

DENIS DA CRUZ PINHA

**ESCALONAMENTO ÓTIMO BASEADO
NA TEORIA DE CONTROLE SUPERVISÓRIO
APLICADO A UM ESTALEIRO DE REPARO NAVAL**

**FLORIANÓPOLIS
2010**

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO
EM ENGENHARIA DE AUTOMAÇÃO E SISTEMAS

ESCALONAMENTO ÓTIMO BASEADO
NA TEORIA DE CONTROLE SUPERVISÓRIO
APLICADO A UM ESTALEIRO DE REPARO NAVAL

Dissertação submetida à
Universidade Federal de Santa Catarina
como parte dos requisitos para a
obtenção do grau de Mestre em Engenharia de Automação e Sistemas.

DENIS DA CRUZ PINHA

Florianópolis, Julho de 2010.

ESCALONAMENTO ÓTIMO BASEADO NA TEORIA DE CONTROLE SUPERVISÓRIO APLICADO A UM ESTALEIRO DE REPARO NAVAL

Denis da Cruz Pinha

‘Esta Dissertação foi julgada adequada para obtenção do Título de Mestre em Engenharia de Automação e Sistemas, Área de Concentração em Controle, Automação e Sistemas e aprovada em sua forma final pelo Programa de Pós-Graduação em Engenharia de Automação e Sistemas da Universidade Federal de Santa Catarina. ’

Prof. Max Hering de Queiroz, Dr. - DAS/UFSC
Orientador

Prof. José Eduardo Ribeiro Cury, Dr. - DAS/UFSC.
Coordenador do Programa de Pós-graduação em Engenharia de
Automação e Sistemas

Banca Examinadora:

Max Hering de Queiroz, Dr. - DAS/UFSC
Presidente

José Eduardo Ribeiro Cury, Dr. - DAS/UFSC
Co-orientador

Eduardo Camponogara, Dr. - DAS/UFSC

Ricardo Hiroshi Caldeira Takahashi, Dr. – MAT/UFMG

Ricardo José Rabelo, Dr. - DAS/UFSC

*A minha mulher, Carolina, pelo amor, carinho e apoio nessa trajetória.
Aos meus pais, Cesar e Vera, pelo exemplo e amor.
Aos meus irmãos, Andréia, César e Daniel pela eterna amizade e amor.
Ao meu sobrinho, Victor, pelo amor e interesse surpreendente em diversos assuntos.
Ao meu mais novo sobrinho e afilhado, Felipe, que fez um ano e traz consigo uma alegria enorme.*

Agradecimentos

Ao meu orientador Prof. Max Hering de Queiroz, que me recebeu de braços abertos desde o início e com ele certamente pude amadurecer como pesquisador e profissional. Além dos valiosos conhecimentos adquiridos, tive a oportunidade de receber uma orientação com um equilíbrio medido entre liberdade versus responsabilidade, o que me possibilitou alcançar este importante objetivo.

Ao meu co-orientador Prof. José Eduardo Ribeiro Cury, por seus conhecimentos e presença em todos os momentos decisivos de nossa pesquisa. Sua contribuição para este trabalho é inestimável. As nossas reuniões presenciais ou à distância foram sempre para mim um motivo de grande prazer.

Aos professores Eduardo Camponogara, Ricardo Takahashi e Ricardo Rabelo, pela honrosa presença como membros da banca examinadora.

Aos professores Ricardo Costa e Eduardo Jardim, pela oportunidade dada e ainda presente de vivenciar inúmeros trabalhos no Brasil e no exterior. Sem dúvida foram experiências enriquecedoras que contribuíram para o meu crescimento profissional desde 2002 e que me ajudaram a conquistar também este objetivo. Agradeço também pela oportunidade, ao longo de 3 anos, de fazer o mestrado em tempo parcial e poder continuar trabalhando. Foi uma gerência complicada, mas valeu!

Aos colegas de trabalho Leandro Gabriel, Sandro Santos, Leandro Jardim, Igor Peres e Igor Leão da equipe técnica da empresa “Trilha da Inovação – Projetos Tecnológicos e Educacionais” que me ajudaram e souberam dar o real valor ao meu mestrado. Todas as ajudas e ajustes de nossas tarefas diárias me permitiram concluir mais esta meta.

À equipe administrativa da Trilha: Teresa, Heloísa, Sandra e Caroline pela amizade e por sempre nos apoiar.

Aos amigos Andréa Regina e Manoel Saísse do Instituto Nacional de Tecnologia, pela amizade, aprendizado, contribuição e convívio ao longo de quase 8 anos. Certamente minha vida profissional não seria a mesma se não tivesse tido a oportunidade de conhecê-los.

A todos os professores do DAS da UFSC, por suas aulas e conhecimentos transmitidos.

À Nelly Brandt, da secretaria do DAS, pela atenção fora do comum oferecida aos alunos de mestrado e doutorado.

Resumo da Dissertação apresentada à UFSC como parte dos requisitos necessários para a obtenção do grau de Mestre em Automação e Sistemas.

ESCALONAMENTO ÓTIMO BASEADO NA TEORIA DE CONTROLE SUPERVISÓRIO APLICADO A UM ESTALEIRO DE REPARO NAVAL

Denis da Cruz Pinha

Julho, 2010

Orientador: Prof. Max Hering de Queiroz, Dr.

Área de Concentração: Controle.

Palavras-chave: Controle supervisão, sistemas a eventos discretos temporizados, escalonamento da produção com múltiplos objetivos, autômatos temporizados, estaleiro de reparo naval

Número de Páginas: xiv + 98

RESUMO: A Teoria de Controle Supervisório (TCS) permite a síntese automática de supervisores não bloqueantes que habilitem todas e apenas as sequências que satisfaçam especificações de segurança para um sistema a eventos discretos temporizado. O supervisor ótimo que satisfaz as especificações de recursos, roteiros e prazos para problema do tipo *jobshop* contém todas as soluções de escalonamento possíveis. No entanto, o crescimento do número de estados dos modelos pode inviabilizar a solução para problemas reais. Nessa pesquisa, uma nova proposta de modelagem dos autômatos temporizados é desenvolvida com o objetivo de reduzir o tamanho dos modelos. Propõe-se também um algoritmo eficiente para síntese de escalonamento baseada na composição incremental dos roteiros de produção e prazos das tarefas e um método de bissecção para minimização do tempo de produção global e também dos tempos de produção de cada tarefa. Este método é aplicado a um estaleiro de reparo naval para o escalonamento das atividades nos cinco recursos principais para execução de dez obras distintas. Também foi desenvolvido um sistema que integra o planejamento da produção com uma ferramenta de síntese automática de supervisores para que o usuário não precise estar familiarizado com a TCS.

Abstract of Dissertation presented to UFSC as a partial fulfillment of the requirements for
the degree of Master in Automation and System Engineering

OPTIMIZED SCHEDULING BASED ON THE SUPERVISORY CONTROL THEORY APPLIED IN A REPAIR SHIPYARD

Denis da Cruz Pinha

July, 2010

Advisor: Prof. Max Hering de Queiroz, Dr.

Area of Concentration: Control.

Keywords: Supervisory control, Timed Discrete Event Systems, Multiobjective Scheduling, Timed Automata, Repair shipyard

Number of Pages: xiv + 98

ABSTRACT: The Supervisory Control Theory (SCT) allows automatic synthesis of nonblocking supervisors that ensures safety specifications to a timed discrete event system. The optimal supervisory that ensures the resources specifications, production routers, and due dates to the problem of jobshops provides all the possible solutions of scheduling. However, the size of the state space of the models can make impracticable the solution of such a problem. In this dissertation, a new modeling approach is proposed for the timed automata models in order to expressively reduce the size of the models. Also, it is proposed an efficient algorithm for the optimal schedules based on an incremental synthesis of the production routers and due dates. A method of bisection was developed to minimize of total production time and the lead times of jobs as well. This method is applied to a repair shipyard to schedule its activities in the five main resources and ten orders. From the research was developed a system that integrates the production planning with a tool of automatic synthesis of supervisors in order to make the interface an easier place for those users who are not used to SCT.

Sumário

SUMÁRIO.....	IX
LISTA DE FIGURAS.....	XII
CAPÍTULO 1 INTRODUÇÃO	15
1.1 ESCALONAMENTO DA PRODUÇÃO	16
1.2 PROBLEMAS DE ESCALONAMENTO.....	18
1.3 ALGUNS MÉTODOS DE ESCALONAMENTO ENCONTRADOS NA LITERATURA	20
1.3.1 <i>Métodos Construtivos</i>	20
1.3.2 <i>Programação Matemática</i>	21
1.3.3 <i>Programação Dinâmica</i>	22
1.3.4 <i>Regras de Filas</i>	22
1.3.5 <i>Sistemas Especialistas</i>	23
1.3.6 <i>Redes Neurais</i>	24
1.3.7 <i>Algoritmos Genéticos</i>	24
1.3.8 <i>Sistemas Multiagentes (MAS)</i>	24
1.3.9 <i>OPT (Optimized Production Technology)</i>	25
1.3.10 <i>Programação por Capacidade Finita</i>	26
1.4 JUSTIFICATIVA E OBJETIVOS.....	27
1.5 ORGANIZAÇÃO DO DOCUMENTO	29
CAPITULO 2 ESCALONAMENTO ATRAVÉS DO CONTROLE SUPERVISÓRIO TEMPORIZADO	30
2.1 SISTEMAS A EVENTOS DISCRETOS TEMPORIZADOS (SEDT)	30
2.2 CONTROLE SUPERVISÓRIO DE SEDTs.....	36
2.3 APLICAÇÃO DA TCS AO ESCALONAMENTO.....	38
2.3.1 <i>Problema introdutório 5/4/J/F</i>	38
2.3.2 <i>Modelagem da planta</i>	39
2.3.3 <i>Modelagem das especificações</i>	45
2.3.4 <i>Síntese do supervisor</i>	48

CAPÍTULO 3 MÉTODO PROPOSTO PARA O ESCALONAMENTO..... 51

3.1	AUTÔMATOS TTGs DOS RECURSOS.....	52
3.2	AUTÔMATOS TTGs DAS ESPECIFICAÇÕES DOS ROTEIROS DOS PRODUTOS	56
3.3	AUTÔMATOS TTGs DAS ESPECIFICAÇÕES DOS PRAZOS DE ENTREGAS.....	58
3.4	ALGORITMO INCREMENTAL PARA A SÍNTESE	60
3.5	OTIMIZAÇÃO DO SUPERVISOR COM TEMPO MÍNIMO GLOBAL	61
3.6	OTIMIZAÇÃO DO SUPERVISOR PARA TEMPO MÍNIMO POR PRODUTO	63
3.7	ESCOLHA DE UMA CADEIA DE EVENTOS NO SUPERVISOR	64
3.8	ALGORITMO PARA GERAÇÃO DO ESCALONAMENTO.....	65
3.9	RESULTADOS ALCANÇADOS	67
3.9.1	<i>Gráfico de Gantt.....</i>	<i>67</i>
3.9.2	<i>Comparação entre os métodos de síntese</i>	<i>68</i>

CAPÍTULO 4 ESCALONAMENTO DAS ATIVIDADES DE UM ESTALEIRO DE REPARO NAVAL..... 70

4.1	DESCRIÇÃO DO ESTALEIRO	70
4.1.1	<i>Recursos</i>	<i>71</i>
4.1.2	<i>Obras do estaleiro.....</i>	<i>74</i>
4.2	APLICAÇÃO DO MÉTODO AO ESTALEIRO	77
4.2.1	<i>Autômatos dos recursos</i>	<i>77</i>
4.2.2	<i>Autômatos dos roteiros das obras.....</i>	<i>80</i>
4.2.3	<i>Autômatos de prazos para as obras</i>	<i>86</i>
4.2.4	<i>Síntese do supervisor.....</i>	<i>90</i>
4.2.5	<i>Gráfico de Gantt e escalonamento encontrado para o estaleiro.....</i>	<i>92</i>

CAPÍTULO 5 A FERRAMENTA MDC..... 98

5.1	APRESENTAÇÃO DA FERRAMENTA.....	98
5.1.1	<i>Definição dos modelos dos recursos.....</i>	<i>98</i>
5.1.2	<i>Definição dos modelos de prazos de entregas</i>	<i>99</i>
5.1.3	<i>Definição das operações</i>	<i>100</i>
5.1.4	<i>Integração com o software TTCT.....</i>	<i>101</i>
5.1.5	<i>Gráfico de Gantt.....</i>	<i>102</i>

5.1.6	<i>Relatório de sequência dos recursos</i>	103
CAPÍTULO 6	CONCLUSÃO E TRABALHOS FUTUROS	104
REFERÊNCIAS BIBLIOGRÁFICAS	107

