

**ALGORITMO GENÉTICO-TABU
PARA A PROGRAMAÇÃO REATIVA
DA PRODUÇÃO EM UM SISTEMA
DE MANUFATURA COM RECURSOS
COMPARTILHADOS**

UNIVERSIDADE FEDERAL DE SÃO CARLOS
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**Algoritmo Genético-Tabu para a Programação Reativa da Produção em
um Sistema de Manufatura com Recursos Compartilhados**

Ageu Morais Duarte

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do Título de Mestre em Ciência da Computação.

Orientador: Prof. Dr. Orides Morandin Junior

SÃO CARLOS
2009

**Ficha catalográfica elaborada pelo DePT da
Biblioteca Comunitária da UFSCar**

D812ag

Duarte, Ageu Moraes.

Algoritmo genético-tabu para a programação reativa da produção em um sistema de manufatura com recursos compartilhados / Ageu Moraes Duarte. -- São Carlos : UFSCar, 2010.

120 f.

Dissertação (Mestrado) -- Universidade Federal de São Carlos, 2009.

1. Inteligência artificial. 2. Programação da produção. 3. Algoritmos genéticos. 4. Busca tabu. I. Título.

CDD: 006.3 (20^a)

Universidade Federal de São Carlos
Centro de Ciências Exatas e de Tecnologia
Programa de Pós-Graduação em Ciência da Computação

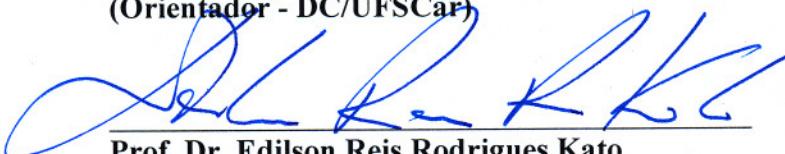
**“Algoritmo Genético – Tabu para a Programação
Reativa da Produção em um Sistema de
Manufatura com Recursos Compartilhados”**

AGEU MORAIS DUARTE

Dissertação de Mestrado apresentada ao
Programa de Pós-Graduação em Ciência da
Computação da Universidade Federal de São
Carlos, como parte dos requisitos para a
obtenção do título de Mestre em Ciência da
Computação

Membros da Banca:


Prof. Dr. Ofrides Morandin Júnior
(Orientador - DC/UFSCar)


Prof. Dr. Edilson Reis Rodrigues Kato
(DC/UFSCar)


Prof. Dr. Mario Luiz Tronco
(IBILCE/UNESP/São José do Rio Preto)

São Carlos
Novembro/2009

*Dedico este trabalho aos
meus pais Durval e Maria Luiza,
aos meus irmãos Amós e Agnê
e à minha noiva Inarcina.
Pelo apoio e pela dedicação
de todos os dias.*

“A vida é para quem topa qualquer parada, e não para quem para em qualquer topada.”
Bob Marley

“Qualquer tolo consegue escrever código que um computador entenda. Bons programadores escrevem código que humanos possam entender.”
Martin Fowler

“Dar menos que o seu melhor é sacrificar o dom que recebeu.”
Steve Prefontaine

Agradecimentos

Primeiro e, principalmente, a Deus por ter me dado tudo o que tenho.

Aos meus pais, Durval e Maria Luiza, e aos meus irmãos, Agnê e Amós, pela ajuda e conselhos em todos os momentos.

À minha noiva Inarcina, pela constante ajuda, incentivo e carinho nos momentos difíceis.

Ao meu orientador, Prof. Dr. Orides Morandin Jr., pela oportunidade oferecida, ensinamentos transmitidos, companheirismo e amizade durante todo esse tempo.

Aos meus amigos Adinovam e Flávio, pela amizade, conhecimentos compartilhados, grande ajuda durante o mestrado e, principalmente, pelo apoio nesta fase final.

Aos amigos da República São Lucas, pela amizade construída e compreensão.

Aos professores do Departamento de Computação da Universidade Federal de São Carlos, pelo apoio e disposição.

A todas as outras pessoas que de forma, direta ou indireta, contribuíram para a realização deste trabalho.

Resumo

Os sistemas de manufatura com recursos compartilhados têm sido projetados para atender às necessidades do mercado, que demanda produtos que apresentem elevado padrão de qualidade, baixo custo e garantia das condições acordadas com o cliente. A programação da produção é uma tarefa complexa, pois é de natureza combinatória. Várias pesquisas apontam para a utilização de métodos de busca, como Algoritmos Genéticos (AGs) e Buscas Tabu (BTs), aplicados ao refinamento do problema da programação da produção. Os AGs apresentam como desvantagens a convergência prematura, a falta de mecanismos de intensificação da busca nas regiões promissoras e a falta de mecanismos que mantenham a diversificação da população. Para que os AGs possam ser mais efetivos em sua busca, de forma a evitar sua convergência prematura e garantir a diversidade populacional, acrescentam-se aos AGs, mecanismos que ajustam seus parâmetros no decorrer do processo de busca, os chamados Algoritmos Genéticos Adaptativos (AGAs). Esses mecanismos, no entanto, não garantem a intensificação da busca nas regiões promissoras encontradas. A BT, por sua vez, possui mecanismo de intensificação e diversificação da busca, porém o seu tempo computacional depende do quanto ótimo é a sua solução inicial (solução pela qual se inicia o processo de busca). A fim de superar as limitações dos métodos de busca tradicionais, desenvolveram-se os Algoritmos Híbridos que agregam (ou associam) um método a outro, de forma que um auxilia o outro em sua “deficiência”. Neste trabalho, propõe-se o desenvolvimento de um Algoritmo Genético-Tabu (AGT) aplicado ao problema da programação reativa da produção em um sistema de manufatura com recursos compartilhados, a fim de garantir um bom compromisso entre valores de *makespan* e o tempo de obtenção da resposta. A BT será uma funcionalidade agregada ao AG e ao AGA, ou seja, é um procedimento que realizará o refinamento do(s) indivíduo(s) da população inicial e o refinamento do indivíduo obtido pelos métodos de busca. Realizam-se testes para definir qual método de seleção (roleta ou torneio) é mais bem aplicado ao problema, definição da estrutura da vizinhança e definição do momento de aplicação do operador de mutação. Além disso, foram também realizados testes com as diferentes formas de cálculo do *makespan*, sendo uma delas proposta neste trabalho e as demais propostas por Deriz (2007) e Sanches (2008).

Palavras-Chave: Programação Reativa da Produção, Compartilhamento de Recursos, Algoritmo Genético, Busca Tabu, Sistemas de Execução da Manufatura.

Abstract

Manufacturing systems with shared resources have been designed to meet the market needs, which require high quality products, low cost and the guarantee of the conditions agreed with the customer. The production scheduling in a manufacturing system is a complex task due to its combinatorial nature. Several studies show the use of search methods, such as Genetic Algorithms (GAs) and Tabu Search (TS), applied to the refinement of the production scheduling problem. The GAs present as disadvantages, the premature convergence, the lack of search intensification mechanisms in promising regions and also the lack of mechanisms which maintain the diversification of the population. In order to make GAs more effective in their search, by avoiding the premature convergence and ensuring the population diversity, some mechanisms are added to them so as to adjust their parameters during the search process, the so called Adaptative Genetic Algorithms (AGAs). However, these mechanisms do not guarantee the search intensification in the promising regions found. On the other hand, the TS presents mechanisms of search intensification and diversification, although its computational time depends on how optimal its initial solution is (solution by which the search process starts). In order to overcome the limitations of the traditional search methods, the Hybrid Algorithms (HAs) have been developed. They consist of the association of one method with another so that one helps the other in its “deficiencies”. This dissertation proposes the development of a Genetic-Tabu Algorithm (GTA) applied to the problem of the production reactive scheduling in a manufacturing system with shared resources, in order to ensure good compromise between makespan values and feedback time. The TS will be a functionality added to the AG and to the AGA, in other words, it will be a procedure to refine the individual(s) of the initial position and also to refine the individual obtained by the search methods. Tests have been conducted to determine which selection method (roulette or tournament) is more adequate for the definition of the neighborhood structure and also for the definition of the time to apply the mutation operator. Besides, other tests have been conducted by using different ways to calculate the makespan; one of them proposed in this dissertation and the others by Deriz (2007) and by Sanches (2008).

Keywords: Production Reactive Scheduling, Shared Resources, Genetic Algorithm, Tabu Search, Manufacturing Execution Systems.

Lista de Figuras

Figura 2-1 - Fluxograma: Algoritmo Genético, adaptado de Carvalho; Braga e Ludermir (2003)	26
Figura 2-2 - População de Cromossomos.....	27
Figura 2-3 - Roleta.....	29
Figura 2-4 - Seleção por Torneio.....	30
Figura 2-5 - Tipos de Cruzamentos: (A) um ponto, (B) dois pontos e (C) uniforme	32
Figura 2-6 - Tipos de Mutação: (A) um ponto, (B) dois pontos e (C) uniforme	33
Figura 2-7 - Fluxograma: Busca Tabu, adaptado de Glover e Laguna (1997).....	45
Figura 3-1 - Representação completa do cromossomo/solução (Morandin et. al., 2008a)	53
Figura 3-2 - Representação simplificada do cromossomo/solução (Morandin et. al., 2008a)	53
Figura 3-3 - Cromossomo/Solução, adaptado de Morandin et. al. (2008a).....	54
Figura 3-4 - Cromossomo/Solução simplificado, adaptado de Morandin et. al. (2008a).....	54
Figura 3-5 - Primeira versão do AG/AGA (V1).....	55
Figura 3-6 - Segunda versão do AG/AGA (V2).....	56
Figura 3-7 - Aplicação do ajuste das taxas dos operadores genéticos.....	57
Figura 3-8 - Cruzamento, adaptado de Sanches (2008).....	57
Figura 3-9 - Mutação, adaptado de Sanches (2008)	58
Figura 3-10 - 1 ^a Vizinhança	61
Figura 3-11 - 2 ^a Vizinhança	62
Figura 3-12 - Fluxograma: cálculo do <i>makespan</i> – 1 ^a versão	65
Figura 3-13 - Fluxograma: cálculo do <i>makespan</i> – 2 ^a versão	67
Figura 3-14 - Estrutura representativa do controle das operações sobre os produtos	69
Figura 3-15 - Estrutura representativa do controle dos produtos	69
Figura 3-16 - Estrutura representativa do controle das máquinas	70
Figura 3-17 - Estrutura representativa do controle dos AGVs	70
Figura 3-18 - Fluxograma do cálculo do makespan – 3 ^a versão.....	70
Figura 4-1 - Layout do Sistema Produtivo, adaptado de Morandin (1999).....	74
Figura 4-2 - Roteiros de Produção.....	75
Figura 4-3 - Tempo de Produção	75
Figura 4-4 - Tempo de Transporte	75
Figura 5-1- Roleta.....	82
Figura 5-2 - Torneio	82
Figura 5-3 - AG_V1	86
Figura 5-4 - AG_V2	86
Figura 5-5 - População Final (AG_V1 x AG_V2)	86
Figura 5-6 - AGA_V1	88
Figura 5-7 - AGA_V2	88
Figura 5-8 - População Final (AGA_V1 X AGA_V2)	88
Figura 5-9 - Tipos de vizinhança.....	94
Figura 5-10 - Exemplo da aplicação da vizinhança.....	94
Figura 8-1- AG_V1	117
Figura 8-2 - AG_V2	117
Figura 8-3 - População Final (AG_V1 X AG_V2)	117
Figura 8-4 - AGA_V1	119
Figura 8-5 - AGA_V2	119
Figura 8-6 – População Final (AGA_V1 X AGA_V2).....	119

Lista de Tabelas

Tabela 2-1 - População	28
Tabela 2-2 - Posição na Roleta	29
Tabela 2-3 - Principais Características da Busca Tabu, adaptado de Glover (1995)	37
Tabela 3-1- Produtos e Roteiros de Fabricação, adaptado de Morandin et. al. (2008a).....	54
Tabela 5-1 - Testes dos cálculos dos <i>makespans</i>	80
Tabela 5-2 - Roleta X Torneio.....	83
Tabela 5-3 - Aplicação dos métodos de seleção ao AG e AGA.....	83
Tabela 5-4 -AG_V1 X AG_V2	87
Tabela 5-5 - AGA_V1 X AGA_V2.....	89
Tabela 5-6 – AGs/AGAs - Resultados Obtidos (30 indivíduos)	92
Tabela 5-7 - AGs/AGAs - Resultados Obtidos (100 indivíduos).....	93
Tabela 5-8 - Exemplo de aplicação das vizinhanças	95
Tabela 5-9 - BTs: Resultados Obtidos.....	97
Tabela 5-10 -BTs: Resultados Obtidos.....	97
Tabela 5-11 - AGT: Otimização da população inicial.....	100
Tabela 5-12 - AGT: Otimização do resultado final do AGs	101
Tabela 5-13 - AGT: Otimização do resultado final do AGAs.....	102
Tabela 8-1 - AG_V1 X AG_V2	118
Tabela 8-2 AG_V1 X AG_V2.....	120

Lista de Abreviaturas e Siglas

ABNT	Associação Brasileira de Normas Técnicas
AE	Algoritmo Evolutivo
AG	Algoritmo Genético
AGA	Algoritmos Genéticos Adaptativos
AGH	Algoritmo Genético Híbrido
AGV	Veículos Auto Guiados (do inglês: “ <i>Automated Guided Vehicle</i> ”)
AH	Algoritmo Híbrido
AGT	Algoritmo Genético-Tabu
BT	Busca Tabu
C	Estação de Carga
C/D	Estação de Carga e Descarga
CE	Computação Evolutiva
D	Estação de Descarga
EE	Estratégia Evolutiva
FMS	Sistema de Manufatura Flexível (do inglês “ <i>Flexible Manufacturing System</i> ”)
M	Máquina
MES	Sistemas de Execução da Manufatura (do inglês: “ <i>Manufacturing Execution Systems</i> ”)
MKP	Do inglês: “ <i>Makespan</i> ”
NP	Do inglês, “ <i>Nondeterministic Polynomial Time</i> ”
P	Produto
PCP	Planejamento e Controle da Produção
PN	Redes de Petri (do inglês: “ <i>Petri Nets</i> ”)
RV	Veículo Aleatório (do inglês: “ <i>Random Vehicle Rule</i> ”)
STT/D	Menor Tempo de Viagem (do inglês: “ <i>Shortest Travel Time/Distance</i> ”)

Sumário

1	Introdução	14
1.1	Objetivos	18
1.1.1	Objetivos Gerais	18
1.1.2	Objetivos Específicos	18
1.2	Resultados Alcançados	19
1.3	Metodologia	20
1.4	Estrutura do trabalho	22
2	Revisão	24
2.1	Algoritmos Genéticos	24
2.1.1	Fundamentos	25
2.1.2	Parâmetros Genéticos	26
2.1.3	População	27
2.1.4	Avaliação da Aptidão	28
2.1.5	Métodos de Seleção	28
2.1.6	Mecanismo de Reprodução	30
2.1.7	Algoritmos Genéticos Adaptativos	34
2.1.8	Algoritmos Genético Híbridos	34
2.2	Busca Tabu	35
2.2.1	Fundamentos	36
2.2.2	Memória Adaptativa	38
2.2.3	Estratégias	42
2.2.4	O algoritmo	44
2.3	Aplicação dos métodos ao problema da programação da produção	45
3	Proposta	49
3.1	Caracterização e restrições do problema	50
3.2	Codificação da solução e do cromossomo	53
3.2.1	Exemplo de codificação	54
3.3	Modelagem dos AGs e AGAs	55
3.3.1	Cruzamento	57
3.3.2	Mutação	58
3.3.3	Elitismo	58
3.3.4	Seleção	58
3.3.5	Função Objetivo	59

3.4	Modelagem das BTs-----	59
3.5	Modelagem dos AGTs -----	63
3.6	Formas de cálculo do Makespan -----	64
4	Cenário de Testes e Desenvolvimento -----	74
4.1	Cenário -----	74
4.2	Descrição dos testes do sistema de validação da proposta -----	76
4.3	Descrição dos testes de validação da proposta -----	77
5	Análise e discussão dos resultados -----	79
5.1	Versões de cálculo do <i>makespan</i>-----	79
5.2	Seleção por Roleta X Seleção por Torneio -----	81
5.3	Modelagem dos AGs e AGAs -----	84
5.4	AGs X AGAs -----	90
5.5	Tipos de Vizinhança -----	94
5.6	Buscas Tabu (BTs) -----	96
5.7	Algoritmos Genético-Tabu (AGTs)-----	98
6	Conclusões e trabalhos futuros -----	103
6.1	Conclusões -----	103
6.2	Trabalhos Futuros -----	106
7	Referências Bibliográficas -----	108
8	Apêndices-----	115
8.1	Apêndice A – Métodos de Seleção (Roleta e Torneio) -----	115
8.2	Apêndice B – Resultados dos AGs e AGAs-----	117

1 Introdução

O atual cenário industrial, sujeito às conjecturas econômicas que exigem alta produtividade com elevado padrão de qualidade e baixo custo, e os avanços das técnicas e tecnologias de processos utilizadas por elas, levam a um aumento da quantidade de informações coletadas a respeito dos processos produtivos e, para aumentar a competitividade das indústrias, gerou-se a necessidade de tratar essas informações.

Para isso, surgiram os Sistemas de Execução da Manufatura (MES, do inglês *Manufacturing Execution Systems*), que integram os Sistemas de Chão de Fábrica com os sistemas corporativos. Os Sistemas de Execução da Manufatura são Sistemas de Informação integrados, ou seja, são sistemas que tornam as informações sobre o sistema de manufaturas disponíveis em tempo real, eles acompanham o processo produtivo, desde a emissão da ordem de serviço até que o produto saia da linha de produção e esteja pronto para ser entregue ou armazenado. (QUI; ZHOU, 2004)

Segundo Reddy e Rao (2006), as indústrias de manufatura têm sofrido enormes transformações nos últimos anos devido à utilização crescente de tecnologias de processos, tais como: automação de máquinas, ferramentas de controle numérico computadorizado, robôs industriais, sistemas de manipulação e transporte de materiais.

Esse contexto evidencia uma mudança no enfoque que coloca a manufatura não mais como uma simples área de suporte, mas como integrante do grupo de fatores determinantes para a competitividade. Nesse novo ambiente, a manufatura deve ser avaliada em suas dimensões estratégicas, como: custo, qualidade, serviço e flexibilidade. (MONTEVECHI; MORANDIN; MIYAGI, 2007)

Segundo SLACK *et. al.* (2002), a flexibilidade significa a capacidade de mudar a operação, ou seja, a possibilidade de alteração do que a operação faz, como faz ou quando faz. Assim sendo, um sistema de manufatura que possua certa flexibilidade deve proporcionar pelo uma das seguintes características: possibilitar a realização da produção em roteiros distintos; possibilitar que uma mesma máquina possa realizar diferentes operações sobre qualquer produto, etc.

Para fazer um melhor proveito da flexibilidade e utilizar de forma eficiente dos recursos do sistema produtivo, é necessário realizar uma boa programação da produção, que corresponde à definição da ordem e momento da alocação de recursos na sequência correta, além da alocação das máquinas e a alocação do sistema de transporte, cujo resultado é a

conclusão do conjunto de tarefas, ou seja, a finalização da produção. Geralmente aperfeiçoa-se o tempo de conclusão das tarefas em relação a alguns critérios como: *due date*, *lead time*, *makespan*, etc.

De acordo com Zhou *et. al.* (2001) e Chan *et. al.* (2006), a programação da produção é um problema de otimização combinatória, classificado como NP-Difícil. Portanto, em uma tarefa como a programação da produção, testar todas as suas possíveis combinações em um tempo hábil é uma tarefa que acarreta alto dispêndio de esforço computacional, que cresce exponencialmente com o aumento do tamanho do problema.

Os problemas considerados NP (do inglês, *Nondeterministic Polynomial Time*), são problemas onde o tempo computacional do melhor algoritmo conhecido para sua solução, cresce exponencialmente em função ao seu tamanho, e não se conhece ou não há a garantia da existência de algoritmos melhores, para a sua solução.

Uma das dificuldades encontradas na programação da produção é o uso simultâneo de máquinas e sistemas de transporte. O sistema de transporte em um Sistema de Manufatura Flexível (FMS, do inglês, *Flexible Manufacturing System*) geralmente é constituído de Veículos Auto-Guiados (AGV – do inglês *Automated Guided Vehicle*). Segundo Slack *et. al.* (2002), AGVs são veículos pequenos e autônomos e que movem materiais entre vários pontos do chão de fábrica.

Em Sistemas de Manufatura Flexíveis com recursos compartilhados, geralmente um produto possui vários roteiros e é manufaturado em máquinas distintas, não sequências, separadas geograficamente umas da outras, necessitando, assim, de um meio de transporte para movimentá-los. Nesse caso, o sistema de transporte pode ser constituído por AGVs, carregadeiras, ou outros veículos não automatizados que necessitem de um operador, etc.

Mesmo que a programação da produção seja realizada da forma mais otimizada possível, não se pode garantir que essa seja cumprida, pois o chão de fábrica está sujeito a variáveis que podem interferir no processo produtivo, atrasando a produção, ou até mesmo paralisando o sistema produtivo. Essas variáveis são conhecidas como ocorrências (eventos) não programadas, que podem ocorrer devido à quebra de máquinas, quebra do AGV, falta de matéria prima, falta de operadores, etc.

Na tentativa de minimizar os efeitos dessas ocorrências, é necessário que a fábrica seja monitorada e que as ocorrências dos eventos inesperados sejam tratadas e contornadas, sem que o sistema produtivo seja prejudicado, mantendo a produção em alta. Para isso, é realizada a reprogramação da produção, também conhecida como programação reativa da produção.

Segundo Li e Ierapetritou (2008), programação reativa da produção é a atividade responsável por monitorar a execução da programação previamente planejada, lidando com eventuais problemas em sua execução.

Os eventos não esperados trazem como consequência a parada do sistema produtivo, maior tempo de ociosidade das máquinas e até mesmo o não cumprimento da entrega dos produtos nas datas devidas, atrapalhando, assim, o relacionamento com os clientes.

A programação reativa da produção, se bem executada, pode assegurar uma boa qualidade em um curto período de tempo, melhorando a competitividade, caso encontre uma programação que gere um melhor desempenho do sistema produtivo, segundo Sun e Xue (2001), e, Tang e Wang (2008).

Na literatura são encontradas diversas heurísticas e meta-heurísticas (ou métodos de busca), que são aplicadas à programação e reprogramação da produção em sistemas de manufatura, o que indica a grande eficácia dos seguintes algoritmos de busca: Algoritmos Genéticos, Busca Tabu, *Simulated Annealing*, Busca Dispersa, entre outras, assim como também se encontram diversos Algoritmos Híbridos aplicados ao problema em questão.

Os Métodos Híbridos associam uma meta-heurística com um método de busca (sendo esse método uma heurística ou uma meta-heurística), objetivando o refinamento da solução (RENDERS; FLASSE, 1996).

Os Algoritmos Genéticos (AGs) são algoritmos de otimização global que encontram regiões promissoras no espaço de busca. Mas esses não realizam a busca exaustiva na região encontrada; muitas vezes, eles também não conseguem sair da região selecionada, devido à falta de diversidade da população. Para aumentar a diversidade da população e explorar novas regiões do espaço de busca, desenvolveram-se os Algoritmos Genéticos Adaptativos, porém não se pode ainda garantir que o indivíduo indicado seja o melhor da vizinhança e nem mesmo que os mecanismos genéticos (cruzamento e mutação) direcionem a uma região mais promissora.

A Busca Tabu (BT) é uma meta-heurística de otimização local, que possui a característica de ultrapassar os ótimos locais e se aproximar do ótimo global, e inicia seu processo de busca de uma solução inicial (região inicial selecionada) em torno da qual realiza uma busca exaustiva. Ao encontrar o ótimo local desta região a BT direciona a busca a outra região do espaço de busca, encontrando, assim, novos ótimos locais e se aproximando ou até mesmo encontrando a melhor solução, o ótimo global.

Com base nessas características, este trabalho propõe um modelo de combinação (ou hibridização) dos métodos de busca conhecidos como Algoritmo Genético e Busca Tabu, aplicados ao Problema de Programação Reativa da Produção. Isso será realizado na tentativa de suprir as desvantagens dos AGs citadas anteriormente. Para tanto, os métodos de busca foram estudados, de forma a alcançar uma melhor combinação dos AGs (tradicional e adaptativo) com a BT (estratégias). Nesse caso, a BT irá dar prosseguimento ao processo de busca dos AGs, usando como parâmetros as regiões já visitadas pelos AGs, ou seja, as boas soluções. Ela também será usada na otimização da população inicial dos mesmos.

O grupo de pesquisa do TEAR, Laboratório de Pesquisa e Inovação em Tecnologias e Estratégias de Automação, investiga aplicações em indústrias de manufatura e de processos que buscam a incorporação de questões estratégicas de produção e estratégias competitivas nos projetos. Os temas macro abordados pelo grupo são: planejamento reativo de produção, automação da planta industrial e integração entre estratégias, planejamento e chão de fábrica. Os problemas, as oportunidades e as necessidades das manufaturas são pontos de partida para as investigações do grupo, que tem como objetivo a busca por maior eficiência do sistema produtivo.

Uma das linhas de pesquisa é a Programação Reativa da Produção com técnicas de Inteligência Artificial. Nessa linha, destacam-se os trabalhos realizados por Maggio (2005) e Morandin *et. al.* (2007a), que usam uma heurística de busca aplicada a um modelo de Rede de Petri para gerar uma programação da produção. Ainda nessa linha de pesquisa, Deriz (2007), Morandin *et. al.* (2007b) e Morandin *et. al.* (2008a) propõem uma forma de modelagem de Algoritmo Genético para resolver o problema da Programação da Produção em Sistemas de Manufatura com Recursos Compartilhados, cujos critérios para medir a eficiência da programação gerada são o *makespan* e o tempo de obtenção da resposta (tempo computacional).

Também nessa mesma linha de pesquisa Morandin *et. al.* (2008b), Sanches *et. al.* (2008) propuseram um Algoritmo Genético Adaptativo aplicado ao problema da Programação Reativa da Produção em um Sistema de Manufatura com Recursos Compartilhados, levando em consideração os Sistemas de Transporte, cujos critérios para medir a eficiência da programação gerada são o *makespan* e o tempo de obtenção da resposta.

Sendo assim, procurou-se dar continuidade aos trabalhos do grupo na linha de programação reativa da produção com técnicas de inteligência artificial com a utilização de um Algoritmo Híbrido.

Para a validação do modelo, inicialmente é necessário legitimar os módulos referentes aos métodos de busca utilizados, assim como também as formas de cálculo do *makespan* utilizados no trabalho. Após a validação dos módulos, todos os casos de uso levantados são testados, avaliando o comportamento do modelo e verificando se ele atende ou não o caso de uso em questão.

1.1 Objetivos

1.1.1 Objetivos Gerais

Em função do problema abordado neste trabalho, é proposta uma Programação da Produção com Uso Simultâneo de Máquinas e Sistemas de Transporte em um Sistema de Manufatura com Recursos Compartilhados usando um Algoritmo Genético-Tabu.

Quanto ao sistema de transporte, realizar um estudo sobre o adiantamento de suas tarefas, pois os trabalhos realizados até o momento no grupo TEAR somente ativam o sistema de transporte quando a necessidade de se movimentar um produto no chão de fábrica.

Quanto ao desenvolvimento dos AGs, é estudar suas variações, e as formas de aplicação de seus mecanismos evolutivos que melhor se aplicam ao problema. Já quanto ao desenvolvimento da BT, é estudar as formas de configuração de sua vizinhança, e a codificação de seus parâmetros, e definir a forma de aplicação da BT que é mais bem empregada, ou seja, a forma de aplicação da BT, que gera bons resultados sem prejudicar o tempo computacional.

Com base no problema e no cenário de estudo associar o AGs e a BT de forma a gear o AGT, por meio do qual é possível obter um melhor resultado de *makespan*, sem que haja perda de desempenho do sistema produtivo e em um tempo computacional bem menor que os demais algoritmos.

1.1.2 Objetivos Específicos

- Propor um método de busca usando Algoritmo Genético-Tabu para a Programação da Produção em Sistemas de Manufatura com Recursos Compartilhados;
- Implementar esse método para gerar a programação da produção segundo o método proposto;
- Desenvolver um sistema de testes para validar o método proposto;

- Implementar os cálculos do *makespan* que Deriz (2007) e Sanches (2008) propuseram, fazendo uso do MATLAB®.
- Com base no trabalho de Sanches (2008), realizar o adiantamento das tarefas do sistema de transporte e, assim, desenvolver uma nova forma de cálculo do *makespan*;
- Comparar os métodos de cálculo do *makespan*.
- Comparar os métodos de seleção (Roleta X Torneio);
- Definir a forma mais adequada de desenvolvimento dos AG's e AGA's, as quais se diferenciam no momento em que se aplica o operador de mutação;
- Comparar as variantes dos AG's, AGA's e das BT's, que foram desenvolvidos no MATLAB®:
 - Não se fez uso das *toolkits* encontradas no MATLAB®.
- Desenvolver o AGT e suas variantes:
 - Comparar seus resultados com os resultados obtidos pelos demais algoritmos.
- Averiguar qual algoritmo, dentre os algoritmos desenvolvidos, é mais bem aplicado ao problema.
- Comparar os resultados do método proposto com os resultados de outras abordagens que tratam da programação da produção, sendo essas abordagens os trabalhos propostos por Deriz (2007) e Sanches (2008).

1.2 Resultados Alcançados

Neste trabalho é elaborado e testado um método/modelo para a geração do Algoritmo Genético-Tabu (AGT) aplicado ao problema da Programação Reativa da Produção em um Sistema de Manufatura com Recursos Compartilhados. O AGT corresponde a um algoritmo que faz uso de AGs e de BTs, cujo foco é realizar a Programação Simultânea de Máquinas e Sistemas de Transporte. A alocação do sistema de transporte é realizada simultaneamente ao cálculo do *makepan*, isto ocorre, pois o tempo de transporte, ou seja, a programação do sistema de transporte influência de forma significativa, o valor final do *makespan*.

Para alcançar um bom comprometimento entre o valor do *makespan* e o tempo computacional, foi estudadas formas de associação das BTs ao AGs, tanto no algoritmo

tradicional quanto no adaptativo. Assim como também se estudou algumas variações de desenvolvimento dos AGs, AGAs e BTs.

O resultado desta abordagem foi testado para uma série de problemas (quebra de máquina e falta de matéria prima), e seus resultados comparados com as abordagens propostas por DERIZ (2007) e SANCHES (2008). Outro resultado é a definição de qual algoritmo é mais bem aplicado ao problema: AG, AGA, BT clássica, BT com estratégias de diversificação ou o AGT.

- Quanto à aplicação dos AGs e AGAs ao problema, os seguintes resultados foram obtidos:
 - definição do melhor método de seleção que se aplica ao problema, dentre os utilizados por Deriz (2007) e Sanches (2008);
 - definição do momento de aplicação da mutação, aplicada aos indivíduos da população atual ou aos filhos obtidos por meio do operador de cruzamento e aos indivíduos que serão copiados para a próxima população, a fim de manter o seu tamanho.
- Quanto à aplicação das BTs ao problema se obteve os seguintes resultados: definição da(s) vizinhança(s) e das estratégias a serem utilizadas;
- Quanto à aplicação do AGT ao problema se obteve como resultados a melhor forma de aplicação da BT ao AG e ao AGA.

O desempenho de cada algoritmo (AG, AGA, BT e AGT), assim como, a definição de seus parâmetros serão descritos de forma sucinta nos próximos capítulos

1.3 Metodologia

Este trabalho foi desenvolvido utilizando aspectos de três métodos científicos, a saber: o indutivo, o hipotético-dedutivo e o dedutivo, conforme definido por Gil e Lakatos; Marconi. (*apud* SILVA; MENEZES, 2001)

O método indutivo é utilizado à medida que parte de observações de casos da realidade particular de sistemas de manufatura para desenvolver um produto aplicado para programação da produção em Sistemas de Manufatura com Recursos Compartilhados.

Utiliza-se também o método hipotético-dedutivo, pois foi criada a hipótese de que uma busca utilizando Algoritmo Genético-Tabu para a resolução do problema de Programação da Produção com uso Simultâneo de Máquinas e Sistemas de Transporte para

Ambientes de Manufatura com Recursos Compartilhados pode gerar soluções mais eficientes do que outros métodos de programação da produção. Portanto, pretende-se confirmar a hipótese e não torná-la falsa, aspecto que remete ao método dedutivo.

Segue, em ordem cronológica, uma lista das atividades que foram desenvolvidas ao longo deste trabalho para obter os resultados finais:

- **Levantamento bibliográfico**

- Foi feito o levantamento bibliográfico sobre o que está sendo desenvolvido na área acadêmica e na comercial para a solução do problema de Programação da Produção com uso Simultâneo de Máquinas e Sistemas de Transporte em Sistemas de Manufatura com Recursos Compartilhados.

- **Modelagem e a resolução do problema**

- Foram utilizados um AG, um AGA, uma BT e BT com estratégias de intensificação e diversificação, a fim de modelar o problema e posteriormente aplicar uma busca sobre este modelo na tentativa de encontrar uma boa solução para o problema proposto.

- **Sistema Desenvolvido**

- O sistema foi desenvolvido por meio do MATLAB®.
- A fim de testar a validade do código e garantir a qualidade do sistema desenvolvido, testes unitários de cada módulo foram realizados.

- **Testes**

- Os testes foram realizados em um cenário, de tamanho fixo (9×9 – máquinas x produtos), no grupo de pesquisa TEAR, do Departamento de Computação da Universidade Federal de São Carlos.
- O tamanho do sistema de transporte é sempre fixo. Neste trabalho foram utilizados três veículos (3 AVGs) para movimentar o produto, no sistema produtivo.

- **Análise dos Resultados**

- A partir dos resultados obtidos, buscou-se observar se o trabalho proposto obteve um melhor *makespan* na maioria dos casos

testados quando comparado com os resultados obtidos por Deriz (2007) e por Sanches (2008).

- Como um dos objetivos deste trabalho é dar continuidade aos trabalhos do grupo de pesquisa do TEAR, em especial os da linha de Programação Reativa da Produção, o tempo de obtenção de resposta (tempo de computação/computacional) foi também observado nos resultados, já que para uma Programação Reativa é necessário que o valor desta variável seja baixo.
- Outro ponto analisado foi o desempenho do Algoritmo Genético-Tabu, em relação às seguintes abordagens: AG, AGA, BT e BT com estratégias de intensificação e diversificação. Assim, buscou-se, com o uso dos AGs e das BTs, alcançar soluções melhores que as já encontradas para o problema.

1.4 Estrutura do trabalho

Este trabalho está dividido da seguinte forma: no segundo capítulo é apresentada uma contextualização sobre os métodos de busca utilizados, no caso, AGs (conceitos, funcionalidades, abordagem tradicional e adaptativa) e BTs (conceitos, funcionalidades, características e estratégias), para que esses possam ser modelados, desenvolvidos e utilizados no processo de busca de uma solução ideal em um tempo hábil. Também são apresentados alguns trabalhos relacionados ao tema, alguns em domínios diferentes da proposta, porém com pontos importantes a contribuir para este trabalho e outros em domínios correlatos ao da proposta.

No terceiro capítulo descreve-se um modelo de Algoritmo Híbrido e apresenta-se a estratégia utilizada para modelar a programação reativa da produção. Nesse capítulo também é abordada a construção dos módulos referentes aos métodos de busca utilizados (AGs e BT), apresentando-se as decisões tomadas no decorrer do projeto, a modelagem dos AGs e das BTs, assim como a descrição de como as BTs foi combinada com os AGs.

No quarto capítulo é definido o cenário para a aplicação do modelo e as suas características, como a fábrica, a caracterização dos produtos, as máquinas e o sistema de transporte. Além disso, o método de validação do modelo e dos algoritmos desenvolvidos no trabalho é também apresentado.

No quinto capítulo são apresentados os resultados obtidos, assim como a análise dos resultados da aplicação do modelo nos casos de uso. Finalmente, no sexto capítulo, encontram-se as conclusões e os trabalhos futuros.

Após isso, são apresentadas as referências bibliográficas utilizadas e os apêndices contendo os códigos fonte desenvolvidos em MATLAB®.

2 Revisão

2.1 Algoritmos Genéticos

Algoritmos Genéticos (AGs) são métodos heurísticos de otimização pertencentes à classe dos Algoritmos Evolutivos (AEs), também conhecidos como Computação Evolutiva (CE). Esses métodos se baseiam na ideia da adaptação e da evolução das espécies, inspiradas nos estudos de Charles Darwin, publicados em seu livro de 1859, "A Origem das Espécies por Meios da Seleção Natural ou a Preservação de Raças Favorecidas na Luta pela Vida" (do original, em inglês, *On the Origin of Species by Means of Natural Selection or the Preservation of Favoured Races in the Struggle for Life*).

Os AE's são considerados técnicas de buscas globais que operam sobre uma população de soluções candidatas e exploram áreas distintas no espaço de busca. Portanto tais métodos (ou técnicas, ou algoritmos) de busca e otimização possuem maiores chances de atingir áreas mais promissoras no espaço de busca do que os métodos tradicionais. Um dos métodos mais difundidos dentre os AEs são os AGs.

No início dos anos 50, vários pesquisadores já realizavam estudos sobre os métodos de CE baseado na ideia de que o mecanismo de evolução poderia ser utilizado em problemas de otimização. Esses estudos eram realizados de formas distintas e individuais e aplicados a apenas alguns problemas específicos, porém, nas décadas seguintes, estes estudos tornaram-se abrangentes e seus conceitos se difundiram, o que gerou varias subáreas de estudos como: Algoritmos Genéticos (AGs), propostos por Holland (1962) e Bremermann (1962); Estratégias Evolutivas (EEs), propostas por Rechenberg (1973); Programação Evolutiva (PE), proposta por Fogel (1962) e Programação Genética (PG), proposta por Koza (1992). Segundo Srinivas e Patnaik (1994b), essas subáreas possuem características em comum, porém cada modelo possui diferenças significantes em seu modo de operação.

Segundo Lacerda e Carvalho (1999), os métodos de busca e otimização tradicionais, geralmente são constituídos por: um espaço de busca, obtido de forma aleatória, no qual se encontram as possíveis soluções do problema; e por uma função que avalia e associa valores numéricos às soluções contidas no espaço de busca.

