArray(7) 'Maq 1

oModule.Data("Value(1,2)") = StaticArray(14) 'Maq 2
oModule.Data("Value(1,3)") = StaticArray(25) 'Maq 4
oModule.Data("Value(1,4)") = StaticArray(31) 'Maq 5
oModule.Data("Value(1,5)") = StaticArray(47) 'Maq 8
'
End Sub

```

O simulador Arena® faz as replicações determinadas, grava o novo arquivo de desempenho e o envia para o AG.

```

Private Sub ModelLogic_RunEndReplication()
Dim oModel As Arena.Model
Dim oSIMAN As Arena.SIMAN
Dim oMSIndex As Long
Dim oTAIndex As Long
Dim oNAIndex As Long
Dim nValue1 As Integer
Dim nValue2 As Long
Dim nValue3 As Long
'Faz a escrita em arquivo
Dim sFileName As String
Dim iFileNum As Integer
Dim i As Integer
Dim sBuf As String
Dim dVal As Double

'Aponta o modelo atual
Set oModel = ThisDocument.Model
'Aponta SIMAN
Set oSIMAN = oModel.SIMAN
'Pega index "makespan"
oMSIndex = oSIMAN.SymbolNumber("Makespan")
oTAIndex = oSIMAN.SymbolNumber("Tempo Total de Atraso")
oNAIndex = oSIMAN.SymbolNumber("Numero de trabalhos atrasados")
'Pega valores
nValue1 = oSIMAN.TallyAverage(oMSIndex)

```

```
nValue2 = oSIMAN.CounterValue(oTAIndex)
nValue3 = oSIMAN.CounterValue(oNAIndex)
'Create (or open) output file
sFileName = "C:/Users/Marilda/DissertacaoExe/samples_c/outputArena.txt"

iFileNum = FreeFile()
Open sFileName For Append As iFileNum
'

'Write #iFileNum, nValue1; nValue2; nValue3
Print #iFileNum, nValue1; Spc(2); nValue2; Spc(2); nValue3
' close the file
Close iFileNum
'

End Sub
```

ANEXO B

Trabalhos publicados oriundos desta dissertação, por ordem cronológica crescente.

SILVA, M. F. S., BARBOSA, E., GONÇALVES M., CALARGE, F. A., PEREIRA, F. H. Otimização multiobjetivo de regras heurísticas de sequenciamento em sistemas de manufatura *job shop* por meio de simulação computacional e algoritmos genéticos. In: **V Coloquio dos Pesquisadores de Ciências Exatas da Uninove**, 2010.

BARBOSA, E., GONÇALVES, M., SILVA, M., PEREIRA, F. H. Evaluation of Production Sequencing Rules in Jop Shop Environment through computer simulation. In: **V Coloquio dos Pesquisadores de Ciências Exatas da Uninove**, 2010.

SILVA, M. F. S., BARBOSA, E., GONÇALVES M., CALARGE, F. A., PEREIRA, F. H. Otimização multiobjetivo de regras heurísticas de sequenciamento em sistemas de manufatura *job shop* por meio de simulação computacional e algoritmos genéticos. In: **XLII SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL- SBPO, 30/08 a 03/09, Bento Gonçalves, 2010.**

BARBOSA, E., GONÇALVES, M., SILVA, M., PEREIRA, F. H. Evaluation of Production Sequencing Rules in Jop Shop Environment through computer simulation. In: **XLII SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL- SBPO, 30/08 a 03/09, Bento Gonçalves, 2010.**

PEREIRA, F.H., BARBOSA, E., SILVA, M. F. S., COSTA, M.G Evaluation of Job Shop Scheduling Rules in Due Date Performance Optimization. In: **European Conference on Operational Research**, EURO XXIV, 2010, Lisboa.

PEREIRA, F.H., SILVA, M. F. S., CALARGE, F.A., BARBOSA, E., COSTA, M.G.A multi-objective optimization approach in a job-shop scheduling rules simulation In: **2nd International Conference on Engineering Optimization EngOpt 2010, 06 a 08/09, Lisboa. Book of Abstracts**, 2010.

PEREIRA, F.H., SILVA, M. F. S. Otimização de Parâmetros de Controle da Produção em Sistemas Flexíveis de Manufatura por meio de Simulação Computacional. In: **IV Colóquio dos Pesquisadores de Ciências Exatas da Uninove**, 2009.