Lista de Figuras

Figura 1: <i>ATG R1</i>	32
Figura 2: <i>ATG R2</i>	32
Figura 3: <i>TTG R1</i>	33
Figura 4: <i>TTG R2</i>	33
Figura 5: <i>TTG R1 TTG R2</i>	35
Figura 6: Resultado com a função <i>Comp</i>	35
Figura 7: Autômatos <i>ATGs</i> das máquinas	40
Figura 8: Autômato <i>TTG</i> da máquina <i>M1</i>	42
Figura 9: Autômato <i>TTG</i> da máquina <i>M2</i>	43
Figura 10: Autômato <i>TTG</i> da máquina <i>M3</i>	44
Figura 11: Autômato <i>TTG</i> da máquina <i>M4</i>	44
Figura 12: <i>SA</i> : Autômato <i>TTG</i> da especificação de roteiro do produto <i>A</i>	45
Figura 13: <i>SB</i> : Autômato <i>TTG</i> da especificação de roteiro do produto <i>B</i>	46
Figura 14: <i>SC</i> : Autômato <i>TTG</i> da especificação de roteiro do produto <i>C</i>	46
Figura 15: <i>SD</i> : Autômato <i>TTG</i> da especificação de roteiro do produto <i>D</i>	46
Figura 16: <i>SE</i> : Autômato <i>TTG</i> da especificação de roteiro do produto <i>E</i>	47
Figura 17: <i>STempG(35)</i> : Exemplo do autômato <i>TTG</i> da especificação temporal.....	48
Figura 18: <i>ATG M4</i>	52
Figura 19: <i>TTG M4</i> com problema	53
Figura 20: Autômato <i>TTG</i> da máquina <i>M4</i>	53
Figura 21: Autômato <i>TTG</i> da máquina <i>M1</i>	55
Figura 22: Autômato <i>TTG</i> da máquina <i>M2</i>	55
Figura 23: Autômato <i>TTG</i> da máquina <i>M3</i>	56
Figura 24: <i>SA</i> : Autômato <i>TTG</i> da especificação de roteiro de <i>A</i>	57
Figura 25: <i>SB</i> : Autômato <i>TTG</i> da especificação de roteiro de <i>B</i>	57
Figura 26: <i>SC</i> : Autômato <i>TTG</i> da especificação de roteiro de <i>C</i>	58
Figura 27: <i>SD</i> : Autômato <i>TTG</i> da especificação de roteiro de <i>D</i>	58
Figura 28: <i>SE</i> : Autômato <i>TTG</i> da especificação de roteiro de <i>E</i>	58

Figura 29: <i>STempE</i> : Exemplo do autômato <i>TTG</i> da especificação de prazo de entrega do produto E	59
Figura 30: Gráfico de Gantt.....	68
Figura 31: Layout do estaleiro.....	71
Figura 32: Equipe de Reparo Estrutural	77
Figura 33: Equipe de Controle de Qualidade	78
Figura 34: Equipe de Hidrojato	79
Figura 35: Equipe de Mecânica	79
Figura 36: Equipe de Pintura	80
Figura 37: Obra P1- Navio Barão de Mauá.....	80
Figura 38: Obra P2- Navio Presidente Juscelino.....	81
Figura 39: Obra P3- Navio Supply SC Lancer	82
Figura 40: Obra P4 - Navio Petroleiro Jose do Patrocínio	82
Figura 41: Obra P5 - Navio Petroleiro Lambari	83
Figura 42: Obra P6 - Navio de Pesquisa Scan Nordic.....	83
Figura 43: Obra P7 - Navio Graneleiro Mv Sky L O.S1	84
Figura 44: Obra P8 - Navio Graneleiro Navision Bulker.....	84
Figura 45: Obra P9 - Navio de Pesquisa.....	85
Figura 46: Obra P10 - Navio Supply	85
Figura 47: Prazo para a obra P1 (12 semanas)	87
Figura 48: Prazo para a Obra P2 (8 semanas)	87
Figura 49: Prazo para a Obra P3 (10 semanas)	88
Figura 50: Prazo para a Obra P4 (14 semanas)	88
Figura 51: Prazo para a Obra P5 (8 semanas)	88
Figura 52: Prazo para a Obra P6 (24 semanas)	89
Figura 53: Prazo para a Obra P7 (16 semanas)	89
Figura 54: Prazo para a Obra P8 (20 semanas)	89
Figura 55: Prazo para a Obra P9 (26 semanas)	90
Figura 56: Prazo para a Obra P10 (23 semanas)	90
Figura 57: Escalonamento da fase 1	93
Figura 58: Escalonamento da fase 2.....	94
Figura 59: Escalonamento final do estaleiro	96

Figura 60: Sequência das atividades para a equipe de reparo estrutural	97
Figura 61: Interface de criação dos modelos dos recursos	98
Figura 62: Interface de criação dos modelos das especificações de prazo de entrega	99
Figura 63: Interface de criação dos modelos das especificações dos roteiros	100
Figura 64: Interface de importação do supervisor	102
Figura 65: <i>Algoritmo/Com.TTCT</i>	102
Figura 66: Gráfico de Gantt do sistema MDC	102
Figura 67: Relatório de sequência dos recursos	103

CAPÍTULO 1

Introdução

Segundo pesquisa do Centro de Estudos em Gestão Naval (CEGN) da Universidade de São Paulo (Goldberg, 2007), a indústria de reparos navais representa uma fonte importante de receitas para os estaleiros não só no Brasil, mas em todo o mundo. Em 2006, esse segmento representou 13% do total arrecadado pela indústria naval mundial, e tem se mantido neste percentual desde então. O Brasil precisa se estabelecer neste mercado, pois se essa oportunidade não for aproveitada perderemos divisas potenciais e nossos estaleiros desperdiçarão a chance de operar em uma atividade lucrativa, que representa a porta de entrada e continuidade para o desenvolvimento da indústria da construção naval.

Os setores de reparo naval e *offshore* envolvem estruturas de grande porte, elevado número de componentes, grande flexibilidade de serviços, prazos reduzidos e alto valor financeiro. Estima-se um mercado em grande expansão, principalmente no Brasil. Além disso, o atraso na entrega de um navio causa uma perda de arrecadação muito grande para os armadores (clientes). Com as plataformas de petróleo, esse prejuízo é ainda maior, sem contar com as altas multas contratuais. Isso torna o processo de planejamento complexo e determinante para os ganhos deste setor, que demanda eficácia (requisitos) e eficiência (recursos).

Os pontos chaves para lucratividade de um estaleiro que trabalha com reparo são volume e gestão. O ganho de escala só é viável com o aumento de volume de obras, dada a possibilidade de otimizar a ocupação dos ativos, das equipes de trabalho e da cadeia de fornecedores. A captura de todo o potencial criado pelo volume só é possível com grande habilidade de planejamento, programação (escalonamento das atividades), e controle de obras.

1.1 Escalonamento da produção

A definição de escalonamento é bastante vasta na literatura. Quando associado a um domínio de aplicação, pode-se dizer que o escalonamento é a priorização das atividades que devem ser realizadas em uma sequência conhecida para atender um ou mais objetivos da empresa. Em outras palavras, é a execução coordenada através de procedimentos padrões e decisões de múltiplas tarefas realizadas por diferentes recursos produtivos, de forma a atender às demandas que lhe são requisitadas. Segundo o ANSI (*American National Standard Institute*), escalonamento é “um plano que autoriza a fábrica a manufaturar certa quantidade de itens em um determinado intervalo de tempo” (Tersine, 1985).

Os planos e as correspondentes ações de produção devem ainda respeitar determinadas restrições impostas não apenas pela demanda, mas também pela própria estrutura interna do sistema produtivo. Slack (1995) classifica estas restrições em quatro tipos: i) Restrições de Custo, ii) Restrições de Capacidade, iii) Restrições de Tempo e iv) Restrições de Qualidade. Slack (1995) observa que as fronteiras teóricas e práticas que separam o escalonamento do controle não são bem definidas, porém alguns aspectos estão mais caracteristicamente afetos a um conceito do que ao outro. Assim, o escalonamento estaria mais ligado a uma ideia de intenção e, conseqüentemente, seria a materialização das intenções futuras. Mas também pode ser visto de forma discreta, como uma rede de intenções interligadas, em que, para que uma determinada intenção possa ser realizada, é preciso que outras que a antecedem no tempo (ou que acontecem em paralelo a ela) tenham sido (ou estejam sendo) executadas conforme estava estabelecido previamente no plano. Cabe ao controle então, além de fazer a coordenação dos subsistemas envolvidos, acompanhar a execução das intenções contidas nos planos, corrigir desvios e sinalizar ao planejador a detecção de desvios que justifiquem uma reestruturação do escalonamento original.

A notação utilizada para identificar os problemas de escalonamento é a proposta por French (1982) e se caracteriza por ser a mais difundida entre os pesquisadores da teoria de escalonamento. Este esquema de notação é estruturado a partir de 4 caracteres separados por barras da seguinte forma A/B/C/D. Os quatro caracteres correspondem a :

A - caractere numérico que corresponde ao número de itens envolvidos no problema (n para qualquer número)

B - caractere numérico que corresponde ao número de máquinas envolvidas no problema (n para qualquer número)

C - este caractere descreve o tipo de fluxo de produção do problema e pode assumir os seguintes valores: Problemas de um único estágio de fabricação (Vazio), problema de tipo *jobshop* (J), problema de tipo *flowshop* (F).

D - Descreve o indicador que será avaliado no problema, como exemplo, pode-se citar: Tempo de espera mínimo (W_{min}), tempo médio de espera (W), atraso médio (T), número de itens atrasados (N_t) e tempo de fluxo mínimo (F).

Para o fluxo de produção do tipo *flowshop*, n tarefas devem ser processadas por um conjunto de m máquinas distintas, sendo que todas as tarefas têm fluxos de processamentos iguais nas máquinas. No tipo *jobshop*, n tarefas também devem ser processadas por um conjunto de máquinas distintas, mas ao contrário do tipo *flowshop*, o fluxo de processamento das tarefas nas máquinas é variado, ou seja, cada *job* (tarefa) tem sua ordem ou roteiro de produção.

Tipicamente o fluxo em aplicações de estaleiros navais pode ser caracterizado como *jobshop* e para este tipo há as seguintes observações: (i) as máquinas não são iguais; (ii) os tempos das operações podem ser diferentes para cada *job*; (iii) cada trabalho é escalonado somente uma única vez; (iv) cada trabalho pode ser escalonado para qualquer máquina e em qualquer ordem; (v) cada máquina pode processar somente um trabalho por vez e (vi) nenhuma máquina pode ser liberada antes de finalizada a operação (preempção não permitida).

É importante ressaltar que, a qualidade de uma solução num problema de escalonamento é verificada em cima de indicadores e critérios de desempenho. Estes estão diretamente relacionados com variáveis tais como: atendimento aos prazos dos clientes, a utilização dos equipamentos, tempos de produção, custos de produção e outros. Encontrar um bom critério de otimização, que traduza o objetivo do escalonamento, pode ser uma tarefa difícil, por várias razões. Em primeiro lugar, alguns objetivos podem ser subjetivos e difíceis de serem quantificados, como é o caso de aspectos relacionados à satisfação dos

clientes. Em segundo lugar, muitos objetivos são diferentes por natureza e até conflitantes, isto é, a melhoria de um traduz-se na deterioração de outro. No ambiente de produção *Just in time (JIT)*, por exemplo, terminar tarefas demasiadamente cedo pode representar acúmulo de material nos *buffers* intermediários, ou seja, estoques em processo sem necessidade, o que é altamente desvantajoso. Além disso, o cliente pode não se interessar em receber suas encomendas antes do prazo estipulado. Muitos deles naturalmente, não querem receber suas encomendas com atraso, mas também não gostariam de recebê-las cedo de mais.

Em função da dificuldade de encontrar somente um indicador de desempenho de escalonamento, que represente a realidade e o objetivo de uma corporação, o indicador escolhido para este trabalho é típico para uma aplicação em estaleiro de reparo naval e contém múltiplos objetivos. Geralmente o primeiro objetivo do indicador buscado é entregar todas as tarefas envolvidas nos seus respectivos prazos, o segundo objetivo é fazer isso no menor tempo possível e o terceiro é reduzir os *lead times*¹ (período entre o instante mais cedo possível de início de produção e o instante de final de produção de um *job*) para tarefas prioritárias. Os indicadores escolhidos estão essencialmente voltados para cumprir com os desejos dos clientes. Seguramente indicadores voltados para a empresa também podem ser abordados, tais como indicadores relacionados com a ociosidade dos recursos, estoques em processos e outros. No entanto, a escolha dos três indicadores acima tem foco no ambiente de reparo naval, onde ganhos de eficiência operacional não se justificam frente aos altos valores das multas contratuais provocados por atrasos.

1.2 Problemas de escalonamento

Problemas de escalonamento têm sido extensivamente estudados há algum tempo, começando com o trabalho pioneiro de Johnson (1954), que apresentou um algoritmo para o problema de n tarefas em 2 máquinas. Jackson (1955) e Smith (1956) apresentaram

¹ *Lead time* ou **tempo de atravessamento** é o período entre o início de uma atividade, produtiva ou não, e o seu término. A definição mais convencional para *lead time* é o tempo entre o momento de entrada do material até a sua saída do inventário (Lambert et al., 1998, p. 347, pp. 503-506, pp. 566-576).

regras otimizantes para problemas de máquina única. Desde então, os problemas de sequenciamento passaram a atrair grande atenção de pesquisadores de diferentes áreas, como de pesquisa operacional, ciência da computação e de diversas engenharias. Shah (1992) afirma que "*Pesquisadores propuseram milhares de algoritmos para várias simplificações do problema geral de sequenciamento jobshop. Praticamente todos os fascículos de periódicos tais como Management Science, International Journal of Production Research ou Operations Research contêm ao menos um artigo comentando ou propondo uma heurística ou algoritmo de escalonamento*". A quantidade atual de trabalhos sobre o tema, apresentados em periódicos, ainda é uma constante.