Apesar de as teorias sobre os AGs terem sido desenvolvidas no início dos anos 60, somente nos anos 80 é que a ideia foi popularizada, graças a um aluno de John Holland,

chamado David Goldberg, que obteve sucesso na geração da primeira aplicação industrial com AG's e na publicação de seu livro em 1989.

Segundo Mitchell (1997), os AGs desde então vêm sendo utilizados com sucesso na solução de problemas de otimização, aprendizado de máquina, desenvolvimento de estratégias e fórmulas matemáticas, análise de modelos econômicos, problemas de engenharia e diversas aplicações na Biologia como: simulação de bactérias, sistemas imunológicos, ecossistemas, descoberta de formato e propriedades de moléculas orgânicas.

Segundo Goldberg (1989), os AGs diferem dos métodos tradicionais de busca e otimização nos seguintes aspectos:

- trabalham com uma codificação do conjunto de parâmetros e não com os próprios parâmetros;
- trabalham com uma população e não com um único ponto;
- utilizam informações de custo ou recompensa e não informações derivadas ou outro conhecimento auxiliar;
- utilizam regras de transição probabilísticas e não determinísticas.

Os AGs são algoritmos de busca (busca global), porém não são métodos de busca puramente aleatórios, pois combinam elementos de métodos de busca diretos e estocásticos. São algoritmos probabilísticos baseados na teoria de seleção natural e da genética (hereditariedade), ou seja, dada uma população, os indivíduos com características genéticas melhores, ditos aptos, irão sobreviver, enquanto indivíduos menos aptos tendem a desaparecer. A cada nova geração, novos indivíduos são gerados, os quais substituem os que não sobrevivem.

2.1.1 Fundamentos

Segundo Mitchell (1997), os AGs operam por meio de evoluções e, a cada iteração (geração), a população é atualizada. Essa é formada por um conjunto aleatório de indivíduos que podem ser vistos como possíveis soluções do problema.

Nos AGs, a cada iteração (ou geração), isto é, no decorrer do processo evolutivo, todo indivíduo (cromossomo) pertencente à população é avaliado pela função de aptidão que mensura seu grau de adaptação a determinado ambiente. Após está avaliação são aplicados os mecanismos de reprodução assexuada e sexuada (seleção e cruzamento) e o mecanismo de variabilidade genética (mutação), gerando, assim, os descendentes da próxima geração e simulando o processo natural de evolução descrito por Darwin. Esses mecanismos objetivam

a exploração do espaço de busca (seleção de regiões promissoras) e a obtenção de uma solução com alto grau de adequação. Esses mecanismos são aplicados até que uma solução satisfatória seja encontrada ou algum critério de parada seja satisfeito. A Figura 2-1 apresenta um fluxo básico dos AG's.

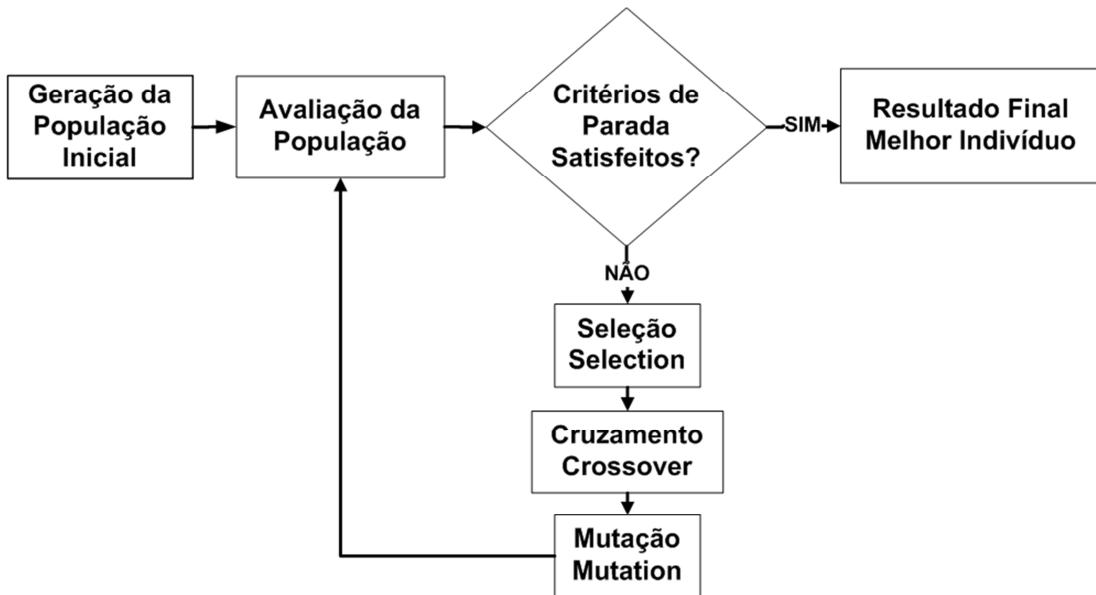


Figura 2-1 - Fluxograma: Algoritmo Genético, adaptado de Carvalho; Braga e Ludermir (2003)

2.1.2 Parâmetros Genéticos

O desempenho de um AG é fortemente influenciado pela definição dos parâmetros a serem utilizados. Segundo Srinivas e Patnaik (1994b), a escolha dos parâmetros pode tornar-se um problema complexo de otimização e, além disso, a escolha desses parâmetros está relacionada à natureza da função objetivo.

A escolha inadequada dos valores desses parâmetros pode causar uma série de problemas como a convergência prematura. Portanto, é importante analisar como os parâmetros podem ser utilizados diante das necessidades do problema e dos recursos disponíveis. A seguir serão apresentados alguns critérios para a escolha dos principais parâmetros:

- **Tamanho da População:** O tamanho da população afeta o desempenho e a eficiência dos AGs. Em uma população pequena o AG é mais rápido, mas seu desempenho pode cair, pois fornece uma pequena cobertura do espaço de busca. Uma população grande geralmente fornece uma cobertura representativa do espaço de busca, no entanto, para se trabalhar com grandes populações, são necessários maiores recursos computacionais ou que o algoritmo trabalhe por um maior período.

- **Taxa de Cruzamento:** Quanto maior a taxa de cruzamento, mais rapidamente novos atributos são introduzidos na população, mas pode-se perder indivíduos mais aptos. Caso essa taxa seja muito pequena, a busca pode se tornar muito lenta.
- **Taxa de Mutação:** Uma baixa taxa de mutação previne que a busca fique estagnada em subregiões do espaço de busca. Caso essa taxa seja muito alta, a busca se torna aleatória.
- **Critério de Parada:** Diferentes critérios podem ser utilizados para terminar a execução de um AG. Por exemplo, após um dado número de gerações ser atingido, quando não há melhora após certo número de iterações, quando a aptidão média do melhor indivíduo parar de melhorar ou quando o desvio padrão da população é inferior a certo valor definido. Assim evidencia-se e caracteriza-se uma homogeneidade da população.

2.1.3 População

Nos AGs a população representa as possíveis soluções do problema e é formada por indivíduos, ou seja, cromossomos. Cada gene do cromossomo está associado a um atributo da solução. Cada gene possui alelos, ou seja, possíveis valores que o gene pode assumir. A população (ou cada indivíduo) pode ser gerada de uma forma aleatória, por meio de heurísticas construtivas ou ainda de forma mista. Em problemas com restrições deve-se tomar cuidado para não gerar indivíduos inválidos. Na Figura 2-2 é exemplificada uma população com codificação binária.

	Indivíduos				
	CRM-01	CRM-02	CRM-03	CRM-04	CRM-05
CRM-01	1	0	1	0	1
CRM-02	0	1	1	0	0
CRM-03	1	1	0	1	0
CRM-04	1	0	1	1	1
CRM-05	0	0	0	1	1

Figura 2-2 - População de Cromossomos

Nos cromossomos (ou indivíduos) está contida a codificação (genótipo) de uma possível solução do problema (fenótipo). Os cromossomos são constituídos por genes, os quais são definidos pelos alelos - características dos problemas, ou seja, são eles que definem os valores de cada gene.

Os alelos podem ser constituídos por números binários, números inteiros, números de pontos flutuantes ou estruturas complexas, dependendo do genótipo e fenótipo a serem

representados. A definição dos alelos (codificação cromossômica) deve ser realizada de forma simples, que represente o problema como um todo, pois esta definição influenciará diretamente na estruturação e codificação dos mecanismos de reprodução, que podem gerar soluções inválidas.

2.1.4 Avaliação da Aptidão

A terminologia Função Objetivo é muito comum em problemas de otimização, mas, em se tratando de AGs, ela passa a se chamar Função de Aptidão ou Função de Fitness, é responsável por mensurar o grau de adaptação de um indivíduo em um determinado ambiente.

Esta função realiza no AG o mesmo papel que o meio ambiente teria em relação aos organismos vivos. Ou seja, como se ela definisse quem é o "rei", o "chefe", o indivíduo mais "forte" e assim por diante. Portanto, ela simula e demonstra uma avaliação do indivíduo perante o meio, define e mede sua qualidade, seu valor e seu lugar perante um conjunto de indivíduos.

2.1.5 Métodos de Seleção

Este mecanismo simula o processo de reprodução assexuada e é aplicado após o cálculo da aptidão de todos os indivíduos da população, pois este mecanismo proporciona maior prioridade aos indivíduos mais aptos. Aos indivíduos selecionados são aplicados o mecanismo de reprodução sexuada e o mecanismo de variabilidade genética. Existem várias formas de selecionar os indivíduos aptos.

Porém, independentemente de como o mecanismo de seleção é modelado, ele deve garantir que os indivíduos mais aptos sobreviverão sempre. Os mais utilizados são os métodos da roleta, do torneio e da amostragem universal estocástica.

Tabela 2-1 - População

Indivíduo	Adequação	Posição na Roleta	Faixa na Roleta
CRM-01	13	13°	0° – 13°
CRM-02	37	36°	13° – 49°
CRM-03	65	64°	50° – 103°
CRM-04	12	12°	104° – 125°
CRM-05	73	72°	126° – 197°
CRM-06	45	44°	198° – 242°
CRM-07	24	24°	243° – 265°
CRM-08	15	15°	266° – 280°
CRM-09	27	27°	281° – 307°
CRM-10	54	53°	308° – 360°

A Tabela 2-1 apresenta a relação dos indivíduos de uma determinada população com a função de adequação (*fitness*) para um problema de maximização, assim como as informações que são utilizadas para a exemplificação dos métodos de seleção apresentados a seguir.

2.1.5.1 Roleta

Neste método os indivíduos são selecionados para a próxima geração (ou população intermediária) por meio de uma roleta, a qual é acionada várias vezes até que todos os indivíduos da próxima geração sejam selecionados. Segundo Holland (1975), neste método a probabilidade de um indivíduo ser selecionado é diretamente proporcional ao seu valor de aptidão e essa probabilidade é dada por (1).

(1)

Neste método, cada indivíduo recebe uma fatia proporcional ao seu índice de adequação, ou seja, os indivíduos mais aptos possuem maiores chances de serem selecionados. A fatia (probabilidade de seleção) de cada indivíduo pode ser definida de forma aleatória ou por meio da equação (1), onde: F_i representa a fatia do indivíduo na roleta ; é o valor da função de aptidão do indivíduo ; é a soma de todas as aptidões dos indivíduos dessa população.

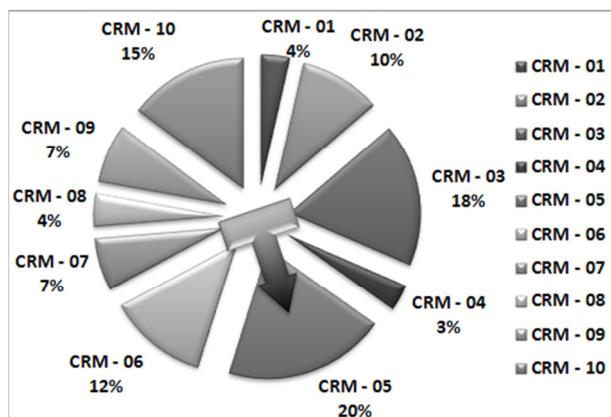


Figura 2-3 - Roleta

Tabela 2-2 - Posição na Roleta

Números Aleatórios (Grau)	Indivíduos
171°	CRM – 05
259°	CRM – 07
90°	CRM – 03
230°	CRM – 06
162°	CRM – 05
360°	CRM – 10

A Figura 2-3 ilustra a posição e a fatia pertencente a cada indivíduo na roleta, na qual os indivíduos estão dispostos no sentido horário. Para montar a roleta da Figura 2-3 foram utilizadas as informações da Tabela 2-1.

A Tabela 2-2 relaciona os indivíduos selecionados pela roleta. Para realizar a seleção de cada indivíduo são gerados números aleatórios entre “0” e “360”. A seguir, verifica-se à qual faixa da roleta o número sorteado pertence e, então, seleciona-se o indivíduo (pertencente àquela faixa) e o copia para a próxima geração.

Segundo Carvalho, Braga e Ludermir (2003), os indivíduos com maior fatia na roleta possuem maior chance de serem selecionados. A roleta não funciona com aptidões negativas e pode gerar problemas, como a convergência prematura.

2.1.5.2 Torneio

Neste método um número de indivíduos da população é escolhido de modo aleatório e com a mesma probabilidade. O indivíduo com maior aptidão é selecionado para a próxima geração. Esse processo se repete até que a população intermediária seja preenchida. Esse método de seleção tem como objetivo destacar os melhores indivíduos de uma população na esperança de se obter descendentes mais aptos que seus antecessores. Esse método é ilustrado na Figura 2-4, onde se encontra a seleção de dois (2) indivíduos (CRM-10 e CRM-05).

CRM-01	13	CRM-05	73
CRM-04	12	CRM-04	12
CRM-07	24	CRM-01	13
CRM-10	54	CRM-10	54

Figura 2-4 - Seleção por Torneio

Segundo Carvalho; Braga e Ludermir (2003), geralmente são selecionados apenas três (3) indivíduos para o torneio e, a partir desses, seleciona-se o que apresentar melhor aptidão. Este processo é repetido até que se forme a população da próxima geração.

2.1.6 Mecanismo de Reprodução

Os mecanismos reprodução são responsáveis por, gerar uma nova geração de indivíduos (nova população), por meio, da simulação dos mecanismos de reprodução biológicos. Os mecanismos frequentemente utilizados são: elitismo, cruzamento e mutação, que serão descritos nas subseções a seguir. Outro mecanismo utilizado, é a cópia ou a geração

de clone dos indivíduos, mas somente para que garantir que a nova população tenha o mesmo número de indivíduos que a população atual.

2.1.6.1 Elitismo

Segundo Lacerda e Carvalho (1999), elitismo é o mecanismo utilizado para prevenir que os melhores indivíduos não desapareçam da população pela manipulação dos mecanismos genéticos. Isso é necessário para garantir que as características genéticas do(s) melhor(es) indivíduo(s) não sejam alteradas. É interessante transferir o(s) melhor(es) indivíduo(s) de uma geração para a outra sem alterações, pois, assim, pode-se aumentar o desempenho do AG. No entanto, existe o perigo de uma convergência prematura para mínimos locais. Por isso, não se deve transferir muitos indivíduos de uma população para outra. A transferência de indivíduos deve ser realizada de maneira ponderada, de forma que não prejudique ou influencie o AG em seu processo de busca.

2.1.6.2 Cruzamento

Este mecanismo que simula o processo de reprodução sexuada é aplicado após o mecanismo de reprodução assexuada (Seleção), sendo conhecido como recombinação. Na biologia, é o fenômeno que ocorre durante a meiose, onde os cromossomos homólogos se pareiam antes de se segregarem para gametas diferentes. Basicamente, uma parte do cromossomo pode ser trocada por outra parte do cromossomo homólogo, fazendo com que alelos, que antes faziam parte do mesmo cromossomo, passem a ficar em cromossomos separados.

O cruzamento direciona o processo de busca a regiões que ainda não foram visitadas. Ele é responsável pela recombinação das características dos pais durante. E é considerado o mecanismo predominante e, por isso, é aplicado com alta taxa de probabilidade de ocorrência. Os valores da taxa de cruzamento, usualmente utilizados variam de $0,6 \leq T_c \leq 0,99$. (CARVALHO; BRAGA; LUDELMIR, 2003)

Existem várias formas de modelar e simular o cruzamento em um AG. Abaixo encontram-se relacionadas as formas mais usadas. A Figura 2-5 ilustra alguns tipos de cruzamento:

- **Cruzamento de Um Ponto:** Um ponto é escolhido aleatoriamente e os pais são selecionados nesse ponto. Assim, o segundo seguimento de ambos

os pais são trocados, gerando dois novos indivíduos. Esse mecanismo é ilustrado na Figura 2-5(A).

- **Cruzamento de Múltiplos Pontos:** Este operador escolhe "K" pontos de cruzamento e divide os cromossomos pais em "K + 1" partes. Essas partes são trocadas entre os cromossomos pais gerando, assim, os filhos, como ilustrado na Figura 2-5(B).
- **Cruzamento Uniforme:** Gera-se uma máscara para cada par de cromossomos pais ou uma única máscara para todos os pares a serem cruzados. Esta máscara define quais genes do pai o cromossomo filho herdará, como ilustrado na Figura 2-5(C). O valor “1”, na máscara, indica que o gene correspondente ao 1º pai será herdado pelo 1º filho e o gene correspondente ao 2º pai será herdado pelo 2º filho. Para um valor igual a “0” na máscara, ocorre o inverso.

The diagram illustrates three types of crossover:

- A**: One-point crossover. Shows two parents (Pais) with chromosomes [1, 0, 0] and [1, 0, 1]. They produce two offspring (Filhos) with chromosomes [1, 0, 1] and [1, 1, 0].
- B**: Two-point crossover. Shows two parents (Pais) with chromosomes [1, 0, 0] and [1, 0, 1]. They produce two offspring (Filhos) with chromosomes [1, 0, 1] and [1, 1, 0].
- C**: Uniform crossover. Shows a mask (Máscara) with values [0, 1, 0, 1, 0, 1]. It is applied to two parents (Pais) with chromosomes [1, 0, 0] and [0, 1, 1]. The resulting offspring (Filhos) have chromosomes [0, 0, 1, 1, 1, 0] and [1, 1, 0, 1, 0, 0].

	Pais					
A	[1, 0, 0]	[1, 0, 1]		[1, 0, 1]	[1, 1, 0]	
	[1, 0, 0]	[1, 0, 1]		[1, 0, 1]	[1, 1, 0]	
	Filhos					
B	[1, 0, 0]	[1, 0, 1]		[1, 0, 1]	[1, 1, 0]	
	[1, 0, 0]	[1, 0, 1]		[1, 0, 1]	[1, 1, 0]	
	Pais					
C	[1, 0, 0]	[0, 1, 1]		[1, 0, 0]	[0, 1, 1]	
	[1, 0, 0]	[0, 1, 1]		[1, 0, 0]	[0, 1, 1]	
	Filhos					
Máscara	[0, 1, 0, 1, 0, 1]					
	[0, 1, 0, 1, 0, 1]					

Figura 2-5 - Tipos de Cruzamentos: (A) um ponto, (B) dois pontos e (C) uniforme

2.1.6.3 Mutação

É o mecanismo de variabilidade genética e deve ser aplicado após o mecanismo de reprodução assexuada (Seleção) e após o mecanismo de reprodução sexuada (Cruzamento). Na biologia a mutação é frequentemente definida como mudanças na

sequência de pares de base de um gene, mas esse termo também é usado de uma forma mais ampla, que inclui mudanças no número cromossômico ou na estrutura dos cromossomos.

A mutação consiste em alterar aleatoriamente uma ou várias partes dos genes de cada indivíduo. Ela é responsável pela introdução e manutenção da diversidade genética e é considerado um mecanismo secundário e, por isso, é aplicado com baixa taxa de probabilidade de ocorrência. Esta probabilidade é dada pela taxa de mutação, que deve ser menor que a taxa de cruzamento. Os valores usualmente utilizados para a taxa de mutação variam de $0,001 \leq T_m \leq 0,01$. (CARVALHO; BRAGA; LUDERMIR, 2003)

Existem várias formas de modelar e simular mutação em um AG, como as relacionadas a seguir, que são as mais usadas. A Figura 2-6 ilustra alguns tipos de mutação:

- **Mutação de Um Ponto:** Escolhe-se um dos cromossomos da população e de forma aleatória seleciona-se um de seus genes. O gene a sofrer mutação também pode ser selecionado de acordo com alguma função probabilística. A Figura 2-6(A) apresenta este tipo de mutação.
- **Mutação de Múltiplos Pontos:** Selecionam-se vários genes do cromossomo, os quais sofrerão a mutação, ou seja, a troca de seus genes, como ilustra a Figura 2-6(B).
- **Mutação Uniforme:** Gera-se uma máscara para cada cromossomo que sofrerá a mutação ou uma única máscara para o conjunto de indivíduos. Essa máscara define quais genes do cromossomo sofrerão alterações. Quando na máscara ocorrer o valor “1”, significa que o indivíduo terá o seu gene mudado nessa posição, caso o valor do gene da máscara seja “0”, ocorre o inverso. Este tipo de mutação é apresentado na Figura 2-6(C).

A	Original	1	0	0	1	0	1
	Mutação	1	0	1	1	0	1
B	Original	1	0	0	1	0	1
	Mutação	1	1	0	1	1	0
C	Máscara	0	1	0	1	0	1
	Original	1	0	0	1	0	1
	Mutação	1	1	0	0	0	1

Figura 2-6 - Tipos de Mutação: (A) um ponto, (B) dois pontos e (C) uniforme

2.1.7 Algoritmos Genéticos Adaptativos

Os AGs Tradicionais exigem a definição de diversos parâmetros, os quais permanecem fixos durante toda a execução do algoritmo. Também exercem grande influência na capacidade do algoritmo explorar eficientemente o espaço de busca. A escolha de valores inadequados para esses parâmetros pode causar uma série de problemas, como a convergência prematura, causada pela perda da diversidade populacional, que faz com que o algoritmo convirja prematuramente para pontos de máximo ou mínimo local.

Os parâmetros permanentemente fixos podem causar diferença de desempenho, pois alguns valores podem ser eficientes para um problema específico, porém não para outros. Encontrar os valores apropriados para esses parâmetros é uma tarefa computacionalmente custosa, pois não há uma metodologia eficiente para a definição dos mesmos. A mais utilizada, no entanto, é a de tentativa e erro.

Uma tentativa de melhorar a forma que o algoritmo explora o espaço de busca se dá pelo ajuste dos valores de seus parâmetros em tempo de execução. Para isto desenvolveu-se os AGAs, que possuem a capacidade de alterar os valores iniciais de seus parâmetros durante seu processo busca. Esta alteração é realizada, com base no atual estado da população em termos de diversidade genética. Assim, não se alteram arbitrariamente esses valores, garantindo que os algoritmos encontrem uma solução de qualidade e também sejam eficientes (tempo de processamento é uma das formas de se medir eficiência) na busca dessa solução.

Na literatura são encontrados vários trabalhos sobre ajuste dinâmico dos parâmetros dos AGs. O principal motivo para esse ajuste é a convergência prematura. Nesse contexto destacam-se os trabalhos de Srinivas e Patnaik (1994a) e Jerald *et. al.* (2006), nos quais os autores, por meio do ajuste dinâmico dos parâmetros do AG, que assim realizam uma busca mais profunda nas regiões de busca, por ele apontadas, e evitam a heterogeneidade da população.

2.1.8 Algoritmos Genético Híbridos

O hibridismo nos AGs surgiu para suprir a deficiência de não possuírem mecanismos efetivos de intensificação de busca, o que contribui para a sua convergência prematura, onde o AG estabiliza-se em um determinado ponto das regiões de busca e os melhores indivíduos não são mais encontrados e nem se faz o refinamento apropriado da região apontada.

Segundo Zhou *et. al.* (2001), o AG é eficaz em várias situações incluindo a programação e o sequenciamento da produção. Apesar de eficaz, não é considerado satisfatório, pois não abrange todas as regiões de busca, e sim algumas regiões aleatórias. Davis (1991) afirma que, embora sejam robustos, os AGs geralmente não são os algoritmos mais bem sucedidos. Segundo Whitley (1994), Davis argumenta que os AGHs são mais bem sucedidos. Segundo Sinha e Goldberg (2003), uma forma de tornar o AG mais robusto é combiná-lo com outros métodos de busca.

Nos AGs, a cada iteração, a população tende a se tornar homogênea, com perda significativa de informações sobre o espaço de busca como um todo. Além disso, a cada geração, o melhor indivíduo da população pode possuir uma solução próxima melhor, o que torna necessária a incorporação de mecanismos que proporcionem a heterogeneidade da população e de mecanismos que intensifiquem o processo de busca, de forma a encontrar um resultado melhor do que o já encontrado pelo AG.

Várias heurísticas vêm sendo usadas no processo evolutivo dos AGs, como na geração inicial dos indivíduos; cruzamentos mais inteligentes, os quais mesclam nos descendentes as melhores partes dos pais; incorporação técnicas de busca local ou ferramentas de outras heurísticas ou meta-heurísticas. Esses AGs Híbridos combinam, de forma equilibrada, o caráter genérico e simplista dos AGs convencionais com técnicas especializadas e sofisticadas.

2.2 Busca Tabu

Segundo Glover (1995), os métodos clássicos de busca frequentemente encontram grandes dificuldades para resolver problemas de otimização do mundo real (negócio, engenharia, economia e ciência), pois essas aplicações requerem uma esperança razoável de sucesso, dentro de horizontes de tempo práticos e viáveis.

Para suprir essa necessidade Glover em 1986, propôs o método de busca conhecido como Busca Tabu (do inglês, *Tabu Search*), que é uma técnica adaptativa, que utiliza estrutura de memória para guiar métodos de busca local na contínua exploração do espaço de busca. Em seu processo de exploração a BT realiza movimentos sucessivos, que levam o processo de busca de uma solução para outra, a nova solução é o melhor vizinho contido no conjunto de vizinhança pertencente à solução corrente. A seguir o processo de busca da BT é melhor explicado.

2.2.1 Fundamentos

A Busca Tabu (BT) é definida como um método inteligente probabilístico de programação matemática pertencente à classe de métodos de busca local. Possui uma memória que armazena as soluções elites visitadas, assim como os movimentos (operações sobre a vizinhança) que caracterizaram a geração das soluções se tornam tabus, permanecendo inacessíveis durante um determinado prazo (prazo tabu). A condição tabu pode ser revogada caso algum movimento atenda a algum critério de aspiração. Isso é o movimento tabu pode ser realizado na tentativa de escapar dos ótimos. A condição tabu de um movimento serve para evitar a ciclagem, ou seja, o retorno a soluções já visitadas.

Segundo Glover (1997), a BT é uma meta-heurística que orienta um procedimento heurístico de busca local, que explora as soluções e vai além de seu ótimo local. A palavra Tabu (ou *Taboo*) originou-se do Tongan, idioma da Polinésia, onde era usado pelos aborígenes da ilha de Tonga. A palavra tabu possui os seguintes significados:

- Objetos ou coisas que não podem ser tocados, pois são sagrados.
- De acordo com o Dicionário Webster's, significa:
 - Uma proibição imposta por costumes sociais, proporcionando proteção;
 - Proibir por constituir um risco. Esta definição é a que mais está de acordo com o termo "Busca Tabu".
 - O risco significa uma sequência a ser evitada, pois leva a um curso pouco produtivo, ou seja, leva a uma situação da qual não se pode fugir.
- Segundo o Dicionário Aurélio, é a proibição ou restrição de natureza ritual e religiosa que determina que certos objetos, indivíduos, lugares ou atos sejam evitados, por serem considerados sagrados ou especialmente impuros e perigosos e, como instituição social, está associada a fortes sanções e à crença de que sua violação traz castigo o sobrenatural.

No contexto social, as proibições são ultrapassadas quando a ocasião exigir. Os "tabus" da BT são anulados quando o tabu proporcionar/gerar um vizinho mais atraente. Esses tabus são transmitidos por meio de uma memória social que está sujeita a modificações. Modifica-se o estado dos elementos proibidos com a evolução da memória, que ocorre de acordo com o tempo e as circunstâncias. (GLOVER, 1995)

Na BT existem dois grandes e importantes componentes que são a estratégia de intensificação e a estratégia de diversificação. Na estratégia de intensificação regressa-se a regiões mais atraentes a fim de pesquisá-las mais profundamente. Para tanto, as soluções elite devem ser armazenadas E sua pesquisa se concentra no exame dos vizinhos dessas soluções. A diversificação incentiva a pesquisa em outras regiões, visitando assim soluções divergentes.

A BT possui aplicações em planejamento de recursos, análise financeira, programação, sequenciamento, logística, classificação de padrões, indústria, entre outros problemas de otimização. Segundo Glover (1995), a BT pode ser aplicada a problemas como: programação da produção, telecomunicações, definição de rotas, otimização de grafos e problemas de otimização combinatória em geral. Ela pode ser classificada como inteligente, pois incorpora a memória adaptativa (lembança seletiva dos movimentos chave), que proporciona uma pesquisa econômica (espaço mínimo de memória) e efetiva (exploração de uma grande parte do espaço de busca), além da exploração responsiva (habilidade de realizar escolhas estratégicas), que explora as boas soluções enquanto procura espaços de busca que sejam mais promissores. Esses elementos básicos da BT possuem várias características importantes e encontram-se resumidos na Tabela 2-3. A BT tenta encontrar formas novas e mais eficazes para tirar proveito desses conceitos, além de também identificar e associar princípios que possam ampliar sua base de pesquisa. Com isso, ocorre a mistura de estratégias e novas ideias emergem e conduzem a melhores soluções.

Tabela 2-3 - Principais Características da Busca Tabu, adaptado de Glover (1995)

Memória Adaptativa
Seletividade (incluindo estratégias esquecidas)
Abstração e Decomposição (por memória explícita e atributiva)
Tempo/Período:
eventos recentes
eventos frequentes
diferenciação entre curto e longo prazo
Qualidade e Impacto:
alternativas de escolhas com grande atratividade
mudanças na estrutura ou nas relações restritivas
Contexto:
interdependência regional
interdependência estrutural
interdependência sequencial
Exploração Responsiva
Restrições estrategicamente impostas e induções. (condições tabu e nível de aspiração)
Grande foco em boas regiões e em características de boas soluções. (intensificação)
Caracterização e exploração de novas regiões promissoras. (diversificação)
Procuram padrões não repetitivos. (oscilação estratégica (movimentos estratégicos))
Soluções de integração e extensão. (reconexão de caminhos)

A BT vai além da busca local, pois no decorrer do seu processo de busca, emprega estratégias de modificações na vizinhança. Uma delas é a estrutura de memória, que serve para determinar a nova vizinhança e, consequentemente, organizar o espaço de busca da melhor forma. Outra é a definição dos movimentos de geração da vizinhança, os quais correspondem às operações sobre os elementos da solução.

Frequentemente, o termo vizinhança ocorre somente em métodos que transformam progressivamente uma solução em outra. Vizinhança é uma região no espaço onde a BT realiza uma pesquisa local para gerar uma nova solução e que essa seja melhor que as já visitadas. (GLOVER, 1996)

Segundo Glover e Laguna (1997), o conceito de vizinhança em BT é diferente do conceito empregado em busca local, pois a BT adota os tipos de movimentos usados em métodos heurísticos construtivos e destrutivos, ou seja, seus movimentos frequentemente correspondem à inserção de um elemento na solução ou à troca da posição entre dois elementos, etc.

2.2.2 Memória Adaptativa

É o principal elemento da BT e é basicamente dividida em memória de curto prazo e memória de longo prazo, que podem ser definidas como:

- **Explícita:** registra na memória todas as soluções elas visitadas durante o processo de busca. Guarda soluções completas, que geralmente consistem nas melhores soluções encontradas no decorrer da busca.
- **Atributiva:** registra os atributos das soluções, ou seja, os movimentos que levam de uma solução a outra. Guarda informações sobre atributos que mudam de uma solução para outra.

Segundo Glover e Laguna (1997), uma distinção importante em BT é a diferenciação entre memória de curto prazo e memória de longo prazo. Cada tipo de memória é acompanhado por suas próprias estratégias. Geralmente, a memória de curto prazo mantém informações das soluções visitadas recentemente. Em algumas aplicações, a BT usando memória de curto prazo é suficiente para produzir soluções de ótima qualidade. Porém a BT é significativamente mais eficaz quando se faz uso de uma memória de longo prazo. Outra distinção importante é a definição do tipo de estrutura de memória a ser utilizada.

A estrutura de memória é utilizada para evitar a revisita das soluções (prevenir ciclos) e auxilia na exploração das regiões que ainda não foram visitadas. Com a utilização da

memória, a BT “aprende” com as experiências (soluções) passadas, o que melhora o processo de seleção das regiões a serem visitadas.

A memória armazena informações sobre os atributos das soluções, os quais podem ser os movimentos que levam de uma solução a outra, soluções completas, pontuações das soluções, posições de elementos na solução, ou seja, toda informação relevante para a construção da vizinhança e para o processo de busca.

2.2.2.1 Estrutura de Memória

A definição da estrutura da memória a ser utilizada é um passo de grande importância, pois nela será armazenado o histórico da busca a partir da qual o processo será guiado de forma adaptativa, prevenindo os pontos de atração dos ótimos locais ou a realização de ciclos (ciclagem – retornar a soluções já visitadas).

Segundo Glover (1997), dentre os quatro tipos principais de estrutura de memória, os mais comuns são recentidade e frequência. Os tipos de estrutura de memória são apresentados a seguir:

- **Recentidade (do inglês, *recency*):** armazena a solução e atributos visitados há pouco tempo;
- **Frequência:** armazena quantas vezes a solução ou um atributo foi visitado; baseia-se na frequência de residência e na frequência de transição;
- **Qualidade:** define a importância de uma solução ou atributo durante o processo de busca. Identifica elementos comuns nas soluções elite ou o caminho que levou a elas;
- **Influência:** determina a influência, o impacto de uma escolha no processo de busca (atributo).

2.2.2.2 Memória de Curto Prazo

A memória de curto prazo é baseada em recentidade e foi desenvolvida para superar as limitações dos métodos de busca local. É, portanto, um componente básico do algoritmo da BT, e visa eliminar ou reduzir os riscos de ciclagem e direcionar o processo de busca. Para isso, incorpora uma estrutura de memória que proíbe ou penaliza certos atributos da solução, isto é, os torna tabus ativo. Esses atributos provocariam o retorno a soluções já visitadas e, consequentemente, a permanência naquela região.

A estrutura que armazena os itens tabus-ativo é conhecida como lista tabu (do inglês, *tabu list*) e ela armazena o histórico das soluções visitadas recentemente. Em geral, armazena somente alguns atributos das soluções (geralmente movimentos que geraram a solução), ao invés da solução completa, o que seria computacionalmente mais custoso. Ela é uma lista circular, ou seja, quando um novo atributo é adicionado, um atributo antigo é descartado. Uma solução é considerada tabu quando, o(s) atributo(s) necessário(s) para gerá-la envolver(em) elemento(s) tabu(s)-ativo pertencentes à lista tabu.

A eficácia desta estratégia depende das regras que tornam um atributo tabu, assim como o tempo que esse permanece na lista tabu (período tabu). Portanto, é de grande importância que se crie regras que permitam a revogar a condição de tabu-ativo dos atributos. Isto permite o alcance de novas regiões de busca, e a saída de um ótimo local, esta nova região pode ser mais promissora que atual. Caso a nova região não seja considerada boa, ou seja, não se encontrou alguma solução, que seja melhor que as contidas no conjunto de soluções elite, pode-se retornar a região anterior e prosseguir com o processo de busca.

Os parâmetros da lista tabu são descritos a seguir:

- **critério(s) de ativação tabu (do inglês, *tabu active*):** regras que determinam quando um atributo se tornará tabu;
- **critério(s) de aspiração do(s) período(s) tabu (do inglês, *aspiration criteria*):** permite(m) revogar a condição tabu associada ao(s) atributo(s) presente(s) na lista tabu. Para isso ocorrer, esse movimento precisa ser considerado interessante para o prosseguimento da busca;
- **período(s) tabu (do inglês, *tabu tenure*):** tempo em que um atributo permanece na lista tabu.

As regras dos parâmetros acima variam em relação às peculiaridades de cada problema. Essas regras podem estar relacionadas ao número de iterações, a posição de um elemento na solução, os movimentos de troca de dois elementos, entre outros.

Um período tabu muito longo pode resultar em uma convergência prematura da busca e pode ser muito útil quando há a necessidade de fugir de ótimos locais. Já um período muito curto pode ocasionar o eventual retorno a um ótimo local já visitado, mas, por outro lado, permite a exploração de soluções vizinhas (próximas) a um ótimo local, caso ainda não o tenha encontrado. Portanto, o período tabu deve ser um valor intermediário entre os dois extremos. Para que ocorra esse equilíbrio é necessário que haja uma variação período tabu no decorrer do processo de busca, o que permite a exploração exaustiva de uma determinada região do espaço de busca e ao mesmo tempo procura regiões mais promissoras.

2.2.2.3 Memória de Longa Prazo

A memória de curto prazo é suficiente para garantir uma solução de qualidade em muitas aplicações, mas, para se obter ganhos adicionais no desempenho da busca, aplica-se a memória de longo prazo. Esse tipo de memória completa as informações fornecidas pela memória de curto prazo. Nela não estão presentes os critérios de aspiração que removem seus atributos, ou seja, os atributos ficam armazenados por tempo ilimitado.

Segundo Glover (1995), a inserção desse tipo de memória no processo de busca da BT não acarreta aumento visível, significativo ou relevante ao esforço computacional.

Seus principais componentes são a intensificação e a diversificação. As estratégias de intensificação tentam garantir a busca exaustiva nas regiões promissoras selecionadas, já as estratégias de diversificação buscam por regiões ainda não visitadas que sejam mais promissoras que a região atual. Esses são aplicados por meio da análise do conjunto de soluções elite.

Essa memória é comumente baseada em dois tipos de frequência: frequência de transição e frequência de residência. A frequência de transição corresponde ao número de vezes em que um atributo foi alterado (número de vezes que uma determinada posição da solução troca seu valor) e a frequência de residência armazena o número de vezes e o tempo em que um atributo esteve presente em um conjunto de soluções (verificação da ocorrência do atributo nas soluções elite). A seguir são apresentadas as características de cada frequência:

- Frequência de Transição:
 - **Alta:** se o atributo está em constante alteração significa que está ocorrendo o ajuste sistemático da solução, ou seja, ocorre a busca por um ótimo local;
 - **Baixa:** caso o ajuste esteja sendo finalizado, o atributo com baixa frequência de transição certamente estará presente na solução final. Isso significa que a busca está próxima do ótimo local.
- Frequência de Residência:
 - **Alta:** se um determinado atributo aparece várias vezes durante a busca, esse é altamente atraente para a geração da solução;
 - **Baixa:** se um determinado atributo aparece poucas vezes durante o processo de busca, pode indicar que esse é pouco atraente para a geração da solução ou também que a busca se encontra em um ótimo local e que não consegue sair do mesmo, necessitando,

então, de um mecanismo de diversificação para encontrar uma nova região, podendo esta ser mais promissora que a atual.