A razão do vasto número de pesquisas e trabalhos sobre o problema de escalonamento advém do fato de que a organização da execução de um conjunto de tarefas, por um grupo finito de recursos, encontra-se presente em quase todo o tipo de atividade produtiva e ao mesmo tempo, sua complexidade matemática é extremamente desafiante. Um problema clássico de escalonamento de tipo *jobshop* pode possuir até $(n!)^m$ soluções (French, 1982), onde n é o número de tarefas e m o número de máquinas. Biegel e Davem (1990) apresentam o seguinte argumento para ilustrar o grau de complexidade dos problemas de escalonamento: a terra tem aproximadamente 20 bilhões de anos ou $6,31152 \times 10^{23}$ microsegundos de idade. Se existisse um supercomputador capaz de avaliar uma solução de escalonamento a cada microsegundo desde o nascimento de nosso planeta, ele só seria capaz de avaliar $6,2 \times 10^{23}$ ou $24!$ soluções, o que corresponde ao universo de soluções do problema de escalonamento de 24 tarefas numa única máquina. Este grau de complexidade, aliado ao fato de que nem todos os problemas da classe NP são redutíveis, faz com que quase todos os problemas de escalonamento de tipo *jobshop*, que têm como objetivo a otimização de medidas de desempenho, sejam classificados como *NP-Hard*, o que indica que não é possível desenvolver algoritmos que possam resolvê-los em tempo polinomial.

Para resolver o problema de escalonamento, inúmeros autores contribuíram com diferentes trabalhos ao longo do tempo, entre eles, podemos citar: **Programação Matemática** com pesquisas de Van De Velde (1991), Appelgate et al (1991), Roshanaei (2010), Teixeira et al (2010), Chen e Hall (2008); **Regras de Fila** com trabalhos de Montazer et al (1990), Dorndorf et al (1993), Grabot (1994) e Huang et al (2010);

Sistemas Especialistas com Fox (1987), que desenvolveu o primeiro sistema especialista (ISIS) voltado especificamente para solução de um problema de escalonamento da produção. Modificações e aperfeiçoamentos no ISIS deram origem à família OPIS de sistemas de escalonamento da produção (Ow et al, 1988) e ao sistema CORTES (Fox et al, 1990). Pesch et al (1996) citam ainda os sistemas SOJA (Lepape,1995), e OPAL (Bensana et al, 1988); **Redes Neurais** através de trabalhos de Cheung (1994), Dagli (1995), Osman et al (1996), Chaudhuri et al (2010); **Algoritmo Genético** com Saïsse (2001), Ishibuchi et al (2003), Kesen et al (2010); **Sistemas Multi-Agentes (SMA)** com Rabelo (1997); Zhou et al (2010); **Optimized Production Technology** de Goldratt (1988); **Sistema de Capacidade Finita**, Costa (1996), Pedroso e Correa (1996) citam mais de 90 sistemas comerciais de planejamento e controle da produção baseados em técnicas de programação por capacidade finita; **SWAEP (Sistemas web de apoio ao escalonamento da produção)**, algumas pesquisas têm oferecido contribuições baseadas em serviços web para a utilização de métodos de escalonamento distribuídos e implementados globalmente por fontes diversas. A proposta deste serviço é disponibilizar os conhecimentos acumulados de forma distribuída, para que cada utilizador (setores industrial e acadêmico) possa usar de forma variável e de acordo com a sua necessidade no momento (Varela, 2007). Na seção 1.3 será descrito, de forma breve, um panorama geral com alguns métodos para obtenção do escalonamento de tarefas.

1.3 Alguns métodos de escalonamento encontrados na literatura

1.3.1 Métodos Construtivos

Os métodos chamados construtivos atingem a solução ótima para o problema seguindo uma série de passos pré-estabelecidos que, ao final do método, determinam a ordem de processamento das tarefas. O primeiro método construtivo para tratar problemas de escalonamento foi proposto por Johnson (1954), que aborda um problema de tipo $n/2/F/F_{max}$. Outros métodos construtivos foram propostos por Akers (1956) para problemas de tipo $2/m/O/J$, e Jackson (1956) para problemas de tipo $n/2/O/J$. A busca de métodos construtivos foi perdendo seu espaço quando muitos problemas simples foram resolvidos

com heurísticas rudes, mas eficientes. Apesar dos progressos alcançados, resultantes de inúmeras tentativas, não foi possível encontrar métodos construtivos eficientes para problemas que englobam simultaneamente mais de 3 máquinas e mais de 3 operações. French (1982) afirma que, para a grande maioria dos problemas de escalonamento, nenhum algoritmo construtivo será encontrado.

1.3.2 Programação Matemática

A programação matemática consiste numa metodologia de formulação na qual o problema é transformado numa função a ser otimizada e cujas variáveis independentes estão sujeitas a um conjunto de restrições. Um tipo de programação matemática que foi muito aplicada a problemas de escalonamento é a chamada programação inteira. Neste método, todas as variáveis independentes da função que se pretende otimizar são confinadas ao conjunto dos números inteiros, sendo que, em geral, este confinamento se restringe aos números 0 e 1, que representam a presença ou ausência de uma determinada propriedade. Quando apenas algumas das variáveis independentes são restritas ao confinamento descrito, a formulação é chamada programação inteira mista. Uma vez formulado desta maneira, o problema pode ser resolvido por métodos enumerativos tais como *Branch and Bound* (Baker, 1974). Giglio e Wagner (1964) resolveram seis problemas de tipo $6/3/O/C_{\max}$ utilizando formulações de programação inteira, porém French (1982) observa que, neste caso, o número de interações necessárias se aproxima do número de soluções possíveis para o problema. Giffler e Thompson (1960) observam que a programação inteira oferece pouca praticidade, enquanto que French (1982) afirma que as formulações resultantes deste método são muito pesadas, mesmo quando aplicadas a computadores rápidos. Os casos mais bem sucedidos que tomam como base de formulação a programação matemática estão associados a Multiplicadores de Lagrange (Fisher, 1973), (Van DeVelde, 1991), e a Métodos de Decomposição (Ashour, 1967), (Appelgate e Cook, 1991) e mais recente, o trabalho de Teixeira et al (2010). Chen e Hall (2008) propõem algoritmos ótimos com o objetivo de maximizar o lucro em função do escalonamento realizado. Para este fim, os autores desenvolvem uma função objetivo a ser maximizada com condições de contorno a serem obedecidas e ao final, mostram os impactos financeiros através de decisões operacionais.

1.3.3 Programação Dinâmica

A programação dinâmica é uma técnica que se inicia com uma busca em profundidade a partir de uma função objetivo, que usa um critério de dominância para introduzir restrições de precedência e outras funções auxiliares para construir soluções. Este método parte o problema de escalonamento em um conjunto de pequenos subproblemas aninhados sendo que a solução de cada problema é obtida em função da solução do problema anterior. Held e Karp (1962) desenvolveram um método de programação dinâmica para problemas de máquina única onde a função objetivo pode ser qualquer combinação linear de funções não decrescentes com o tempo total de processamento. Baker e Schrage's(1978) foram os primeiros a aplicar a programação dinâmica a problemas de escalonamento sujeitos a restrições de precedência, introduzindo a noção do critério de dominância. French (1982) observa que a transposição de problemas envolvendo grande número de restrições de precedência para a abordagem de programação dinâmica é extremamente complexa, pois as restrições se tornam mais difíceis de serem controladas quando o problema é partido em subproblemas menores.

1.3.4 Regras de Filas

Regras de fila foram e ainda são extensamente aplicadas a problemas de escalonamento da produção. Este tipo de metodologia atribui um valor de prioridade a cada uma das tarefas envolvidas no problema e monta a sequência de processamento segundo a ordem crescente ou decrescente destes valores, levando também em consideração as restrições estruturais de capacidade e precedência. Uma das primeiras e mais importantes compilações analíticas sobre regras de fila, onde são apresentadas, analisadas e classificadas 113 diferentes regras, foi publicada por Panwalkar e Iskander (1977). Mais tarde, Blackstone et al.(1982), Haupt(1989) e Bahaskaran e Pinedo(1991) também apresentaram importantes compilações adicionando novas regras.

Nos últimos anos, diferentes regras de fila foram estudadas com o uso de técnicas de simulação computacional (Montazer e Van Wassenhove, 1990). Um trabalho recente com uso desta técnica foi proposto por Huang et al (2010). Estes estudos visam determinar quais regras de fila são mais adequadas para otimizar determinados objetivos de desempenho. Métodos sofisticados baseados em Algoritmos Genéticos (Dorndorf e Pesch,

1993) e lógica *fuzzy* (Grabot e Geneste, 1994) foram utilizados para montar escalonamentos determinando uma regra de fila específica para cada momento de escolha de uma nova tarefa a ser seqüenciada. As regras de fila, tomadas isoladamente, não são suficientes para gerar soluções bastante próximas das melhores possíveis para problemas práticos de escalonamento da produção. Por outro lado, elas podem ser utilizadas em conjunto com outras técnicas e heurísticas sempre que houver necessidade de se decidir, num determinado instante, qual operação, dentre duas ou mais, deverá ser processada por um determinado recurso.

1.3.5 Sistemas Especialistas

A partir da década de 80, uma série de novas técnicas para solução de problemas complexos, denominadas genericamente de Sistemas Especialistas, começaram a ser aplicadas a problemas de escalonamento. Estas técnicas buscam transformar relações complexas, que incluem informações quantitativas e qualitativas, em estruturas de dados organizadas ligadas a poderosos mecanismos de inferência. Os Sistemas Especialistas buscam modelar o problema de escalonamento a partir de uma base de conhecimento. Os modelos de representação mais difundidos são as regras de produção, os quadros (*frames*) e as redes semânticas. A base de conhecimento contém informações sobre as regras de funcionamento do sistema que se pretende estudar. No caso da escalonamento da produção, nesta base deverão estar representadas as restrições, flexibilidades e metodologias de sequenciamento utilizadas no processo produtivo. Aliado à esta base de conhecimento, está o algoritmo de inferência responsável por realizar as pesquisas em busca de uma solução que esteja de acordo com os objetivos estabelecidos pelo usuário.

ISIS (Fox, 1987) foi o primeiro sistema especialista voltado especificamente para solução de um problema de escalonamento da produção. ISIS utiliza uma estratégia baseada em três categorias de restrições: objetivos organizacionais, limitações físicas, restrições causais. Modificações e aperfeiçoamentos no ISIS deram origem à família OPIS de sistemas de escalonamento da produção (Ow e Smith, 1988) e ao sistema CORTES (Fox e Sycara, 1990). Pesch e Tetzlaff (1996) citam ainda os sistemas SOJA (Lepape, 1995), e OPAL (Bensana et al, 1988) como exemplos de técnicas de *constraint satisfaction* aplicadas a problemas de programação da produção, porém estes mesmos

autores indicam que estes sistemas não constroem o programa de produção propriamente dito, limitando-se a dar informações de alto nível de agregação para o planejador.

1.3.6 Redes Neurais

Cheung (1994) apresenta alguns dos principais modelos de redes neurais aplicados a problemas de escalonamento. O método de redes neurais proposto por Hopfield (Hopfield and Tank, 1985), na qual o problema é usualmente representado através de um modelo de programação linear mista, foi utilizado por Zhou (1990) para resolver problemas envolvendo 4 operações e 3 máquinas e 10 operações e 10 máquinas. Dagli (1991) combina redes neurais com algoritmos genéticos. Osman e Kelly (1996) afirmam que as redes neurais não são competitivas quando comparadas com as melhores heurísticas para as principais classes de problemas de otimização. Chaudhuri et al (2010) apresentam um trabalho com a combinação de lógica *fuzzy* com redes neurais.

1.3.7 Algoritmos Genéticos

No início da década de 70, John Holland (1975), inspirado no mecanismo da seleção natural, propôs uma técnica denominada algoritmo genético para resolver problemas de otimização combinatória que envolve espaços de busca complexos. Na técnica desenvolvida por Holland, um grupo de possíveis soluções para o problema a ser tratado é escolhido para formar a *população inicial*. Cada solução membro desta população é denominada um *indivíduo* e é caracterizada por um valor de *função objetivo*.

Desde as proposições iniciais de Holland esta técnica tem se desenvolvido enormemente, caracterizando-se numa ferramenta robusta para resolver problemas práticos tão distintos. Inúmeros autores, como Davis (1991), Saísse (2001) e Kesen et al (2010) trabalharam especificamente com algoritmos genéticos para encontrar soluções em problemas de escalonamento de tarefas.