2.2.2.3.1 Intensificação

A intensificação direciona o processo de busca da BT, de forma que se realize uma busca exaustiva na região, no intuito de alcançar outra solução melhor, ou seja, de forma a exaurir a região e encontrar seu ótimo local. Geralmente, a estratégia de intensificação baseia-se no cálculo da frequência de transição dos atributos nas soluções elite, ou seja, verifica o quanto um determinado atributo das soluções elite é alterado.

Estratégias comuns de intensificação consistem do retorno a regiões atraentes visitadas recentemente, por meio do reinício do processo de busca com uma solução contida no conjunto de soluções elite ou por meio do cálculo da frequência de transição dos atributos. Os atributos mais frequentes tornam-se tabus, e reinicia-se o processo de busca. Estas duas formas podem ser aplicadas de forma conjunta.

2.2.2.3.2 Diversificação

A diversificação direciona o processo de busca da BT na visita de regiões ainda não visitadas, ou seja, é a tentativa de direcionar a busca a novas regiões e também de sair de ótimos locais. Geralmente, a estratégia de diversificação baseia-se no conceito do cálculo da frequência de residência dos atributos nas soluções elite. Com isso, os atributos que ocorrem com alta frequência são penalizados e os com baixa frequência são incentivados. Ela é aplicada geralmente quando o algoritmo de busca exauriu a análise naquela região, ou seja, para dada solução, não existem movimentos que provoquem ou tornem possível algum tipo de melhora.

A diversificação pode ser baseada em métodos simples, como a reinicialização, onde se reinicia o processo de busca de forma parcial ou total, ou por meio da modificação das regras de escolha dos movimentos. Na reinicialização parcial a nova solução mantém algumas características das soluções elite, já a reinicialização total a nova diverge completamente das soluções contidas no conjunto de soluções elite.

2.2.3 Estratégias

Na literatura encontram-se várias estratégias que visam à aplicação de intensificação e diversificação e objetivam alcançar uma boa combinação dessas estratégias. Estas estratégias auxiliam a BT no seu processo de busca, elas atuam na minimização das

desvantagens da BT, que originalmente não é um método de busca global e sim local. As técnicas de intensificação auxiliam no processo de busca local o tornando mais rápido e eficiente, já as técnicas de diversificação direcionam a BT a novas regiões de busca, este direcionamento não é aleatório e ao acaso, mas sim de forma probabilística, oriundas das soluções contidas no conjunto de soluções elite.

Segundo Glover (1995), a estratégia de diversificação não deve ser aplicada de forma arbitrária, mas, sim, quando não houver melhora na solução durante certo período de tempo. E a estratégia de intensificação deve-se tomar cuidado para não haja muita perda no tempo computacional.

A seguir, são apresentadas algumas das técnicas mais usadas presentes na literatura.

2.2.3.1 Reconexão de Caminhos (Path Relinking)

Esta técnica, proposta por Glover (1996), é uma estratégia de intensificação que explora a trajetória que originou as soluções contidas no conjunto de soluções elite. Para isso, ela realiza uma combinação entre os atributos das soluções elite, ou seja, seleciona os melhores atributos das soluções elites e os introduz na solução corrente. Esta técnica seleciona os atributos das soluções elites, de forma a guiar a solução no processo de busca. Este método é desenvolvido por meio do uso dos conceitos de memória atributiva.

2.2.3.2 Oscilação Estratégica (Strategic Oscillation)

Segundo Glover e Laguna (1997), esta estratégia está relacionada às origens da BT. Ela provê meios para que haja uma combinação ou intercâmbio das técnicas de intensificação e diversificação e orienta os movimentos em relação a uma fronteira de oscilação (nível crítico) que representa o ponto onde o método deveria ser interrompido. Ao invés de parar quando a fronteira for atingida, esta estratégia modifica as regras que selecionam os movimentos, permitindo, assim, que uma região definida pelo nível crítico seja ultrapassada. Esse nível pode ser um estágio de construção ou um intervalo de valores escolhido para uma função.

A estratégia prossegue até certa profundidade e volta. Então, o limite da oscilação é alcançado novamente e atravessado, mas, desta vez, na direção oposta. Este processo gera um comportamento oscilatório, o qual dá nome à estratégia. Este mecanismo se repete até que

o número máximo de iterações seja atingido. A possibilidade de ciclagem é evitada por mecanismo tabu, como aqueles utilizados na memória de curto prazo.

2.2.3.3 Busca Tabu Reativa (Reactive Tabu Search)

Battiti e Tecchiolli (1994-5) e Battiti (1996) apresentaram uma variação da BT, chamada Busca Tabu Reativa, cujo objetivo é alcançar a combinação entre intensificação e diversificação e almejaram também alcançar uma alternativa para se detectar a ciclagem.

Para detectar a ciclagem é utilizado um mecanismo que verifica a frequência da repetição de atributos que se diferem nas soluções elites. Portanto, poucas soluções devem possuir os mesmos atributos. Para realizar o controle da frequência de repetição dos atributos, utiliza-se uma estrutura de tabela *hashing* no processo de busca, de forma a otimizar o processo de controle das repetições. A estratégia armazena as soluções completas, assim como as iterações em que estas foram encontradas.

A Busca Tabu Reativa proíbe soluções completas, pois o processo analisa a repetição de uma solução, calcula o número de iterações entre as duas e reage automaticamente incrementando ou decrementando a duração tabu. Quando um conjunto ou grupo de soluções atinge certo nível de repetição, ativa-se o mecanismo de diversificação, pois o processo de busca exauriu aquela região. O método de diversificação consiste na realização de movimentos aleatórios.

2.2.4 O algoritmo

Segundo Glover e Laguna (1997), o método recebe como entrada uma solução x inicial dita x^* , a cada iteração. É gerado um subconjunto V de soluções vizinhas $N(x)$ a partir da solução corrente x . Esse conjunto é então avaliado por meio de uma função de avaliação $f(x)$. A função de avaliação retorna uma solução x' que é um dos vizinhos de x , e é obtido por movimentos m os quais constam na lista tabu T . Portanto $x' \in V$, sendo que x' oferece o menor custo segundo $f(x')$. Obtém-se a nova solução corrente x^* desconsiderando $f(x)$ a solução anterior. Ainda que $f(x^*) \geq f(x)$, é então selecionada tornando-se a solução corrente. Vale destacar que a melhor solução obtida, $f(x^*)$, sob todos os efeitos continua armazenada, pois somente soluções $f(x) \leq f(x^*)$ a substituirão, sendo essa (a melhor solução x , não necessariamente se) a solução final do método.

Glover e Laguna (1997), as principais propriedades da Busca Tabu são: o comprimento da lista tabu $|T|$ (tamanho da memória que armazena os movimentos proibidos),

a função de aspiração (critérios utilizados para tornar um movimento tabu), o comprimento do conjunto (tamanho do subconjunto da vizinhança de) e o número máximo de iterações em melhora . Essas propriedades devem estar bem definidas para que solução de qualidade seja alcançada ao final do método.

A condição de parada do método é comumente estabelecida de duas formas: a primeira é quando o método atingir um número máximo de iterações sem que haja melhora ; a segunda ocorre quando o método atingir um valor melhor que , a qual representa o valor mínimo conhecido para a função de avaliação , ou seja, uma solução considerada boa. Na Figura 2-7 é apresentado o fluxograma da BT.

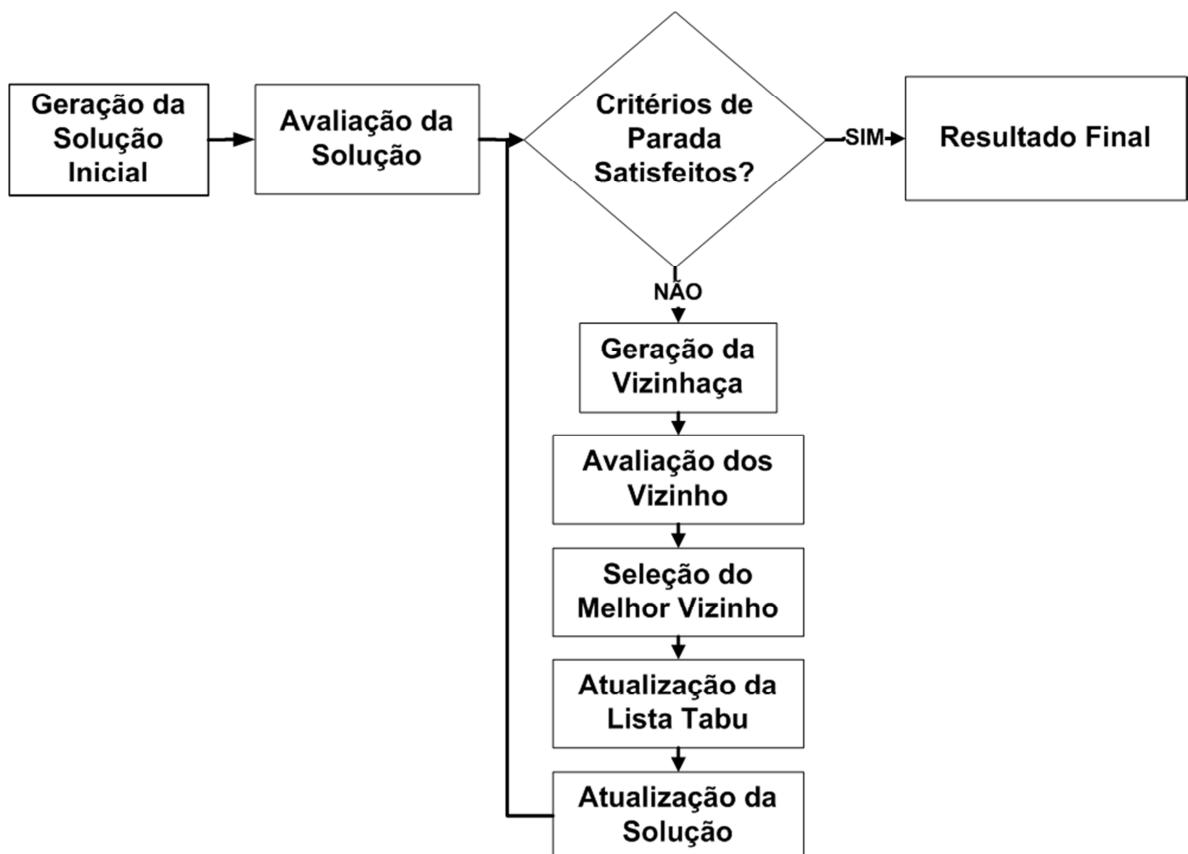


Figura 2-7 - Fluxograma: Busca Tabu, adaptado de Glover e Laguna (1997)

2.3 Aplicação dos métodos ao problema da programação da produção

Geralmente quando se trata de problemas de otimização em um *job shop* este é representado por um grafo disjuntivo (representa as restrições de rotas) e conjuntivo (representa as restrições de recursos). Neste tipo de grafo, geralmente os movimentos

consistem na troca da direção dos arcos disjuntivos, na troca de sequência de operações de uma mesma máquina ou na busca de máquinas alternativas para a operação, etc.

O critério de parada geralmente utilizado é o alcance de um determinado número de iterações sem que se encontre uma solução melhor que as já visitadas. Caso se conheça o valor mínimo esperado para o problema e o valor obtido se aproxime desse valor, o processo de busca pode ser interrompido. Geralmente esse valor é obtido por algoritmos de programação matemática, como o *branch-and-bound*, entre outros.

No entanto, nenhum se aproxima da modelagem utilizada neste trabalho, pois não se utiliza a modelagem em grafos. Portanto, se buscou nos documentos encontrados na literatura, formas de modelagem e desenvolvimento da BT, assim como a configuração de seus parâmetros.

Foram selecionados e estudados os seguintes trabalhos:

- Vilcot e Billaut (2008) utilizaram BT na otimização da população inicial do AG e a lista tabu com tamanho fixo.
- Eswaramurthy e Tamilarasi (2007) realizaram o hibridismo da BT como a heurística SPT (do inglês *Shortest Processing Time* - Menor Tempo de Processamento) e permitiram certo dinamismo no tamanho da lista tabu, que varia de acordo com o número de iterações da BT.
- Geyik e Cedimoglu (2004) realizaram testes com quatorze (14) diferentes métodos de definição do tamanho da lista tabu. Desses métodos, alguns possuem valores fixos, outros possuem certa flexibilidade, variando em uma determinada faixa de valores e outros dependem da quantidade de máquinas, da quantidade de tarefas a serem realizadas ou do tamanho da vizinhança. O autor obteve resultados satisfatórios com o tamanho da lista tabu fixado em seis (6).
- Scrich *et. al.* (2004) utilizaram estratégias de intensificação e diversificação baseadas na frequência da ocorrência dos atributos nas soluções elites.

Como se observa nos trabalhos acima listados, a construção de uma boa solução inicial para a BT gera bons resultados. Portanto, neste trabalho o AG gerará a solução inicial, que será usada como solução inicial da BT. A BT também será usada na otimização da população inicial dos AGs e a lista tabu possuirá tamanho fixado em seis (6). Além disso,

serão realizados testes com as estratégias de diversificação e intensificação, as quais são geralmente baseadas em frequência de residência e em frequência de transição.

No grupo TEAR desenvolvem-se pesquisas voltadas para a programação da produção. Geralmente aplicam-se AGs aos problemas, porém neste trabalho o objetivo é aplicar a BT ao problema e, só então, aplica-lá, aos AGs. Realizou-se, então, apenas uma revisão de textos recentes da aplicação dos AGs ao problema, como os trabalhos do TEAR são recentes. As referências utilizadas por Deriz (2007) e por Sanches (2008.) são mantidas.

A seguir, são apresentados os trabalhos desenvolvidos no grupo TEAR que são correlatos à proposta.

Morandin *et. al.* (2007a) propuseram uma heurística para a programação de um Sistema de Manufatura Flexível (FMS) usando modelagem em *Petri Nets* virtuais temporizadas e também uma estratégia de busca aplicada à modelagem *Petri Nets*. Na proposta, assume-se que os elementos no chão de fábrica (máquinas, AGVs) nunca falham e todos os recursos envolvidos na produção estão disponíveis durante o processo. O tempo de transporte é considerado desprezível e a capacidade de armazenamento de peças em *buffers* é infinita. Assume-se, ainda, que um recurso realiza uma operação por vez e retorna um único lote e, uma vez iniciada uma operação, essa não é interrompida. Os tempos de configuração são embutidos no tempo de operação e os tempos de processamento de operações são previamente conhecidos. A estratégia de busca proposta é uma variante do A^* . A função heurística é definida como uma estimativa do custo mínimo de produção remanescente, onde o custo é definido como o tempo corrente, contado a partir do início da primeira operação sobre os produtos envolvidos na modelagem do problema. Assim, é possível encontrar a sequência de operações que resulta no menor tempo total de produção. As experiências realizadas apresentam uma redução do número de nós explorados e um ganho considerável em relação ao tempo de resposta.

Deriz (2007) propôs uma modelagem de AG para resolver programação da produção de maneira eficiente, tendo como critério de desempenho o *makespan* mínimo. Um AG é utilizado com o objetivo de encontrar uma ótima programação. Para tanto, um cromossomo é representado por um vetor de tamanho $m + m * n$, onde m é o número de produtos a serem produzidos e n é o número máximo de operações em todos os possíveis roteiros de fabricação para os m produtos, os quais são subdivididos em m subvetores de tamanho $n + 1$. Cada subvetor representa um produto P_i a ser produzido e o primeiro gene do subvetor é o índice i , que identifica o produto desse subvetor e os genes seguintes representam as operações do roteiro de fabricação escolhido para o produto. A fim de evitar

genes com valores vazios, onde não há mais operações no roteiro de fabricação para o produto (determinando, assim, que o produto está concluído), foi definido que as máquinas nas quais não há o processamento daquele produto possuirão o valor “0”. A operação de cruzamento é definida por meio da escolha de dois cromossomos pais, onde é trocado um subvetor referente ao mesmo produto em cada um dos cromossomos pais, gerando assim os cromossomos filhos. A operação de mutação é definida como a troca de um roteiro de um produto de forma aleatória por outro roteiro escolhido aleatoriamente. A função de *fitness* usada para avaliar as soluções para a programação da produção é o mínimo *makespan*.

Sanches (2008) procurou abordar a integração da programação da produção com a programação dos AGVs. Um Algoritmo Genético Adaptativo foi desenvolvido para tratar o problema de Programação da Produção com o uso Simultâneo de Máquinas e Sistemas de Transporte. O critério de desempenho utilizado foi o *makespan* e os resultados obtidos foram comparados aos de outras duas abordagens. Em seu trabalho foi utilizado o modelo proposto por Deriz (2007), no qual se insere a modelagem dos AGVs e se utiliza uma abordagem adaptativa do AG. Para os testes de validação, os resultados obtidos por Sanches foram comparados aos resultados de Deriz (2007) e também aos valores obtidos por meio de um software comercial de simulação. O Algoritmo Genético Adaptativo obteve melhor resultado que o algoritmo proposto por Deriz (2007) e se aproximou, e em alguns casos se igualou, aos resultados fornecidos pelo software de simulação. Sanches utilizou a seleção por torneio, sendo que cinco candidatos da população foram selecionados aleatoriamente e o melhor indivíduo, baseado no valor da função de *fitness*, foi copiado para a população intermediária. A seleção por torneio é repetida até que toda a população intermediária seja preenchida. A partir da sequência dos produtos fornecida pelo AG, realiza-se a programação dos Sistemas de Transportes.

No TEAR possui vários trabalhos onde se aplica AGs na resolução da programação da produção, mas nenhum deles focou-se em determinar a melhor forma de definição do desenvolvimento dos AGs e dos seus respectivos parâmetros. Por exemplo, tanto Deriz (2007) quanto Sanches (2008) utilizam mecanismos distintos na seleção dos indivíduos, e também utilizam valores distintos, para os parâmetros genéticos. Na literatura também não foram encontradas indicações de uma melhor forma de desenvolvimento de AG. Portanto neste trabalho também será realizada um estudo para averiguar qual o método de seleção dentre os já utilizados para o problema em questão, e também será averiguado qual a melhor forma e momentos de se aplicar o operador de mutação se nos indivíduos da população corrente os nos filhos oriundos do operador de cruzamento.

3 Proposta

Sistemas de Manufatura com Compartilhamento de Recursos são definidos como Sistemas Flexíveis de Manufatura, cujas máquinas podem ser utilizadas em várias operações distintas. Os produtos podem ser processados em roteiros de produção distintos e diferentes meios de transporte podem ser utilizados para movimentar o produto no sistema produtivo. No entanto, essa flexibilidade exige um esforço considerável para definir a programação da produção e utilizar de forma eficaz os recursos disponíveis ao longo do tempo com o objetivo de satisfazer certos critérios de desempenho.

No grupo TEAR, há algum tempo têm-se desenvolvido pesquisas relacionadas à Programação Reativa da Produção em Sistemas de Manufatura com Recursos Compartilhados, com uso de AGs. A primeira foi a de Deriz (2007), que faz uso de um AG tradicional para a resolução do problema da programação da produção. Deriz codifica todo o problema no cromossomo, a programação da produção é obtida por meio da leitura dos seus genes, no entanto, não trata o problema do transporte. O segundo trabalho foi o de Sanches (2008a), que percebendo a falta do sistema de transporte, propôs o seu cálculo associando, à função de *fitness*, um procedimento que realiza a programação do sistema de transporte utilizando regras de despacho de veículos. Para garantir a diversidade da população do AG, foi proposto o uso do AGA. Tanto Deriz (2007) quanto Sanches (2008a) utilizaram como critérios de desempenho o valor do *makespan* e o tempo computacional do algoritmo. Nenhum deles, porém, preocupou-se com a verificação da vizinhança das regiões selecionadas pelos AGs ou utilizou nenhum método para otimizar o processo de busca dos mesmos (AG e AGA).

Sendo assim, neste trabalho, propõe-se a combinação de AGs com as BTs, a fim de realizar a verificação das regiões apontadas pelos AGs. A BT foi escolhida por incorporar estruturas de memória em seu processo de busca, já que sua busca é baseada em vizinhança, e também por possuir a característica de exaurir as regiões de busca alcançadas e de escapar dos ótimos locais.

Segundo Glover (1997), esse tipo de busca se dá por meio dos métodos heurísticos construtivos e destrutivos, que tornam a vizinhança maleável, o que possibilita a geração de vizinhanças otimizadas que minimizam a busca exaustiva realizada pela busca local.

Neste trabalho, delineiam-se os seguintes objetivos: definir qual dos métodos de seleção utilizados se aplica melhor ao problema, de forma a garantir maior diversidade genética; realizar o adiantamento das tarefas do sistema de transporte, pois Sanches (2008) o ativa somente quando há a necessidade de movimentar um produto no sistema de manufatura; definir a melhor forma de desenvolvimento dos AGs e dos AGAs, gerando duas variantes de ambos os algoritmos, onde a principal diferença entre elas está no momento de aplicação do operador de mutação. Isso foi realizado de formar a definir uma variante que favoreça a diversidade populacional dos algoritmos (AGs e AGAs).

Quanto à aplicação da BT, são averiguadas duas formas de vizinhança. A primeira é baseada na troca sucessiva dos roteiros de produção, sem alterar a ordem de produção dos mesmos. Já a segunda baseia-se na troca sucessiva de pares de produtos. Os pares são formados por dois produtos, seguidos de seu respectivo roteiro de produção. Também é averiguada uma forma de aplicação da memória de curto prazo e da memória de longo prazo.

Os critérios adotados para definir o desempenho da proposta são baseados na obtenção do mínimo *makespan* e no tempo computacional que o algoritmo leva para convergir em uma boa solução. O sistema de transporte é constituído por 3 AGVs. A programação da produção é realizada no nível de controle e não no nível de planejamento. Portanto, não é necessário fazer o tratamento das ocorrências dos eventos inesperados, pois estas são tratadas no momento em que ocorrem para isto solicita-se a reprogramação da produção.

Nas próximas seções são apresentadas a modelagem dos algoritmos (AGs, AGAs e BTs), a modelagem da solução e, consequentemente, a codificação do cromossomo, além da forma de cálculo do *makespan* proposto e a remodelagem das formas de cálculo que Deriz (2007) e Sanches (2008) propuseram.

Todos os algoritmos desenvolvidos (AGs, AGAs, BTs e AGTs), assim como as formas de cálculo do *makespan*, foram desenvolvidos através do MATLAB®.

3.1 Caracterização e restrições do problema

A Programação da Produção em Sistema de Manufatura com Recursos Compartilhados e com uso Simultâneo de Máquinas e Sistema de Transporte envolve a tomada de decisão que implica na alocação dos recursos de produção e do sistema de transporte, assim como a escolha dos roteiros de manufatura a serem seguidos no processo

produtivo. O valor do *makespan* é comumente utilizado para definir o desempenho da produção em relação a essas decisões.

Dado um sistema de manufatura qualquer, supõe-se que sua meta seja fabricar certo número de produtos em um determinado tempo de produção. Para que se alcance essa meta, necessita-se determinar a alocação dos recursos, a ordem de fabricação dos produtos e a ordem das operações sobre cada produto. As dificuldades no cumprimento dessa meta estão em determinar, previamente, quais operações devem ser executadas e quando devem ser processadas, de forma que todos os produtos sejam finalizados no menor tempo possível. Devido a isso, é importante definir as características e restrições do problema a ser tratado, assim como definir a regras de negócio para a realização da programação reativa da produção. A seguir, são apresentadas as características do problema:

- I. Inicialmente, nenhum produto está sendo produzido;
- II. Um tipo de produto é produzido seguindo um único roteiro de fabricação, ou seja, se houver vários roteiros possíveis, apenas um deve ser selecionado;
- III. Um determinado produto deve passar por todas as máquinas do roteiro de fabricação escolhido, sem alteração da ordem definida;
- IV. Uma operação é executada em apenas uma máquina, ou seja, um produto passa por apenas uma máquina em determinado tempo;
- V. Uma máquina processa apenas um produto por vez;
- VI. Os tempos de operação dos produtos nas máquinas são previamente conhecidos;
- VII. Máquinas falham;
- VIII. O sistema de transporte falha;
- IX. Há falta de matéria prima;
- X. Uma vez que uma determinada operação é iniciada (que um determinado produto passa por uma máquina), ela não é interrompida até que seja concluída;
- XI. O número de veículos é conhecido e os veículos são iguais, em relação à velocidade e características de transporte de cargas;
- XII. Os veículos partem inicialmente de um ponto caracterizado como estação de Carga/Descarga;
- XIII. Um veículo carrega somente um produto por vez;
- XIV. Os tempos de transporte em cada ponto do chão de fábrica são conhecidos;

- XV. As máquinas possuem *buffers* (entrada e saída) de tamanhos infinitos;
 XVI. Tempos de *setup* de máquinas estão incluídos nos tempos de operação.

A programação do sistema de transporte é necessária no momento em que um determinado produto precisa ser transportado até outra máquina, pois mais de um elemento do sistema de transporte pode estar disponível ou nenhum deles. Então, o problema é definir qual veículo deve ser selecionado.

Para tratar esse problema, Sanches (2008) utilizou duas regras de despacho, a STT/D (*Shortest Travel Time/Distance* – Menor tempo de Viagem) e a RV (*Random Vehicle Rule* – Veículo Aleatório). Essas regras serão utilizadas da seguinte forma:

- **STT/D:** utilizada quando existe mais de um veículo disponível e esses não se encontram na estação de carga/descarga.
- **RV:** utilizada quando todos os sistemas de transporte estão disponíveis

No entanto, apenas essas duas regras de despacho não são suficientes para otimizar a seleção dos elementos do sistema de transporte a ser utilizado. Para isso, neste trabalho foi desenvolvida uma estratégia heurística que realiza a previsão de quais máquinas estarão disponíveis e necessitarão de transporte. A lógica de execução do sistema de transporte é apresentada a seguir:

- (a) Mover todos os produtos para a estação de Carga/Descarga de acordo com a ordem fornecida pelo cromossomo.
- (b) Verificar qual produto deve ser transportado, segundo sua prioridade.
 - (i) Verificam-se quais os veículos encontram-se disponíveis e se aplica a regra de despacho apropriada.
 - (ii) Move-se o veículo, a partir do ponto corrente (onde ele se encontra), até o ponto onde ele está sendo solicitado. O veículo move o produto para a próxima máquina designada, conforme cadastrada no roteiro do produto.
 - (iii) Após transportar um determinado produto, o veículo fica estacionado no ponto de estacionamento da máquina ou no ponto da estação Carga/Descarga no qual o produto foi entregue. No entanto, neste trabalho, o veículo vai para o ponto de estacionamento da próxima máquina/operação, que irá solicitar o sistema de transporte.
- (c) Voltar à etapa (b). Este processo se repete até que todos os produtos tenham sido produzidos e entregues à estação Carga/Descarga.

3.2 Codificação da solução e do cromossomo

Tanto a codificação da solução inicial, que é o ponto de partida para a BTs, quanto a codificação dos cromossomos dos AGs e, consequentemente, do AGTs, baseiam-se nos trabalhos de Deriz (2007), Morandin *et. al.* (2007a), Morandin *et. al.* (2007b), Morandin *et. al.* (2008a), Morandin *et. al.* (2008b), Sanches (2008) e Sanches *et. al.* (2009). A fim de facilitar a aplicação da BT e reduzir o número de movimentos realizados por ela em seu processo de busca, e para minimizar a quantidade de memória utilizada pelos AGs e AGAs, utilizou-se, como nos trabalhos mencionados anteriormente, a representação simplificada da solução. Segundo os autores, a codificação dos cromossomos é definida da seguinte forma: seja i o número de produtos a ser fabricados; j o número máximo de roteiros possíveis para cada produto; e k o número máximo de operações em todos os possíveis roteiros de fabricação para os i produtos.

Além disso, um cromossomo (ou solução) é representado(a) por um conjunto de valores inteiros de tamanho $i \times j \times k$, subdividido em conjuntos de genes de tamanho $j \times k$, sendo que cada conjunto representa um produto a ser processado, seguido de seu roteiro de fabricação, em que o primeiro gene desse conjunto é o produto e os genes seguintes representam as máquinas do roteiro de fabricação escolhido para aquele produto. A Figura 3-1 apresenta esse cromossomo. Para que cada conjunto possua um tamanho fixo, há a necessidade da inserção de genes vazios, os quais assumem o valor “0” a partir do gene da última máquina do roteiro.

Figura 3-1 - Representação completa do cromossomo/solução (Morandin et. al., 2008a)

A fim de facilitar a modelagem da proposta, os autores redefinem a modelagem do cromossomo de maneira mais simplificada, que contém, em cada conjunto de genes, o produto a ser produzido e o índice do roteiro escolhido, conforme apresentado na Figura 3-2.

Figura 3-2 - Representação simplificada do cromossomo/solução (Morandin et. al., 2008a)

Defina-se i como o produto a ser processado, j como uma máquina do roteiro de fabricação desse produto, onde j representa a ordem da máquina no roteiro de produção e k é o roteiro a ser utilizado no processo produtivo.

3.2.1 Exemplo de codificação

Um cromossomo representa quais produtos devem ser produzidos, seguido de seus respectivos roteiros de fabricação. A seguir, durante a etapa de cálculo do fitness, os produtos são alocados para as suas respectivas máquinas através do uso de um veículo em um determinado momento, a fim de se obter uma programação da produção.

Morandin *et. al.* (2008a) e Deriz (2007) apresentam um exemplo de codificação do cromossomo, em que consideram uma fábrica com “6” máquinas $M_1, M_2, M_3, M_4, M_5, M_6$ e “3” tipos de produto P_1, P_2, P_3 . Na Tabela 3-1 estão relacionados os roteiros de fabricação possíveis para os produtos.

Tabela 3-1- Produtos e Roteiros de Fabricação, adaptado de Morandin et. al. (2008a)

Produto	Roteiros de Fabricação
P_1	$R_1(M_1, M_2, M_6)$
	$R_2(M_4, M_5, M_6)$
P_2	$R_1(M_1, M_2, M_5, M_6)$
	$R_2(M_3, M_4, M_5, M_6)$
P_4	$R_1(M_4, M_3, M_2)$
	$R_2(M_4, M_1, M_5)$

Na Figura 3-3 é apresentada a codificação completa de um cromossomo, gerada a partir dos produtos e dos roteiros de fabricação relacionados na Tabela 3-1. O primeiro gene indica o produto “1” e os “4” genes seguintes indicam as máquinas do roteiro de fabricação do produto “1”. Lembrando que os genes vazios são codificados com o valor “0”.

P1 M1 M2 M6 0 P3 M4 M1 M5 0 P2 M1 M2 M5 M6

Figura 3-3 - Cromossomo/Solução, adaptado de Morandin et. al. (2008a)

Na Figura 3-4 é apresentado o cromossomo da Figura 3-3 com a codificação simplificada, na qual o primeiro e o segundo genes indicam que o produto P_1 será produzido pelo roteiro R_1 , o terceiro e o quarto genes indicam que o produto P_3 será produzido pelo roteiro R_2 , e o quinto e o sexto genes indicam que o produto P_2 será produzido pelo roteiro R_1 .

P1 R1 | P3 R2 | P2 R1

Figura 3-4 - Cromossomo/Solução simplificado, adaptado de Morandin et. al. (2008a)

3.3 Modelagem dos AGs e AGAs

Foram modeladas e desenvolvidas duas variantes do AG. Com isso, objetiva-se encontrar uma variante que favoreça a diversidade populacional. A principal diferença entre essas versões é a forma de geração dos indivíduos da próxima geração. As versões divergem no momento de aplicação do operador de mutação. A estas variantes aplicou-se o elitismo.

Na primeira versão, os indivíduos que sofrem mutação são oriundos da população anterior, ou seja, a população corrente. Já na segunda versão, eles são oriundos dos indivíduos cruzados e do processo de cópia, ou seja, os indivíduos sofrem mutação devido ao operador de cruzamento e à cópia dos indivíduos. A primeira versão é exemplificada na Figura 3-5. A população de sua próxima geração é obtida da seguinte maneira:

- Verifica-se a quantidade de indivíduos que são gerados por cruzamento, por mutação e selecionados por elitismo:
 - caso não se tenha completado toda a população, seleciona-se os indivíduos necessários por meio do mecanismo de seleção, os quais são copiados (clonado) para a população da próxima geração. Esses indivíduos não estão sujeitos à mutação;
 - tanto os indivíduos que cruzam, quanto os indivíduos que sofrem mutação são escolhidos pelo mecanismo de seleção e são oriundos da população da geração anterior, ou geração corrente.

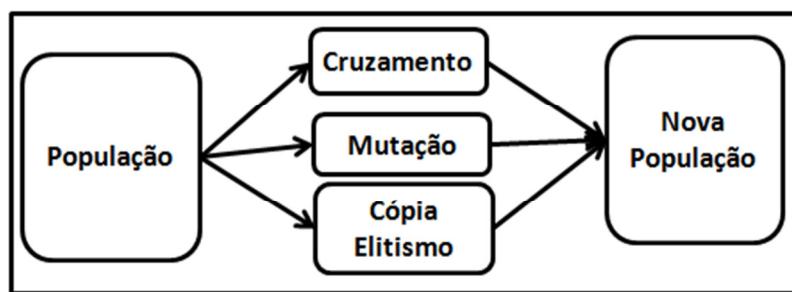


Figura 3-5 - Primeira versão do AG/AGA (V1)

A segunda versão é exemplificada na Figura 3-6. A população de sua próxima geração é obtida da seguinte maneira:

- Verifica-se a quantidade de indivíduos que são gerados por cruzamento, por mutação e selecionados por elitismo:
 - caso não se tenha completado toda a população, seleciona-se os indivíduos necessários por meio do mecanismo de seleção, os quais são copiados (clonado) para a população da próxima geração;

- tanto os filhos gerados pelo cruzamento, quanto os indivíduos copiados estão sujeitos à mutação e é somente sobre eles que a mutação será aplicada.

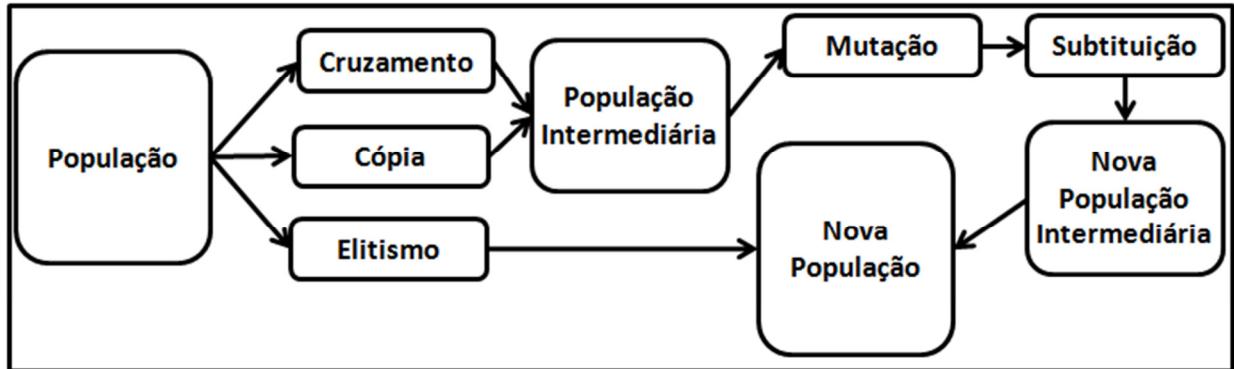


Figura 3-6 - Segunda versão do AG/AGA (V2)

O método cópia é utilizado para garantir que a próxima população tenha o mesmo número de indivíduos que a população anterior. Isso é necessário, pois nem sempre os operadores de cruzamento e mutação geram uma população que possua um mesmo número de indivíduos que a população anterior.

A essas versões foi aplicado o mecanismo que garante a diversidade populacional apresentado em Sanches (2008). Ele realiza o ajuste das taxas de cruzamento e a mutação por meio das seguintes regras:

- I. Se a porcentagem de melhora da média do fitness dos cromossomos filhos gerados for maior ou igual a 10% do correspondente aos cromossomos pais, então a probabilidade de ocorrência do operador genético deve ser aumentada em 0.05 para operações de cruzamento e em 0.005 para operações de mutação.
- II. Se a porcentagem de piora da média do fitness dos cromossomos filhos gerados for maior ou igual a 10% do correspondente aos cromossomos pais, então a probabilidade de ocorrência do operador genético deve ser diminuída em 0.05 para operações de cruzamento e em 0.005 para operações de mutação.
- III. Se a porcentagem de melhora/piora da média do fitness dos cromossomos filhos em relação à média do fitness dos cromossomos pais não se aplicar as regras I e II, então a probabilidade de ocorrência do operador genético não deve ser alterada.

- IV. As taxas devem estar entre o mínimo de 0.5 e o máximo de 1.0, para o operador de cruzamento, e entre o mínimo de 0.00 e o máximo de 0.10, para o operador de mutação.

Como ilustrado na Figura 3-7, a taxa de aplicação dos operadores genéticos é adaptada da seguinte forma: calcula-se a média da aptidão dos pais e a média da aptidão dos filhos e a diferença entre elas; caso essa diferença provoque uma melhora (valor negativo – possível melhora) superior a 10% em relação aos pais, incrementa-se o operando; caso provoque uma piora (valor positivo – possível piora) superior a 10% em relação aos pais, decrementa-se o operando.