1.3.8 Sistemas Multiagentes (MAS)

O conceito básico de uma abordagem MAS está relacionado na resolução de problemas de forma distribuída, onde é adotada a idéia de “dividir para conquistar”. Desta maneira o problema é dividido em subproblemas e cada um é executado separadamente

por um agente, cada um desses comunicando ou cooperando entre si. Ao final, a soma dos resultados locais corresponde à solução do problema geral.

A definição de Agente permanece ainda sem um consenso claro na comunidade de Inteligência Artificial Distribuída, mas segundo Rabelo (1997), que fez um resumo das definições existentes na literatura, um agente é: “um sistema computacional que habita um ambiente com alguma dinâmica e complexidade, sente e age autonomamente neste ambiente e, desta forma, é capaz de atingir, executar um conjunto de tarefas para o qual foi desenvolvido.”

Segundo Rabelo (1997), a proposta de resolver o escalonamento com uma abordagem Multiagente se deve ao fato de que o problema de escalonamento é: intrinsecamente distribuído, requer uma junção de diferentes domínios de conhecimento, requer aplicação de diferentes sistemas e que sejam integrados num mesmo ambiente, além disso, inclui diferentes níveis de autonomia nas decisões, é dinâmico e por fim, é extremamente conflituoso, o que requer variados níveis de cooperação e negociação.

1.3.9 OPT (*Optimized Production Technology*)

O OPT ou *Optimized Production Technology* é uma abordagem de gerência de atividades produtivas desenvolvida originalmente por Eliyahu Goldratt, que dá grande ênfase à forma como o processo produtivo é gerenciado (Goldratt, 1988). Em geral, esta abordagem é aplicada com base num sistema computacional fechado que se utiliza de algoritmos de simulação e métodos heurísticos para balancear o fluxo de produção em função dos “gargalos”. Apesar da característica fechada do sistema, que não permite acesso à estrutura dos algoritmos e heurísticas utilizados, os princípios gerais que regem seu projeto são amplamente divulgados e vêm sendo aplicados em diferentes ambientes de produção com aparente sucesso.

Uma descrição geral do funcionamento do sistema OPT é apresentada por Meleton (1986). O OPT é baseado numa arquitetura de banco de dados denominada de BUILDNET que organiza todas as redes de informações necessárias para o controle do fluxo de produção. Uma vez de posse das informações, um módulo denominado SERVE realiza uma avaliação de carga calculando para cada um dos recursos um índice de utilização

média. Os recursos são então seqüenciados em ordem decrescente de utilização. Quando os gargalos são identificados, um módulo denominado SPLIT entra em ação classificando os recursos em críticos e não críticos. Os recursos críticos são então programados a partir de um procedimento de programação para frente com capacidade finita. Este procedimento, que é a parte mais secreta e importante do sistema, também determina o tamanho dos lotes de transferência e de processamento para cada operação. Por fim, o módulo SERVE programa os recursos não críticos de forma a que eles possam "servir" adequadamente aos recursos críticos.

1.3.10 Programação por Capacidade Finita

O método de Programação por Capacidade Finita tem como objetivo construir um programa detalhado de execução das operações a serem executadas que respeite uma disponibilidade previamente estabelecida de capacidade de produção. Durante o carregamento das operações nos recursos, as relações precedência entre as operações devem ser consideradas. Quando duas ou mais operações competem por um mesmo recurso num mesmo instante, deve-se estabelecer uma regra de prioridade para determinar qual deverá ser a ordem de carregamento. A Programação por Capacidade Finita aparece freqüentemente associada (e muitas vezes até confundida) com técnicas de simulação discreta (Roy e Meikle,1995).

Os modernos sistemas de programação e controle da produção baseados em técnicas de escalonamento de capacidade finita utilizam um algoritmo que simula a ocorrência das ações operacionais necessárias à fabricação de um determinado conjunto de produtos. Durante a simulação, o algoritmo consulta um banco de dados central que contém informações acerca das características funcionais e disponibilidades temporais dos recursos disponíveis bem como dados relevantes dos produtos cuja fabricação será simulada.

Atualmente, é possível armazenar os dados mais relevantes acerca do processo produtivo de uma fábrica de médio porte num único microcomputador. O desenvolvimento dos processadores permitiu a construção de algoritmos, tais como os propostos por Costa (1996), capazes de simular o funcionamento de uma fábrica real de produção, sob

encomenda, com uma carteira de pedidos composta por diferentes equipamentos razoavelmente complexos, em intervalos de tempo da ordem de segundos.

Desde o seu surgimento, vários sistemas de planejamento e controle da produção baseados em simuladores determinísticos com programação por capacidade finita têm sido lançados comercialmente. Numa pesquisa conduzida em indústrias na Grã-Bretanha, Little e Jarvis (1993) observaram que muitos sistemas de programação por capacidade finita são usados em conjunto com sistemas do tipo MRP/MRP II para auxiliar na programação e escalonamentos diários da produção. Pedrosa e Correa (1996) citam mais de 90 sistemas comerciais de planejamento e controle da produção baseados em técnicas de programação por capacidade finita, disponíveis em vários países.

1.4 Justificativa e Objetivos

Em vista da grande flexibilidade do ambiente produtivo (entrada de novos recursos, novas restrições, novos produtos, quebras de máquinas, etc) frente à especificidade das modelagens características das técnicas descritas na seção 1.2, as alterações na maioria dos casos requerem ajustes de programação e devem ser feitas por especialistas, o que tornam estas técnicas inflexíveis do ponto de vista do usuário. Além disso, elas tomam tempos de processamento relativamente altos e em muitos casos apresentam objetivos que não tem fidelidade com o ambiente produtivo, por isso elas se tornam de difícil aplicação em sistemas reais.

Por outro lado, a Teoria de Controle Supervisório (TCS) oferece um método formal baseado na teoria de autômatos e linguagens para gerar um agente de controle chamado supervisor, que coordena o sistema de acordo com regras de controle (Wonham, 2009). Este método determina o controle menos restritivo possível para a planta, elemento a ser controlado, pois permite a ocorrência de todos os eventos que não se oponham ao comportamento especificado. Brandin e Wonham (1992) desenvolvem uma abordagem para o controle de sistemas a eventos discretos temporizados (SEDT) e aplicam a teoria para resolver um problema de controle supervisório em uma célula de manufatura com 2 máquinas e 2 produtos. Essa aplicação sugere uma abordagem para a síntese automática de um supervisor que habilite todas e apenas as sequências controláveis de um sistema a

eventos discretos temporizado que satisfaçam as especificações de escalonamento no menor tempo possível.

Assim a TCS oferece uma estrutura flexível, através da modelagem de autômatos e da síntese dos supervisores para construir os modelos da planta e de suas restrições. Todas as mudanças necessárias, presentes num sistema de produção, podem ser feitas somente a partir da composição de novos modelos, não havendo necessidade de alterações em códigos computacionais.

No entanto, a síntese de supervisores para SEDT é limitada pela explosão do número dos estados dos modelos, decorrente da incorporação do evento do tempo na estrutura de transição e da composição dos subsistemas. Saadatpoor (2004) propõe tratar este problema pelo uso de uma estrutura eficiente de representação baseada em BDD (Binary Decision Diagram). Um trabalho desenvolvido por Park e Yang (2009) aplica o controle supervísório temporizado para gerar um escalonamento em tempo real de tarefas periódicas e esporádicas a fim de cumprir os seus respectivos *deadlines*.

A proposta desta pesquisa é reduzir o custo computacional e o tamanho dos supervisores a partir de modificações nos modelos propostos por Brandin e Wonham (1992) e do desenvolvimento de algoritmos, com objetivo de gerar um escalonamento. Em função da relevância do escalonamento para os estaleiros de reparos, aliado ao fato de não haver ferramentas aderentes e em uso para apoiar essa função neste setor, esta pesquisa explora a flexibilidade de modelagem da TCS e propõe uma metodologia para preencher esse hiato. Neste sentido a pesquisa tem três objetivos específicos distintos:

1. Propor uma metodologia de síntese de escalonamento a partir da modificação dos modelos de Brandin e Wonham (1992) e da utilização de algoritmos incrementais.
2. Desenvolver uma ferramenta de escalonamento, que tem objetivo de ser a interface entre a TCS e o PCP (Planejamento e Controle da Produção). Esse desenvolvimento se torna indispensável, pois os profissionais do PCP têm seus padrões de trabalho e de avaliação dos resultados e, não estão familiarizados com autômatos.
3. Aplicar a metodologia a um estaleiro de reparo naval.

1.5 Organização do documento

Esta dissertação está estruturada em seis capítulos. Neste capítulo inicial foram apresentados: o contexto geral do trabalho, a proposta e objetivos da pesquisa. Os demais capítulos encontram-se assim:

O capítulo 2 apresenta um resumo dos conceitos fundamentais da teoria de controle supervisorio temporizado na TCS e um problema 5/4/J/F para demonstrar os passos da síntese do supervisor propostos por Brandin e Wonham (1992).

No capítulo 3 é apresentado o método desta pesquisa para resolver o problema de escalonamento com a TCS. São apresentados os algoritmos de síntese e de geração do escalonamento, a modificação dos modelos dos autômatos para fins de escalonamento e os resultados alcançados com esta metodologia.

No capítulo 4 é apresentado o problema do escalonamento de um estaleiro de reparo naval: seu layout, suas obras, suas operações e recursos relevantes. São apresentados a modelagem e o resultado desta aplicação com a metodologia proposta.

O capítulo 5 apresenta a ferramenta MDC com suas principais interfaces e funções.

O capítulo 6 apresenta a conclusão deste trabalho e lista sugestões para trabalhos futuros.

Capítulo 2

Escalonamento através do controle supervisório temporizado

Os sistemas a eventos discretos (SED) são sistemas onde sua relação e interação com o mundo externo é feita através de estímulos denominados eventos. A ocorrência de um evento, em geral, provoca mudanças instantâneas no sistema que altera o seu estado corrente. Segundo Cury (2001), um SED “*é um sistema dinâmico que evolui de acordo com a ocorrência abrupta de eventos físicos, em intervalos de tempos em geral irregulares e desconhecidos*”.

No modelo temporizado (SEDT), além de considerar a ocorrência dos eventos e as mudanças de estados, é importante saber quando o sistema entrará em um estado ou quanto tempo permanece em um. Para isso, é introduzida a dimensão do tempo, que traz consigo um aumento de complexidade, mas por outro lado, expande a teoria para problemas de grande interesse. Diante desta adição, uma proposta de modelagem e controle para SEDT foi desenvolvida por Brandin e Wonham (1992) e nesse capítulo, será apresentado de forma resumida como o trabalho destes autores pode ser aplicado ao problema de escalonamento. Uma apresentação mais detalhada é feita por Wonham (2009).

2.1 Sistemas a eventos discretos temporizados (SEDT)

Na TCS, considera-se a *planta*, como um conjunto de subsistemas que podem ser equipamentos como: máquinas, robôs, ferramentas, equipes de trabalho, etc. Estes subsistemas isolados têm um comportamento básico original que deve ser restringido, de forma que, quando operando conjuntamente, o sistema respeite um comportamento global especificado.

A planta pode ser obtida com a composição dos comportamentos de cada subsistema isolado e é modelada por uma estrutura formal denominada autômato. O autômato é uma quintupla de forma $G = (\Sigma, Q, \delta, q_0, Q_m)$. Σ representa o conjunto de eventos, Q o conjunto

de estados, $q_0 \subseteq Q$ o estado inicial, $Q_m \subseteq Q$ o subconjunto de estados marcados e $\delta: \Sigma \times Q \rightarrow Q$ a função de transição parcial definida em cada estado de Q para um subconjunto de Σ . Σ^* é o conjunto de todas as cadeias finitas formadas por elementos de Σ , inclusive a cadeia vazia ε . A linguagem da planta denominada $L(G)$ representa todas as sequências possíveis de ocorrer a partir do estado inicial de G , enquanto a linguagem $L_m(G)$ representa as sequências que alcançam um estado marcado a partir do estado inicial.

Os autômatos podem ser graficamente visualizados por um diagrama de transição de estado, que é um grafo direcionado. Os vértices simbolizam os estados e os arcos estão associados com as respectivas funções de transições parciais, relacionados com a ocorrência dos eventos. Uma flecha aponta para o estado inicial e os estados marcados são desenhados como círculos duplos. Os eventos controláveis possuem um traço sobre os arcos de transição.

Para representar a evolução paralela de autômatos, recorre-se ao produto síncrono dos mesmos. No produto síncrono, os eventos comuns aos autômatos somente ocorrem simultaneamente. Se na composição não houver nenhum evento comum aos autômatos, diz-se que o produto é assíncrono.