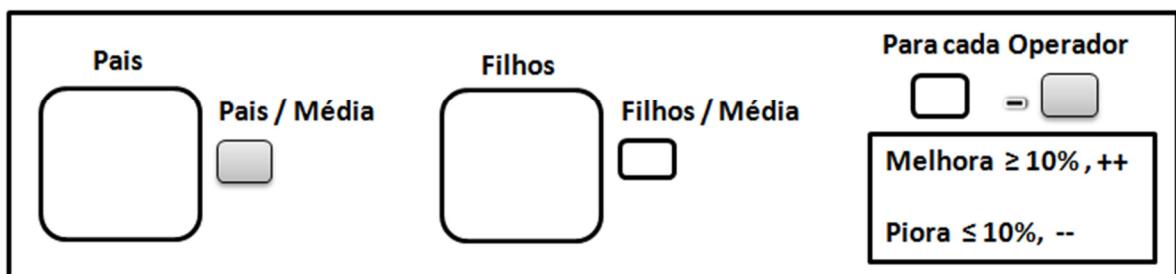


Figura 3-7 - Aplicação do ajuste das taxas dos operadores genéticos

3.3.1 Cruzamento

Segundo Morandin *et. al.* (2008a), o cruzamento é feito entre dois cromossomos pais, pela troca do conjunto de genes referente ao mesmo produto em cada um dos cromossomos, o que gera dois cromossomos filhos, que vão para a nova população. Esse processo é apresentado na Figura 3-8.

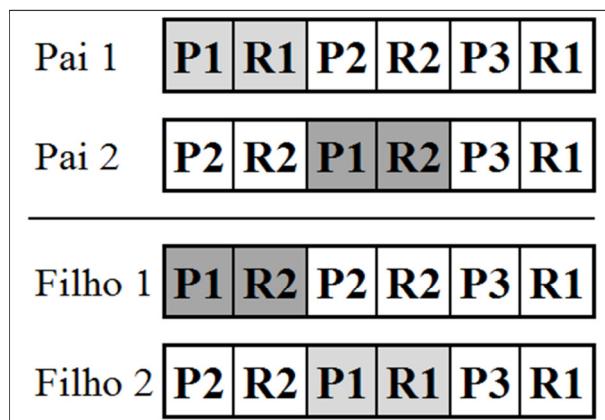


Figura 3-8 - Cruzamento, adaptado de Sanches (2008)

3.3.2 Mutação

Segundo Morandin *et. al.* (2008a), a mutação é realizada em um cromossomo, no qual um determinado produto é escolhido de forma aleatória e o seu roteiro de fabricação é trocado por outro possível roteiro. Esse processo é apresentado na Figura 3-9 e, através dele, é possível observar que ocorre perda de material genético e novas características são introduzidas no cromossomo, o que garante maior diversidade na população.

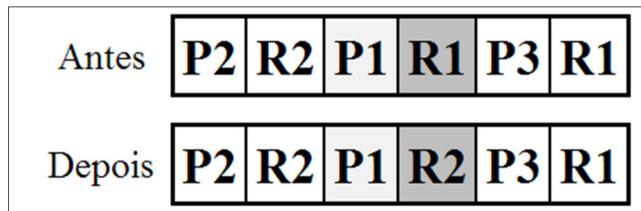


Figura 3-9 - Mutação, adaptado de Sanches (2008)

3.3.3 Elitismo

Aos algoritmos descritos acima é aplicado o mecanismo de elitismo, o qual copia, para a próxima geração (próxima população), o melhor indivíduo, ou seja, o mais apto da geração (população) atual. Esse passa para a próxima geração sem sofrer alteração alguma em sua carga genética. Copia-se apenas um indivíduo, para não prejudicar a convergência natural dos algoritmos e nem a diversidade populacional.

3.3.4 Seleção

Na literatura encontram-se diversos métodos de seleção, dentre os quais se destacam: seleção por roleta, seleção por torneio, seleção por classificação (do inglês, *rank selection*), seleção pelo método da amostragem universal estocástica, seleção de Boltzmann, etc. Esses métodos utilizam estratégias distintas que aumentam ou diminuem a convergência do AG ou tentam proporcionar a diversidade genética. Porém, não há garantias ou indícios de que um ou outro método de seleção para um determinado problema, o que exige sempre uma abordagem empírica. Deriz (2007) utiliza a roleta como método de seleção, enquanto Sanches (2008) utiliza o torneio, porém nenhum deles especifica qual dos métodos é o melhor para o problema em questão. Portanto, neste trabalho, ambos os métodos de seleção são levados em consideração.

Para o método de torneio são selecionados, de forma aleatória, cinco indivíduos candidatos. O indivíduo de melhor aptidão é selecionado e estará à disposição dos mecanismos de reprodução. Esse mecanismo é aplicado até que todos os indivíduos

necessários sejam selecionados. Já pelo método da roleta, todos os indivíduos da população são inseridos na mesma. A faixa de cada indivíduo na roleta é definida pela equação apresentada na seção 2.1.5.1e ela é girada até que todos os indivíduos necessários para a reprodução sejam selecionados.

3.3.5 Função Objetivo

A função de fitness é utilizada a fim de atribuir uma determinada nota aos indivíduos, de acordo com a sua aptidão e se baseia no valor de *makespan*. Para obter tal valor, é necessário definir como as tarefas do sistema de transporte serão programadas/sequenciadas para movimentar os produtos entre as máquinas. Tanto o cálculo do *makespan* (alocação das máquinas, do chão de fábrica), quanto à alocação do sistema de transporte são realizados de forma simultânea. As descrições do cálculo do *makespan* usadas neste trabalho, em Deriz (2007) e em Sanches (2008) são apresentadas na seção 3.6.

3.4 Modelagem das BTs

Foram modeladas e desenvolvidas três variantes da BT e suas diferenças consistem na inserção das seguintes estratégias:

- 1^a versão: BT clássica, que não possui nenhuma estratégia agregada;
- 2^a versão: incorporou-se, à BT clássica, a estratégia conhecida como Primeira Melhora (do inglês, “*First Improvement Method*”). Aqui, interrompe-se o algoritmo de busca quando uma solução melhor é encontrada;
- 3^a versão: incorporaram-se, à BT clássica, as estratégias de intensificação e diversificação.

Tanto a 1^a quanto a 2^a versão fazem uso da memória de curto prazo. Já a 3^a versão faz uso tanto da memória de curto prazo, quanto da memória de longo prazo. A estrutura das memórias utilizadas em cada variante é idêntica, não existem variações na configuração das memórias utilizadas. A memória armazena as informações completas sobre as soluções.

A memória de curto prazo baseia-se em recentidade, ou seja, armazena a solução e atributos visitados há pouco tempo. Portanto, neste trabalho, é utilizada uma lista tabu que armazena as posições da solução que foram alteradas e geraram essa solução. O tempo que um movimento permanece na lista é proporcional ao tamanho da mesma. Portanto, quando

uma nova solução é alcançada, o movimento que está há maior tempo na lista é descartado. Um movimento tabu é permitido somente se ele gerar uma solução melhor que a encontrada até o presente instante.

A memória de longo prazo baseia-se em frequência, ou seja, verifica o número de vezes que uma solução ou um atributo foi visitado. Neste trabalho, a memória armazena a solução completa e seu respectivo valor obtido pela função objetivo, pois eles servem como ponto de partida para as estratégias de intensificação e diversificação. O seu tamanho é indefinido, pois armazena todas as boas soluções visitadas. Nesse tipo de memória, as boas soluções visitadas recentemente ocupam as últimas posições. Isso é importante para que as estratégias de intensificação e diversificação identifiquem de forma rápida essas soluções e realizem suas respectivas operações sob as mesmas.

A intensificação baseia-se na frequência de transição, a qual verifica o quanto os atributos das soluções estão sendo alterados. Caso ocorra um alto índice de alterações, significa que está ocorrendo a busca por um ótimo local; caso o índice de alterações seja baixo, significa que o processo de busca está próximo de um ótimo local. Neste trabalho, a intensificação ocorre por meio do uso da memória de curto prazo. Os atributos mais comuns, ou seja, com baixo índice de alterações, são inseridos na lista tabu e penalizados durante certo período de tempo. Essa operação aumenta o tamanho da lista, que deverá voltar ao normal após a aplicação da estratégia de diversificação. A intensificação será aplicada quando o valor da função objetivo se aproximar, valor da última solução a ser inserida na memória.

A diversificação baseia-se na frequência de residência, que verifica o quanto um atributo está presente nas soluções. Caso esse apareça muitas vezes, significa que ele é altamente atraente, caso ele apareça poucas vezes significa que ele é pouco atraente. A diversificação será aplicada quando o processo de busca atingir certo número de iterações sem melhora na solução. Neste trabalho, a diversificação ocorrerá por meio da troca de posição dos atributos que possuem alto e baixo índice de residência. O atributo que possuir maior índice de residência será trocado pelo atributo que possuir o menor índice e, assim, sucessivamente, até que todos os atributos estejam reorganizados.

Segundo Glover e Laguna (1997), a BT, ao gerar a vizinhança de uma determinada solução, adota os tipos de movimentos usados em métodos heurísticos construtivos e destrutivos, ou seja, seus movimentos frequentemente correspondem à inserção de um elemento na solução, à troca da posição de dois elementos, etc.

Este trabalho propõe duas vizinhanças distintas: a primeira consiste na alteração dos roteiros de produção no vetor de solução, sem alterar a ordem de produção dos mesmos, e

a segunda consiste na troca da posição de pares de atributos no vetor solução. A Figura 3-10 demonstra o fluxo da geração da primeira vizinhança.

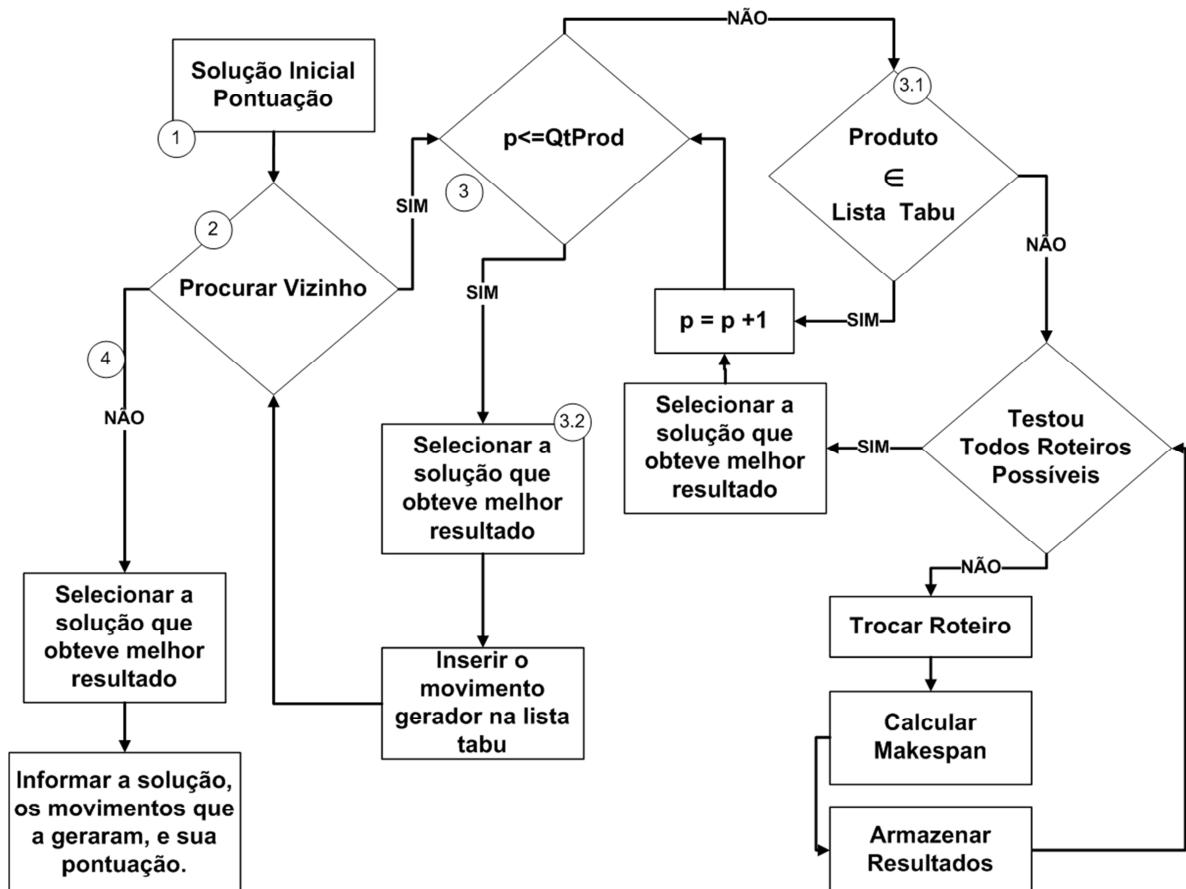


Figura 3-10 - 1^a Vizinhança

Como ilustrado na Figura 3-10, para encontrar o melhor vizinho por meio da 1^a vizinhança, segue-se os passos abaixo relacionados:

- 1) Obter a solução e a sua pontuação, indicada por “score”.
- 2) Verificar se há a necessidade de realizar a busca do vizinho:
 - a. Caso seja necessário, seguir para o passo (3);
 - b. Caso contrário, seguir para o passo (4).
- 3) Verificar se os produtos foram analisados, e se os seus respectivos roteiros foram testados:
 - a. Caso nem todos os produtos tenham sido analisados, verifica-se se os mesmos não se encontram na lista tabu. Caso eles não estejam presentes, troca-se seus respectivos roteiros (isso é realizado para cada roteiro); caso se tenha testado todos os roteiros, seleciona-se a solução que obteve o melhor resultado e retorna-se ao passo (3);

- b. Caso todos os produtos tenham sido analisados, seleciona-se a melhor solução gerada, assim como os movimentos que a geraram e retorna-se para o passo (2).

4) Informar o melhor vizinho encontrado.

Já Figura 3-11 demonstra o fluxo da geração da segunda vizinhança.

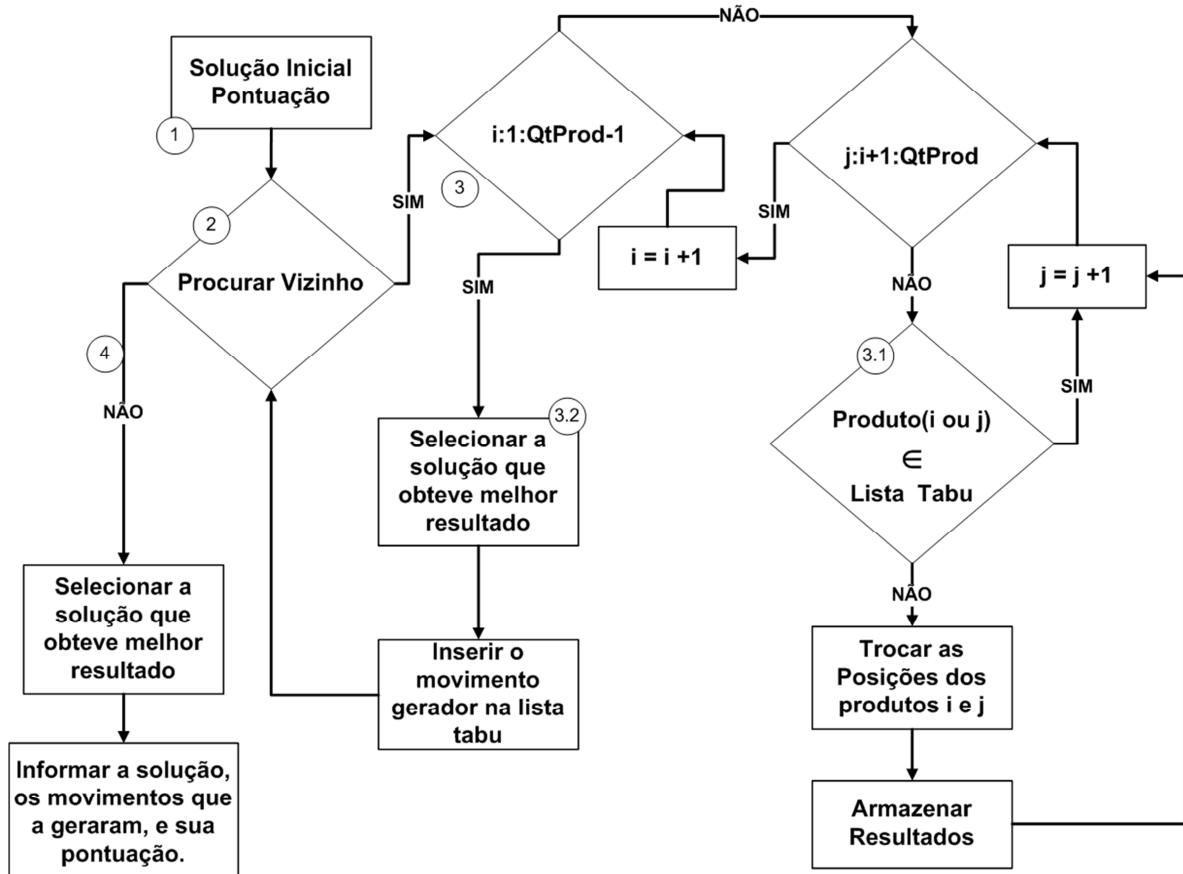


Figura 3-11 - 2^a Vizinhança

Como ilustra a Figura 3-11, para encontrar o melhor vizinho por meio da 2^a vizinhança, segue-se os passos abaixo relacionados:

- 1) Obter a solução e a sua pontuação, indicada por “score”.
- 2) Verificar se há a necessidade de realizar a busca do vizinho:
 - a. Caso seja necessário, seguir para o passo (3);
 - b. Caso contrário, seguir para o passo (4).
- 3) Realiza-se a troca de posição dos produtos na solução, ou seja, altera-se a ordem de processamento dos produtos:
 - a. Caso nem todos os produtos tenham sido analisados, verifica-se se os mesmos não se encontram na lista tabu. Caso eles não estejam presentes, troca-se o produto i pelo produto j; caso todos os roteiros

- tenham sido testados, seleciona-se a solução que obteve melhor resultado e retorna-se ao passo (3);
- b. Caso todos os produtos tenham sido analisados, seleciona-se a melhor solução gerada, assim como os movimentos que a geraram e retorna-se para o passo (2).
- 4) Informar o melhor vizinho encontrado.

3.5 Modelagem dos AGTs

Foram modeladas e desenvolvidas diferentes versões do AGT e suas diferenças consistem na inserção da BT em partes distintas dos AGs e dos AGAs. Inicialmente, a BT é utilizada para otimizar a população inicial dos algoritmos. A seguir, a BT é utilizada para otimizar o resultado gerado pelos AGs e pelos AGAs. A cada um desses algoritmos foram adicionadas as variantes do cálculo do *makespan* desenvolvidas.

A BT é agregada aos AGs e aos AGAs devido às suas características, como memória, estrutura de vizinhança e estratégias de intensificação e diversificação que só agregam valor aos algoritmos. Pois os AGs não possuem um mecanismo que intensificam a busca nas regiões por eles apontadas, o que possibilita ter uma solução próxima muito melhor que a encontrada e seu mecanismo de diversificação é probabilístico. Os AGAs por meio da adaptação de seus parâmetros tentam intensificar a busca nas regiões por eles selecionadas, e também tentam melhorar seu mecanismo de diversificação, mas não garantem que a efetividade destes.

Apesar dos AGs e AGAs terem uma “memória coletiva/populacional”, eles não a utilizam para incrementar seu processo de busca, pois não possuem mecanismos para isto. Possuem apenas mecanismos probabilísticos, de complexa configuração, e não leva em consideração nenhuma informação sobre os espaços de busca por eles visitados, ou seja, os AGs e AGAs não fazem análise de sua população, e não armazena informações dos espaços de busca por eles visitadas, assim não podem utilizar informações do progresso da busca para direcioná-la de forma eficaz. A configuração dos seus parâmetros genéticos é complexa, devido que seus valores geram bons resultados para um determinado problema, mas não necessariamente irá gerar bons resultados para outro problema, ou seja, a configuração dos AGs e AGAs varia de um problema para outro, e geralmente sua configuração de seus parâmetros são feitas na base da tentativa e erro.

3.6 Formas de cálculo do Makespan

Como retratado na seção anterior, os produtos são processados da esquerda para a direita na solução, ou seja, pega-se o elemento mais à esquerda, selecionam-se seus roteiros e alocam-se as respectivas máquinas contidas no roteiro para a manufatura do produto. Esse processo é realizado até que o último produto contido na solução tenha suas operações alocadas em suas respectivas máquinas e, assim, o processo de manufatura também é finalizado.

Segundo Deriz (2007), para garantir o uso simultâneo das máquinas durante o processo de manufatura quando um produto finaliza uma operação em uma máquina, é alocada a esse produto a próxima máquina do roteiro de fabricação. Essa etapa de alocação das máquinas é realizada de acordo com a disponibilidade das mesmas, ou seja, um produto deverá esperar a liberação de uma máquina caso essa se encontre ocupada. Quando uma máquina se torna disponível, o cromossomo é percorrido novamente, seguindo a mesma ordem, até encontrar um produto que necessite dessa máquina para ser processado. Esse processo é realizado até que todas as máquinas de todos os roteiros contidos no cromossomo tenham sido alocadas para os respectivos produtos.

Sanches (2008) diz que, para garantir o uso simultâneo do sistema de transporte, são utilizadas as regras de despacho STT/D e RV, as quais são aplicadas no momento em que uma operação é finalizada, o que torna essa máquina disponível. No entanto, somente essas regras não garantem a otimização do tempo de viagem, ou seja, o sistema de transporte permanece no local de sua última solicitação e deve se deslocar até o ponto em que o transporte foi solicitado.

Portanto, neste trabalho, além dessas regras de despacho, é aplicado um procedimento que prevê as sequências de solicitações realizadas ao sistema de transporte, ou seja, esse procedimento realiza a previsão da próxima solicitação e posiciona o sistema de transporte próximo à máquina que o solicitou, o que diminui o tempo de viagem. As formas de cálculo do *makespan* propostas por Deriz (2007) e Sanches (2008) são remodeladas e implementadas novamente, com a finalidade de avaliar a forma de cálculo proposta neste trabalho. A seguir, encontra-se a descrição das formas de cálculo empregadas.

Para calcular o valor do *makespan* é necessário, primeiramente, obtém-se o tempo de conclusão de cada operação de um produto e, então, pode-se calcular o tempo de conclusão do mesmo. Assim, o *makespan* é o tempo em que a última operação do roteiro do último produto do vetor solução é concluída, ou seja, calcula-se o *makespan*, desde a inserção do

produto no sistema produtivo até que todas as máquinas tenham finalizado as operações alocadas, e não se tenha alguma operação a ser realizada sobre algum produto. A Figura 3-12 ilustra o cálculo do *makespan*, porém esse não leva em consideração o sistema de transporte.

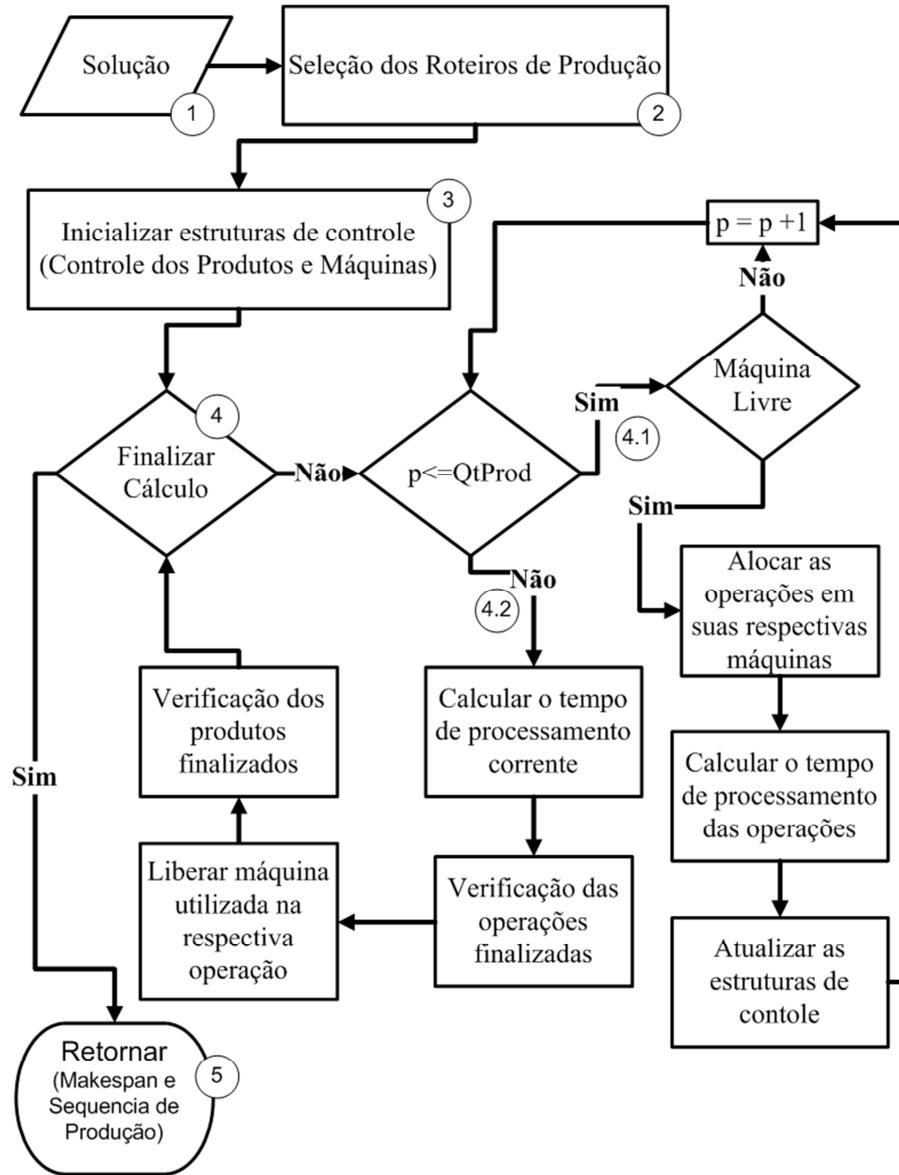


Figura 3-12 - Fluxograma: cálculo do *makespan* – 1^a versão

O cálculo do *makespan* de Deriz (2007) é realizado de acordo com os seguintes passos, ilustrados na Figura 3-12:

1. Obtenção da solução
2. Obtenção dos roteiros de solução, ou seja, da sequência de máquinas a serem utilizadas no processo de manufatura de um determinado produto. Essas informações são armazenadas em uma estrutura que controla a produção. A Figura 3-14 ilustra essa estrutura e nela cada coluna representa um produto da solução e as linhas correspondem às operações sobre cada produto.

3. Inicialização das estruturas de controle.

- 3.1. Controle dos produtos: responsável por armazenar informações sobre o produto, como: a referência à sua posição na solução; a sua posição atual, o seu destino, ou seja, a informação do local onde o produto sofrerá a próxima operação e, por fim, informações sobre as operações já efetuadas sobre o produto. A Figura 3-15 ilustra essa estrutura e nela as colunas representam as informações sobre o produto e as linhas correspondem às referências da posição do produto na solução.
- 3.2. Controle das máquinas: responsável por armazenar informações sobre as máquinas, como: se uma determinada máquina está ocupada ou não; referência sobre o produto a ser manufaturado, ou que já está em processo de manufatura; e o tempo de término da manufatura. A Figura 3-16 ilustra essa estrutura.
4. Verifica-se a finalização da produção, ou seja, se todos os produtos foram manufaturados.
- 4.1. Verifica-se se todos os produtos foram analisados, $p \leq QtProd$, averigua se pode alterar o estado de algum produto no chão de fabrica (transportar, inserir em alguma máquina, iniciar manufatura, etc.).
- 4.1.1. Verifica-se se é possível realizar a próxima operação sobre o produto p e se a máquina necessária está disponível.
- 4.1.1.1. Aloca-se a respectiva máquina para a operação.
- 4.1.1.2. Calcula-se o instante do término da manufatura do produto. Esse instante corresponde ao tempo de processamento corrente acrescido do tempo de manufatura do produto na respectiva máquina.
- 4.1.1.3. Atualizam-se as informações sobre as máquinas e os produtos, ou seja, informa-se que a operação sobre o produto já foi iniciada, busca-se a próxima operação sobre o produto e informa que a respectiva máquina agora está ocupada.
- 4.1.1.4. Passa-se para o próximo produto, $p = p + 1$, e retorna-se ao item (4.1).
- 4.2. Caso não haja mais produtos a serem analisados.
- 4.2.1. Calcula-se o novo tempo de processamento corrente, que corresponde ao menor valor do tempo de término da manufatura dos produtos.
- 4.2.1.1. Verifica-se quais operações foram finalizadas. Caso haja empate, finaliza-se a operação do produto mais à esquerda na solução.
- 4.2.1.2. Libera-se a máquina que realizou a operação.
- 4.2.1.3. Verifica-se se o produto foi finalizado.
- 4.2.1.4. Retorna-se ao item (4).

5. Informa-se o valor do *makespan*.

A fim de realizar o tratamento do sistema de transporte, alterou-se o fluxo do cálculo do *makespan*, ilustrado na Figura 3-12, para o fluxo ilustrado na Figura 3-13. Isso ocorreu devido à incorporação do mecanismo que realiza a programação simultânea do sistema de transporte e das máquinas. Isso foi realizado agregando-se, ao mecanismo de cálculo, as regras de despacho RV e STT/D. Assim, alterou-se a sequência dos passos do cálculo do *makespan*, como indica a Figura 3-13.

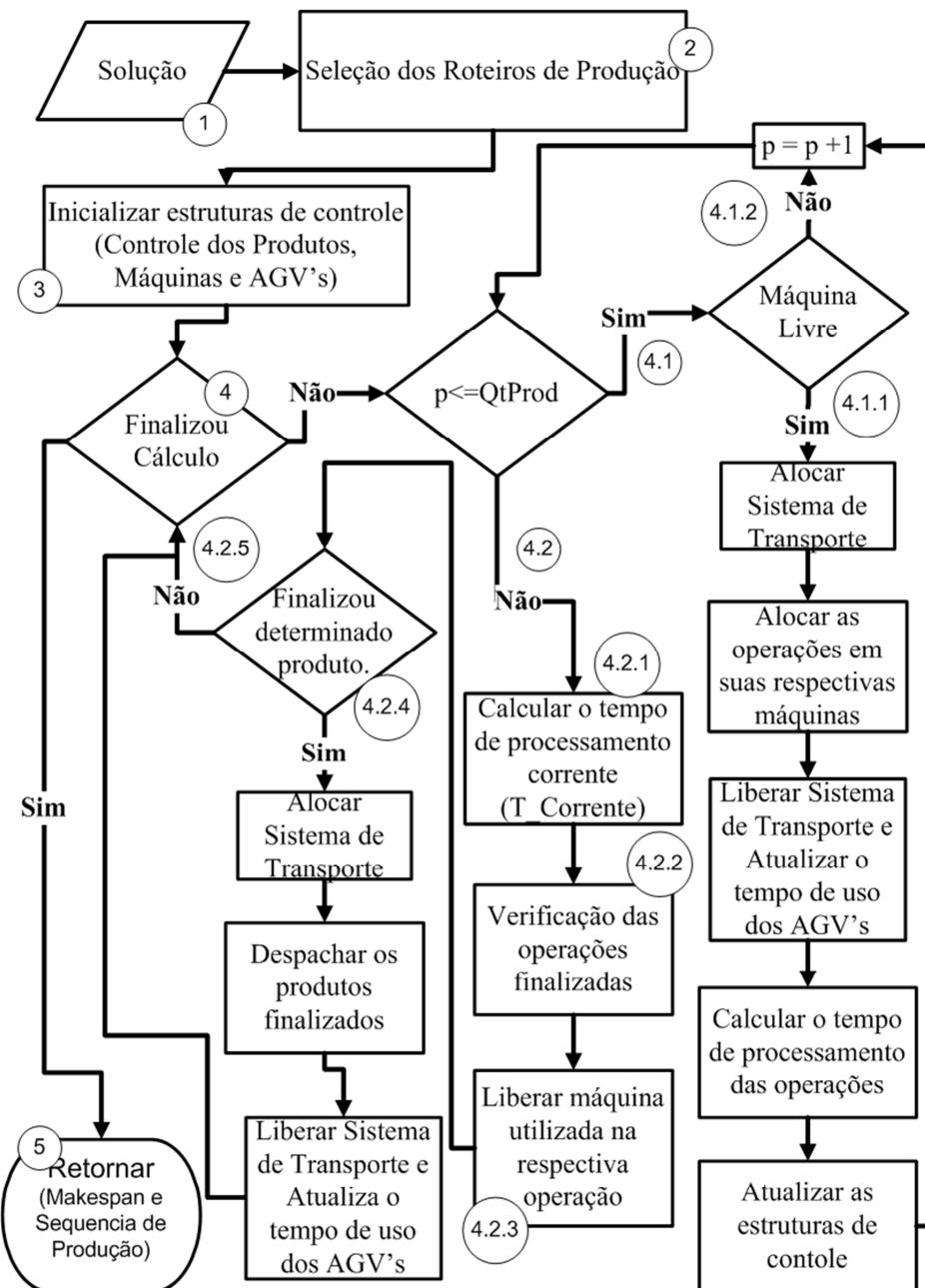


Figura 3-13 - Fluxograma: cálculo do *makespan* – 2^a versão

O cálculo do *makespan* de Sanches(2008) é realizado de acordo com os seguintes passos, ilustrados na Figura 3-13:

1. Obtenção da solução
2. Obtenção dos roteiros de solução, ou seja, da sequência de máquinas a serem utilizadas no processo de manufatura de um determinado produto. Essas informações são armazenadas em uma estrutura que controla a produção. A Figura 3-14 ilustra essa estrutura e nela cada coluna representa um produto da solução e as linhas correspondem às operações sobre cada produto.
3. Inicialização das estruturas de controle.
 - 3.1. Controle dos produtos: responsável por armazenar informações sobre o produto, como: a referência à sua posição na solução; a sua posição atual, o seu destino, ou seja, a informação do local onde o produto sofrerá a próxima operação e, por fim, informações sobre as operações já efetuadas sobre o produto. A Figura 3-15 ilustra essa estrutura e nela as colunas representam as informações sobre o produto e as linhas correspondem às referências da posição do produto na solução.
 - 3.2. Controle das máquinas: responsável por armazenar informações sobre as máquinas, como: se uma determinada máquina está ocupada ou não; referência sobre o produto a ser manufaturado, ou que já esta em processo de manufatura; e o tempo de término da manufatura. A Figura 3-16 ilustra essa estrutura.
 - 3.3. Controle dos AGVs: responsável por armazenar informações sobre os AGVs, como: se determinado AGV está em uso e qual o posicionamento dos mesmos no sistema produtivo. A Figura 3-17 ilustra essa estrutura.
4. Verifica-se a finalização da produção, ou seja, se todos os produtos foram manufaturados.
 - 4.1. Verifica-se se todos os produtos foram analisados, $p \leq QtProd$, averigua se pode alterar o estado de algum produto no chão de fabrica (transportar, inserir em alguma máquina, iniciar manufatura, etc.).
 - 4.1.1. Verifica-se se é possível realizar a próxima operação sobre o produto p e se a máquina necessária está disponível.
 - 4.1.1.1. Aloca-se o sistema de transporte, ou seja, seleciona-se o veículo que fará o transporte do produto em menor tempo.
 - 4.1.1.2. Aloca-se a respectiva máquina para a operação.
 - 4.1.1.3. Libera-se o sistema de transporte e atualiza-se o tempo de uso dos AGVs (T_{AGV}).

4.1.1.4. Calcula-se o instante do término da manufatura do produto. Esse instante corresponde ao tempo de processamento corrente, acrescido do tempo de manufatura do produto na respectiva máquina e do tempo de transporte do respectivo produto até a máquina.

4.1.1.5. Atualizam-se as informações sobre as máquinas e os produtos, ou seja, informa-se que a operação sobre o produto já foi iniciada, busca-se a próxima operação sobre o produto e informa que a respectiva máquina agora está ocupada.

4.1.2. Passa-se para o próximo produto, , e retorna-se ao item (4.1).

4.2. Caso a máquina não esteja disponível, passa-se para o próximo produto, e retorna-se ao item (4.1).

4.2.1. Caso não haja mais produtos a serem analisado.

4.2.1.1. Calcula-se o novo tempo de processamento corrente, que corresponde ao menor valor do tempo de término da manufatura dos produtos.

4.2.1.2. Verificam-se quais operações foram finalizadas. Caso haja empate, finaliza-se a operação do produto mais à esquerda na solução.

4.2.1.3. Libera-se a máquina que realizou a operação.

4.2.1.4. Verifica-se se o produto foi finalizado, caso tenha:

- 1) Aloca-se o sistema de transporte.
- 2) Despacha-se o produto para a estação de descarga.
- 3) Libera-se o sistema de transporte e atualiza-se o tempo de uso dos AGVs, (T_AGV).

4.2.1.5. Retorna-se ao item (4).

5. Informa-se o valor do *makespan*.

Operações Máquinas	Produtos	P1	P2	P3	P4	P5	P6	P7	P8	P9
		X4	X8	X6	X3	X2	X5	X9	X7	X1
	Y1	Y1	Y1	Y1	Y1	Y1	Y1	Y1	Y1	Y1
	Y2	Y2	Y2	Y2	Y2	Y2	Y2	Y2	Y2	Y2
	Y3	Y3	Y3	Y3	Y3	Y3	Y3	Y3	Y3	Y3

	Y7	Y7	Y7	Y7	Y7	Y7	Y7	Y7	Y7	Y7

Figura 3-14 - Estrutura representativa do controle das operações sobre os produtos

Produto	Origem	Destino	Controle
P1	C/D	Y1	2
P2	C/D	Y1	2
P3	C/D	Y1	2
P4	C/D	Y1	2
...
P9	C/D	Y1	2

Figura 3-15 - Estrutura representativa do controle dos produtos

Máquina	Ocupada	Produto	Tempo
M1	1	P1	*
M2	1	P4	#
M3	0	P9	\$
M4	0	-	-
...
MP	0	-	-

Figura 3-16 - Estrutura representativa do controle das máquinas

AGV's Posição Temporada
AGV1
AGV2
AGV3

Figura 3-17 - Estrutura representativa do controle dos AGVs

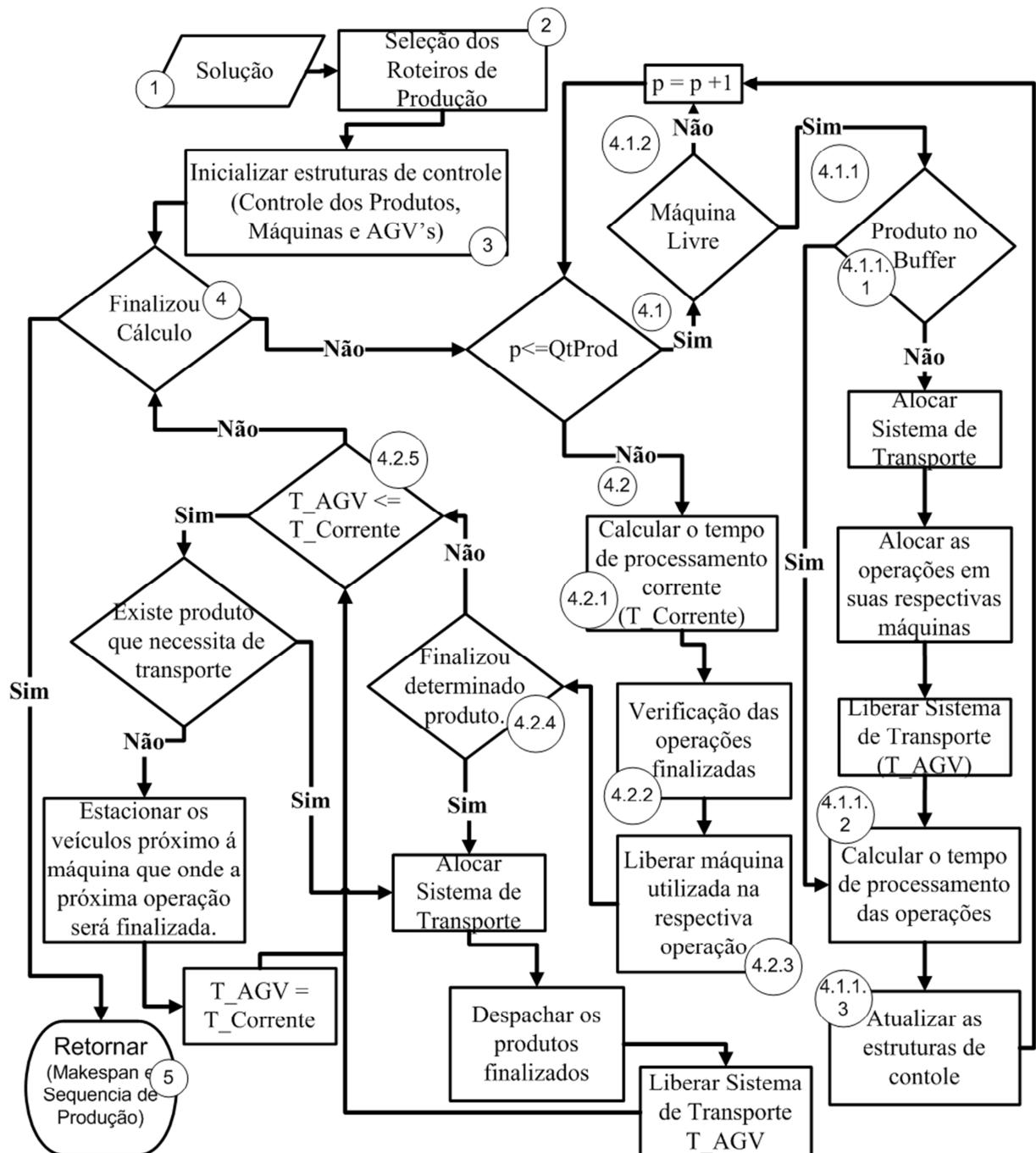


Figura 3-18 - Fluxograma do cálculo do makespan – 3^a versão

Até o momento, nenhuma dos métodos de cálculo do *makespan* não se preocupou em adiantar, predizer as próximas operações do sistema de transporte, de forma a antecipar o transporte do produto ou a operação que será finalizada no instante seguinte e estacionar o veículo o mais próximo possível da máquina que realizará a operação. Assim, alterou-se a sequência dos passos do cálculo do *makespan*, como indica a Figura 3-18.

Assim, o cálculo do *makespan* proposto neste trabalho, e é realizado de acordo com os seguintes passos:

1. Obtenção da solução
2. Obtenção dos roteiros de solução, ou seja, da sequência de máquinas a serem utilizadas no processo de manufatura de um determinado produto. Essas informações são armazenadas em uma estrutura que controla a produção. A Figura 3-14 ilustra essa estrutura e nela cada coluna representa um produto da solução e as linhas correspondem às operações sobre cada produto.
3. Inicialização das estruturas de controle.
 - 3.1. Controle dos produtos: responsável por armazenar informações sobre o produto, como: a referência à sua posição na solução; a sua posição atual, o seu destino, ou seja, a informação do local onde o produto sofrerá a próxima operação e, por fim, informações sobre as operações já efetuadas sobre o produto. A Figura 3-15 ilustra essa estrutura e nela as colunas representam as informações sobre o produto e as linhas correspondem às referências da posição do produto na solução.
 - 3.2. Controle das máquinas: responsável por armazenar informações sobre as máquinas, como: se uma determinada máquina está ocupada ou não; referência sobre o produto a ser manufaturado, ou que já esta em processo de manufatura; e o tempo de término da manufatura. A Figura 3-16 ilustra essa estrutura.
 - 3.3. Controle dos AGVs: responsável por armazenar informações sobre os AGVs, como: se determinado AGV está em uso e qual o posicionamento dos mesmos no sistema produtivo. A Figura 3-17 ilustra essa estrutura.
4. Verifica-se a finalização da produção, ou seja, se todos os produtos foram manufaturados.
 - 4.1. Verifica-se se todos os produtos foram analisados, $p \leq QtProd$, averigua se pode alterar o estado de algum produto no chão de fabrica (transportar, inserir em alguma máquina, iniciar manufatura, etc.).
 - 4.1.1. Verifica-se se é possível realizar a próxima operação sobre o produto p e se a máquina necessária está disponível.

- 4.1.1.1. Aloca-se o sistema de transporte, ou seja, seleciona-se o veículo que fará o transporte do produto em menor tempo.
- 4.1.1.2. Aloca-se a respectiva máquina para a operação.
- 4.1.1.3. Libera-se o sistema de transporte e atualiza-se o tempo de uso dos AGVs (T_{AGV}).
- 4.1.1.4. Calcula-se o instante do término da manufatura do produto. Esse instante corresponde ao tempo de processamento corrente, acrescido do tempo de manufatura do produto na respectiva máquina e do tempo de transporte do respectivo produto até a máquina.
- 4.1.1.5. Atualizam-se as informações sobre as máquinas e os produtos, ou seja, informa-se que a operação sobre o produto já foi iniciada, busca-se a próxima operação sobre o produto e informa que a respectiva máquina agora está ocupada.
- 4.1.1.6. Passa-se para o próximo produto, $p=p+1$, e retorna-se ao item (4.1).
- 4.1.2. Caso a máquina não esteja disponível, passa-se para o próximo produto, $p = p + 1$ e retorna-se ao item (4.1).
- 4.2. Caso não haja mais produtos a serem analisado.
 - 4.2.1. Caso não haja mais produtos a serem analisado.
 - 4.2.1.1. Calcula-se o novo tempo de processamento corrente, que corresponde ao menor valor do tempo de término da manufatura dos produtos.
 - 4.2.1.2. Verificam-se quais operações foram finalizadas. Caso haja empate, finaliza-se a operação do produto mais à esquerda na solução.
 - 4.2.1.3. Libera-se a máquina que realizou a operação.
 - 4.2.1.4. Verifica-se se o produto foi finalizado, caso tenha:
 - 1) Aloca-se o sistema de transporte
 - 2) Despacha-se o produto para a estação de descarga
 - 3) Libera-se o sistema de transporte e atualiza-se o tempo de uso dos AGVs, (T_{AGV}).
 - 4.2.1.5. Verifica-se se há a possibilidade de adiantar o transporte dos produtos, ou seja, se algum produto está na estação de carga ou no buffer de saída das máquinas e pode ser levado para o buffer de entrada da próxima máquina de seu roteiro de produção. Caso seja possível:
 - 1) Aloca-se o sistema de transporte.
 - 2) Despacha-se o produto para o *buffer* de entrada da respectiva máquina.

- 3) Libera-se o sistema de transporte e atualiza-se o tempo de uso dos AGVs (T_{AGV}).

4.2.1.6. Caso não seja possível adiantar o sistema de transporte:

- 1) Estaciona-se os veículos próximos à máquina onde a próxima operação será finalizada e iguala-se o tempo de uso dos AGVs ao tempo corrente.

4.2.1.7. Retorna-se ao item (4).

5. Informa-se o valor do *makespan*.

Segundo Sanches (2008), para encontrar o tempo de conclusão de uma determinada tarefa (operação sobre um produto) é necessário inicialmente encontrar o tempo de conclusão de uma determinada operação de um produto, que é dado por (1). O cálculo do *makespan* com uso simultâneo de sistemas de transporte e máquinas pode ser representado pelas equações (1), (2) e (3), relacionadas a seguir.

$$O_{ij} = T_{ij} + P_{ij} + E_{ij} \quad (1)$$

Com isso, é possível encontrar o tempo total para a obtenção de um determinado produto.

$$P_i = \sum_{j=1}^n O_{ij} \quad (2)$$

Onde: O_{ij} corresponde ao tempo de conclusão da operação sobre um produto, T_{ij} corresponde ao tempo de transporte, P_{ij} é o tempo de processamento da operação, E_{ij} é o tempo de espera que a operação aguardou até o momento de ser processada.

Assim sendo, é possível encontrar o *makespan* (MKP), baseando-se no tempo de obtenção do último produto ou no valor máximo encontrado em P_i .

$$MKP = \text{Max}(P_i) \quad (3)$$

Portanto, para gerar o *makespan* com a programação simultânea do sistema de transporte e das máquinas, inseriu-se somente a variável T , que corresponde ao tempo de transporte na equação (1), proposta inicialmente por Deriz (2007).

Já para o desenvolvimento do cálculo do *makespan* proposto neste trabalho, cuja estrutura de fluxo é apresentada na Figura 3-18 não é acrescentada nenhuma outra variável de controle, mas é inserido, juntamente com as regras de despacho, um mecanismo que realiza o adiantamento das tarefas do sistema de transporte, como demonstrado anteriormente.

4 Cenário de Testes e Desenvolvimento

Neste capítulo é apresentado o cenário de testes (estudo), seguido da descrição dos testes desenvolvidos do sistema, e dos testes dos módulos de busca, dos módulos de cálculo do *makespan* e da descrição dos testes de validação da proposta.

4.1 Cenário

O arranjo físico do sistema de manufatura considerado neste trabalho é apresentado na Figura 4-1. O sistema é composto por nove estações de trabalho, uma estação de carga e descarga, uma estação de manutenção de AGVs e três AGVs. Os produtos iniciam seu processo de produção por meio da estação de carga e retirados do processo através da estação de descarga.

São manufaturados nove produtos distintos ($P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8$ e P_9). As informações referentes a cada um desses produtos, em relação aos roteiros de produção, aos tempos de produção e aos tempos de transporte, são apresentadas a seguir.

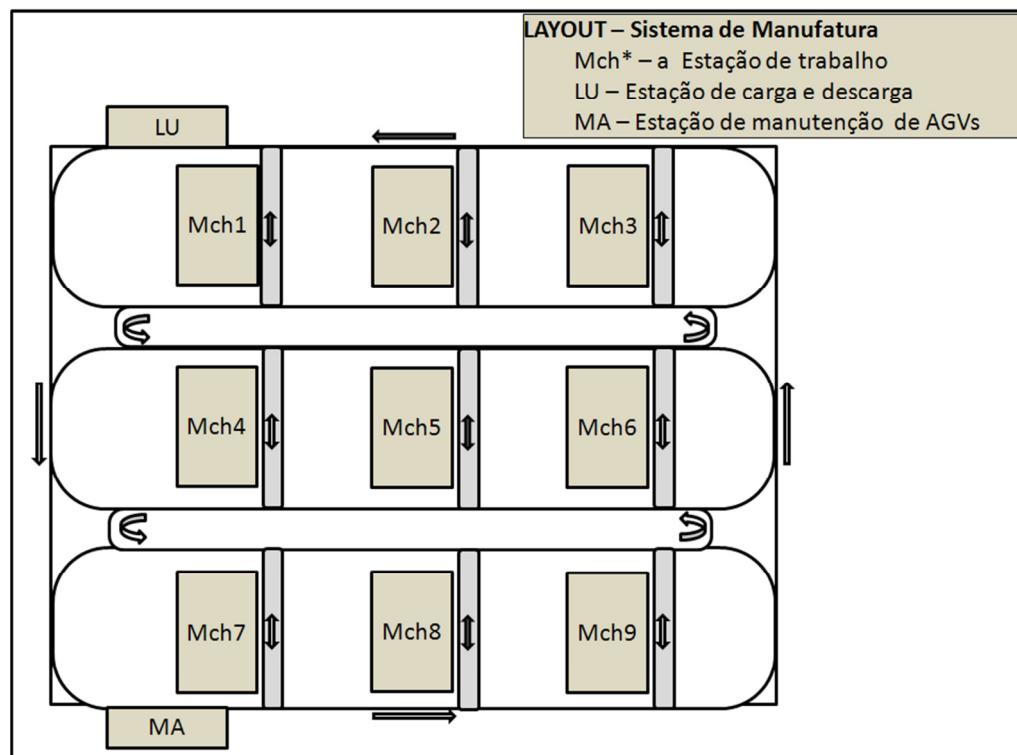


Figura 4-1 - Layout do Sistema Produtivo, adaptado de Morandin (1999)

Os roteiros de produção dos nove produtos pertencentes ao sistema produtivo são demonstrados na Figura 4-2.

Os roteiros de fabricação dos produtos e os tempos de produção foram gerados aleatoriamente. Esses últimos, que variam de 400 a 500 unidades de tempo (u.t.), encontram-se ilustrados na Figura 4-3. Já os tempos de transporte, calculados de forma empírica, encontram-se na Figura 4-4. O tempo de transporte corresponde ao tempo que um veículo leva para movimentar um produto de um ponto a outro do sistema produtivo, ou seja, o tempo em que o veículo leva para alcançar uma máquina ou estação de carga/descarga à partir da sua posição atual.

	Roteiros	Máquinas
Produtos	<i>R1</i>	1 2 4 5 7 9 0
	<i>R2</i>	3 4 5 6 8 9 0
	<i>R1</i>	1 2 3 4 5 6 7
	<i>R2</i>	2 3 5 7 8 9 0
	<i>R1</i>	4 5 6 7 8 0 0
	<i>R2</i>	2 3 7 8 9 0 0
	<i>R1</i>	2 3 4 6 7 0 0
	<i>R2</i>	1 5 6 8 9 0 0
	<i>R1</i>	4 5 7 8 9 0 0
	<i>R2</i>	1 2 3 5 6 0 0
	<i>R1</i>	2 4 5 6 7 8 9
	<i>R2</i>	1 3 6 7 8 9 0
	<i>R1</i>	1 2 4 5 6 9 0
	<i>R2</i>	1 2 3 7 8 9 0
	<i>R1</i>	4 5 6 7 8 9 0
	<i>R2</i>	3 4 5 7 8 9 0
	<i>R1</i>	3 5 6 7 8 9 0
	<i>R2</i>	2 4 6 7 8 9 0

Figura 4-2 - Roteiros de Produção

	P1	P2	P3	P4	P5	P6	P7	P8	P9	
M1	428	439	453	403	481	446	414	491	458	M1
M2	423	433	474	436	440	495	457	419	486	M2
M3	459	487	417	410	477	474	452	435	416	M3
M4	433	405	447	410	442	448	426	491	454	M4
M5	467	447	486	400	450	469	493	495	452	M5
M6	461	497	496	468	468	408	408	452	438	M6
M7	464	495	459	489	436	454	457	477	484	M7
M8	455	469	489	439	486	424	497	452	435	M8
M9	418	439	480	457	435	482	445	408	416	M9
	P1	P2	P3	P4	P5	P6	P7	P8	P9	

Figura 4-3 - Tempo de Produção

	C	M1	M2	M3	M4	M5	M6	M7	M8	M9	D	
C	0	14	5	10	12	14	10	13	11	9	13	C
M1	11	0	11	6	5	6	5	9	11	6	11	M1
M2	6	9	0	7	5	6	10	6	9	11	13	M2
M3	10	7	5	0	11	5	10	12	14	8	6	M3
M4	7	13	8	9	0	10	11	6	14	7	9	M4
M5	9	10	8	11	10	0	7	13	9	12	9	M5
M6	14	11	13	7	14	6	0	12	13	14	8	M6
M7	10	7	9	13	5	8	10	0	8	10	13	M7
M8	11	6	9	6	8	10	12	13	0	9	7	M8
M9	6	5	6	8	9	10	9	5	4	0	8	M9
D	6	5	11	8	10	11	8	5	14	13	0	D
	C	M1	M2	M3	M4	M5	M6	M7	M8	M9	D	

Figura 4-4 - Tempo de Transporte

4.2 Descrição dos testes do sistema de validação da proposta

Para garantir a qualidade do software, ele deve passar por uma sequência de testes. O objetivo do teste neste trabalho é tentar garantir que a proposta desenvolvida esteja livre de eventuais erros nas principais operações do sistema desenvolvido.

Para isso, foram aplicados testes em nível unitário, que segundo Louridas (2005). Este tipo de teste, verifica a implementação de um módulo ou função do software. Assim, é possível garantir que a lógica do programa esteja completa e correta.

Como a atividade de executar testes é uma tarefa exaustiva, na maioria das vezes, uma sugestão seria utilizar a automatização dos mesmos. O uso de testes automatizados permite ao desenvolvedor verificar se mudanças no código fonte não se propagam para outras classes e outros requisitos. Para auxiliar no uso de testes automatizados, neste trabalho foram desenvolvidos módulos no MATLAB®, os quais verificam se o resultado correto é o mesmo obtido após a execução do teste. Se os resultados forem idênticos, o módulo testado está livre de erros; caso contrário, é falho (LOURIDAS, 2005).

Portanto, o unitário foi utilizado neste trabalho para validar as etapas mais importantes do sistema, listadas abaixo:

- Função Fitness,
- Operação de Cruzamento,
- Operação de Mutação,
- Cálculo do *Makespan*,
- Geração do Cromossomo (ou seja, da Solução),
- Geração da População,
- Geração da vizinhança,
- Obtenção do melhor vizinho,
- Intensificação,
- Diversificação.

Os módulos foram testados para 10 exemplos distintos e cada um deles respondeu a todos os testes, corretamente. Portanto, os módulos possuem 0% de erro e 100% de acerto.

4.3 Descrição dos testes de validação da proposta

No próximo capítulo são apresentados os resultados obtidos a partir da proposta deste trabalho, além de algumas análises sobre os resultados de *makespan*, o tempo de resposta da execução e o desempenho dos algoritmos desenvolvidos. O resultado dessas análises é comparado aos resultados de outras abordagens testadas neste trabalho.

Neste contexto, experimentos foram realizados e os resultados encontrados foram comparados com os obtidos por Deriz (2007) e Sanches (2008), nos quais utilizam AG e AGA, respectivamente, como métodos de busca para o problema de programação reativa da produção.

Também foram realizadas as análises dos métodos de seleção a serem utilizados, das formas de aplicação das vizinhanças propostas, dos momentos de aplicação da mutação e dos cálculos dos *makespans* propostos por Sanches (2008) e por Deriz (2007). Vale ressaltar que essas duas propostas de *makespan* foram reimplementadas por meio do MATLAB® e testadas. As análises são apresentadas no próximo capítulo.

Neste trabalho, foi utilizado um cenário específico para a validação do método proposto, já descrito na seção anterior. Para esse cenário, foram definidos 50 testes estatísticos, que visam obter amostragens do desempenho dos algoritmos desenvolvidos, estes são avaliados quanto ao tempo necessário para obtenção da solução e valor de *makespan* fornecido em cada teste. Isto é necessário devido às características não determinísticas dos algoritmos usados na proposta (AG, AGA, BT e AGT), assim os resultados variam para cada teste.

Como os dados obtidos nos testes não pertencem à distribuição normal (conhecida também como Distribuição de Gauss ou Gaussiana) pode-se aplicar testes não paramétricos para a análise dos resultados obtidos. Dentre os possíveis, foi utilizado o teste de Wilcoxon (MARTINS, 2005), para verificar o grau de confiança dos resultados obtidos em relação às outras abordagens testadas.

No teste de Wilcoxon, uma amostra A1 é submetida a um tratamento T1 e tem seu efeito medido. Posteriormente, essa mesma amostra, chamada agora de A2, é submetida a um segundo tratamento T2 e tem seu efeito medido pela mesma variável usada no primeiro tratamento. Comparando-se o efeito dos dois tratamentos em cada elemento da amostra, pode-se verificar se o efeito aumentou, diminuiu ou permaneceu o mesmo. Por meio desse teste, pode-se analisar se os resultados obtidos pelos dois tratamentos aplicados à mesma amostra apresentam diferenças estatisticamente relevantes, levando em conta a magnitude do aumento

ou da diminuição dos resultados e não apenas a direção da variação para mais ou para menos (CAMPOS, 2007).

Portanto, o teste de Wilcoxon foi utilizado para comparar os resultados de *makespan* obtidos neste trabalho com os obtidos por Deriz (2007) e Sanches (2008). Mesmo que o trabalho de Deriz (2007) não considerou o sistema de transporte, a sua avaliação foi necessária, pois somente através dessa análise pôde-se avaliar a influência do sistema de transporte no cálculo do *makespan* e as vantagens do cálculo proposto.

A apresentação dos resultados no próximo capítulo segue a ordem abaixo:

- Apresentação da análise dos métodos de cálculo do *makespan*, sendo que os propostos por Deriz (2007) e por Sanches (2008) foram remodelados e desenvolvidos por meio do MATLAB®;
- Apresentação da análise e definição do melhor método de seleção aplicado ao problema, dentre os utilizados por Deriz (2007) e por Sanches (2008);
- Apresentação da análise e definição do momento de aplicação do operador de mutação, que pode ocorrer em dois momentos distintos;
- Apresentação da análise da aplicação dos AGs e dos AGAs e comparação com os resultados obtidos por Deriz (2007) e Sanches (2008);
- Apresentação da análise da vizinhança e da aplicação das BTs ao problema em questão;
- Apresentação da análise dos AGTs desenvolvidos. A aplicação da BT é analisada em dois momentos distintos: na otimização da população inicial dos AGs e dos AGAs e na otimização dos resultados fornecidos pelos algoritmos.

Em todos os algoritmos desenvolvidos (AGs, AGAs, BTs e AGTs) são aplicados tanto o cálculo do *makespan* proposto, quanto os cálculos propostos por Deriz (2007) e Sanches (2008).

Para cada algoritmo, são realizados 50 testes sucessivos. A partir desses resultados, analisa-se a média e a dispersão desses valores em relação à média. Essa dispersão é obtida por meio do desvio padrão, obtido no conjunto de 50 testes. Esses testes são realizados em uma máquina com a seguinte configuração: Intel® Core™ 2 Duo, de 2.0 GHz, com 2 GB de memória RAM e em um sistema operacional de 32 Bits (Windows Vista™ Home Basic).

5 Análise e discussão dos resultados

Como apresentado no capítulo anterior, o teste de Wilcoxon é utilizado para comparar os resultados obtidos. Para a definição da amostragem é utilizada a amostragem aleatória simples, onde se seleciona, aleatoriamente, um subconjunto do espaço amostral, ao qual são aplicados os mecanismos de testes e a validação da proposta. Para este trabalho se definiu que esse subconjunto teria um tamanho fixo correspondente a 50 indivíduos.

5.1 Versões de cálculo do *makespan*

Neste trabalho, além da forma de cálculo do *makespan* proposto, o qual realiza o adiantamento das tarefas do sistema de transporte, também foram desenvolvidas as propostas de cálculo de Deriz (2007) e de Sanches (2008), fazendo-se o uso do MATLAB®. Isto foi necessário, pois ambos os autores desenvolverem e validaram seus trabalhos fazendo uso da IDE Eclipse (linguagem Java). Além disso, elas precisaram ser desenvolvidas a fim de avaliar a influência do sistema de transporte no cálculo do *makespan* e para validar a forma de cálculo proposta neste trabalho.

Deriz (2007) não faz uso do sistema de transporte, pois considera o tempo de transporte desprezível. Já Sanches (2008) inclui o sistema de transporte no cálculo do *makespan*. Porém ele somente é iniciado quando um determinado produto necessita ser movimentado de um lugar para outro no sistema produtivo. A antecipação das tarefas do sistema de transporte pode, muitas vezes, antecipar as operações sobre algum produto, de forma a gerar um menor valor *makespan*. Devido a isso, fez-se necessário o desenvolvimento de uma forma de cálculo que antecipa as tarefas do sistema de transporte.

A Tabela 5-1 ilustra os resultados obtidos por meio da aplicação dos métodos de cálculo desenvolvidos no MATLAB®, a um conjunto de dados gerados aleatoriamente. Cada forma de cálculo é aplicada a todos os indivíduos pertencentes ao conjunto.

Na tabela, observa-se que a antecipação do sistema de transporte aproximou o valor do *makespan* obtido ao de Deriz (2007). No pior teste, reduziu-se pela metade a diferença entre os valores obtidos por Deriz (2007) e Sanches (2008) e, no melhor caso, os resultados obtidos ficaram muito próximos aos de Deriz (2007). Isso porque as operações de transporte e o uso das máquinas ocorrem em paralelo, diferentemente do trabalho de Sanches (2008), que só ativa o transporte após o término do processamento de uma determina

operação em uma determinada máquina. Também se observa que a aplicação do sistema de transporte provoca alterações significativas no valor do *makespan*.

Tabela 5-1 - Testes dos cálculos dos *makespans*

	Deriz (2007) Makespan(u.t.)	Sanches (2008) Makespan(u.t.)	Proposta Makespan(u.t.)
1	5489	5785	5561
2	5873	6137	5900
3	5860	6464	6205
4	5870	6077	5900
5	5881	6026	5893
6	5643	5899	5679
7	5928	6174	5948
8	5442	5697	5460
9	5447	5665	5459
10	5908	6176	5926
11	5838	6094	5852
12	5933	6191	5951
13	6322	6566	6341
14	5942	6225	5954
15	5816	6164	5900
16	6366	6653	6386
17	5998	6268	6010
18	5884	6135	5900
19	5469	5707	5481
20	6282	6578	6294
21	5475	5711	5525
22	5152	5357	5172
23	5987	6245	5998
24	5489	5721	5515
25	5480	5707	5490
26	6230	6478	6266
27	5375	5655	5413
28	6293	6536	6302
29	6297	6540	6307
30	5447	5659	5462
31	5881	6026	5893
32	5643	5899	5679
33	5928	6174	5948
34	5442	5697	5460
35	5447	5665	5459
36	5908	6176	5926
37	5838	6094	5900
38	5933	6191	5951
39	6322	6566	6341
40	5942	6225	5954
41	5816	6164	5900
42	6230	6478	6266
43	5375	5655	5413
44	6293	6536	6302
45	6297	6540	6307
46	5447	5659	5462
47	5881	6026	5893
48	5643	5899	5679
49	5870	6174	5900
50	5881	6094	5900
Média	5813,87	6076,33	5848,33

5.2 Seleção por Roleta X Seleção por Torneio

Para testar os métodos de seleção, primeiramente, gerou-se uma população com 30 (trinta) indivíduos, a seguir, calculou-se o fitness de cada indivíduo e, de posse desses valores, deu-se início a aplicação sucessiva dos métodos de seleção. Cada método é utilizado na seleção de 30 (trinta) indivíduos e, a cada seleção, são calculadas as seguintes informações: a média aritmética, o *makespan* do melhor e do pior indivíduo, assim como a divergência inferior e superior da média, obtida por meio do desvio padrão. O valor do desvio padrão é adicionado ao valor da média, de forma a se obter a divergência superior (limite superior). Além disso, o valor do desvio padrão é subtraído do valor da média, o que permite obter a divergência inferior (limite inferior). Essa divergência informa a faixa de valores na qual se encontra a maior parte dos indivíduos da amostra (da população).

Na Figura 5-1 e na Figura 5-2 são demonstrados os desempenhos dos métodos em questão. Como se pode observar, o método da roleta realiza uma seleção representativa da população inicial, pois não há grandes variações em seus pontos médios. Já o método do torneio, que geralmente seleciona os melhores indivíduos, provoca mudanças nas médias da próxima população. Esse último, porém, apesar de selecionar os melhores indivíduos para a reprodução, possui alguns inconvenientes quando aplicado por muitas gerações, pois torna a população elitista, homogênea.

A Tabela 5-2 apresenta a população inicial à qual foram aplicados os métodos de seleção, assim como os indivíduos selecionados na segunda execução dos métodos. Como se observa ao se comparar as médias das seleções entre os métodos, o torneio seleciona os indivíduos mais aptos, reduzindo, assim, a média geral da população, porém a roleta seleciona uma população com maior diversidade entre os indivíduos, o que favorece a diversidade genética da população.

Na Tabela 5-3 são demonstrados os resultados obtidos à partir da aplicação de cada método tanto ao AG, quanto ao AGA, cada um executado 30 (trinta) vezes. Ao se comparar a média aritmética dos resultados, observa-se que tanto a Seleção por Torneio, quanto a Seleção por Roleta, obtiveram resultados aproximados. A cada teste, os AGs tiveram ponto de partida comum, ou seja, iniciaram seus processos de busca por uma mesma população, assim como também iniciaram-se com os mesmos parâmetros genéticos.

Como observado, o método de torneio torna a população elitista, isto é, homogênea, pois sempre privilegia os melhores indivíduos. A aplicação do mecanismo de diversificação não provocou mudanças significativas na diversidade da população, pois

grande parte dos indivíduos da população final dos algoritmos possuía um mesmo valor de aptidão, ou valores muito próximos, geralmente se distanciando do melhor indivíduo. O contrário ocorreu na aplicação do método da roleta, que favorece a heterogeneidade da população. Portanto, neste trabalho, utiliza-se o método de seleção por roleta, pois se visa uma população mais heterogênea. Os códigos desses métodos de seleção, assim como a suas descrições, encontram-se no Apêndice A – Métodos de Seleção (Roleta e Torneio).

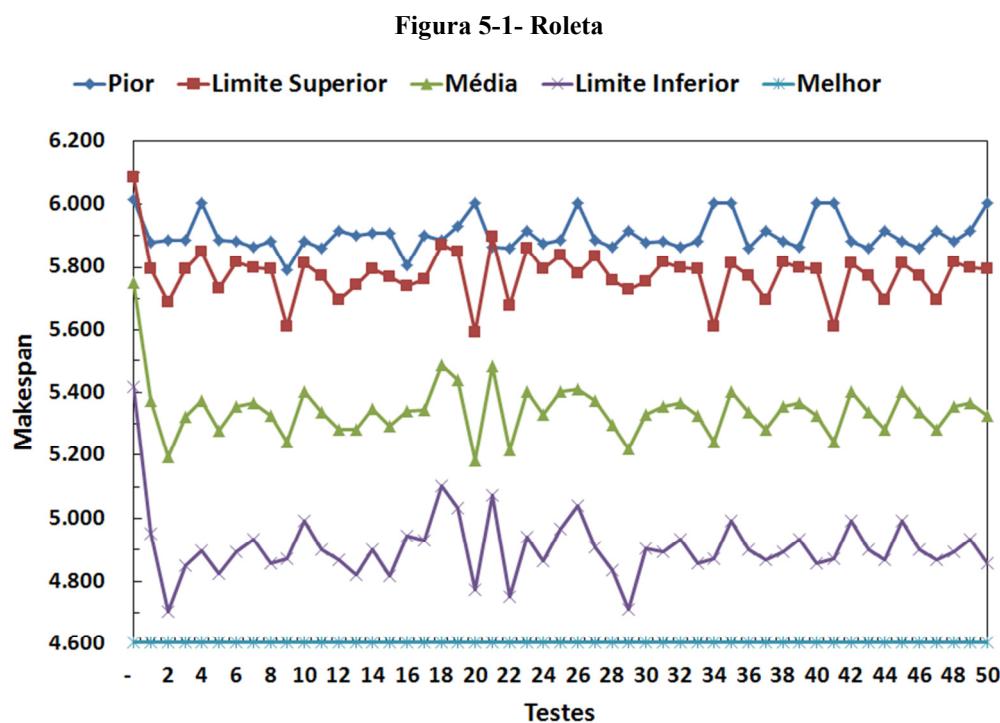
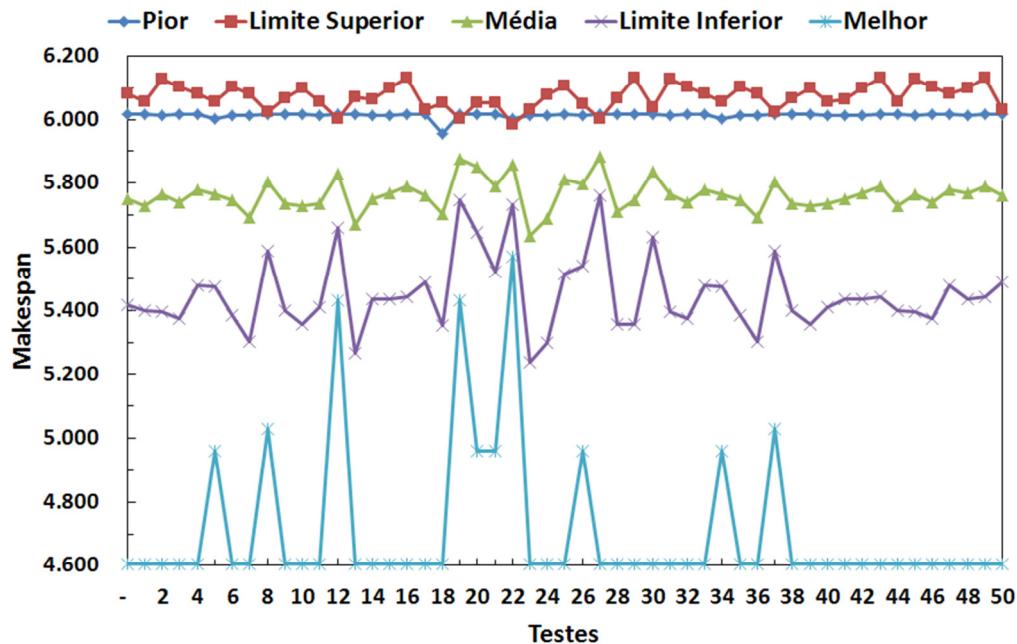


Tabela 5-2 - Roleta X Torneio

População	Roleta	Torneio
4606	6013	4606
5907	5741	5884
5595	5860	5791
5433	5741	4961
5931	5741	5595
5887	5902	5726
5791	5772	4961
5807	5875	5570
5902	5917	5433
4961	5875	4606
5877	6013	5433
5884	5931	4606
5939	5860	4961
5726	6013	4606
5917	5907	4606
5917	5961	5029
5839	5887	5433
5860	5839	5029
6015	5839	5772
6003	4961	4606
6013	6013	5887
5954	5772	4606
5961	5839	5863
5570	5807	5772
5875	4606	4606
5029	5961	5741
5931	4606	5029
5741	5902	4961
5772	5839	5595
5863	5931	4606
5931	5741	5595
5887	5902	5726
5791	5772	4961
5807	5875	5570
5902	5917	5433
4961	5875	4606
5877	6013	5433
5884	5931	4606
5939	5860	4961
5726	6013	4606
5877	6013	5433
5884	5931	4606
5939	5860	4961
5726	6013	4606
5917	5907	4606
5917	5961	5029
5839	5887	5433
5860	5839	5029
6015	5839	5772
6003	4961	4606
6013	6013,00	5887,00
5.750,2	5.764,1	5.196,0

Tabela 5-3 - Aplicação dos métodos de seleção ao AG e AGA

AG		AGA	
Roleta	Torneio	Roleta	Torneio
4988	4990	4976	4956
4933	4990	4595	4610
4968	4586	5000	4956
4975	4586	4834	4956
4900	4586	4600	4956
4637	4586	4600	4956
4590	4586	4956	4600
4970	4586	4927	4600
4993	4946	4595	4956
4834	4990	4956	4956
4637	4900	4950	4956
4946	4987	4995	4595
4586	4586	4989	4600
4937	4586	4595	4956
4945	5019	4600	4956
4594	4586	4981	4981
4900	4900	4966	4600
4937	4586	4937	4956
4968	4586	4595	4600
4966	4990	4955	4600
4994	4946	4981	4610
4905	4586	4927	4600
4608	4586	4956	4956
4586	4990	4927	4956
4937	4990	4955	4956
4905	4990	4600	4956
4987	4590	4956	4981
4608	4990	4955	4956
4673	4946	4956	4981
4586	4586	4974	4956
4993	4946	4595	4956
4834	4990	4956	4956
4637	4900	4950	4956
4946	4987	4995	4595
4586	4586	4989	4600
4937	4586	4595	4956
4945	5019	4600	4956
4594	4586	4981	4981
4900	4900	4966	4600
4937	4586	4937	4956
4608	4586	4956	4956
4586	4990	4927	4956
4937	4990	4955	4956
4905	4990	4600	4956
4987	4590	4956	4981
4608	4990	4955	4956
4673	4946	4956	4981
4586	4586	4974	4956
4993	4946	4595	4956
4834	4990	4956	4956
4.820,4	4.801,0	4.863,7	4.866,6

5.3 Modelagem dos AGs e AGAs

Para testar as variantes de modelagem e de implementação dos algoritmos, as quais se diferenciam no momento de aplicação do operador de mutação, primeiramente, gerou-se uma população com trinta “30” indivíduos, a seguir, calculou-se o fitness de cada indivíduo e, de posse desses valores, deu-se início ao processo de busca de cada algoritmo. Em cada teste, a população inicial fornecida a cada algoritmo foi idêntica, assim como a configuração de seus parâmetros. O melhor indivíduo de cada geração foi transferido para a próxima por meio de elitismo (somente um indivíduo é copiado); a taxa de cruzamento foi de 80%; a taxa de mutação foi de 1%; a população possuía tamanho fixo (trinta indivíduos); o número de gerações também era fixo (cinquenta gerações) e o critério de parada utilizado foi o alcance do número máximo de gerações permitido, correspondente a cinquenta “50”.

Tanto o número de gerações, quanto o tamanho da população dos algoritmos foram definidos de forma empírica. Testes com diferentes valores para esses parâmetros foram realizados e definiu-se como padrão os valores que proporcionaram melhores resultados em relação ao problema em questão. Para o tamanho da população foram testados os seguintes valores: 30, 40, 50, 60, 70, 80, 90, 100, 150 e 200. Já para a quantidade de gerações foram testados os seguintes valores: 30, 50, 60, 70, 80, 90, 100 e 200.