Sejam os autômatos $A_1 = (\Sigma_1, Q_1, \delta_1, q_{01}, Q_{m1})$, $A_2 = (\Sigma_2, Q_2, \delta_2, q_{02}, Q_{m2})$, $\Sigma_1(q_1)$ é o conjunto de eventos ativos (conjunto de eventos habilitados) no estado q_1 do autômato A_1 , $\Sigma_2(q_2)$ é o conjunto de eventos ativos no estado q_2 do autômato A_2 e $Ac(G)$ o autômato resultante da evolução paralela entre eles. A composição síncrona destes autômatos, simbolizada por $A_1 \parallel A_2$ é definida da seguinte maneira (Wonham, 2009):

$$A_1 \parallel A_2 = Ac(\Sigma_1 \cup \Sigma_2, Q_1 \times Q_2, \delta_{1\parallel 2}, (q_{01}, q_{02}), Q_{m1} \times Q_{m2})$$

sendo:

$$\delta_{1\parallel 2}((q_1, q_2), \sigma) = \begin{cases} (\delta_1(q_1, \sigma), \delta_2(q_2, \sigma)) & \text{se } \sigma \in \Sigma_1 \cap \Sigma_2 \text{ e } \sigma \in \Sigma_1 \text{ e } \sigma \in \Sigma_1(q_1) \cup \Sigma_2(q_2) \\ (\delta_1(q_1, \sigma), q_2) & \text{se } \sigma \in \Sigma_1 \text{ e } \sigma \notin \Sigma_2 \text{ e } \sigma \in \Sigma_1(q_1) \\ (q_1, \delta_2(q_2, \sigma)) & \text{se } \sigma \in \Sigma_2 \text{ e } \sigma \notin \Sigma_1 \text{ e } \sigma \in \Sigma_2(q_2) \\ \text{indefinida} & \text{caso contrário} \end{cases}$$

A composição síncrona é uma operação comutativa e associativa (Cassandras e Lafortune, 1999) e quando se deseja compor vários autômatos a operação é feita repetitivamente aos pares.

A representação do tempo em SEDTs proposta por Brandin e Wonham (1992) é feita a partir da introdução de um evento denominado *tick*. Como já escrito, eventos ocorrem

instantaneamente e de periodicidade irregular, entretanto, quando da ocorrência do evento *tick* é convencionada uma medida de duração, que guarda relação com o relógio global. A contagem do tempo (t) pode ser descrita como $\{(t) = n \mid n \leq t < n+1, n \in \mathcal{N}\}$. A escolha do valor do *tick*, ou seja, sua duração, é uma decisão de modelagem. Por exemplo, em um sistema bancário, o tempo de atendimento mínimo de um cliente em um caixa toma cinco minutos, já para uma companhia de navegação, que transporta cargas para diferentes continentes, seu tempo de transporte ente as cidades de origem e destino são no mínimo de três semanas. Para o modelo do sistema bancário o valor do *tick* poderia assumir valor igual a cinco minutos, enquanto para a empresa de navegação seria de três semanas. Estes valores devem representar o máximo divisor comum (mdc) entre os tempos das operações do sistema, com o valor do tick estipulado, a contagem do tempo é sempre múltipla deste valor.

Na abordagem de Brandin e Wonham (1992), dois tipos de estruturas foram desenvolvidos para um SEDT: *i*) ATGs (*Activity Transition Graphs*), que são estruturas modeladas por autômato onde cada evento σ do conjunto Σ_{act} do ATG (Σ_{act} representa o conjunto de eventos do ATG) é adicionado de informações relativas ao tempo, um *lower time bound* (l_σ) e um *upper time bound* (u_σ). Esses limites correspondem aos tempos mínimos e máximos de ocorrência do evento σ , uma vez que ele esteja habilitado no autômato; *ii*) a segunda, os TTGs (*Timed Transition Graphs*), são estruturas de transição também representadas por autômatos, porém neste caso introduz-se a noção do tempo explicitamente na estrutura de transição do autômato através da adição do evento *tick*, que modela o avanço de uma unidade de tempo. A seguir, nas figuras 1 e 2 são apresentados dois exemplos de autômatos ATGs.

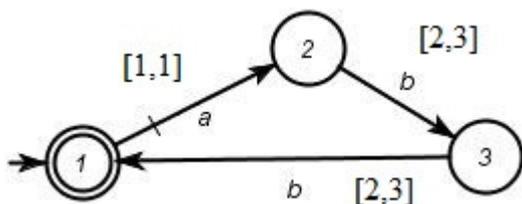


Figura 1: ATG R1

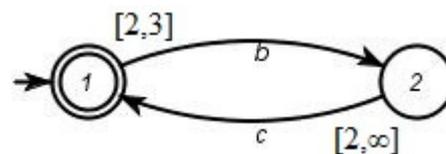


Figura 2: ATG R2

Para cada evento de um ATG está associado um intervalo de tempo [*lower time bound* (l_σ), *upper time bound* (u_σ)]. Quando *upper time* tem valor igual a ∞ (infinito), por

convenção, significa que este evento pode ocorrer em qualquer instante ou provavelmente nunca. No caso dos autômatos das figuras 1 e 2, o intervalo associado ao evento a é $[1,1]$, para o b é $[2,3]$ e para o evento c é $[2, \infty]$. Para o evento a , a interpretação do seu intervalo de tempo diz que em exatamente uma unidade de tempo (uma ocorrência do evento *tick*) a partir do estado inicial, o evento a está habilitado, alcançou seu *deadline* e por isso, sua ocorrência é obrigatória antes de qualquer outro evento do relógio (*tick*). Para o evento b , a interpretação do intervalo $[2,3]$ representa a necessidade de no mínimo duas ocorrências do evento *tick* após a ocorrência do evento a , para que b esteja habilitado e no máximo três ocorrências do evento *tick*, após a ocorrência do evento a , para que o evento b ocorra de forma mandatória antes de qualquer outro evento do relógio (*tick*), *deadline* alcançado. Por último, a interpretação do intervalo $[2, \infty]$ de c , mostra que esse evento está habilitado em no mínimo duas unidades de tempo após a ocorrência de b ou em qualquer instante depois desse tempo. Nas figuras 3 e 4 são mostrados os autômatos *TTGs* correspondentes.

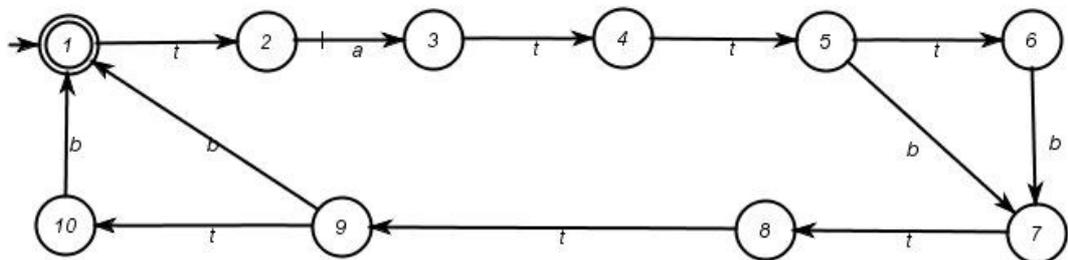


Figura 3: *TTG R1*

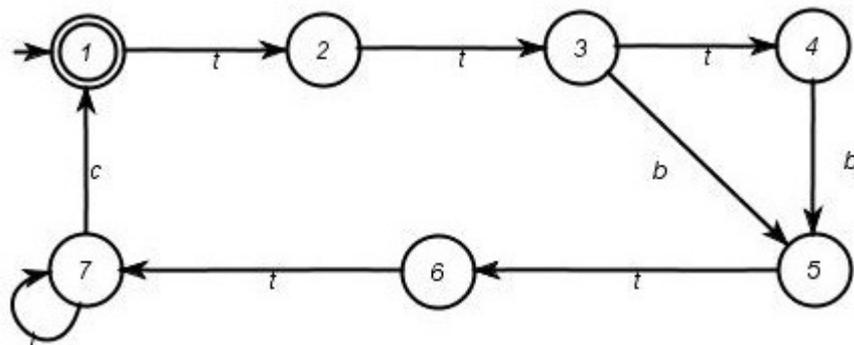


Figura 4: *TTG R2*

O Autômato *TTG R1* da figura 3 é a tradução temporizada do autômato *ATG R1* da figura 1, a mesma coisa acontece para o *TTG R2* (Figura 4) e o *ATG R2* (Figura 2). Como

pode ser visto nas figuras 3 e 4, a estrutura lógica (sequencial) da ocorrência dos eventos é mantida com seus respectivos *ATGs*, mas é introduzido o evento *tick*, representado pela letra *t* na estrutura de transição dos autômatos *TTGs*.

Para a obtenção do modelo da planta, das especificações e para realizar a síntese dos supervisores em SEDTs para esta pesquisa é usado o software TTCT (Wonham, 2009). Este sistema foi desenvolvido na universidade de Toronto e estende as funcionalidades encontradas no sistema TCT desenvolvido para sistemas a eventos discretos não temporizados para temporizados. A extensão da teoria de Ramadge e Wonham (1989) foi proposta por Brandin e Wonham (1992) e primeiramente implementada por O'Young (1992).

Para realizar a composição do modelo da planta *G* em SEDTs é usada a função *Comp* do TTCT (Wonham, 2009). Esta função representa o comportamento do produto síncrono dos autômatos *ATGs* $A1_{act}$ e $A2_{act}$ com a composição dos intervalos de tempo $[l_{\sigma}, u_{\sigma}]$ de cada evento dos autômatos. O conjunto de eventos de $A1_{act}$ é representado por Σ_{A1} e o conjunto de eventos de $A2_{act}$ é representado por Σ_{A2} . Cada evento $\sigma \notin (\Sigma_{A1} \cap \Sigma_{A2})$, o intervalo $[l_{\sigma}, u_{\sigma}]$ permanece sem alteração, enquanto que se $\sigma \in (\Sigma_{A1} \cap \Sigma_{A2})$, o intervalo é definido como $[l_{\sigma}, u_{\sigma}] = [\max(l_{A1,\sigma}, l_{A2,\sigma}), \min(u_{A1,\sigma}, u_{A2,\sigma})]$.

Em Brandin e Wonham (1992), para obter o modelo da planta é necessário construir primeiramente os modelos dos autômatos *ATG* de cada recurso envolvido e então, executar a função *Comp* do TTCT com estes autômatos. Esta função garante o comportamento real do sistema, mesmo quando há eventos comuns entre diferentes *ATGs*. Nessa situação, quando há eventos em comum entre os *ATGs*, não seria possível obter o *TTG* global (*Planta*) a partir do *TTG* de cada subsistema, pois a sincronização do evento *tick* pode gerar inconsistências no modelo resultante. Com o objetivo de esclarecer o problema do produto síncrono com autômatos *TTGs* para obtenção do modelo da planta, será mostrada a figura 5 que contém o resultado de $TTG R1 \parallel TTG R2$.

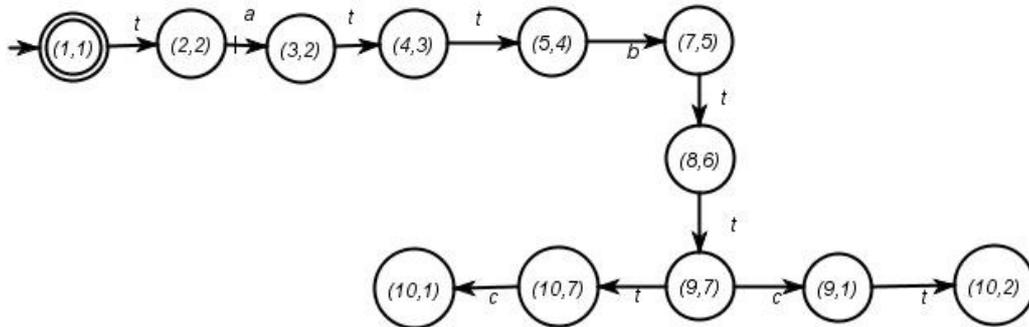


Figura 5: *TTG R1 || TTG R2*

O resultado obtido não é realístico, pois além do evento tick (t), o evento b é um elemento em comum aos dois autômatos das figuras 3 e 4. Como exemplo, na planta *TTG R1 || TTG R2* consideremos a ocorrência da cadeia de eventos *tattbtcc*, com esta sequência é alcançado o estado de número 10 no *TTG R1* e o estado de número 1 no *TTG2 R2*. Nesta situação é constatado que o evento *b* e também o evento *tick* não podem mais ocorrer de forma síncrona, o estado (10,1) do produto síncrono estaria com bloqueio (*deadlock*). Este resultado não representa a realidade do sistema, o resultado correto, que mostra o comportamento real é apresentado na figura 6.

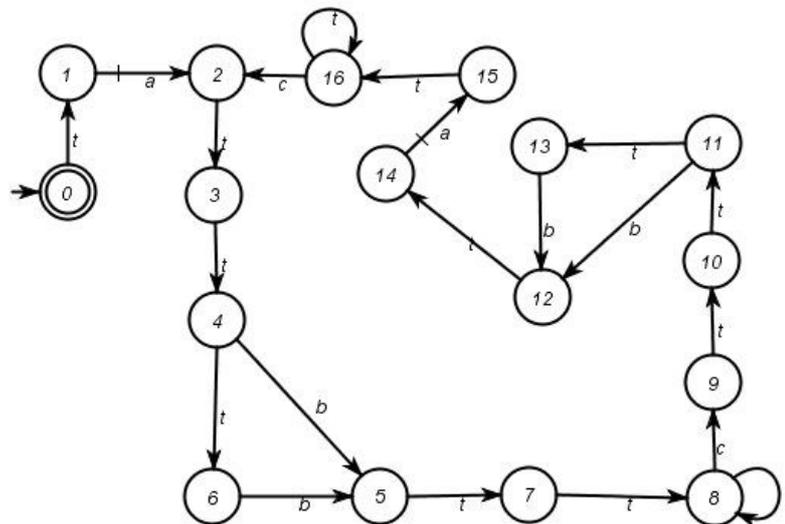


Figura 6: Resultado com a função *Comp*

Como demonstrado, o produto síncrono dos autômatos *TTG* da planta com eventos em comum pode acarretar em um *deadlock* e um travamento do relógio, o que representa um comportamento não real do sistema.