Foram realizados cinquenta testes com cada variante aplicada tanto a um AG, quanto a um AGA (AG_V1, AG_V2, AGA_V1 e AGA_V2). A cada geração foram calculadas e armazenadas as seguintes informações: a média aritmética, o melhor e o pior indivíduo, assim como a divergência inferior e superior da média obtida por meio do desvio padrão. Esses resultados são demonstrados na Tabela 5-4 e na Tabela 5-5. O valor do desvio padrão é adicionado ao valor da média, de forma a se obter a divergência superior (limite superior). Além disso, o valor do desvio padrão é subtraído do valor da média, o que permite obter a divergência inferior (limite inferior). Essa divergência informa a faixa de valores na qual se encontra a maior parte dos indivíduos da amostra (da população).

A Figura 5-5 e a Figura 5-8 apresentam a população final obtida na última geração dos AGs do último teste realizado. Já as Figuras 5-3, 5-4, 5-6 e 5-7 apresentam as análises das gerações do último teste realizado.

Como se observa na Figura 5-5, a população da segunda variante possui maior diversidade que a da primeira, o que se comprova na Figura 5-3 e na Figura 5-4. Na Figura 5-3 observa-se que a média da população, a cada geração, se distancia da melhor solução logo

no início do processo de busca. Já na Figura 5-4 observa-se que essa divergência aumenta e diminui a cada geração e está sempre próxima da melhor solução, até a geração na qual o algoritmo converge (alcança um ponto ao qual não consegue ultrapassar).

Na Figura 5-8, observa-se que o mecanismo provocou certa diversificação da população em ambas as versões (AGA_V1 e AGA_V2), mas essa diversificação foi mais perceptível na segunda versão, já que na primeira, à medida que as gerações avançam, a população tendia à homogeneidade, pois o algoritmo convergiu em um determinado ponto e não se gerou nenhuma melhora. Já a segunda versão manteve a heterogeneidade populacional por uma quantidade maior de gerações, mesmo já tendo收敛ido.

Isso pode ser observado tanto na Figura 5-6, quanto na Figura 5-7. Na Figura 5-6 observa-se que, a partir da quinta geração, os demais indivíduos da população começam a se distanciar da melhor solução. Já na Figura 5-7 observa-se o contrário, porém, essa divergência não é tão perceptível quanto na anterior, podendo, assim, existir vários indivíduos com a pontuação próxima à da solução de melhor pontuação.

Como esperado, independentemente da variante utilizada, os AGAs (AGA_V1 e AGA_V2) obtiveram melhores resultados que os AGs (AG_V1 e AG_V2), o que comprovou a necessidade da aplicação do mecanismo de adaptação, pois, por meio dessa versão, pôde-se realizar uma exploração mais abrangente do espaço de busca, alcançando regiões que não foram encontradas pelos AGs.

Na implementação dos algoritmos não foi utilizado nenhum elemento pertencente ao seu toolkit no MATLAB®. A implementação dos algoritmos partindo do zero foi necessária devido à dificuldade em se aplicar os mecanismos de adaptação no toolkit do MATLAB®. Realizaram-se testes a fim de comparar os resultados da implementação realizada com os da implementação obtida por meio do toolkit do MATLAB®, o qual baseia seu desenvolvimento na primeira variante apresentada.

Devido ao fato de o toolkit ser muito genérico, o tempo computacional do algoritmo se tornou muito alto e, por isso, a alteração da implementação do toolkit se torna mais complicada, não sendo viável. No que se refere à resposta obtida, tanto os algoritmos desenvolvidos, quanto o toolkit, obtiveram resultados muito próximos.

Portanto, neste trabalho, não se utiliza o toolkit do MATLAB®. No que se refere à variante para validar a proposta, decidiu-se pela utilização da segunda - AG_V2 e AGA_V2, pois essa proporciona uma melhor heterogeneidade da população, alcançando, assim, mais regiões que a outra. No Apêndice B encontram-se outros testes nos quais se aumentam o número de gerações (200 gerações) e o tamanho da população (100 indivíduos).

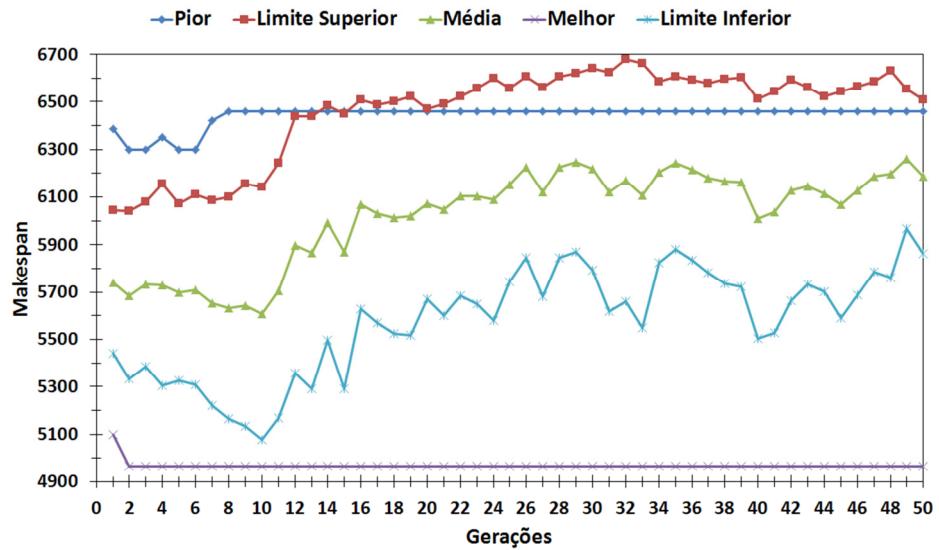


Figura 5-3 - AG_V1

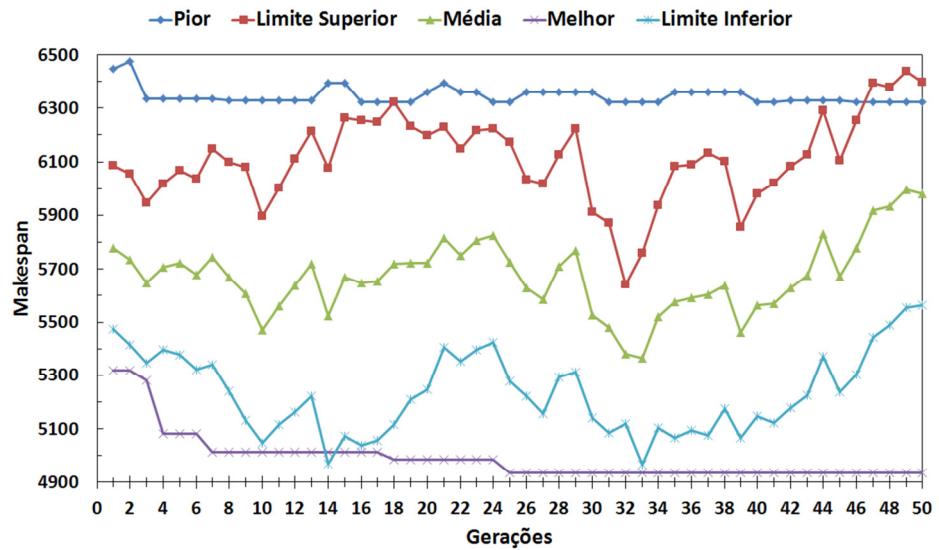


Figura 5-4 - AG_V2

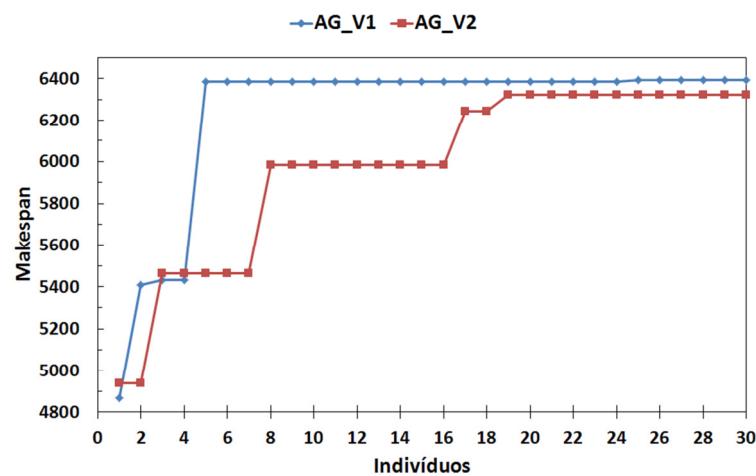


Figura 5-5 - População Final (AG_V1 x AG_V2)

Tabela 5-4 -AG_V1 X AG_V2

-	AG_V1		AG_V2	
	Makespan (u.t.)	Tempo (s)	Makespan (u.t.)	Tempo (s)
1	4965	3,3320	4965	3,4622
2	4990	3,4861	5008	3,4930
3	4937	3,4050	4992	3,3904
4	5045	3,5069	4973	3,4024
5	4975	3,5328	4975	3,4975
6	4964	3,3799	4970	3,5235
7	4949	3,4306	4955	3,4142
8	4983	3,4666	5007	3,6400
9	4977	3,4112	4937	3,4474
10	4965	3,4715	4965	3,4682
11	4936	3,4438	4963	3,4594
12	4962	3,6749	5017	3,4969
13	4950	3,3839	4905	3,3956
14	4954	3,4444	4655	3,4095
15	4963	3,4468	4963	3,4508
16	4959	3,3634	4986	3,4159
17	4927	3,3390	4927	3,4450
18	4952	3,3500	4603	3,4039
19	4937	3,4513	4937	3,5539
20	4963	3,3344	4933	3,5145
21	5045	3,4053	5008	3,3880
22	4975	3,4153	4963	3,4193
23	4937	3,4046	4976	3,3699
24	4985	3,3150	4995	3,3280
25	4914	3,3265	4914	3,3380
26	4566	3,3776	4970	3,3892
27	4990	3,3365	4972	3,3965
28	4952	3,3845	4952	3,3839
29	4937	3,3290	4991	3,4111
30	4936	3,4815	5022	3,3865
31	4955	3,5224	4700	3,4224
32	4614	3,3119	4607	3,7683
33	4957	3,5513	4937	3,5299
34	5017	3,3534	5017	3,3712
35	4968	3,3956	4968	3,3248
36	4975	3,3583	4975	3,4096
37	4952	3,4735	4906	3,6079
38	4970	3,3619	4914	3,3306
39	5031	3,3818	4970	3,3592
40	4963	3,3605	4966	3,3119
41	4917	3,3673	4941	3,3887
42	4610	3,3203	4937	3,3575
43	4890	3,3198	4890	3,3454
44	4993	3,3646	4594	3,4730
45	4635	3,3840	4549	3,3599
46	4618	3,4744	4618	3,4569
47	4900	3,4345	4645	3,3921
48	4944	3,5173	4931	3,3598
49	4593	3,3213	4973	3,4850
50	4866	3,4040	4937	3,3930
Média	4917,16	3,4082	4907,48	3,4308

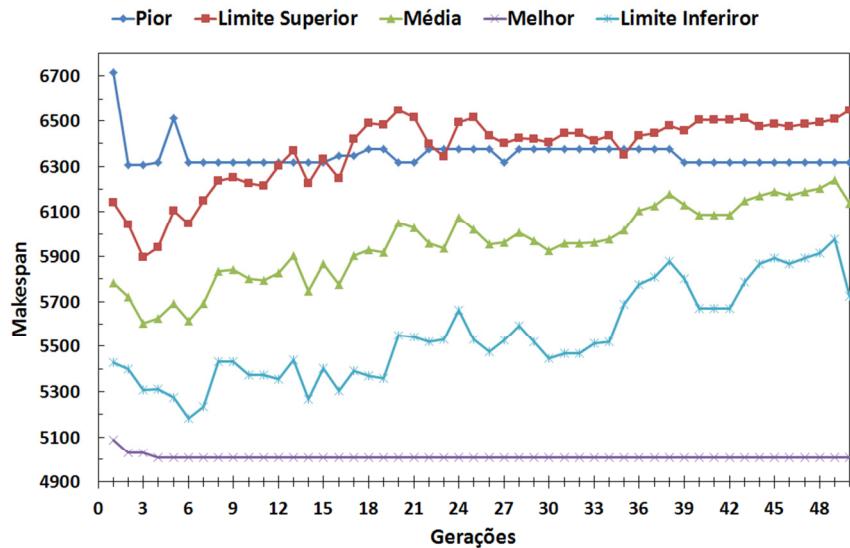


Figura 5-6 - AGA_V1

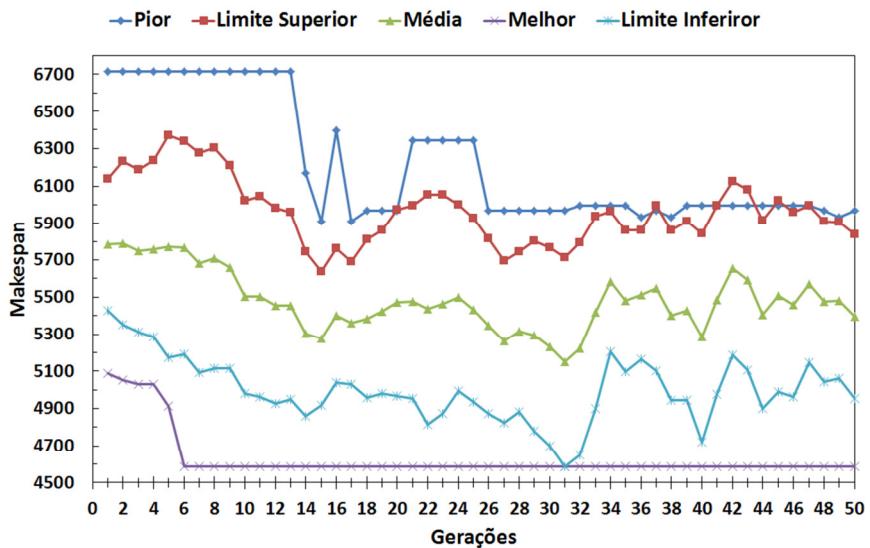


Figura 5-7 - AGA_V2

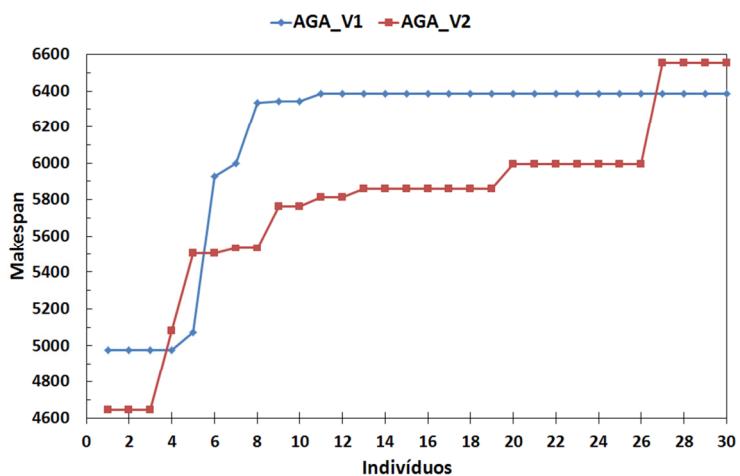


Figura 5-8 - População Final (AGA_V1 X AGA_V2)

Tabela 5-5 - AGA_V1 X AGA_V2

-	AG_V1		AG_V2	
	Makespan (u.t.)	Tempo (s)	Makespan (u.t.)	Tempo (s)
1	5071	3,4315	5039	4,1887
2	5008	3,8874	4586	4,0896
3	4989	4,2168	5067	3,7179
4	4645	3,8651	4937	3,7507
5	4948	3,9416	4948	4,3635
6	4949	3,7406	4862	4,1030
7	4990	4,2635	5017	3,6794
8	4945	3,8508	5003	3,7913
9	5007	3,4172	4579	3,7138
10	4993	3,5238	4978	2,9859
11	4899	4,2727	4899	3,3596
12	5010	3,4594	4584	3,5430
13	4921	4,0514	4608	3,8702
14	4905	4,1746	4976	3,5358
15	4999	3,5343	5010	3,1314
16	4588	4,0483	4573	3,5948
17	5044	3,7588	5063	3,2611
18	4931	3,7398	4931	3,0682
19	4986	4,0405	4986	3,9288
20	4977	3,5713	4591	3,8221
21	5015	3,1863	4618	3,6633
22	4944	3,5990	4830	4,1751
23	4937	4,1157	4646	4,1104
24	4958	3,6962	4949	3,2519
25	5024	3,7034	4963	3,5217
26	4973	3,2384	4927	3,4024
27	4977	3,1816	4900	3,7357
28	4931	3,5549	4931	3,6598
29	5022	3,4873	4583	3,8640
30	5001	3,5360	4994	3,6807
31	4937	3,3005	4970	3,8021
32	4976	3,4039	4976	3,6373
33	4928	3,5328	4928	3,4739
34	4862	4,1372	4996	3,9510
35	4976	3,0594	4975	3,2539
36	4973	3,7185	4622	4,1735
37	5049	3,2041	4579	4,0159
38	5035	3,0931	4937	3,5178
39	4618	3,3488	4618	3,2512
40	4684	3,5715	4914	3,3448
41	4990	3,7986	4965	3,5977
42	4926	3,7998	4957	3,3366
43	4962	3,7481	5005	3,6920
44	4972	3,4691	4978	4,2968
45	5020	3,6498	4991	3,3595
46	4606	3,5269	4606	4,0367
47	4988	3,7663	4964	4,2510
48	4945	3,9413	4594	3,6397
49	4937	3,6908	4937	3,3608
50	4975	4,1111	4643	4,3308
Média	4938,92	3,6792	4854,66	3,6977

5.4 AGs X AGAs

A seguir, realiza-se a comparação dos resultados obtidos por meio das implementações realizadas no MATLAB® (AGs, AGAs e as três variações do cálculo do *makespan*) com os resultados obtidos por Deriz (2007) e por Sanches (2008). Nem Deriz (2007), nem Sanches (2008) utilizaram o MATLAB® no desenvolvimento dos seus algoritmos; utilizaram a IDE Eclipse da linguagem Java. Portanto, para realizar a comparação de seus resultados com os obtidos pelas implementações realizadas neste trabalho, desconsiderou-se o tempo computacional obtido por seus algoritmos e levou-se em consideração somente os valores de *makespan* alcançados por suas aplicações. Já na análise do desempenho dos algoritmos desenvolvidos no MATLAB® levou-se em consideração tanto o tempo computacional dos algoritmos quanto os valores de *makepan* fornecido por eles. Tanto Deriz (2007) quanto Sanches (2008) baseiam seus experimentos na primeira variante de implementação dos algoritmos, apresentada anteriormente.

Nesta análise são realizados cinquenta “50” testes com cada algoritmo, cujos pontos de partidas são a mesma população, gerada de forma aleatória, e a população inicial fornecida a cada algoritmo é alterada a cada teste. A população inicial utilizada como ponto de partida nos trabalhos de Deriz (2007) e Sanches (2008) são distintas em cada teste, não havendo equivalência entre os testes realizados no MATLAB® e no Eclipse. O critério de parada, tanto dos algoritmos desenvolvidos no MATLAB®, quanto dos trabalhos de Deriz (2007) e Sanches (2008), é o alcance de cinquenta “50” gerações.

Deriz (2007) utilizou os parâmetros genéticos a seguir para desenvolver seus testes: taxa de cruzamento de 80%; taxa de mutação de 20% e o tamanho da população igual a trinta “30”. Já Sanches (2008) utilizou os seguintes parâmetros genéticos: taxa de cruzamento iniciando em 60% e variando entre 50% e 100%; taxa de mutação iniciando em 0.5% e variando entre 0.5% e 1%, e o tamanho da população igual a trinta “30”.

Neste trabalho foram utilizados os seguintes parâmetros genéticos: taxa de cruzamento iniciando em 80% e taxa de mutação iniciando em 1% para o AG. Já para o AGA mantiveram-se os mesmos parâmetros iniciais, mas o valor da taxa de cruzamento variando entre 0.5% e 1% e o da taxa de mutação variando entre 0.5% e 2%. O tamanho da população foi igual a trinta “30”. Além disso, neste trabalho foi desenvolvido um algoritmo para cada forma de cálculo do *makespan*, sendo que cada algoritmo possui os mesmos parâmetros apresentados.

Na Tabela 5-6 e na Tabela 5-7 encontram-se os resultados obtidos por meio das implementações realizadas neste trabalho, assim como os resultados obtidos pelas implementações de Deriz (2007) e de Sanches (2008). Vale lembrar que os trabalhos de Deriz (2007) e de Sanches (2008) foram remodelados e novamente desenvolvidos por meio do MATLAB®.

São realizados dois tipos de testes, os quais se diferenciam pelo tamanho da população. Os resultados do primeiro teste, para uma população de tamanho trinta “30”, encontram-se na Tabela 5-6. Já os resultados do segundo teste, para uma população de tamanho cem “100”, encontram-se na Tabela 5-7. Como se pode observar, a variação do tamanho da população não implicou no alcance de melhores resultados, mas aumentou, de forma considerável, o tempo computacional dos algoritmos. Portanto, neste trabalho a população tem seu tamanho fixado em trinta “30” indivíduos.

Na análise da média dos testes observou-se que os algoritmos desenvolvidos no MATLAB® obtiveram um melhor resultado que os algoritmos de Deriz (2007) e de Sanches (2008). Os métodos desenvolvidos divergem cerca de 16% da média dos resultados de Sanches (2008) e cerca de 3% da média dos resultados de Deriz (2007). Vale ressaltar que os resultados de Deriz (2007) podem ser comparados somente com os resultados obtidos pelo método proposto por ela para o cálculo do *makespan* e não com o cálculo da proposta e nem com o cálculo proposto por Sanches (2008), pois, em seu trabalho, a autora não leva em consideração o sistema de transporte, o que não ocorre nos demais cálculos.

Além disso, o método de cálculo do *makespan* proposto, que realiza o adiantamento do sistema de transporte, também obteve melhor resultado que ambos e, neste trabalho, é indicado para a realização da programação da produção.

O tamanho da população que obteve melhor desempenho para o problema em questão é a população com trinta “30” indivíduos. Já a quantidade de gerações necessárias para a obtenção de uma boa resposta sem influência significativa no tempo computacional é de cinquenta “50” gerações.

A redução do tamanho da população do algoritmos, e a redução do número de gerações a serem alcançadas, influenciam diretamente no tempo computacional da obtenção da resposta, pois isto implica na realização de uma menor quantidade de cálculos e inferências.

Assim conclui-se que a proposta, de adiantamento do sistema de transporte, é de boa valia e bem aplicada ao problema, e que a sua inserção não altera o comportamento do sistema, ou seja, não ocorre mudança significativa no tempo computacional.

Tabela 5-6 – AGs/AGAs - Resultados Obtidos (30 indivíduos)

-	Desenvolvidos no MATLAB®												AGA_V1		AG_V1						
	AG_V2				Deriz (2007)				Proposta				Sanches (2008)				Deriz (2007)				
	Proposta		Sanches (2008)		Deriz (2007)		Proposta		Sanches (2008)		Deriz (2007)		Proposta		Sanches (2008)		Deriz (2007)		Makespan(u.t.)	Makespan(u.t.)	
	Makespan (u.t.)	Tempo (s)	Makespan (u.t.)	Tempo (s)	Makespan (u.t.)	Tempo (s)	Makespan (u.t.)	Tempo (s)	Makespan (u.t.)	Tempo (s)	Makespan (u.t.)	Tempo (s)	Makespan (u.t.)	Tempo (s)	Makespan (u.t.)	Tempo (s)	Makespan (u.t.)	Tempo (s)			
1	4975	7,490	5170	6,704	4708	3,559	4946	9,085	5221	7,051	4937	3,415	5147	5031	5147	5031	5147	5031			
2	5016	8,301	5163	7,303	4952	5,050	4966	7,104	4786	8,247	4931	3,659	5156	4937	5156	4937	5156	4937			
3	5032	8,417	5154	7,382	4708	3,583	5032	9,107	5161	6,473	4590	3,897	5164	5031	5164	5031	5164	5031			
4	4858	8,727	5200	8,879	4937	3,985	5041	8,373	5190	7,661	5011	3,635	5232	4571	5232	4571	5232	4571			
5	4966	8,397	5214	7,204	4993	3,649	4935	8,273	5187	6,499	4609	3,687	5232	4590	5232	4590	5232	4590			
6	5047	8,174	5264	7,318	5064	3,723	4949	8,726	5192	5,855	5005	3,091	5144	5000	5144	5000	5144	5000			
7	4637	7,702	4786	7,265	4997	3,616	4931	7,669	4832	6,450	4695	3,567	5116	4945	5116	4945	5116	4945			
8	5016	7,709	5192	6,560	4998	3,341	5026	8,045	5160	7,647	4836	4,896	5141	5031	5141	5031	5141	5031			
9	4949	8,459	4820	7,651	4608	3,402	4642	10,693	5198	7,867	4988	3,755	5141	5016	5141	5016	5141	5016			
10	5038	8,774	5169	7,664	4708	3,869	5034	8,986	5186	6,647	5009	3,080	5116	4971	5116	4971	5116	4971			
11	5027	8,703	4763	7,204	5007	3,698	5038	9,553	5176	8,342	4583	3,967	5063	4964	5063	4964	5063	4964			
12	4685	9,528	5192	9,068	4949	5,009	4988	10,304	5153	8,273	4937	3,777	5105	4984	5105	4984	5105	4984			
13	4949	8,940	5048	8,447	4937	4,172	4712	8,917	4770	9,161	4980	4,172	5164	4980	5164	4980	5164	4980			
14	5073	8,787	5204	7,645	5015	3,829	4993	8,524	4806	7,981	4975	4,585	5164	5014	5164	5014	5164	5014			
15	4966	9,217	5107	7,558	4974	3,889	4979	11,656	5155	9,064	4586	5,249	5106	4965	5106	4965	5106	4965			
16	4979	8,842	5164	8,167	4957	4,724	5032	10,154	5009	7,120	4618	4,379	5125	4945	5125	4945	5125	4945			
17	4642	9,344	4785	8,723	4862	3,859	4926	9,366	5190	9,336	4590	4,431	5156	5050	5156	5050	5156	5050			
18	4966	10,259	5125	7,612	4595	4,157	5014	9,202	5125	8,002	4913	5,932	5048	5031	5048	5031	5048	5031			
19	4949	8,940	4790	8,257	4586	4,328	4642	10,807	4742	7,074	4610	4,937	5232	4937	5232	4937	5232	4937			
20	5020	8,850	5185	7,687	4990	3,971	4547	10,709	4742	8,709	4538	6,126	5182	5050	5182	5050	5182	5050			
21	4955	13,861	5149	10,780	4945	5,901	4970	8,769	4928	9,556	4945	6,676	5121	4708	5121	4708	5121	4708			
22	4955	9,410	5173	7,661	4937	3,822	4955	7,988	5173	6,804	4937	3,873	5156	4937	5156	4937	5156	4937			
23	4982	7,725	5183	6,836	5010	3,395	5027	7,567	5179	6,805	4937	3,712	5187	4959	5187	4959	5187	4959			
24	4590	9,306	4812	7,708	4538	4,687	4621	8,983	4802	6,271	4976	4,202	5164	5023	5164	5023	5164	5023			
25	4901	10,563	5187	8,859	4978	4,232	5012	10,027	5168	9,354	5007	3,847	5156	4914	5156	4914	5156	4914			
26	5015	9,349	5169	8,441	4617	4,277	4622	8,325	4809	10,132	4617	4,786	5164	4937	5164	4937	5164	4937			
27	4987	9,274	5147	7,813	4975	4,553	4987	9,979	4750	8,988	4975	4,731	5232	5050	5232	5050	5232	5050			
28	5003	9,004	4799	8,559	4640	4,975	4608	10,396	4796	9,032	4603	3,700	5232	5031	5232	5031	5232	5031			
29	5016	9,659	5209	8,249	4937	4,873	5042	10,983	5209	8,172	4990	4,014	5147	4950	5147	4950	5147	4950			
30	4955	9,244	5188	7,728	4952	3,974	4939	9,827	5170	9,758	4920	4,316	5180	4976	5180	4976	5180	4976			
31	4638	9,373	5173	8,738	4608	4,304	4645	9,305	5173	8,092	4952	4,661	5040	4966	5040	4966	5040	4966			
32	4602	8,569	5168	7,578	4938	4,057	4907	7,988	4787	8,196	4952	3,968	5120	4931	5120	4931	5120	4931			
33	5007	9,010	4750	7,693	4999	4,474	5055	8,363	5119	10,906	4999	3,534	5109	4931	5109	4931	5109	4931			
34	4943	9,031	5145	7,729	4912	3,880	4925	8,464	5175	8,729	4879	4,884	5113	4684	5113	4684	5113	4684			
35	4988	9,791	5150	8,437	4927	4,003	4961	9,072	5219	9,337	4900	5,128	5232	4600	5232	4600	5232	4600			
36	4642	9,031	5173	7,809	4900	3,997	4986	11,389	4862	7,775	4922	3,463	5232	5022	5232	5022	5232	5022			
37	5034	9,186	4901	7,961	4995	4,100	5033	9,011	5227	8,608	5021	4,607	5105	5050	5105	5050	5105	5050			
38	5038	9,545	5202	8,053	4894	4,963	4996	9,967	5185	10,061	5015	4,029	5107	4900	5107	4900	5107	4900			
39	4988	9,293	5207	8,108	4850	3,991	4992	9,774	5176	8,694	4955	4,347	5090	5050	5090	5050	5090	5050			
40	4712	9,200	4867	7,880	4708	3,832	4602	9,442	4867	7,225	4708	3,673	5129	5000	5129	5000	5129	5000			
41	4993	9,037	4787	7,944	4980	4,097	4975	10,043	5150	9,627	4590	4,680	5120	5020	5120	5020	5120	5020			
42	4979	9,342	4934	7,924	4937	4,023	4978	9,626	4745	9,101	4985	4,622	5170	4972	5170	4972	5170	4972			
43	4549	8,175	5235	8,079	5034	4,262	4998	9,912	5150	7,762	4996	3,626	5141	4978	5141	4978	5141	4978			
44	5016	10,288	5033	11,215	4862	4,723	5019	13,690	4766	8,414	4618	4,640	5160	5050	5160	5050	5160	5050			
45	4949	7,643	5158	8,083	4937	4,834	5013	8,115	5160	7,110	4917	3,583	5117	4896	5117	4896	5117	4896			
46	4940	7,498	5197	6,523	4965	3,276	5004	7,600	5206	7,990	4965	3,276	5130	5050	5130	5050	5130	5050			
47	4969	7,585	5190	6,634	4994	3,302	4917	7,619	5154	7,422	4937	3,352	5232	4972	5232	4972	5232	4972			
48	5059	7,583	5137	6,647	4862	3,269	5060	7,950	5257	6,687	4579	3,347	5232	4975	5232	4975	5232	4975			
49	4904	7,569	5184	6,637	4949	3,279	5014	6,578	5335	6,665	4590	3,357	5164	4938	5164	4938	5164	4938			
50	4902	7,513	4901	6,686	4913	3,272	4902	7,701	4901	7,637	4913	3,052	5105	4977	5105	4977	5105	4977			
	Média	4920,02	8,904	5080,06	7,866	4884,76	4,081	4922,76	9,194	5051,58	8,087	4844,82	4,158	5148,42	4949,80						

Tabela 5-7 - AGs/AGAs - Resultados Obtidos (100 indivíduos)

-	Desenvolvidos no MATLAB®												AGA_V1 Sanches (2008) Makespan(u.t.)	AG_V1 Deriz (2007) Makespan(u.t.)
	Proposta Makespan (u.t.)		AG_V2 Sanches (2008) Makespan (u.t.)		Deriz (2007) Makespan (u.t.)		Proposta Makespan (u.t.)		AGA_V2 Sanches (2008) Makespan (u.t.)		Deriz (2007) Makespan (u.t.)			
	Tempo (s)		Tempo (s)		Tempo (s)		Tempo (s)		Tempo (s)		Tempo (s)		Tempo (s)	
1	4554	118,06	5131	104,45	4538	41,69	4883	126,22	4732	103,89	4538	56,39	5232	4962
2	4592	147,31	4771	99,36	4573	45,56	4592	132,56	4762	146,12	4712	73,34	4756	4945
3	5068	119,17	4832	102,58	4934	46,10	4748	111,69	4818	119,40	4571	54,69	5164	4952
4	5146	113,85	4769	97,887	4593	45,06	5110	136,76	4752	118,31	4617	55,25	5232	4962
5	4785	115,88	4742	98,113	4608	45,71	5150	113,69	5138	100,98	4579	45,71	5035	4914
6	4750	116,15	4784	99,143	4890	44,62	4764	124,42	4786	115,48	4932	43,56	5232	4962
7	4608	116,66	5046	97,792	4618	45,46	5147	117,98	5113	117,45	4590	49,65	4753	4952
8	4917	114,66	5099	100,25	4614	45,67	5147	136,06	5099	100,29	4900	48,88	4806	4962
9	4628	114,72	4737	100,28	4586	45,50	5038	112,26	5036	115,95	4862	52,26	4683	4899
10	4962	115,33	4755	98,516	4973	44,91	4962	115,91	4806	85,23	4587	50,63	4722	4995
11	4650	114,17	5132	98,935	4608	45,96	4659	131,33	4790	104,32	4815	49,50	4743	4962
12	5046	115,76	4818	99,403	4617	47,31	4969	139,93	5149	114,50	4890	46,67	5101	4932
13	5099	115,42	4787	98,749	4607	45,43	4638	125,73	4744	100,48	4549	46,71	5164	4995
14	4737	117,96	5143	100,38	4937	45,70	4637	130,65	5114	108,05	4618	48,67	5091	4915
15	4988	115,58	4802	100,19	4934	45,75	4946	118,60	4847	111,62	4915	54,81	4774	4914
16	4642	115,55	5135	99,189	4579	45,72	4608	116,55	4769	116,27	4938	48,84	5005	4949
17	4652	115,25	4781	98,818	4948	45,60	4652	131,47	4806	111,43	4598	52,30	5232	4962
18	4858	114,6	5125	100,78	4937	46,36	4582	129,21	5110	102,03	4938	55,18	5232	4995
19	4939	116,94	5150	99,381	4889	45,32	4629	138,45	5093	109,31	4618	50,88	5164	4952
20	4629	114,81	4759	100,43	4608	45,54	4629	116,44	4759	99,37	4946	48,69	5071	4952
21	4592	116,77	5036	98,916	4583	45,63	4549	117,73	4748	111,65	4708	46,48	4759	4932
22	4590	116,12	5068	100,47	4590	45,88	4910	106,07	5110	105,10	4892	51,55	4771	4995
23	4949	114,93	5146	99,14	4937	45,35	4949	123,51	5150	96,06	4937	49,03	5129	4952
24	4592	116,67	4785	99,542	4937	46,66	4622	138,90	4764	115,93	4899	46,98	5164	4900
25	4606	117,27	4750	100,41	4910	46,49	4642	129,72	5147	101,25	4538	48,51	5082	4945
26	4949	114,11	5142	100,19	4644	44,93	4549	117,30	5147	117,92	4937	54,26	5089	4871
27	4636	114,87	5038	100,21	4945	46,25	4940	116,90	5038	101,73	4684	49,70	4729	4914
28	4602	116,26	4784	99,499	4594	46,42	4866	128,89	5147	108,24	4573	51,58	5232	4962
29	4949	115,33	4758	100,84	4937	46,11	4651	138,56	4769	101,06	4960	46,76	5030	4598
30	4606	116,07	5160	99,175	4596	45,64	4608	128,96	5024	101,44	4614	46,89	5076	4645
31	4943	116,1	5143	99,87	4590	45,81	4924	120,09	4826	101,19	4590	49,82	5039	4922
32	5046	116,2	5148	101,06	4603	46,60	4608	138,38	5143	106,73	4906	53,15	4806	4995
33	5099	118,37	4867	99,114	4900	45,88	4670	129,98	4813	96,86	4566	50,90	4777	4645
34	4737	116,05	4794	99,117	4594	45,68	4622	116,99	4794	115,74	4594	51,78	4746	4995
35	4637	115,86	5126	99,2	4937	45,73	4620	132,34	4729	118,20	4538	55,51	4768	4871
36	4949	116,68	5144	100,97	4684	45,36	4949	136,89	5150	100,47	4970	49,88	5232	4637
37	4925	116,71	5113	99,26	4928	45,41	4940	125,09	5157	112,14	4928	54,25	5071	4959
38	4636	114,86	4775	100,78	4958	44,77	4955	129,71	5147	110,75	4900	49,01	5164	4545
39	4946	116,17	5161	100,69	4933	46,07	4588	133,14	4837	102,55	4934	49,71	5071	4917
40	4606	117,59	5079	99,09	4856	45,15	4748	127,68	4825	118,59	4608	46,42	4740	4965
41	4943	115,22	4790	101,24	4890	45,54	5110	135,94	5148	112,10	4618	48,84	5232	4645
42	4911	115,03	5154	98,987	4571	45,80	4764	134,19	4832	100,93	4590	52,31	4728	4871
43	4672	116,05	4743	99,222	4598	45,61	4764	116,60	5040	114,13	4602	49,29	5010	4915
44	4886	116,44	5130	98,826	4633	45,17	5147	117,62	4733	116,32	4937	52,84	5164	4995
45	4682	116,94	5142	99,95	4607	45,69	5147	123,90	5142	102,96	4937	46,35	5085	4945
46	4592	116,06	4747	99,797	4975	46,80	5038	138,70	4747	105,40	4862	50,69	4751	4937
47	4634	114,77	5123	99,997	4932	46,80	4623	137,10	5180	109,55	4989	54,13	5016	4850
48	5046	115,11	5093	99,148	4892	45,94	4904	130,71	5093	109,40	4892	49,62	5164	4995
49	5099	116,21	5137	101,16	4570	43,36	4937	125,05	4760	107,35	4906	51,69	5232	4945
50	4737	101,63	4794	87,538	4594	40,57	4622	103,33	4794	87,08	4594	41,83	5097	4645
Média	4800,94	116,29	4963,36	99,56	4750,18	45,56	4809,12	126,00	4943,14	108,19	4758,32	50,65	5002,92	4898,82

5.5 Tipos de Vizinhança

Segundo Glover (1997), a vizinhança na BT é formada por meio dos métodos construtivos e destrutivos. Neste trabalho, foram utilizadas duas vizinhanças específicas: na primeira, troca-se o roteiro de produção e, na segunda, altera-se a ordem do produto na solução, o que, consequentemente, altera sua ordem de produção. A Figura 5-9 ilustra essas vizinhanças e a Figura 5-10 ilustra a aplicação das mesmas ao problema.