2.2 Controle supervisorio de SEDTs

Para o controle supervisorio, a *planta* representa o sistema a ser controlado e as especificações são restrições impostas na planta, de forma a impedir algumas sequências de eventos indesejadas. O papel do supervisor, quando operado conjuntamente com a planta, é garantir que o comportamento especificado seja respeitado.

Para fins de controle, Brandin e Wonham (1992) classificam os eventos como não-controláveis e controláveis. Os eventos não controláveis são eventos que não podem ser desabilitados, por exemplo, eventos de quebra de máquina e o evento do relógio (*tick*). Já os eventos controláveis são divididos em duas categorias: uma se refere aos eventos que podem ser desabilitados, mas não forçados por um supervisor externo (proibitivos) e a segunda categoria representa o conjunto de eventos que podem ser desabilitados ou forçados por um supervisor externo (forçáveis). É entendido que forçar um evento implica sua habilitação. Por definição, o conjunto de eventos controláveis (Σ_c) representa a união do conjunto dos eventos proibitivos (Σ_p) com os forçáveis (Σ_f), $\Sigma_c = \Sigma_f \cup \Sigma_p$ e o Σ_u , representa o conjunto de eventos não controláveis. O conjunto de todos os eventos para esta abordagem é então definido como $\Sigma = \Sigma_c \cup \Sigma_u \cup \{tick\}$.

As especificações são modeladas por um conjunto de autômatos que determinam como o sistema deve operar. Uma especificação global K é encontrada através da composição das especificações impostas no sistema. Um supervisor S é uma função que observa os eventos da planta e define os eventos que devem ser forçados ou proibidos, de modo que o comportamento em malha fechada S/G atenda uma especificação global K . Diz-se que o supervisor S é não bloqueante se ao partir de qualquer estado acessível de S/G for possível alcançar um estado marcado.

A condição para a existência de um supervisor não bloqueante S que, atuando sobre a planta G atenda de forma exata ao conjunto de especificações K ($L_m(S/G)=K$), está diretamente associada à noção de controlabilidade de linguagens. Na abordagem de Brandin e Wonham (1992) a noção de controlabilidade para SEDTs se faz como a seguir.

Seja $K \subseteq L(G)$ uma linguagem e $s \in \bar{K}$ uma cadeia pertencente ao prefixo-fechamento de K ². K é dita ser controlável em relação a $L(G)$ se, i) a possibilidade de ocorrência de um evento não-controlável μ em $L(G)$, após uma cadeia s , seja tal que mantenha o comportamento de G na especificação K , ou seja $s\mu \in \bar{K}$, e, ii) a ocorrência de um evento *tick* que resulte fora da especificação K possa ser *preemptado* pela ocorrência de eventos forçáveis dentro de K . A noção de controlabilidade para SEDT é estendida e definida como: Seja s uma cadeia $\in \bar{K}$, $A_g(s) = \{\sigma \in \Sigma \mid s\sigma \in L(G)\}$ e $A_k(s) = \{\sigma \in \Sigma \mid s\sigma \in \bar{K}\}$, K é dita ser controlável em relação a $L(G)$ se:

$$(\forall s \in \bar{K}) (\Sigma_u \cap A_g(s) \subseteq A_k(s) \wedge [tick \notin A_g(s) - A_k(s) \vee \Sigma_f \cap A_k(s) \neq \emptyset]).$$

Expresso em palavras, $A_g(s)$ representa o conjunto de eventos possíveis da cadeia σ que são fisicamente possíveis de ocorrer na planta após a cadeia s e $A_k(s)$ tem uma interpretação similar para a linguagem definida em K . A proposta de controlabilidade para os SEDT suporta então dois mecanismos de controle: a prevenção de cadeias ilegais a partir da desabilitação de eventos e da utilização de eventos forçáveis.

Nem sempre uma especificação K atende a condição de controlabilidade. Nesse caso, Brandin e Wonham (1992) demonstram a existência de uma máxima sublinguagem controlável de K para SEDT. Esta linguagem é dada por $SupC(K, G)$. O supervisor S tal que $L_m(S/G) = SupC(K, G)$ representa a lógica ótima de supervisão na qual o comportamento da planta é minimamente restringido para atender a especificação global K .

A complexidade computacional do cálculo do $SupC(K, G)$ é polinomial no produto de estados de G e K . Isto significa que o tempo de processamento, sendo proporcional ao número de operações, não é exponencial. No entanto, o número de estados G ou K cresce exponencialmente com o aumento do número de subsistemas ou restrições. Na próxima seção será apresentado o método de síntese proposto pelos autores.

² Prefixo-Fechamento: Seja uma linguagem K , então o prefixo-fechamento de K , denotado por \bar{K} , é definido por: $\bar{K} = \{s \in \Sigma^* : \exists t \in \Sigma^* (st \in K)\}$.

Em palavras, \bar{K} consiste de todas as cadeias de Σ^* que são prefixos de K . Em geral, $K \subseteq \bar{K}$. K é dita prefixo-fechada se $K = \bar{K}$. Uma linguagem K é prefixo-fechada se qualquer prefixo de qualquer cadeia de K é também uma cadeia de K (Cury, 2001).

2.3 Aplicação da TCS ao escalonamento

Apesar da síntese de Brandin e Wonham (1992) não ter o objetivo de gerar um escalonamento, o autômato do supervisor encontrado contém várias cadeias de eventos e, ao escolher uma delas a partir do estado inicial até um estado marcado é possível obter um escalonamento. Os autores apresentam uma proposta de síntese de um supervisor para uma célula de manufatura com dois produtos, com o objetivo de controlar a ordem de produção deles nas máquinas e o tempo máximo de execução de todas as tarefas na célula.

A seguir, ilustra-se a aplicação da metodologia empregada por Brandin e Wonham (1992) através de um estudo de caso introdutório. Será mostrado também neste capítulo, que o TTCT não permite calcular um escalonamento para um problema 5/4/J/F com a metodologia proposta pelos autores.

2.3.1 Problema introdutório 5/4/J/F

O centro tem quatro máquinas que são numeradas de 1 a 4 (M_1, \dots, M_4) e cada uma delas pode processar 5 produtos diferentes com variados tempos de manufatura. Cada produto deve ser processado nas quatro máquinas para que seja possível entregar ao cliente. O objetivo do problema é produzir e entregar todos os produtos do centro de trabalho no menor prazo possível. A ordem de manufatura e os tempos de máquina de cada produto são apresentados na tabela 1.

Será empregada na dissertação, como descrito no capítulo 1, a notação proposta por French (1982) para identificar os problemas de escalonamento. Para o problema em questão temos 5/4/J/F, ou seja, um problema com 5 produtos, 4 máquinas, um ambiente *jobshop* e com objetivo de fluxo mínimo de tempo total de produção. A dimensão deste problema, conforme French (1982), contém $(5!)^4 = 2,1 \times 10^8$ soluções diferentes de escalonamento.

Tabela 1: Ordem e tempos de processamentos dos produtos

Produto	Ordem			
	Tempo de operação em minutos			
A	M4 70	M3 10	M2 10	M1 10
B	M3 30	M2 10	M4 20	M1 10
C	M2 10	M3 10	M4 10	M1 50
D	M1 10	M4 10	M3 10	M2 20
E	M4 10	M3 10	M2 30	M1 10

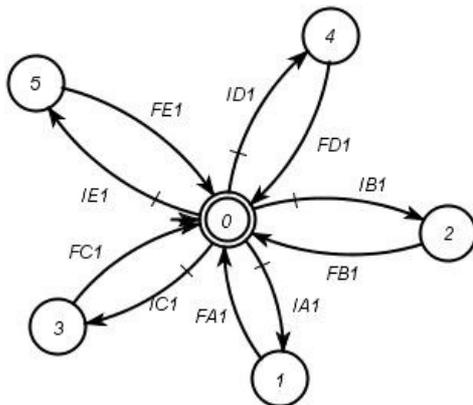
No exemplo da tabela 1, o produto A começa seu processo na máquina M4 (duração de 70 minutos), a segunda operação é feita na máquina M3 (10 minutos), em seguida realiza sua terceira operação na máquina M2 (10 minutos) e finaliza seu processo na máquina M1(10 minutos). Para todos os produtos, antes de começar o processamento nas máquinas, existe um tempo de preparação (*setup*) de 10 minutos, esse tempo é necessário para ajustar o equipamento conforme características específicas de cada produto.

2.3.2 Modelagem da planta

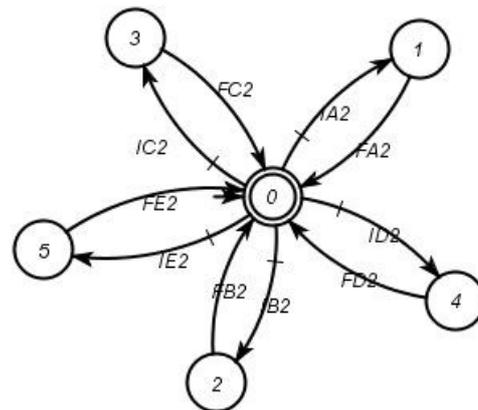
Para a modelagem da planta, o valor do *tick* escolhido foi de 10 minutos, que corresponde ao máximo divisor comum a todos os tempos. O *ATG* é utilizado para construir cada máquina envolvida, ou seja, um *ATG* para cada máquina do centro de trabalho. Os modelos dos autômatos *ATGs* das máquinas estão apresentados na figura 7. Estes autômatos contêm todos os eventos dos 5 produtos que estão relacionados com cada máquina e o estado de número 0 é o estado inicial e marcado em todos. Os eventos de cada uma delas estão descritos na tabela 2 mais abaixo.

Os eventos da planta serão estruturados e nomeados da seguinte forma: os eventos de inícios de operação têm *nomes* que começam com I e os eventos de fim de operação começam com F. O evento IA1, por exemplo, corresponde ao evento de início de operação do produto A na máquina 1 (I=Início de operação, A=Produto A, 1=Máquina 1), o evento

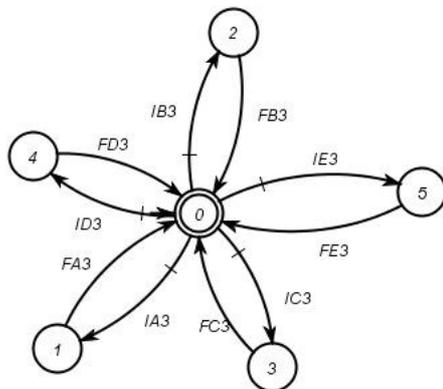
FE3 corresponde ao evento de fim de operação do produto E na máquina 3 (F=Fim de operação, E=Produto E, 3=Máquina 3). De forma genérica a codificação do nome dos eventos é dada por: I ou F + Nome do Produto + Número da Máquina. Todos os eventos de início de operação são modelados como forçáveis e os eventos de fim de operação com não controláveis.



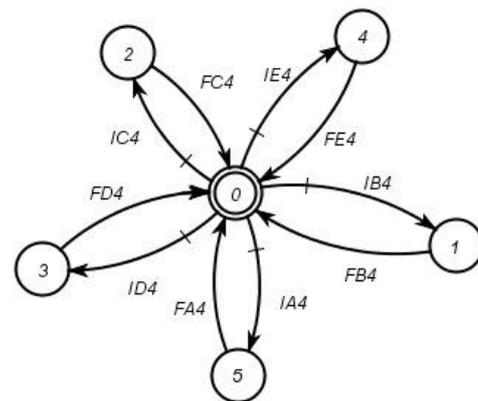
a) Autômato ATG da Máquina M1



b) Autômato ATG da Máquina M2



c) Autômato ATG da Máquina M3



d) Autômato ATG da Máquina M4

Figura 7: Autômatos ATGs das máquinas

Tabela 2: Classificação dos eventos e seus intervalos [l,u]

E.	Tipo do evento M1	[l,u]	E.	Tipo do evento M2	[l,u]
IA1	Forçável	[1, ∞]	IA2	Forçável	[1, ∞]
FA1	Não-controlável	[1,1]	FA2	Não-controlável	[1,1]
IB1	Forçável	[1, ∞]	IB2	Forçável	[1, ∞]
FB1	Não-controlável	[1,1]	FB2	Não-controlável	[1,1]
IC1	Forçável	[1, ∞]	IC2	Forçável	[1, ∞]

FC1	Não-controlável	[5,5]	FC2	Não-controlável	[1,1]
ID1	Forçável	[1, ∞]	ID2	Forçável	[1, ∞]
FD1	Não-controlável	[1,1]	FD2	Não-controlável	[2,2]
IE1	Forçável	[1, ∞]	IE2	Forçável	[1, ∞]
FE1	Não-controlável	[1,1]	FE2	Não-controlável	[3,3]
E.	Tipo do evento M3	[l,u]	E.	Tipo do evento M4	[l,u]
IA3	Forçável	[1, ∞]	IA4	Forçável	[1, ∞]
FA3	Não-controlável	[1,1]	FA4	Não-controlável	[7,7]
IB3	Forçável	[1, ∞]	IB4	Forçável	[1, ∞]
FB3	Não-controlável	[3,3]	FB4	Não-controlável	[2,2]
IC3	Forçável	[1, ∞]	IC4	Forçável	[1, ∞]
FC3	Não-controlável	[1,1]	FC4	Não-controlável	[1,1]
ID3	Forçável	[1, ∞]	ID4	Forçável	[1, ∞]
FD3	Não-controlável	[1,1]	FD4	Não-controlável	[1,1]
IE3	Forçável	[1, ∞]	IE4	Forçável	[1, ∞]
FE3	Não-controlável	[1,1]	FE4	Não-controlável	[1,1]

Tabela 3: Conjuntos dos eventos

Descrição do Conjunto	Conjunto	Eventos
Eventos do Produto A	$\sum PA$	{ IA1, IA2, IA3, IA4, FA1, FA2, FA3, FA4 }
Eventos do Produto B	$\sum PB$	{ IB1, IB2, IB3, IB4, FB1, FB2, FB3, FB4 }
Eventos do Produto C	$\sum PC$	{ IC1, IC2, IC3, IC4, FC1, FC2, FC3, FC4 }
Eventos do Produto D	$\sum PD$	{ ID1, ID2, ID3, ID4, FD1, FD2, FD3, FD4 }
Eventos do Produto E	$\sum PE$	{ IE1, IE2, IE3, IE4, FE1, FE2, FE3, FE4 }
Eventos da Planta	\sum	$\sum PA \cup \sum PB \cup \sum PC \cup \sum PD \cup \sum PE \cup \{tick\}$

Observa-se na figura 7a que a máquina M1 está disponível somente no estado 0. Os estados 1,2,3,4 e 5 representam situações em que a máquina está processando os produtos A,B,C,D e E respectivamente. Após a ocorrência de qualquer evento de início de operação, a máquina só volta a ficar disponível após o tempo necessário para que ocorra o evento de fim de operação correspondente. Para as outras 3 máquinas do centro de trabalho, a modelagem do ATG é semelhante, diferenciadas somente pelos nomes dos eventos e pelos intervalos de tempos associados a cada um deles.

Com os autômatos ATGs das máquinas modelados, a planta ATG é obtida com a composição desses, através da função *Comp* do sistema TTCT explicado na seção 2.1,

$$Planta\ ATG = Comp(M1, M2, M3, M4)$$

(Passo 1 da abordagem)

(1.080, 7.128)

A figura 8 representa o autômato *TTG* da máquina M1 com 16 estados e 21 transições. O tamanho dos autômatos na dissertação será indicado pela notação (estados, transições), para o caso (16,21). Este autômato foi obtido através da função *Timed Graph*³ do software TTCT, a partir das informações lógicas e dos intervalos de tempos $[l,u]$ associados a cada evento do autômato *ATG* correspondente.

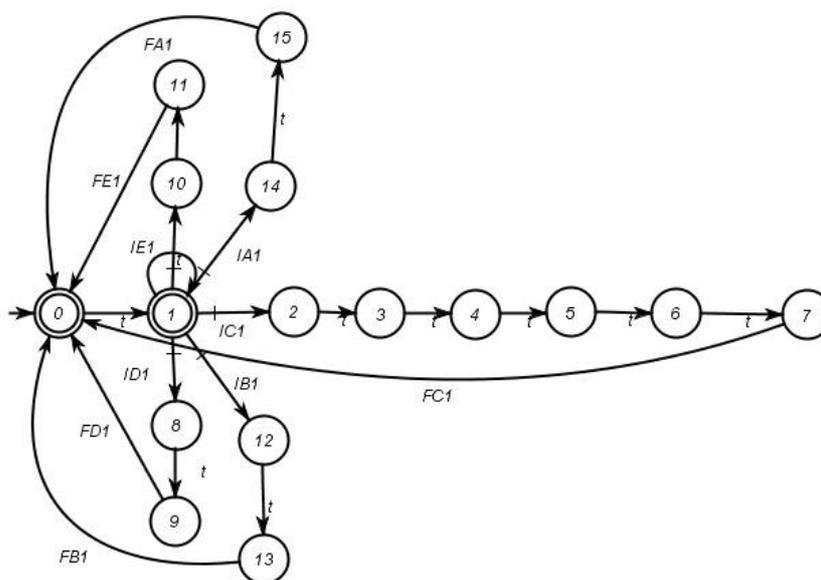


Figura 8: Autômato TTG da máquina M1

O autômato apresentado na figura 8 também ilustra o crescimento do número de estados do *TTG* quando comparado ao *ATG* correspondente. A relação entre o número de estados e transições do *ATG* da planta global e o seu correspondente *TTG* é ainda mais expressiva. No caso do exemplo, o *ATG* da planta tem tamanho (1.080, 7.128) e seu *TTG* (36.960, 107.950).

No autômato da figura 8, podemos observar que existe explicitamente a introdução do evento tick (t) em sua estrutura. Cada tick do relógio, como dito anteriormente, representa 10 minutos de trabalho para este problema. Podemos notar que no autômato da figura 8,

³ Para detalhes da função Timed Graph, verificar em (Wonham, 2009).

entre o eventos ICI e FC1, há 5 ocorrências do evento tick (*lower time* e *upper time* iguais a 5), ou seja, o tempo entre o início de operação e o fim de operação do produto C na máquina M1 está vinculado à ocorrência de exatamente 5 eventos ticks ou 50 minutos.

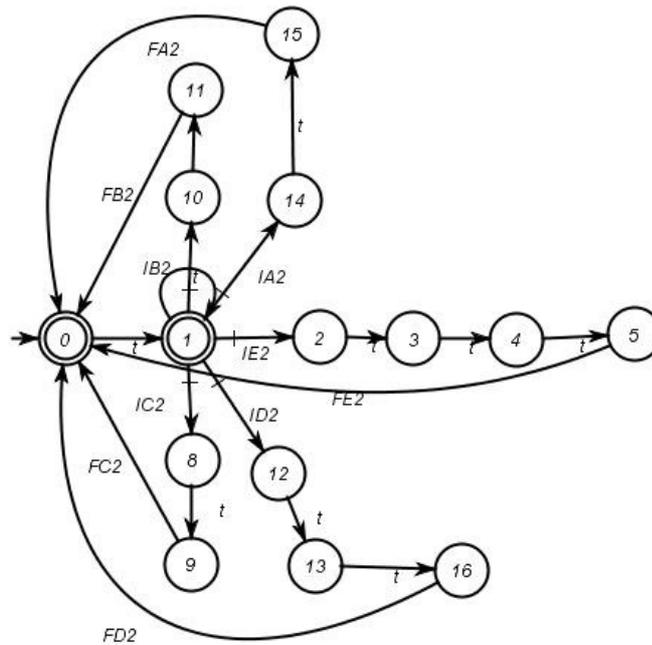


Figura 9: Autômato TTG da máquina M2

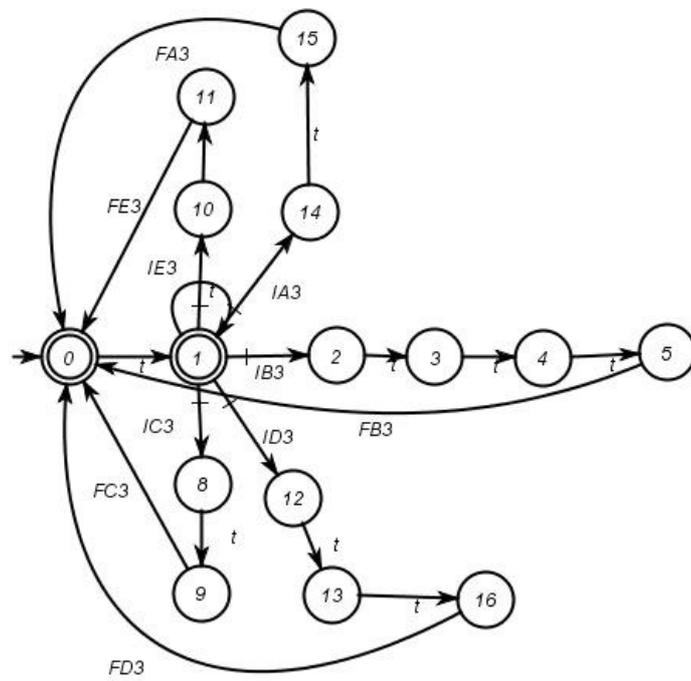


Figura 10: Autômato TTG da máquina M3

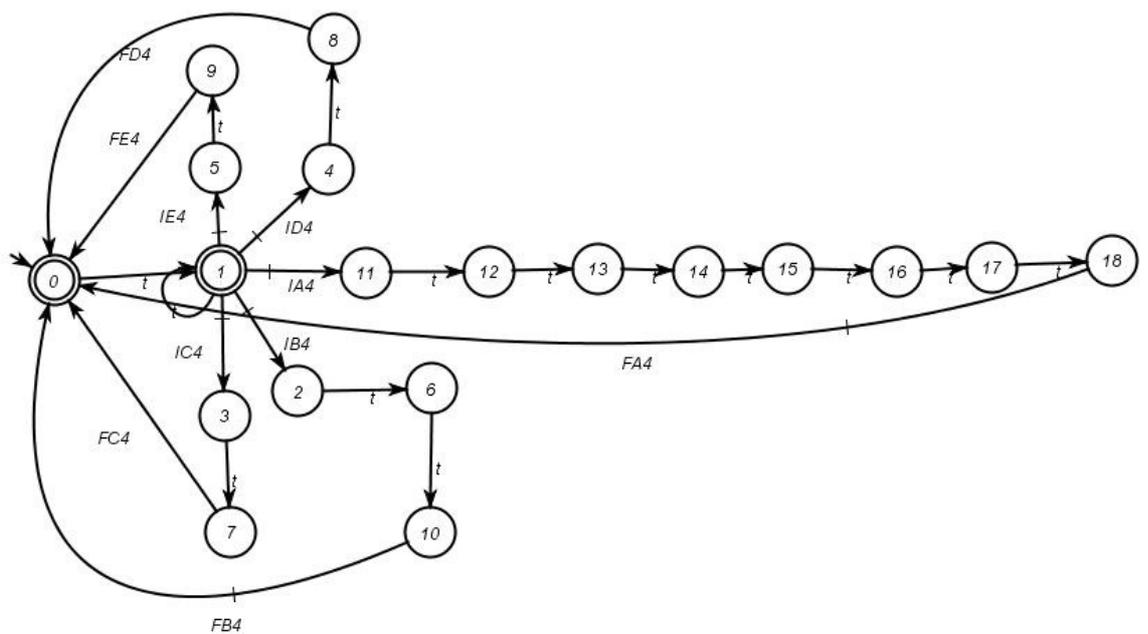


Figura 11: Autômato TTG da máquina M4

Nas figuras 9, 10 e 11 são apresentados os modelos dos autômatos *TTGs* das máquinas M2, M3 e M4 respectivamente. Esses modelos também foram gerados por seus respectivos *ATGs* a partir da função *Timed Graph* do software TTCT.

Na figura 11, por exemplo, podemos notar que entre os eventos IA4 e FA4, há 7 ocorrências do evento tick (*lower time* e *upper time* iguais a 7). Baseado nessa informação, a duração da operação de A na máquina M4 é de 7 unidades de tempo ou 70 minutos.

O objetivo da apresentação dos *TTGs* foi demonstrar visualmente os modelos temporais e o aumento do número de estados frente a seus correspondentes *ATGs*. É importante ressaltar que os *TTGs* de cada subsistema isolado não farão parte da síntese de Brandin e Wonham (1992), mas sim, a planta com seu modelo temporizado. Para obtenção da planta com seu modelo temporizado é executada a função *Timed_Graph* do TTCT em cima da Planta *ATG*, calculada no passo 1 da abordagem com a função *Comp*. O resultado da função *Timed Graph* representa a tradução do *ATG* em seu *TTG* correspondente.

$$\begin{aligned}
 \text{Planta } TTG &= \text{Timed_Graph}(\text{Planta } ATG) && (\text{Passo 2 da abordagem}) \\
 &(36.960, 107.950)
 \end{aligned}$$

O autômato *TTG* calculado no passo 2 é o modelo final da planta que fará parte da síntese.

2.3.3 Modelagem das especificações

Para a modelagem das especificações, Brandin e Wonham (1992) propõem a modelagem diretamente através do autômato *TTG*. O autômato da figura 12 representa a especificação da ordem pela qual o produto A deve ser processado nas máquinas.