Soluções Inicial												Makespan
4 2 8 1 6 2 3 2 2 1 5 2 9 2 7 1 1 2												5027
1ª Vizinhança												
4 2 8 2 6 2 3 2 2 1 5 2 9 2 7 1 2												5013
2ª Vizinhança												
4 2 7 1 6 2 9 2 2 1 5 2 3 2 8 1 1 2												4915

Figura 5-9 - Tipos de vizinhança

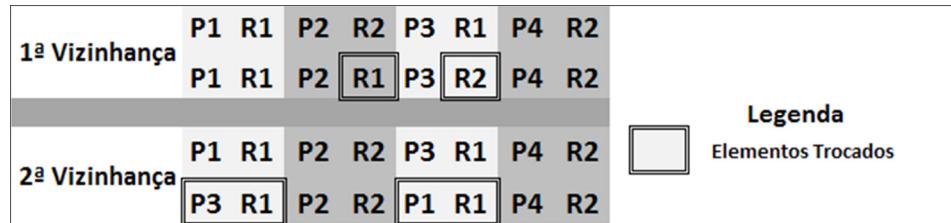


Figura 5-10 - Exemplo da aplicação da vizinhança

Foram realizados testes para verificar a melhor forma de se aplicar essas vizinhanças, como se observa na Tabela 5-8. Os testes realizados estão listados a seguir:

- **V1:** aplicação da primeira vizinhança;
- **V2:** aplicação da segunda vizinhança;
- **V1→V2:** aplicação das vizinhanças em sequência;
- **V2→V1:** aplicação das vizinhanças em ordem inversa;
- **V1→V2→V1:** aplicação da primeira vizinhança ao resultado da aplicação das vizinhanças em sequência;
- **V2→V1→V2:** aplicação da segunda vizinhança ao resultado da aplicação das vizinhanças em ordem inversa.

A aplicação das vizinhanças em sequência foi o teste que apresentou os melhores resultados. Isso pode ser confirmado pela média dos testes que, no melhor caso, diverge cerca de 14% da média do pior caso.

Tabela 5-8 - Exemplo de aplicação das vizinhanças

Iteração	-	V1	V2	V1_V2	V2_V1	V1_V2_V1	V2_V1_V2
1	5972	4965	5426	4913	4995	4971	4987
2	5611	5362	5375	4598	5295	4865	4865
3	5449	4937	5431	4937	4976	5245	4945
4	5860	5060	5013	5016	5001	4965	4905
5	5850	5101	5083	4993	5046	4964	4932
6	5010	4931	4931	4579	4931	4906	4595
7	5430	4981	5403	4949	4990	4937	4949
8	5819	4969	5362	4952	4987	5404	4967
9	5499	4937	4994	4937	4937	5000	4937
10	5463	4965	5002	4965	4913	4927	4579
11	5502	5100	5458	5016	5451	4965	5438
12	5865	4614	5865	4545	4961	4978	4927
13	5441	4937	4594	4937	4693	5050	4626
14	5874	5372	5014	5013	5018	5001	4955
15	5503	4944	5450	4579	4994	4912	4624
16	5467	4937	5391	4937	4937	4937	4937
17	5913	5763	5736	5415	5400	5408	4604
18	5481	5050	5423	4995	5415	4937	4937
19	6652	4990	5070	4603	4964	4640	4952
20	5977	5026	5484	4670	4976	4608	4945
21	5482	5273	5482	5021	5273	5358	5021
22	4880	4890	4618	4890	4590	4896	4564
23	5400	4981	5322	4618	5052	4836	4967
24	5976	5459	5482	5391	5050	5324	4988
25	5355	5059	4615	4579	4927	4587	4594
26	5564	4828	5461	4828	4834	4645	4828
27	5944	5091	5373	4594	4954	4660	4945
28	5897	5058	5456	4984	5035	4937	5008
29	5816	4975	5391	4937	5027	4610	4976
30	5902	5001	5479	4937	5029	4971	4971
31	5503	4944	5450	4579	4994	4912	4624
32	5467	4957	5391	4937	4957	4957	4957
33	5913	5763	5736	5415	5400	5408	4604
34	5481	5050	5423	4995	5415	4937	4945
35	6652	4990	5070	4603	4964	4640	4952
36	5977	5026	5484	4670	4976	4608	4945
37	5482	5273	5482	5021	5273	5358	5391
38	4880	4890	4618	4890	4590	4896	4564
39	5400	4981	5322	4618	5052	4836	4967
40	5976	5459	5482	5391	5050	5324	4988
41	5355	5059	4615	4579	4927	4587	4594
42	5503	4944	5450	4579	4994	4912	4624
43	5467	4957	5391	4937	4957	4957	4957
44	5913	5763	5736	5415	5400	5408	4604
45	5481	5050	5423	4995	5415	4937	5391
46	6652	4990	5070	4603	4964	4640	4952
47	5977	5026	5484	4670	4976	4608	4945
48	5400	4981	5322	4618	5052	4836	4967
49	5976	5459	5482	5391	5050	5324	4988
50	5400	4981	5322	4618	5052	4836	4967
Média	5674,18	5081,98	5298,74	4877,04	5042,18	4947,30	4887,88

5.6 Buscas Tabu (BTs)

A seguir, realiza-se a comparação dos resultados obtidos por meio das implementações realizadas no MATLAB® (BTs e as três variações do cálculo do *makespan*). Nesta análise, são realizados cinquenta “50” testes com cada algoritmo, cujos pontos de partida são uma mesma solução, gerada de forma aleatória, e a solução inicial fornecida a cada algoritmo é alterada a cada teste. O critério de parada utilizado é o alcance de 50 iterações sem a ocorrência de melhora na solução; caso não ocorra melhora em quinze “15” iterações, reinicia-se a BT com uma nova solução gerada de forma aleatória. A partir da 15^a iteração sem melhora, aplica-se o reinício total da busca a cada cinco iterações sem melhora em todas as versões desenvolvidas.

Ao se comparar as médias obtidas pelas BT com a média obtida pelos AGs e pelos AGAs, nota-se que a BT apresenta melhor resultado que os AGs e os AGAs. A inserção das estratégias de intensificação e diversificação não implicou na melhora da solução encontrada e nem na redução do número de iterações de forma significativa, mas apenas aumentou o tempo de execução dos algoritmos. Isso pode ser observado pela análise das médias apresentadas na Tabela 5-9 e na Tabela 5-10.

Para o problema em questão, devido ao seu tamanho e complexidade, a BT clássica, sem nenhuma estratégia agregada (diversificação e intensificação), obteve um melhor desempenho que a BT com estratégia de intensificação e diversificação. Comparando-se o desempenho das BTs com o desempenho dos AGs e dos AGAs, as BTs saíram melhor que ambos, sendo que elas obtiveram um melhor *makespan* em todos os testes. Quanto ao tempo computacional, o da BT foi superior ao do AG e do AGA, porém ele não é tão expressivo a ponto de prejudicar o bom funcionamento do sistema produtivo.

Portanto, para o problema em questão, é indicada a aplicação da BT clássica com os parâmetros aqui apresentados.

Tabela 5-9 - BTs: Resultados Obtidos

	AG_V1		Busca Tabu Sanches (2008)		Deriz (2007)	
	Makespan (u.t.)	Tempo (s)	Makespan (u.t.)	Tempo (s)	Makespan (u.t.)	Tempo (s)
1	4577	15,732	4724	15,492	4549	6,618
2	4561	16,813	4732	26,235	4538	7,566
3	4547	24,560	4729	20,199	4538	12,515
4	4591	24,522	4724	18,868	4538	6,778
5	4547	17,530	4724	28,701	4927	5,749
6	4622	15,342	4724	42,733	4565	7,940
7	4547	23,948	4724	38,336	4538	13,433
8	4614	15,723	4759	13,745	4538	13,452
9	4592	16,314	4724	18,879	4915	7,161
10	4547	19,046	4724	38,725	4565	12,117
11	4547	20,487	4723	39,285	4571	6,410
12	4547	18,879	4724	17,396	4579	5,503
13	4547	18,277	4742	15,358	4579	9,647
14	4941	18,338	4742	19,420	4538	10,383
15	4543	16,938	4724	32,371	4538	5,730
16	4547	23,455	4723	17,078	4545	6,397
17	4547	24,646	4724	25,047	4538	10,022
18	4561	26,427	4724	24,796	4538	7,515
19	4543	29,398	4724	15,180	4565	7,297
20	4547	32,623	4724	33,183	4538	14,141
21	4554	27,201	4724	15,324	4538	10,964
22	4561	46,075	4812	13,387	4538	19,074
23	4554	26,681	4727	19,700	4538	8,484
24	4547	22,849	4724	28,419	4538	12,531
25	4547	35,826	4724	15,035	4579	7,732
26	4543	19,089	4734	40,580	4570	14,320
27	4618	29,875	4724	20,381	4538	6,177
28	4543	30,748	4724	18,244	4565	6,738
29	4543	26,651	4742	14,413	4538	11,023
30	4547	28,777	4724	19,789	4538	7,157
31	4547	25,414	4724	15,558	4538	9,822
32	4547	42,710	4741	35,249	4545	6,804
33	4554	25,793	4733	18,395	4538	7,032
34	4547	38,344	4742	23,977	4538	10,152
35	4547	26,014	4723	19,310	4565	14,151
36	4547	42,592	4727	17,436	4538	8,724
37	4572	35,248	4724	14,494	4594	9,634
38	4572	26,456	4723	37,453	4538	6,806
39	4572	25,340	4727	20,800	4538	7,726
40	4547	43,704	4724	20,194	4538	10,737
41	4547	51,475	4724	25,627	4538	13,098
42	4572	32,444	4731	29,582	4565	7,184
43	4547	36,685	4724	19,746	4538	6,992
44	4547	44,476	4724	36,143	4571	7,319
45	4547	31,427	4738	15,216	4538	8,610
46	4543	25,065	4724	31,520	4538	7,514
47	4543	38,914	4724	17,247	4538	6,031
48	4585	27,500	4738	29,477	4538	10,351
49	4591	19,606	4724	33,858	4545	10,282
50	4572	35,849	4763	23,258	4565	7,914
Média	4566,36	27,757	4730,46	23,817	4563,28	9,149

Tabela 5-10 -BTs: Resultados Obtidos

	Busca Tabu - Diversificação e Intensificação					
	Proposta	Sanches (2008)	Deriz (2007)	Makespan	Tempo	Makespan
	(u.t.)	(s)	(u.t.)	(s)	(u.t.)	(s)
1	4547	52,842	4727	27,686	4579	18,974
2	4591	46,248	4724	25,494	4594	20,634
3	4572	29,289	4731	40,286	4579	9,639
4	4591	36,507	4727	31,967	4538	18,892
5	4547	43,787	4724	20,277	4538	10,820
6	4547	33,655	4724	31,710	4538	29,881
7	4572	35,277	4731	32,415	4565	10,018
8	4547	37,768	4724	20,829	4538	8,075
9	4547	46,476	4724	38,143	4571	9,319
10	4547	42,594	4738	26,383	4579	19,777
11	4543	26,148	4724	32,603	4579	8,597
12	4543	42,331	4724	20,664	4538	9,448
13	4585	54,500	4738	56,477	4538	37,351
14	4591	30,773	4724	45,025	4545	21,449
15	4572	56,932	4763	24,341	4565	18,998
16	4577	50,149	4724	49,909	4549	41,034
17	4561	17,146	4732	26,568	4538	17,899
18	4547	25,643	4729	21,282	4538	13,598
19	4591	27,939	4724	22,285	4538	10,195
20	4554	54,201	4724	42,324	4538	37,964
21	4561	57,242	4812	24,554	4538	30,241
22	4554	27,764	4727	20,783	4538	19,568
23	4547	23,182	4724	28,752	4538	12,864
24	4547	27,140	4724	29,452	4579	26,861
25	4543	46,089	4734	67,580	4570	41,320
26	4618	40,875	4724	31,381	4538	17,177
27	4543	34,165	4724	21,661	4565	10,154
28	4543	53,651	4742	41,413	4579	38,023
29	4547	51,610	4724	22,622	4579	10,999
30	4547	56,497	4724	16,641	4538	10,906
31	4547	44,710	4741	37,249	4545	18,804
32	4554	56,876	4733	19,478	4538	18,116
33	4547	41,761	4742	27,394	4538	13,569
34	4547	53,014	4723	46,310	4565	41,151
35	4547	58,613	4724	29,784	4927	16,833
36	4622	18,759	4724	46,150	4565	11,356
37	4547	50,948	4724	65,336	4538	40,433
38	4614	12,633	4731	12,408	4538	12,379
39	4592	16,397	4733	18,962	4915	17,245
40	4547	21,879	4724	41,558	4565	14,950
41	4547	31,654	4723	50,452	4571	17,577
42	4547	19,962	4724	18,479	4579	16,586
43	4547	18,610	4742	15,691	4579	19,981
44	4941	29,505	4812	30,587	4538	21,550
45	4543	17,021	4724	32,454	4538	15,814
46	4547	51,872	4723	45,495	4545	34,814
47	4547	27,479	4812	27,880	4538	12,855
48	4561	26,510	4724	24,879	4538	17,598
49	4543	57,815	4731	43,597	4565	35,713
50	4547	54,753	4724	37,585	4579	35,681
Média	4567,12	38,384	4733,54	32,265	4569,02	20,474

5.7 Algoritmos Genético-Tabu (AGTs)

A seguir, realiza-se a comparação dos resultados obtidos por meio das implementações realizadas no MATLAB® (AGTs e as três variações do cálculo do *makespan*). Para cada teste de todas as versões do AGT, a população inicial é equivalente e gera-se, de forma aleatória, uma nova população para cada versão do AGT.

O AGT proposto neste trabalho, cujos parâmetros e valores encontram-se descritos anteriormente, baseia-se em AGs, AGAs e BTs. Como se pode observar pelos testes das BTs, a BT clássica obteve os melhores resultados em todos eles, portanto, somente ela é agregada ao AG e ao AGA para formar o AGT. Essa agregação ocorre em dois momentos distintos:

- **1º momento:** geração da população inicial dos algoritmos, onde a variante da BT é aplicada em cada indivíduo pertencente à população inicial dos AGs e AGAs.
- **2º momento:** otimização dos resultados fornecidos pelos algoritmos (AGs e AGAs).

É realizada a otimização tanto da população inicial do AG, quanto da população inicial do AGA. Para isso, aplica-se a BT com a estratégia da primeira melhora, com o intuito de reduzir o tempo computacional do AGT. Já para otimizar os resultados finais do AG e do AGA, aplica-se a BT clássica sem nenhuma estratégia agregada.

Para cada versão do AG e do AGA existem três versões distintas, pois tanto a forma de cálculo do *makespan* proposta por Deriz (2007), quanto à forma de cálculo proposta por Sanches (2008), e também a forma de cálculo proposta neste trabalho, são aplicadas aos algoritmos, ou seja, existem “3” versões do AGT são geradas (AGT_V1, AGT_V2 e AGT_V3), e a cada um destas versões são aplicadas as duas BTs, citada acima. Assim gerando “6” algoritmos distintos.

A aplicação da BT na otimização da população inicial dos AGs e dos AGAs obteve bons resultados, porém isso aumentou de forma considerável o seu tempo computacional. Isso pode ser observado ao se analisar os valores da média dos tempos de processamento obtidos na Tabela 5-11.

Já a aplicação da BT na otimização da solução final dos AGs e dos AGAs, além dos bons resultados, teve o seu tempo computacional reduzido, se comparado aos resultados da versão anterior (otimização da população inicial) anterior, o que torna sua aplicação mais

satisfatória. Isso pode ser verificado pela análise das médias dos tempos de processamento obtidos, tanto na Tabela 5-12, quanto na Tabela 5-13.

O AGT reduziu o valor do *makespan* em aproximadamente 10%, o que representa 200 (u.t.), em média. Isto indica que houve uma melhora significativa na solução do problema, pois o valor agregado ao produto final é reduzido e, consequentemente reduziu-se seu valor de manufatura, podendo assim ser vendido há um preço mais competitivo.

Quando se comparam os resultados dos AGTs aos resultados obtidos por meio dos AGs e dos AGAs, observa-se que os AGTs obtiveram uma melhor solução, embora em um tempo computacional um pouco maior que os do AGs e AGAs. No entanto, o tempo de obtenção da resposta dos AGTs não acarretam nenhum dano ao sistema produtivo, o que pode manter o seu alto desempenho.

Quando se comparam os resultados dos AGTs com os das BTs, observa-se que eles são muito próximos, praticamente equivalentes e os resultados dos AGTs apresentam tempo computacional menor aos da BT, o que indica que tanto o AG, quanto o AGA forneceram à BT resultados muito bons.

Portanto, neste trabalho, indica-se a otimização dos resultados fornecidos pelo AGs e pelo AGAs, sendo que as BTs, no melhor dos casos, melhorou a solução em 10% e, no pior dos casos, empatou com os AGs e AGAs.

Nos testes realizados, tanto o AGT baseado no AG, quanto o AGT baseado no AGA obtiveram bons resultados em um tempo computacional equivalente. Portanto, na proposta com a aplicação da BT não se justifica a utilização de um ou de outro, ambos podem ser aplicados. Isso pode ser observado pela análise das médias constantes tanto da Tabela 5-12, quanto da Tabela 5-13. Mas os AGAs, no entanto, exploram o espaço de busca de uma forma mais abrangente que o AG, sendo assim recomenda-se que o AGT, utilize como base o AGA.

A Tabela 5-11 apresenta os resultados obtidos na otimização da população inicial dos AGs. Nela estão relacionados os *makespans* obtidos, assim como o tempo computacional necessário para os cálculos (otimização da população inicial e a execução dos AGs). Já a Tabela 5-12 e a Tabela 5-13 apresentam os resultados obtidos na otimização do resultado final dos AGs. Nelas estão relacionados os *makepans* obtidos por eles, e o tempo computacional da aplicação dos algoritmos, respectivamente desenvolvidos.

Tabela 5-11 - AGT: Otimização da população inicial

	AGT - AG				AGT - AGA								
	Proposta		Sanches (2008)		Deriz (2007)		Proposta		Sanches (2008)		Deriz (2007)		
	Makespan (u.t.)	Tempo (s)											
1	4572	1396,05	4737	1197,96	4549	461,49	4572	1394,27	4737	1196,36	4549	460,67	
2	4547	1395,67	4737	1197,71	4538	460,90	4547	1393,74	4737	1196,04	4538	460,35	
3	4554	1396,56	4742	1199,01	4545	461,67	4554	1395,06	4742	1197,06	4545	461,00	
4	4572	1396,37	4756	1198,34	4538	461,69	4572	1394,24	4756	1196,66	4538	460,94	
5	4547	1395,79	4733	1197,92	4538	461,16	4547	1393,84	4733	1196,24	4538	460,53	
6	4572	1396,22	4724	1198,70	4564	461,53	4572	1394,37	4724	1196,72	4564	460,65	
7	4550	1395,81	4755	1197,83	4545	461,09	4550	1394,00	4755	1196,03	4545	460,52	
8	4576	1396,66	4733	1198,40	4579	461,70	4576	1394,72	4733	1197,06	4579	461,19	
9	4547	1395,90	4724	1197,88	4538	461,15	4547	1393,85	4724	1196,07	4538	460,53	
10	4543	1395,84	4724	1197,66	4538	461,11	4543	1393,89	4724	1196,05	4538	460,43	
11	4547	1396,46	4724	1198,24	4538	461,53	4547	1394,39	4724	1196,46	4538	460,88	
12	4585	1397,30	4727	1198,80	4538	461,31	4585	1394,62	4727	1197,27	4538	462,63	
13	4547	1396,42	4724	1198,43	4538	461,22	4547	1395,13	4724	1196,46	4538	460,71	
14	4547	1397,47	4727	1199,02	4538	461,98	4547	1394,92	4727	1197,49	4538	460,79	
15	4561	1395,80	4731	1197,80	4538	461,05	4561	1393,76	4731	1196,35	4538	460,31	
16	4554	1395,60	4729	1197,57	4545	460,90	4554	1393,81	4729	1196,01	4545	460,25	
17	4547	1396,42	4738	1198,42	4538	461,79	4547	1394,59	4738	1196,67	4538	460,99	
18	4543	1396,42	4724	1198,48	4549	461,70	4543	1394,31	4724	1197,13	4549	461,05	
19	4543	1396,03	4724	1198,03	4538	461,36	4543	1393,97	4724	1196,45	4538	461,01	
20	4572	1396,30	4733	1197,64	4538	460,84	4572	1394,28	4733	1196,13	4538	460,22	
21	4543	1395,99	4732	1197,96	4538	461,07	4543	1394,06	4732	1196,23	4538	460,49	
22	4543	1396,17	4724	1198,11	4538	461,27	4543	1394,84	4724	1196,39	4538	460,64	
23	4547	1396,51	4724	1198,60	4538	461,73	4547	1394,87	4724	1196,70	4538	461,01	
24	4554	1395,82	4724	1197,72	4545	460,98	4554	1393,61	4724	1196,40	4545	460,50	
25	4550	1395,88	4733	1197,70	4570	460,94	4550	1393,65	4733	1195,99	4570	460,22	
26	4585	1395,96	4727	1198,02	4571	461,05	4585	1393,67	4727	1196,12	4571	460,32	
27	4572	1396,34	4742	1198,37	4570	461,61	4572	1394,58	4742	1196,50	4570	460,85	
28	4554	1396,68	4724	1198,49	4579	461,64	4554	1394,73	4724	1197,13	4579	460,94	
29	4547	1396,45	4724	1198,44	4538	461,55	4547	1395,17	4724	1196,59	4538	460,81	
30	4554	1396,17	4724	1198,20	4545	461,39	4554	1394,20	4724	1197,06	4545	460,70	
31	4547	1396,28	4733	1198,40	4538	461,47	4547	1394,37	4733	1196,46	4538	461,25	
32	4561	1396,00	4729	1198,01	4538	461,20	4561	1394,22	4729	1196,29	4538	460,40	
33	4572	1396,50	4727	1198,49	4566	461,62	4572	1394,30	4727	1197,02	4566	460,94	
34	4583	1395,99	4733	1197,97	4538	461,14	4583	1395,26	4733	1197,16	4538	460,34	
35	4547	1396,52	4724	1198,38	4549	461,56	4547	1394,56	4724	1197,24	4549	460,91	
36	4547	1396,00	4724	1197,84	4538	461,04	4547	1394,97	4724	1197,82	4538	460,40	
37	4554	1396,26	4724	1198,15	4565	461,25	4554	1393,92	4724	1196,49	4565	460,54	
38	4554	1396,51	4724	1198,36	4538	461,50	4554	1395,38	4724	1197,17	4538	460,86	
39	4561	1396,66	4733	1198,57	4538	461,65	4561	1394,52	4733	1196,94	4538	461,08	
40	4591	1395,82	4724	1197,86	4545	460,98	4591	1393,86	4724	1195,98	4545	460,31	
41	4547	1396,76	4731	1198,76	4538	461,87	4547	1394,58	4731	1196,86	4538	461,11	
42	4547	1396,60	4733	1198,55	4565	461,61	4547	1394,46	4733	1197,07	4565	461,16	
43	4547	1396,68	4724	1198,16	4570	461,29	4547	1394,09	4724	1196,36	4570	460,64	
44	4547	1395,93	4724	1197,98	4538	461,28	4547	1394,05	4724	1196,23	4538	460,58	
45	4583	1395,91	4755	1197,93	4565	461,17	4583	1393,86	4755	1196,32	4565	460,54	
46	4547	1395,91	4724	1197,93	4571	461,24	4547	1393,81	4724	1196,64	4571	460,36	
47	4547	1396,23	4740	1198,05	4538	461,40	4547	1396,06	4740	1196,62	4538	460,60	
48	4547	1398,04	4724	1199,55	4538	461,73	4547	1394,61	4724	1197,30	4538	461,39	
49	4576	1396,39	4733	1198,54	4549	461,75	4576	1395,24	4733	1196,61	4549	460,98	
50	4547	1396,34	4730	1198,38	4565	461,65	4577	1394,63	4730	1197,01	4565	460,93	
Média		4556,54	1396,29	4730,34	1198,23	4547,84	461,38	4557,14	1394,40	4730,34	1196,63	4547,84	460,75

Tabela 5-12 - AGT: Otimização do resultado final do AGs

-	Proposta			AGT - AG Sanches (2008)			Deriz (2007)		
	AG	BT	Tempo (s)	AG	BT	Tempo (s)	AG	BT	Tempo (s)
	Makespan (u.t.)	Makespan (u.t.)		Makespan (u.t.)	Makespan (u.t.)		Makespan (u.t.)	Makespan (u.t.)	
1	5015	4561	27,581	4790	4742	20,920	4999	4618	9,157
2	4549	4583	41,567	4737	4724	34,984	4949	4545	14,346
3	4648	4592	25,715	5150	4724	32,222	5035	4545	9,279
4	4637	4547	23,582	4784	4724	21,432	4600	4598	9,071
5	4949	4583	22,347	5159	4729	25,410	5004	4538	10,101
6	4978	4547	27,504	5141	4724	41,994	4981	4538	17,441
7	4997	4591	33,776	4781	4724	23,368	4955	4538	10,598
8	4927	4949	21,781	4815	4785	21,583	4996	4538	14,614
9	4602	4579	24,493	4787	4750	20,133	4590	4571	9,072
10	4966	4583	22,672	5157	4724	25,001	4962	4538	9,699
11	4898	4579	22,712	5071	4723	25,005	4884	4538	15,851
12	4628	4543	31,282	5188	4724	29,728	4610	4538	9,282
13	4661	4547	31,031	4742	4742	20,224	4973	4545	12,755
14	4881	4547	33,712	5197	4741	23,171	4982	4538	9,853
15	4886	4572	26,283	5110	4724	26,816	4610	4538	10,818
16	4955	4641	33,761	5253	4724	27,836	4834	4579	15,298
17	5026	4547	33,344	4793	4727	36,527	4587	4579	8,716
18	4931	4547	42,433	5236	4724	34,590	5007	4579	13,189
19	5028	4547	28,234	5196	4733	21,413	4948	4565	14,411
20	5038	4585	23,299	5222	4724	22,251	4586	4565	10,271
21	4681	4591	25,223	5197	4738	24,018	4916	4565	13,904
22	4852	4582	34,249	5023	4724	28,448	4828	4549	11,004
23	4999	4543	23,012	5188	4724	24,079	4656	4538	9,234
24	4917	4591	23,797	4743	4724	34,227	4962	4538	10,008
25	4955	4543	35,421	4813	4724	41,248	4608	4545	12,308
26	4989	4547	23,515	5124	4724	29,702	4564	4538	9,280
27	4597	4547	30,838	4830	4724	32,280	4579	4579	8,251
28	4967	4577	27,425	5231	4724	26,185	4991	4538	10,857
29	5137	4554	22,885	5148	4727	21,347	4975	4545	9,228
30	4628	4543	30,169	5259	4724	28,862	4847	4538	11,143
31	4647	4628	23,605	4818	4742	23,272	4618	4565	12,960
32	5001	4547	32,973	5182	4724	29,094	4996	4549	10,310
33	4860	4547	46,683	4858	4724	40,724	4931	4618	8,974
34	4992	4547	25,281	4734	4724	21,789	4956	4545	10,284
35	5063	4547	31,037	5196	4724	24,797	4900	4594	9,380
36	5014	4543	43,286	5185	4724	35,374	5017	4571	14,882
37	4974	4547	39,423	4772	4758	29,630	4937	4545	19,602
38	4980	4949	23,121	5137	4724	42,773	4937	4937	8,582
39	5068	4949	21,842	5139	4724	30,634	4923	4565	12,244
40	5011	4543	23,848	5109	4727	26,139	5041	4538	13,138
41	4642	4585	23,781	5137	4724	30,677	4884	4571	10,809
42	4931	4543	26,448	4814	4724	22,630	4970	4538	12,080
43	4923	4547	31,914	4784	4724	36,674	4570	4538	12,648
44	4642	4547	24,854	4806	4724	22,188	4930	4538	9,112
45	4634	4547	26,872	4799	4724	23,505	4927	4538	11,899
46	5050	4547	23,110	5259	4724	19,057	4985	4538	9,280
47	5020	4543	26,857	5145	4724	31,600	4990	4538	11,818
48	4976	4547	46,712	4824	4724	38,125	4939	4538	18,922
49	4591	4547	22,547	4782	4724	20,258	4579	4579	8,311
50	4885	4572	22,854	5234	4742	19,654	4608	4565	9,330
Média	4876,52	4585,20	28,814	5011,58	4728,92	27,872	4853,12	4561,04	11,47

Tabela 5-13 - AGT: Otimização do resultado final do AGAs

6 Conclusões e trabalhos futuros

6.1 Conclusões

A programação de produtos com o uso simultâneo de máquinas e sistemas de transporte é caracterizada pela grande quantidade de soluções possíveis. Em especial, para problemas de grande escala que possuem um número considerável de máquinas e produtos para a produção. Por isso, as características dos algoritmos genéticos se tornam apropriadas para tal tarefa, uma vez que esses possuem a capacidade de percorrer de forma global os espaços da busca com as soluções possíveis para o problema proposto. No entanto, os AGs puramente não garantem o quão ótimo é essa solução, podendo existir uma solução próxima, melhor que as encontradas nas regiões apontadas pelos AGs. Assim, inclui-se a BT no processo de busca dos AGs e AGAs, a qual realiza a busca sistemática das regiões por eles apontadas. Com isso, os resultados encontrados estarão mais próximos de uma solução ótima ou quase ótima.

O principal objetivo deste trabalho é propor um algoritmo com base em AGs, AGAs e BTs para realizar a programação da produção. Esse algoritmo gerado é chamado de AGT – Algoritmo Genético-Tabu.

Além da geração e da construção dos AGTs, este trabalho tem também como foco os seguintes aspectos:

- cálculo do *makespan*: adiantar o sistema de transporte, de forma a otimizá-lo e obter o melhor de sua performance;
- geração dos AGs e AGAs: definir suas estruturas de desenvolvimento , os valores de suas taxas, o método de seleção utilizado, formas de aplicação dos mecanismo de cruzamento e mutação, entre outros;
- geração da BT: definir a forma de vizinhança e sua respectiva aplicação, além de levar em consideração as estratégias de intensificação e diversificação;
- geração do AGT, momento de aplicação da BT.

Com relação ao adiantamento do sistema de transporte, observou-se que houve uma melhora significativa nos valores do *makespan* – cerca de 10%, ou seja, aproximadamente 500 u.t., no melhor dos casos. Já no pior caso, houve uma melhora no *makespan* de no mínimo 2%, ou seja, aproximadamente 100 (u.t.). Neste trabalho, além desse

método de cálculo do *makespan*, foram utilizados também os cálculos propostos por Deriz (2007) e por Sanches (2008).

Quanto à definição das estruturas do AG e do AGA, foram desenvolvidas duas versões distintas, sendo que a principal diferença entre elas se dá no momento de aplicação do operador de mutação. A primeira versão aplica o operador de mutação a indivíduos oriundos da população atual. Já a segunda versão aplica o operador de mutação a indivíduos oriundos do operador de cruzamento, ou seja, aos filhos da população atual e também aos indivíduos oriundos do operador que realiza a cópia dos indivíduos da população atual para a próxima população, a fim de conservar e mantê-la com o mesmo número de indivíduos da população anterior. Essa segunda versão favorece a manutenção da diversidade populacional dos algoritmos.

Também foram testados dois métodos de seleção: por roleta e por torneio.

Tanto no AG, quanto no AGA, há a preocupação de proporcionar uma maior diversidade populacional. Este trabalho baseia-se no desenvolvimento dos algoritmos na segunda versão, onde a mutação é aplicada aos filhos do cruzamento e aos indivíduos a copiados, para manter a próxima população com um mesmo número de indivíduos. Esta versão proporciona uma maior diversidade populacional que a primeira versão. E com mesma motivação a manutenção da diversidade populacional, é utilizado neste trabalho como método de seleção a roleta.

Quanto à aplicação da BT, foram desenvolvidas duas vizinhanças distintas: na primeira, troca-se o roteiro de produção, já na segunda, altera-se a ordem da manufatura dos produtos. Foram realizados estudos relacionados à forma de aplicação de vizinhanças propostas ao problema.

Quanto as estratégias agregas a BT, tanto a diversificação, quanto a intensificação são baseadas na frequência das ocorrências de atributos na solução. Na intensificação, os atributos mais frequentes se tornam tabus e se dá prosseguimento na busca. Já na diversificação, os atributos menos frequentes são valorizados e ocupam o lugar dos mais frequentes, gerando, assim, uma nova solução.

A BT demonstrou ser, em média, 7% melhor que o AG e o AGA, mas obteve um tempo computacional maior que ambos. No entanto, seu tempo computacional não interfere no desempenho do sistema produtivo.

A aplicação das estratégias de diversificação e intensificação não demonstrou viabilidade para o problema em questão, pois forneceu o mesmo resultado, porém em um tempo computacional muito maior, comparado ao da BT clássica.

A lentidão da BT será muito maior, caso ocorra o aumento do tamanho do sistema de manufatura, pois seu tempo de execução está diretamente relacionado ao tamanho do problema, o que não ocorre nos algoritmos evolutivos (AG e AGA). No entanto, quando a BT parte de uma boa solução fornecida por algum outro algoritmo, o seu tempo computacional é reduzido.

Mas não se descarta a aplicação da BT ao problema e nem mesmo das estratégias de intensificação e diversificação, pois essas foram propostas com o intuito de reduzir o tempo computacional da BT.

Quanto ao AGT e à sua comparação com os demais algoritmos (BTs, AGs e AGAS), observou-se que o AGT obteve melhores resultados que os demais na maioria dos testes. A BT, no entanto, não demonstrou melhor desempenho, pois ela é mais lenta que o AGT e os demais algoritmos, porém seus resultados se igualaram aos do AGT. Vale ressaltar que os resultados do AGT chegam a ser, em média, 15% melhores que os demais e, nos melhores casos, chegam a 20%. Já nos piores casos os resultados do AGT se igualaram aos da BT.

No AGT, a BT foi aplicada em dois momentos distintos: primeiramente, na otimização da população inicial, o que tornou o tempo computacional muito alto (algoritmo lento) e inviabilizou a sua aplicação. No segundo momento, ela foi aplicada na otimização dos resultados fornecidos pelos algoritmos (AG e AGA), o que melhorou de forma expressiva o tempo computacional de obtenção do *makespan*.

No AGT, ao aplicar a BT nos resultados de *makepan* fornecidos pelos AGs e AGAs, observa-se que a BT melhorou os resultados cerca de 10%, que correspondem, em média, a 200 u.t. A redução do tempo de manufatura de um produto, implica que se reduziu o de manufatura do mesmo, ou seja, reduz no valor agregado ao produto (reduzir o tempo operacional sobre os produtos e a utilização de mão de obra, o que reduzirá os encargos sociais a serem pagos, etc). Assim pode-se fornecer o produto por preços mais competitivos ou até mesmo aumentar a margem de lucro.

Quanto à utilização do AG ou do AGA como base para a aplicação da BT, não se verificou grande influência de um ou de outro para o problema em questão. A vantagem de utilizar o AGA é que ele geralmente fornece uma solução melhor que o AG, pois o AGA explora o espaço de busca melhor que o AG, o que reduz o tempo de aplicação da BT. Em geral, tanto a aplicação da BT ao AG, quanto ao AGA, apresentaram tempos computacionais muito próximos e resultados muito semelhantes.

Com base nesses resultados, observa-se que o AGT, cuja variante corresponde a aplicação da BT ao final do AG ou do AGA, pode ser aplicado a solução de problemas de manufatura com compartilhamento de recursos de pequena, média e grande escala.

Tanto a BT, quanto o AGT, demonstraram-se ser uma aplicação viável ao problema em questão, porém a BT explora uma região do espaço de busca muito menor que a explorada pelo AG e pelo AGA.

Os resultados alcançados pelo AGT aproximaram-se ou até mesmo se igualaram aos obtidos pela BT, mas em um menor tempo computacional, o que tornou o seu uso mais satisfatório. O AGT que obteve o melhor desempenho foi o que teve o seu resultado final aperfeiçoado pela BT, independentemente se o ponto de partida da BT foi obtido pelo AG ou pelo AGA. Portanto, é aconselhável a utilização desse AGT para o problema em questão.

6.2 Trabalhos Futuros

Com base nos resultados obtidos, sugere-se a inclusão de um método construtivo para gerar as soluções iniciais ou a população inicial dos algoritmos, com o objetivo de melhorar o desempenho da busca. Nesse caso, poderia ser incluído o método de busca colônia de formigas ou o método de construção da solução pertencente ao método de busca GRASP, apresentado em vários trabalhos com seu uso agregado tanto nos AGs quanto na BT.

Outra sugestão é a inclusão de lógica *Fuzzy* para trabalhar juntamente com as regras de despacho utilizadas para a programação do sistema de transporte. Nesse caso, outras regras de despacho deveriam ser testadas, a fim de se obter um melhor desempenho para a programação do sistema de transporte e, com isso, alcançar melhores valores de *makespan*.

É interessante também realizar alterações na função objetivo, acrescentando a ela outros parâmetros, ou seja, tornando-a múltiplos objetivos. A seguir, são listadas as possíveis variáveis do chão de fábrica, que podem ser acrescidas na função objetivo, de forma a se aproximar mais de um sistema de manufatura real e a tornar o valor do *makespan* o mais próximo possível do real. Exemplos dessas variáveis seriam: a definição do tamanho do buffer de entrada e de saída das máquinas, os conflitos de rotas, a taxa de utilização de máquinas, o cumprimento de data devida, entre outros. Além disso, pode-se utilizar o algoritmo genético para mais de um objetivo.

Em relação aos AGs, esses poderiam ser testados com outros tipos de operadores de cruzamento, como os utilizados para permutações, como: OBX, PBX, PMX, CX e OX,

que são eficientes para problemas de otimização combinatória. Também poderiam ser testados outros métodos de seleção, como o método de Boltzmann, o método de *rank*, etc.

Sugere-se também que seja alterado o operador de mutação, de maneira que ele opere sobre a alteração da ordem dos produtos, sendo que esses serão escolhidos aleatoriamente no cromossomo e trocados entre si. Após isso, serão trocados os roteiros de fabricação dos produtos escolhidos. A partir daí, o cromossomo terá uma nova ordem de prioridades dos produtos na programação, o que permite a inserção de mais diversidade genética na população.

Quanto à BT, sugerem-se também os testes com outros métodos de diversificação e intensificação, além de testes com outros tipos de vizinhanças e o estudo de métodos que reduzam o tamanho das vizinhanças propostas neste trabalho, o que pode reduzir seu tempo computacional. A BT obteve melhores resultados que os AGs em seus testes, portanto, indicam-se novos estudos de modelagem de estruturas de vizinhança, métodos de diversificação e intensificação, etc.

Sugere-se também o uso de Petri Nets (PNs) não só para a modelagem, mas também para o processo de busca, como por exemplo, a geração da vizinhança, entre outros.

Tais sugestões necessitam ser desenvolvidas isoladamente e testadas para os problemas em questão. Feito isso, deve-se aplicar uma comparação entre os novos resultados e os obtidos neste trabalho, com a finalidade de verificar se houve melhora nas soluções encontradas.

7 Referências Bibliográficas

- BATTITI, R. **Reactive search: Toward self-tuning heuristics.** Modern Heuristic Search Methods, John Wiley & Sons Ltd., p. 61–83, 1996.
- BATTITI, R.; TECCHIOLLI, G. The reactive tabu search. **ORSA Journal on Computing**, v. 6, n. 2, p. 126–140, 1994.
- BATTITI, R.; TECCHIOLLI, G. The Continuous Reactive Tabu Search Blending Combinatorial Optimization and Stochastic Search for Global Optimization. **Annals of Operations Research**, Springer Netherlands, v. 63, n. 2, p. 151–188, 1995.
- BATTITI, R.; BRUNATO, M.; MASCIA, F. **Reactive Search and Intelligent Optimization.** Springer Verlag, 2008. Operations Research/Computer Science Interfaces Series, Vol. 45, Springer, November 2008.
- BENINCASA, A. X. **Um modelo de sistemas Fuzzy para despacho de veículos autoguiados em manufatura integrada.** Dissertação de Mestrado em Ciência da Computação. Universidade Federal de São Carlos, São Carlos, 2003.
- BREMERMANN, H. **Optimization through evolution and recombination.** Self-Organizing Systems, p. 93–106, 1962.
- CAMPOS, M.C. Estatística prática para docentes e pós graduados. Disponível em: <http://www.forp.usp.br/restauradora/gmc/gmc_livro/gmc_livro.html>. Acesso em: 20 nov. 2010.
- CARVALHO, A. C. P. L. F.; BRAGA, A.P.; LUDEMRIR, T. B. Computação Evolutiva. In: REZENDE, S. **O. Sistemas Inteligentes.** 1^a ed. São Paulo: Editora Manole, 2003, p. 225-248.
- CAVALIERI, S. Petri nets and genetic algorithms to increase productivity in FMS. In: **Second International Conference on Knowledge-Based Intelligent Electronic Systems**, Adelaide, Australia, 1998. p. 134-142.
- CHAN, F. T. S.; CHUNG, S. H.; CHAN, P. L. Y. An adaptive genetic algorithm with dominated genes for distributed scheduling problems. **Expert Systems with Applications**, v 29, p364-371, 2005.
- CHAN, F. T. S.; CHUNG, S. H.; CHAN, P. L. Y; FINKE, G; TIWARI, M. K., Solving distributed FMS scheduling problems subject to maintenance: Genetic algorithms approach. **Robotics and Computer-Integrated Manufacturing**, v 22, p. 493-504, 2006.
- CHIU, Y. F.; FU, L. C. A., GA embedded dynamic search algorithm over a petri net model for an FMS scheduling. In: **Proceedings of the 1997 IEEE International Conference on Robotics and Automation**, Albuquerque, New Mexico, April, 1997. v. 1 p.513-518.

DAVIS, L. **Handbook of genetic algorithms.** Van Nostrand Reinhold New York, 1991.

DERIZ, A. C. **Um Método de Busca usando Algoritmo Genético para Programação Reativa da Produção de Sistemas de Manufatura com Recursos Compartilhados.** Dissertação de Mestrado em Ciência da Computação. Universidade Federal de São Carlos, São Carlos, 2007.

ESWARAMURTHY, V.; TAMILARASI, A. Tabu Search Strategies for Solving Job Shop Scheduling Problems. **Journal of Advanced Manufacturing Systems**, v. 6, n. 1, p. 59–75, 2007.

FOGEL, L. **Autonomous automata.** Industrial Research, v. 4, n. 1, p. 14–19, 1962.

GANG, X., WU, Z. Deadlock-free scheduling method using petri net model analysis and GA search. In: **Proceedings of the 2002 IEEE International Conference Control Applications**, Glasgow, Scotland, U.K., Sep, 2002. p. 1153-1158.

GEYIK, F.; CEDIMOGLU, I. The strategies and parameters of Tabu Search for job-shop scheduling. **Journal of Intelligent Manufacturing, Springer**, v. 15, n. 4, p. 439–448, 2004.

GLOVER, F. **Future paths for integer programming and links to artificial intelligence.** Computers and Operations Research, Elsevier Science Ltd. Oxford, UK, UK, v. 13, n. 5, p. 533–549, 1986.

GLOVER, F. **Tabu search and adaptive memory programming—advances, applications and challenges.** Interfaces in Computer Science and Operations Research: Advances in Metaheuristics, Optimization, and Stochastic Modeling Technologies, Kluwer Academic Pub, p. 1–75, 1996.

GLOVER, F.; TAILLARD, E.; TAILLARD, E. A user's guide to tabu search. **Annals of Operations Research, Springer**, v. 41, n. 1, p. 1–28, 1993.

GLOVER, F. Tabu search fundamentals and uses. Graduate School of Business, **University of Colorado**, Boulder, 1995.

GLOVER, F.; KELLY, J.; LAGUNA, M. Genetic algorithms and tabu search: Hybrids for optimization. **Computers and Operations Research, Elsevier**, v. 22, n. 1, p. 111–134, 1995.

GLOVER, F.; LAGUNA, M. **Tabu Search.** Springer, 1997.

GLOVER, F.; HANAFI, S. Tabu search and finite convergence. **Discrete Applied Mathematics, Elsevier**, v. 119, n. 1-2, p. 3–36, 2002.

GOLDBERG, D. E. **Genetic Algorithms in Search, Optimization and Machine Learning.** Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 1989.

GREFENSTETTE, J.J. Optimizations of control parameters for genetic algorithms. **IEEE Trans. Systems, Man, and Cybernetics**, v 16, p. 122-128, 1986.

HAQ, A. N.; KARTHIKEYAN, T.; DINESH, M. Scheduling Decisions in FMS using a Heuristic Approach. **International Journal Advanced Manufacturing Technology**, 2003, v.22, p. 374-379.

HOLLAND, J. Outline for a logical theory of adaptive systems. **Journal of the ACM (JACM)**, ACM Press New York, NY, USA, v. 9, n. 3, p. 297–314, 1962.

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

HUANG, Z., WU, Z. Deadlock-free scheduling method for automated manufacturing systems using genetic algorithm and petri nets. In: **Proceedings of the 2004 IEEE International Conference on Robotics and Automation**, New Orleans, LA, April, 2004. p. 66-571.

JANAK, S. L.; FLOUDAS, C. A.; KALLRATH, J.; VORMBROCK, N. **Production scheduling of a large-scale industrial batch plant. II. Reactive scheduling**. Industrial and Engineering Chemistry Research, Columbus, v. 45, n. 25, p. 8253-8269, December 2006.

JEONG, S. J.; LIM, S. J.; KIM, K. S. Hybrid approach to production scheduling using genetic algorithm and simulation. **International Journal of Advanced Manufacturing Technology**, London, v. 28, p. 129 – 136, 2006.

JERALD, J; ASOKAN P.; SARAVANAN, R.; A. D. C. Simultaneous scheduling of parts and automated guided vehicles in an FMS environment using adaptive genetic algorithm. **The International Journal of Advanced Manufacturing Technology**, Springer, v. 29, n. 5, p. 584–589, 2006.

KOZA, J. Genetic Programming: On the Programming of Computers by Means of Natural Selection. **University of Michigan Press**, 1992.

LACERDA, E.; CARVALHO, A. Introdução aos Algoritmos Genéticos. **Anais do XIX Congresso Nacional da Sociedade Brasileira de Computação**, v. 2, p. 51–126, 1999.

LEE, D. Y. e DICESARE, F. FMS scheduling using petri nets and heuristic search. In: **Proceedings of the 1992 IEEE International Conference on Robotics and Automation, Anais**. Nice, Fr, May 1992. p. 1057-1062.

LEE, D. Y. e DICESARE, F. Scheduling flexible manufacturing systems using petri nets and heuristic search. **IEEE Transactions on Robotics and Automation**, v.10, n. 2, p. 123-133, 1994.

LI, Z.; IERAPETRITOU, M. Process scheduling under uncertainty: Review and challenges. **Computers and Chemical Engineering**, Oxford, v. 32, n. 4-5, p. 715-727, April 2008.

LOURIDAS, P. JUnit: Unit Testing and Coding in Tandem. **IEEE Software**, v. 22(4), p. 12-15, 2005.

MAGGIO, E. G. R. **Uma heurística para a programação da produção de sistemas flexíveis de manufatura usando modelagem em redes de petri**. Dissertação de Mestrado em Ciência da Computação. Universidade Federal de São Carlos, São Carlos, 2005.

MARTINS, G. A. **Estatística geral e aplicada**. 3. Ed. São Paulo: Atlas, 2005.

MICHALEWICZ, Z. **Genetic Algorithms + Data Structures = Evolution Programs**. 3.ed. Springer-Verlag, 1994.

MITCHELL, M. An Introduction to Genetic Algorithms. **University of Michigan Press**, 1998.

MITCHELL, TM. **Machine Learning**. Mac Graw-Hill Science Engineering. 1997.

MONTEVECHI, J.A.B.; MORANDIN JR., O.; MIYAGI, P.E. Sistemas de manufatura In: AGUIRRE, L.A. **Enciclopédia de Automática, Controle & Automação**. 1ed. São Paulo: Editora Blucher, 2007, p. 247-287.

MORANDIN JR, O.; KATO E. R. R.; MAGGIO, E. G. R.; SANCHES, D. S.; DERIZ, A. C. Heuristic based on Petri nets modeling for FMS scheduling problem of makespan minimization, **33rd Annual Conf. of the IEEE Industrial Electronics Society**, 2007a, Taipei – Taiwan, p. 2683-2688.

MORANDIN JR, 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, **XXVII Encontro Nacional de Engenharia de Produção**, 2007b, Foz de Iguaçu – Brasil.

MORANDIN JR, O. ; Deriz A. C. ; SANCHES, D. S. ; KATO E. R. R . A Search Method using Genetic Algorithm for Production Reactive Scheduling Manufacturing Systems. In: **IEEE International Symposium on Industrial Electronics**, 2008a, Cambridge, UK.

MORANDIN JR, O.; SANCHES, D. S.; Deriz A. C.; KATO E. R. R.; TSUNAKI, R. H. An Adaptive Genetic Algorithm Based Approach for Production Reactive Scheduling of Manufacturing Systems. In: **34rd Annual Conf. of the IEEE Industrial Electronics Society**, 2008b, Orlando, USA.

MORANDIN, O. **Metodologia de Modelagem de Sistemas Automatizados de Manufatura**, Utilizando Rede de Petri Virtual. Tese (Doutorado) — Universidade de São Paulo – Escola de Engenharia de São Carlos, São Carlos, 1999.

MURATA, T. Petri Nets: Properties, analysis and applications. In: **Proceedings of the IEEE**. v. 77, n.4, p.541-580, 1989.

PONGCHAROEN P.; HICKS C.; BRAIDEN P. M.; STEWARDSON D.J., Determining optimum genetic algorithm parameters for scheduling the manufacturing and assembly of complex products. **International Journal of Production Economics.** n. 78, p. 311–322, 2002.

PONGCHAROEN, P.; HICKS, C.; BRAIDEN, P.M. The development of genetic algorithms for the finite capacity scheduling of complex products, with multiple levels of product structure. **European Journal of Operational Research.** n. 152, p. 215–225, 2004.

QIU, R.; ZHOU, M. Mighty MESs: state-of-the-art and future manufacturing execution systems. **Robotics & Automation Magazine, IEEE,** v. 11, n. 1, p. 19–25, 2004.

RECHENBERG, I. Evolutions strategie: Optimierung technischer Systeme nach Prinzipien der Biologischen Evolution. Frommann-Holzboog Verlag 1973.

REDDY, B. S. P. e RAO, C. S. P. A hybrid multi-objective GA for simultaneous scheduling of machines and AGVs in FMS. **International Journal Advanced Manufacturing Technology,** v.31, p. 601-613, 2006.

RENDERS, J.; FLASSE, S. Hybrid methods using genetic algorithms for global optimization. **Systems, Man and Cybernetics, Part B, IEEE,** v. 26, n. 2, p. 243–258, 1996.

REYES, A., YU, H. e KELLEHER, G. Advanced scheduling methodologies for flexible manufacturing systems using Petri nets and heuristic search. In: **ICRA 2000: IEEE International Conference on Robotics and Automation. Anais.** San Francisco, CA, USA, 2000. p. 2398-2403

REYES, A., YU, H. e LLOYD, S. An evolutionary hybrid scheduler based in petri net structures for FMS scheduling. In: **IEEE International Conference on Systems Man and Cybernetics,** 2001, v. 4, p. 2516 – 2521.

REYES, A., YU, H. e KELLEHER, G. Hybrid heuristic search for the scheduling of flexible manufacturing systems using Petri nets. **IEEE Transactions on Robotics and Automation,** v.18, n. 2, p. 240-245, 2002.

REYES, A., et al. Integrating petri nets and hybrid heuristic search for the scheduling of FMS, **Computers in Industry,** v.47, n. 1, p. 123-138. 2002.

RUSSEL, S. J. & NORVIG, P. **Inteligência artificial: um enfoque moderno.** 2. Ed. São Paulo: Campus, 2004.

SANKAR, S.S; PONNAMBALAM S. G.; RAJKUMAR R.; GURUMARIMUTHU M., An Intelligent Integrated Scheduling Model for Flexible Manufacturing System, Proceedings of the 2004 **IEEE Conference on Robotics, Automation and Mechatronics,** Singapore, December, 2004, p.1095-1100.

SANCHES, D. S.; MORANDIN JR, O.; DERIZ, A. C.; KATO E. R. R.; TSUNAKI, R. H. An Adaptive Genetic Algorithm Based Approach for Production Reactive Scheduling of Manufacturing Systems. **34rd Annual Conf. of the IEEE Industrial Electronics Society**, Orlando – USA, November, 2008, p.1461-1466.

SANCHES, D. S. **Estratégia de Modelagem por Algoritmo Genético Adaptativo para Programação Reativa da Produção de Produtos com uso Simultâneo de Máquinas e Sistema de Transporte em Sistemas de Manufatura**. Dissertação de Mestrado em Ciência da Computação. Universidade Federal de São Carlos, São Carlos, 2008.

SANCHES, D. S.; MORANDIN JR, O.; MUNIZ, B. D.; KATO E. R. R. Modeling Strategy by Adaptive Genetic Algorithm for Production Reactive Scheduling with Simultaneous use of Machines and AGV's. In: **IEEE International Conference on Industrial Technology - ICIT2009**, 2009, Churchill.

SCRICH, C.; ARMENTANO, V.; LAGUNA, M. Tardiness minimization in a flexible job shop: A tabu search approach. **Journal of Intelligent Manufacturing**, Springer, v. 15, n. 1, p. 103–115, 2004.

SILVA, E. L.; MENEZES, E. M. **Metodologia da pesquisa e elaboração de dissertação**. 3 ed. Florianópolis: Laboratório de Ensino à Distância da UFSC, 2001

SINHA, A.; GOLDBERG, D. E. A Survey of Hybrid Genetic and Evolutionary Algorithms. Illinois : Illinois **Genetic Algorithms Laboratory**, 2003. 22 p.

SINHA, A.; GOLDBERG, D. E. Verification and extension of the theory of global-local hybrids. **Proceedings of the Genetic and Evolutionary Computation Conference - 2001**, 592-598.

SLACK, N.; CHAMBERS S.; JOHNSTON R., **Administração da Produção**. São Paulo: Atlas, 2002.

SRINIVAS, M.; PATNAIK, L.M. Adaptive Probabilities of Crossover and Mutation in Genetic Algorithms. **IEEE Transactions on Systems, Man and Cybernetics**, vol. 24, n. 4, p. 656-667, April, 1994a.

SRINIVAS, M.; PATNAIK, L.M. Genetic Algorithms: A Survey. **IEEE Computer**, vol. 27, n. 6, p. 17-26, June 1994b.

SUN, J.; XUE D. A dynamic reactive scheduling mechanism for responding to changes of production orders and manufacturing resources. **Computers in Industry**, v. 46, n. 2, p. 189-207, September 2001.

TANG, L.; WANG X. A predictive reactive scheduling method for color-coating production in steel industry. **International Journal Advanced Manufacturing Technology**, Guildford, v. 35, n. 7-8, p. 633-645, January 2008.

TAKAHASHI, K., YAMAMURA, M. e KOBAYASHI, S. A GA approach to solving reachability problems for petri nets. In: **IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences. Institute of Electronics, Inf & Commun Engineers of Japan**, v. 79-A, n 11, p 1774-1780, Nov 1996.

TEMPELMEIER, H. e KUHN, H. **Flexible manufacturing systems** – decision support for design and operation. New York, NY: John Wiley & Sons, Inc. 1993.

TUBINO, D. F. **Manual de planejamento e controle da produção**. 2 ed. São Paulo: Atlas, 2000.

ULUSOY, G.; SIVRIKAYA, F.; BILGE, U. A Genetic Algorithm Approach to the Simultaneous Scheduling of Machines and Automated Guided Vehicles. **Computers & Operations Research**, vol. 24, pp. 335-351, 1997.

VALETTE, R., PRANDIN-CHÉZAVIEL, B. e GIRAUT, F. **An introduction to petri net theory**. Physica-Verlag Fuzziness in Petri Nets – Studies in Fuzzyness and Soft Computing. Heidelberg, Germany: CARDOSO, J.; CAMARGO, H., v. 22, 3-24, 1999.

VILCOT, Geoffrey e BILLAUT, Jean-Charles. A tabu search and a genetic algorithm for solving a bicriteria general job shop scheduling problem. **European Journal of Operational Research**, Elsevier, v. 127, n. 2, p. 398-411, 2008.

WHITLEY, D. **A genetic algorithm tutorial**. **Statistics and Computing**, Springer, v. 4, n. 2, p. 65–85, 1994.

YEUNG, W. H. R., MOORE, P. R. Genetic algorithms and flexible process planning in the design of fault tolerant cell control for flexible assembly systems. **International Journal of Computer Integrated Manufacturing**, vol. 13, n 2, p. 157-168, 2000.

YIN, Y.; YIU, J.; CHENG, Z. A Genetic Algorithm Based Approach to Flowshop Scheduling, **Proceedings of the 5" World Congress on Intelligent Control and Automation**, Hangzhou, P.R. China, June, 2004.

ZHANG, L.; WANG, L.; ZHENG, D. An Adaptive Genetic Algorithm with multiple operatorsfor Flowshop Scheduling, **International Journal Advanced Manufacturing Technology**, v. 27, p. 580-587, 2006.

ZHOU, H.; FENG, Y.; HAN, L. The hybrid heuristic genetic algorithm for job shop scheduling. **Computers & Industrial Engineering**. Elsevier, v. 40, n. 3, p. 191–200, 2001.

ZHOU, L.; XIN, S.S. A Self-Adaptive Genetic Algorithm for Tasks Scheduling in Multiprocessor System, **Proceedings of the 2006 IEEE International Conference on Communications**, Circuits and Systems, China, June, 2006, p.2098-2101.

8 Apêndices

8.1 Apêndice A – Métodos de Seleção (Roleta e Torneio)

A seleção por roleta ocorre da seguinte forma:

- Calcula-se a faixa de cada indivíduo (expectativa de seleção) segundo a função $E(i) = F(i) / \text{Soma}(F)$ e, a seguir ,realiza-se uma soma progressiva dos resultados;
- Geram-se números aleatórios no intervalo de 0 a 1 e seleciona-se o indivíduo que está contido na faixa em foco. Realiza-se esse sorteio até que se tenha selecionado todos os pais;
- Para um indivíduo pertencer a uma determinada faixa, o número gerado deve ser menor que o resultado da soma progressiva para esse indivíduo.

A seguir, encontra-se o código fonte da função que realiza a seleção por roleta.

```

function
parents = selection_roulette(Scores,SizePop,qt_Parents)
%% cálculo da faixa de cada indivíduo na roleta
    expectation = zeros(SizePop,1); %% cria-se um vetor para representar a
    roleta
    sum_Scores = sum(Scores); %% soma das pertinências dos indivíduos
    for i = 1:SizePop %%pertinência de cada individuo na roleta p(i) =
    F(i)/Soma(F)
        expectation(i,1) = (Scores(i)/sum_Scores); %%cálculo da
        pertinência de cada indivíduo
    end
    wheel = cumsum(expectation);%% realiza a soma das pertinências
    parents = zeros(1,qt_Parents); %%armazenará os indivíduos selecionados
    for i = 1:qt_Parents
        r = rand; %%geração de um número aleatório
        for j = 1:length(wheel)
            if(r < wheel(j)) %%verificação da faixa de pertinência
                parents(i) = j; %% indica o indivíduos que forá selecionado
                break; %%força a saída do for
            end
        end
    end
end

```

A seleção por torneio ocorre da seguinte forma:

- Gera-se uma matriz P *m por n*; onde “*m*” indica a quantidade de pais a serem selecionados e “*n*” indica a quantidade de vezes que o torneio será realizado;
- Geram-se números aleatórios no intervalo de 1 a TP ; onde “ TP ” é corresponde ao tamanho da população e os números gerados são armazena na matriz P . Isso é realizado até que se complete a matriz;
- A seguir, seleciona-se a pontuação de cada indivíduo selecionado e comparam se os seus respectivos valores, linha a linha. O indivíduo vencedor é aquele com a melhor pontuação para o problema em questão;
- Cada linha da matriz P corresponde a um indivíduo “pai” a ser selecionado.

A seguir encontra-se o código fonte da função que realiza a seleção por torneio.

```
function
parents = selection_tournament (Scores,SizePop,qt_Parents,TournamentSize)
%% gera matriz contendo os indivíduos selecionados para a aplicação do
%% torneio cada linha da matriz corresponde ao individuo que ser
%% selecionado, portanto caso necessite selecionar 30 indivíduos esta
%% matriz conterá um total de 30 linhas
%% a quantidade de colunas que esta matriz possuirá dependo da quantidade
%% de indivíduos que serão selecionados para o torneio
playerlist = ceil((SizePop) * rand(qt_Parents,TournamentSize));
%% seleção dos parentes
parents = zeros(1,qt_Parents);
for i = 1:qt_Parents
    players = playerlist(i,:);
    %%seleção da pontuação dos indivíduos selecionados
    players_scores = Scores(players);
    %%seleção do melhor indivíduo
    winner_players = min(players_scores);
    %%posição do melhor individuo
    winner = find(players_scores == winner_players);
    parents(i) = players(winner(1));
end;
end
```

8.2 Apêndice B – Resultados dos AGs e AGAs

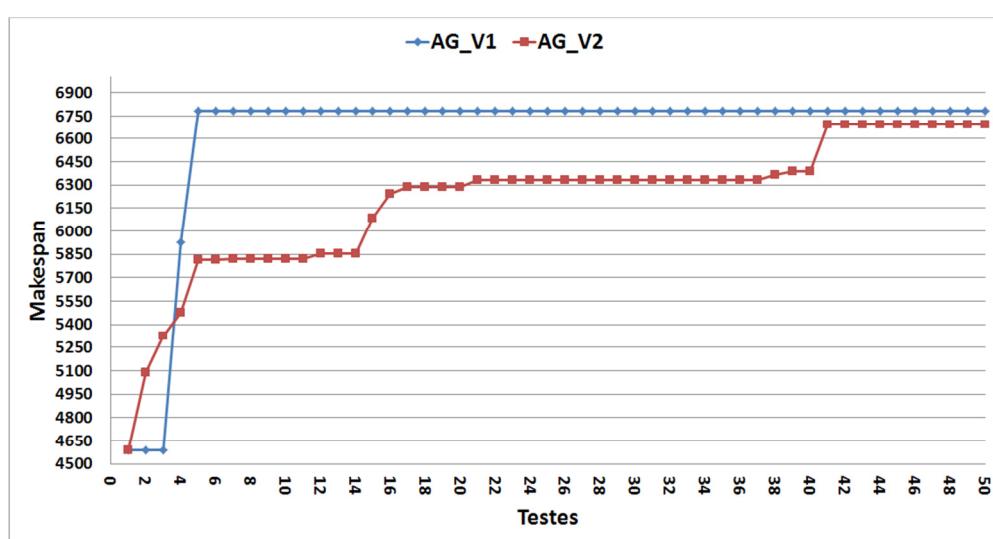
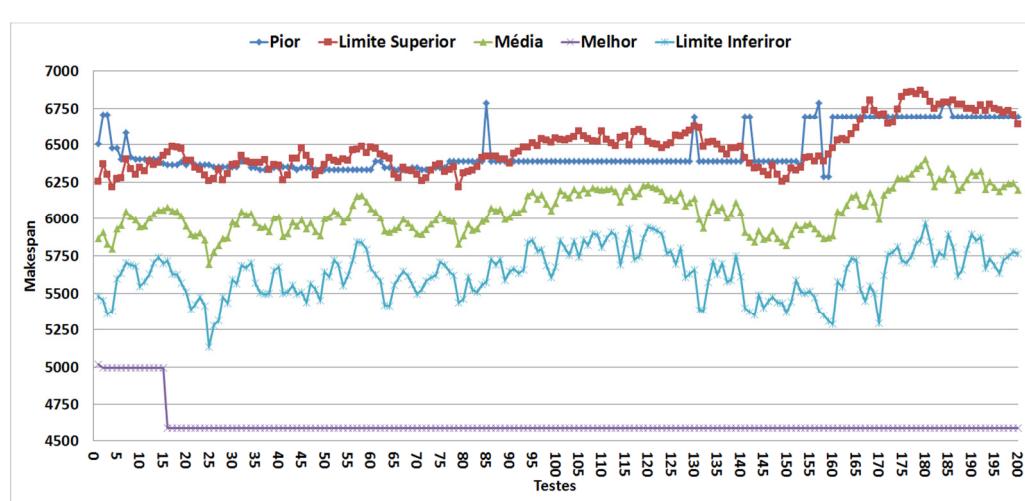
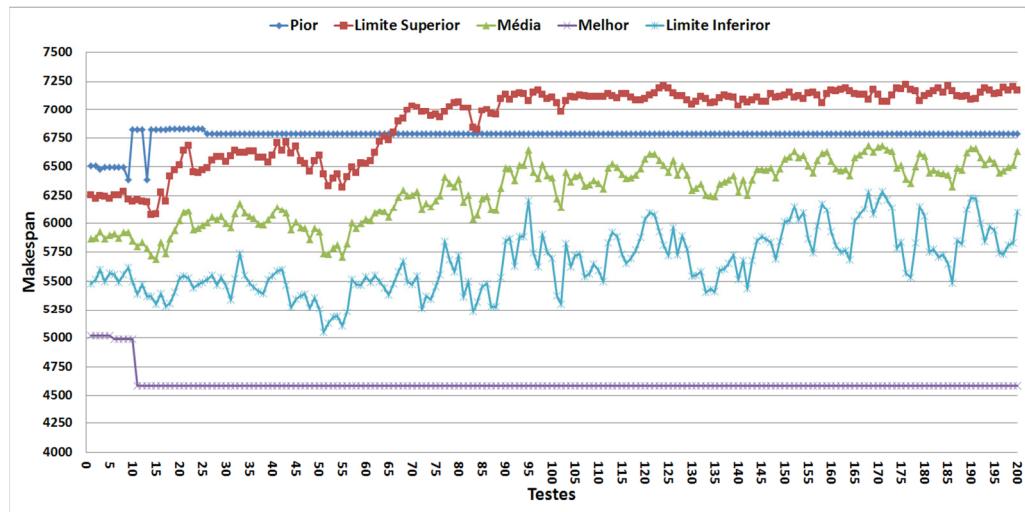


Tabela 8-1 - AG_V1 X AG_V2

-	AG_V1		AG_V2	
	Makespan (u.t.)	Tempo (s)	Makespan (u.t.)	Tempo (s)
1	4905	25,293	4937	22,863
2	4991	23,311	4931	22,922
3	4892	22,766	4684	23,028
4	4948	23,993	4937	23,442
5	4606	24,318	4937	24,534
6	4937	24,948	4645	24,070
7	4970	23,747	4937	23,721
8	4956	23,443	4956	23,742
9	4865	23,648	4992	23,429
10	4595	23,539	4961	23,247
11	4987	23,371	4579	23,609
12	4598	23,799	4881	23,798
13	4915	24,662	4608	25,459
14	4595	24,409	4595	24,958
15	4976	24,885	4662	24,177
16	4618	23,560	4571	24,397
17	4937	24,700	4595	24,582
18	4708	23,600	4939	24,513
19	4937	24,211	4892	24,478
20	4977	23,432	5008	24,792
21	4985	23,729	4834	24,381
22	4937	24,729	4610	25,057
23	4862	24,528	4948	26,083
24	4871	25,512	4871	24,989
25	5014	25,866	5003	26,364
26	4905	25,108	4905	24,561
27	4944	24,454	4610	23,997
28	4938	25,109	5042	28,314
29	4584	27,389	4584	24,704
30	4913	23,038	4910	22,939
31	4976	22,700	4937	23,208
32	4913	23,198	4618	22,597
33	4853	23,339	4890	23,168
34	4937	22,851	4938	22,808
35	4587	23,248	4587	23,047
36	4594	22,589	4862	22,735
37	4637	22,683	4889	23,067
38	4945	22,846	4945	22,842
39	4700	22,607	4937	22,961
40	4538	23,064	4937	22,686
41	4889	22,851	4603	22,927
42	4600	23,181	4937	22,990
43	4605	23,153	4944	23,106
44	4579	22,756	4586	22,669
45	4890	22,662	4969	22,826
46	4865	23,227	4598	22,768
47	4984	22,599	4900	22,722
48	4586	23,283	4586	23,351
49	4618	23,414	4618	23,090
50	4965	23,448	4965	23,305
Média		4822,54	23,776	4816,20
				23,800

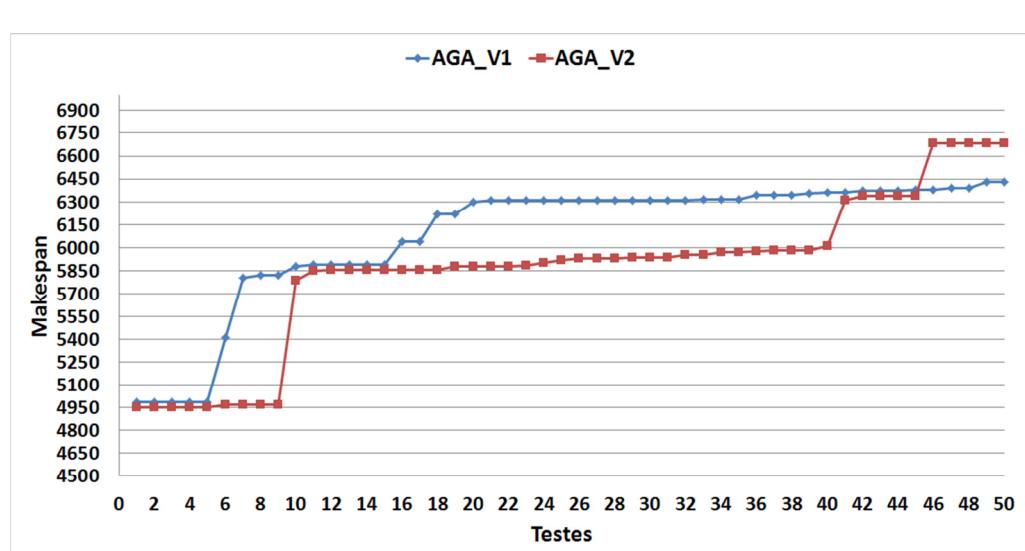
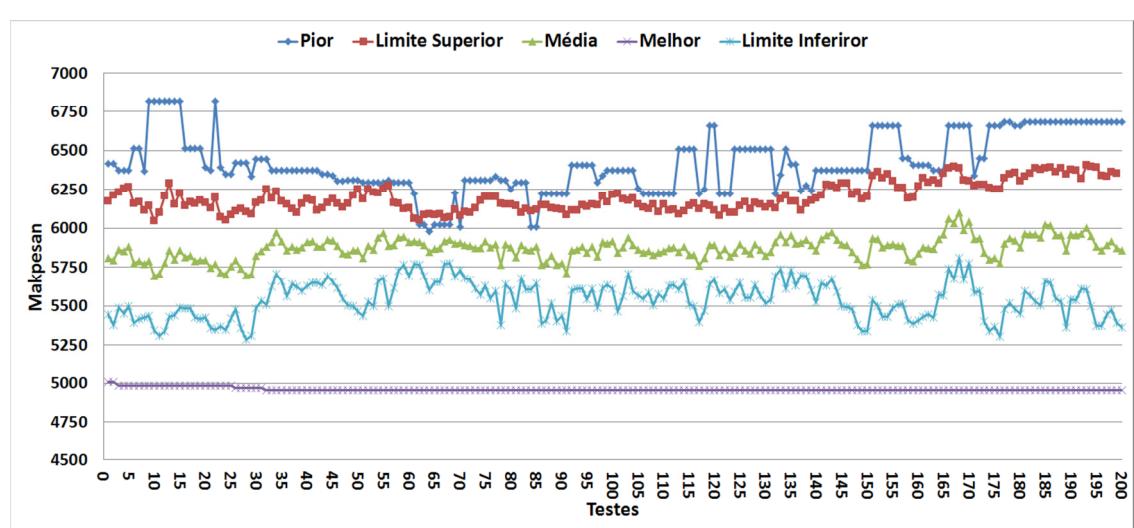
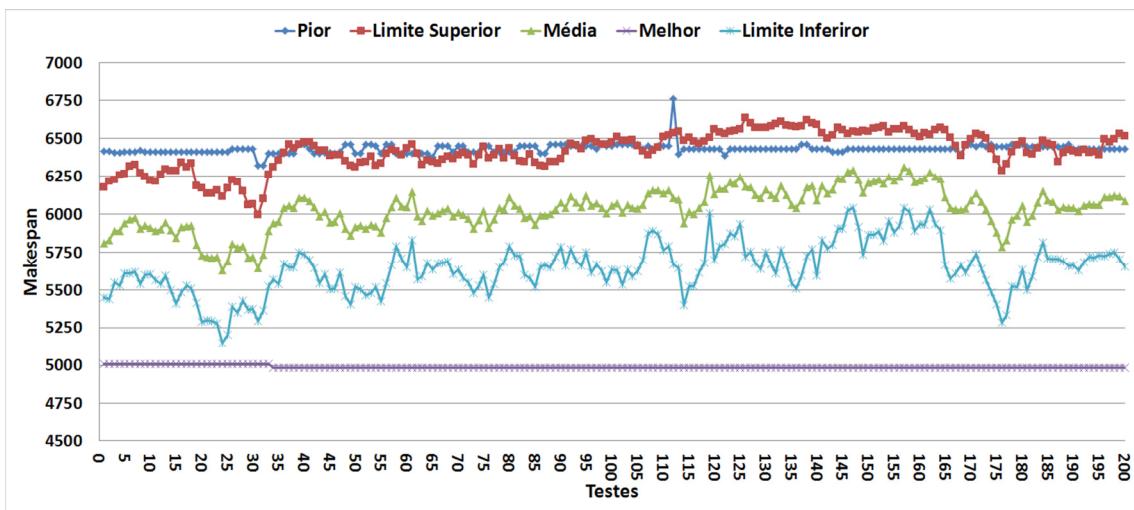


Tabela 8-2 AG_V1 X AG_V2

-	AG_V1		AG_V2	
	Makespan (u.t.)	Tempo (s)	Makespan (u.t.)	Tempo (s)
1	4610	27,0450	4899	22,4540
2	5005	22,0300	4985	25,2610
3	4899	22,6890	4899	26,3980
4	4974	23,4020	4878	23,0150
5	4621	23,1210	4973	24,9380
6	4590	26,5170	4937	21,8530
7	4927	26,1550	4967	25,0350
8	4937	24,0410	4590	22,7920
9	4976	21,5180	4976	19,5610
10	4991	21,8010	4670	20,0590
11	4675	26,0600	4908	18,7720
12	4937	20,3810	4937	24,2860
13	4549	22,6800	4549	26,5580
14	4976	27,3890	4579	28,0090
15	4937	27,4020	4925	28,3450
16	4970	25,4910	4899	26,2030
17	4538	28,1620	4641	24,9950
18	4595	28,3410	4937	24,4570
19	4538	23,0550	4853	27,9350
20	4952	25,7510	4984	18,0320
21	4590	25,5770	4545	28,4650
22	4603	25,8030	4610	27,6580
23	4931	26,9070	4957	26,2370
24	4990	26,8010	4975	26,6840
25	4995	19,1640	5019	24,6470
26	5051	24,2670	4937	26,8960
27	4965	24,0800	4594	25,9930
28	4937	23,8100	4598	27,3260
29	4937	23,6570	5025	21,0620
30	4545	17,3470	4545	20,3660
31	4930	26,1900	4600	26,8560
32	4608	28,0020	4828	27,4510
33	4590	23,3700	4915	24,4790
34	4862	27,8870	4579	27,7140
35	4966	25,8990	4998	22,8060
36	4937	24,4210	4937	23,6140
37	4990	16,9160	4889	23,9450
38	4937	27,9740	4881	26,5250
39	4906	22,8910	5014	22,1400
40	4593	20,9040	4968	22,6150
41	4608	27,9960	4608	22,4760
42	4937	22,9440	4937	27,4070
43	4862	27,4360	4862	25,9350
44	5003	20,7020	4963	26,8380
45	4894	22,5400	4894	27,9280
46	4937	24,1950	4937	26,6370
47	4957	24,7050	5001	26,4640
48	4985	23,5270	4952	23,5510
49	4975	26,2940	4956	24,9680
50	4920	26,0650	4965	25,0190
Média	4842,76	24,4260	4849,50	24,7932