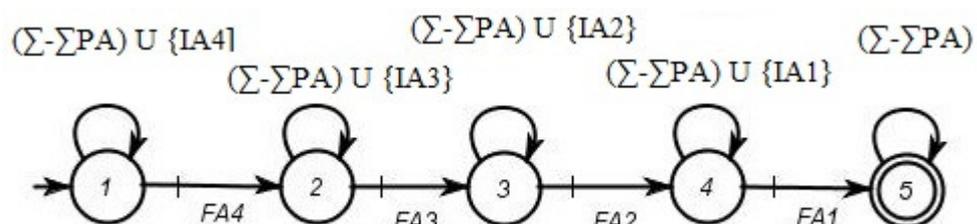


Figura 12: SA: Autômato *TTG* da especificação de roteiro do produto A

Para alcançar o estado marcado 5 no autômato da figura 12, é necessária a ocorrência dos eventos FA4, FA3, FA2 e FA1, conforme ordem estabelecida na tabela 1. Esses eventos representam os términos das operações do produto A nas 4 máquinas. Com essa marcação, um supervisor não bloqueante somente habilita sequências que permitam concluir o roteiro de cada produto.

Os eventos IA1, IA2, IA3 e IA4 são os eventos de início de operação do produto A nas máquinas e eles são modelados no autômato com o objetivo de impedir duas ocorrências do mesmo evento de início de operação antes de se completar a tarefa. Para os outros produtos, a modelagem da ordem de processamento é semelhante, exceto pelos nomes dos eventos e pelas ordens de processamento nas máquinas.

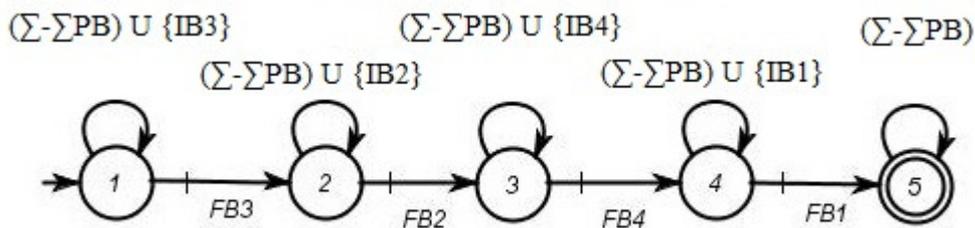


Figura 13: SB: Autômato *TTG* da especificação de roteiro do produto B

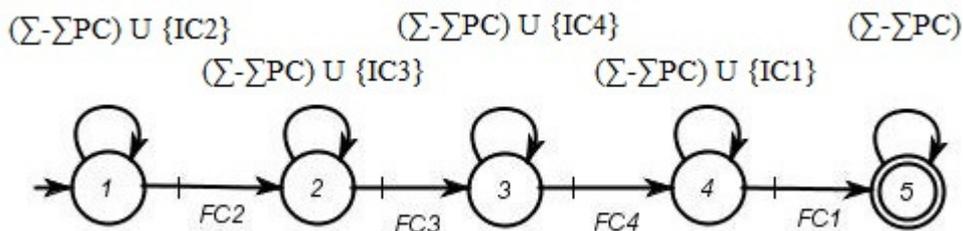


Figura 14: SC: Autômato *TTG* da especificação de roteiro do produto C

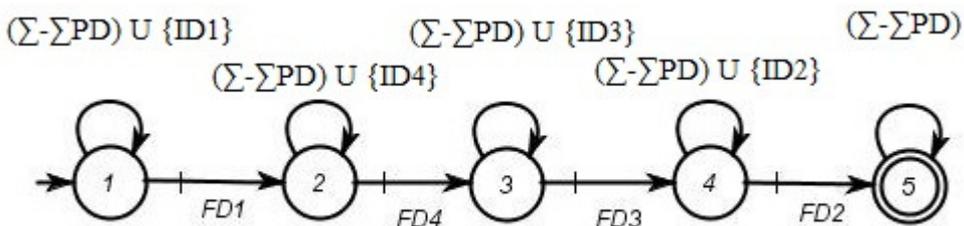


Figura 15: SD: Autômato *TTG* da especificação de roteiro do produto D

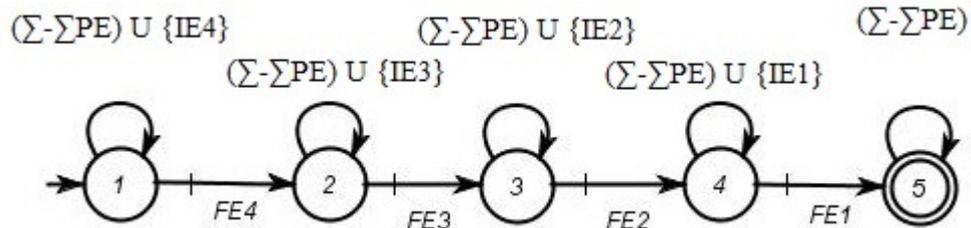


Figura 16: SE: Autômato *TTG* da especificação de roteiro do produto E

Para realizar a composição de uma especificação global K é usada a função *meet* no TTCT. Essa função é similar ao comportamento do produto síncrono, mas faz parte deste resultado somente o comportamento compartilhado entre os autômatos das especificações. As linguagens resultantes de $K = meet(K_1, K_2)$, onde K_1 e K_2 são especificações, podem ser representadas por $L(K) = L(K_1) \cap L(K_2)$ e $L_m(K) = L_m(K_1) \cap L_m(K_2)$. Diferentemente da função *Comp*, a função *meet* **não** tem objetivo de compor os intervalos de tempo dos eventos, mesmo porque, a função *meet* é executada somente para os modelos *TTGs*, onde os eventos do tempo já estão presentes nas transições dos autômatos. É correto afirmar que a função *meet* tem igual resultado ao produto síncrono, quando os conjuntos dos eventos dos autômatos envolvidos são iguais. Como neste problema, os conjuntos dos eventos das especificações são iguais, podemos considerar que o resultado obtido pela função *meet* é o mesmo do que o produto síncrono.

$$Spec = Meet(SA, SB, SC, SD, SE) \quad (Passo 3 da abordagem)$$

(2.500, 22.250)

O autômato *Spec* é obtido a partir dos autômatos das especificações de ordem de processamento dos produtos (roteiros de produção). As especificações SA, SB, SC, SD e SE são respectivamente os autômatos dos roteiros de produção dos produtos A,B,C,D e E.

Para resolver o problema, além das especificações de ordem de processamento acima apresentadas, Brandin e Wonham (1992) desenvolveram outro tipo de especificação relacionada com o tempo. Este tipo será classificado nesta pesquisa como uma especificação de prazo. Essa especificação tem o objetivo de impor um prazo para a execução de todas as tarefas. A figura 17 mostra o autômato que representa este modelo.

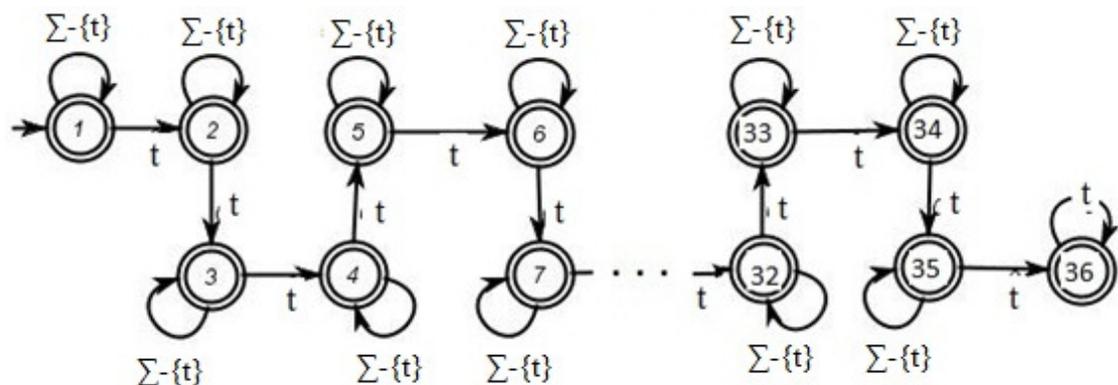


Figura 17: STempG(35): Exemplo do autômato TTG da especificação temporal

Todos os estados do autômato da figura 17 estão marcados e só é possível avançar de um estado para outro através da ocorrência do evento *tick* (t). Do estado 1 até o estado 35, todos os eventos da planta estão habilitados. A partir do estado 35, somente o evento do relógio está habilitado. Desse modo, para atender a especificação, todas as tarefas devem ser completadas em no máximo 35 *ticks*. Esta especificação participa da síntese após o cálculo do supervisor para as especificações de roteiro, por isso será apresentada o seu passo na síntese na próxima seção.

2.3.4 Síntese do supervisor

A síntese do supervisor é feita no software TTCT. Conforme descrito na seção 2.3.2 e 2.3.3, os dois primeiros passos têm objetivo de construir o modelo da planta e o passo 3 constrói a especificação global *Spec* a ser atendida. Com o modelo da planta e o modelo das especificações gerados, a abordagem de Brandin e Wonham (1992) propõe a síntese do supervisor através da função *SupCon* do software TTCT. A função *SupCon* executa a operação $SupC(K, G)$ explicada na seção 2.2, neste caso $G = Planta\ TTG$ e $K = Spec$. Os últimos passos são apresentados a seguir.

$Sup = SupCon (Planta\ TTG, Spec)$

(Passo 4 da abordagem)

(428.038, 879.034)

O supervisor Sup representa o controle menos restritivo diante das restrições de ordem de roteiros impostas. Por último, Brandin e Wonham (1992) adicionam outra especificação chamada $STempG(T)$, apresentada na figura 17, que se relaciona com o tempo máximo T de execução das tarefas. Desta maneira, além de garantir a correta ordem de processamento dos produtos, propõe-se a síntese de um supervisor que respeite a especificação de prazo $STempG(T)$.

$$SupOt = SupCon(Sup, STempG(T)) \quad (\text{Passo 5 da abordagem})$$

(out of memory – Problem too large)

O supervisor $SupOt$ em (5) é o supervisor que além de garantir a ordem correta de processamento dos produtos, estabelece um prazo para a execução de todas as tarefas. Para a abordagem de Brandin e Wonham (1992) não foi possível realizar a operação (5) devido ao tamanho dos autômatos envolvidos. O TTCT enviou a mensagem “*out of memory – Problem too large*”.

Caso um supervisor fosse obtido em (5), Brandin e Wonham (1992), com o objetivo de reduzir o prazo de execução para todas as tarefas, propõem ainda um decréscimo incremental de 1 unidade de tempo na especificação $STempG(T)$ até que não seja mais possível encontrar um supervisor (solução vazia). A solução é considerada vazia, quando não existe um supervisor capaz de executar todas as tarefas no tempo estabelecido pela especificação temporal. Quando isso acontece, a solução com a especificação de prazo do passo imediatamente anterior determina o tempo e o supervisor ótimo.

Observa-se que, qualquer sequência de eventos reconhecida pelo supervisor resultante deve obedecer ao prazo estabelecido pela especificação de prazo. Assim, basta seguir um caminho que leve ao estado final do supervisor para se obter um escalonamento ótimo.

Neste capítulo foi apresentada a proposta de Brandin e Wonham (1992) dos modelos das máquinas, das especificações e a síntese para o controle temporizado numa aplicação. Mostra-se que o tamanho da planta temporizada obtida tem 36.960 estados para um ambiente de 4 máquinas e 5 produtos. O crescimento de estados da planta é consequência do número de máquinas envolvidas e também da quantidade de produtos com seus respectivos tempos de produção. Como não foi possível encontrar um supervisor de prazo

mínimo em função da explosão de estados, a presente pesquisa aprofunda neste tipo de aplicação e propõe outros modelos para construção da planta, das especificações e de uma metodologia própria para síntese do supervisor com o objetivo de gerar um escalonamento.

CAPÍTULO 3

Método proposto para o escalonamento

Este capítulo tem o objetivo de apresentar uma nova proposta de modelagem dos autômatos temporizados a fim de reduzir o tamanho dos modelos. É apresentado um algoritmo para síntese de escalonamento baseada na composição incremental dos roteiros de produção e prazos das tarefas e um método inspirado no da bissecção para minimização do tempo de produção global e também dos tempos de produção de cada tarefa.

Uma alteração significativa da modelagem das máquinas comparada a de Brandin e Wonham (1992) foi a exclusão dos autômatos *ATGs* no processo de construção da planta, eles não serão utilizados em nenhum momento do método. Nesta proposta, a construção dos modelos de cada máquina é feita diretamente pelo autômato *TTG*. Será mostrado, que essa alteração se torna permitida e necessária para alcançar os objetivos desejados com os modelos propostos. Além disso, os eventos de fim de operação serão classificados como forçáveis, diferente de Brandin e Wonham (1992) que eram modelados como não controláveis. Com isso, os eventos de início e fim de operação são forçáveis e somente o evento do tick (t) do relógio é não controlável. Cada alteração será detalhada nas seções seguintes. Apresenta-se também neste capítulo um método de síntese incremental para reduzir a complexidade das operações (Passo 4 de BW 1992) e (Passo 5 de BW 1992). Todas as alterações propostas e o método incremental serão aplicados ao mesmo problema apresentado no capítulo anterior, sendo que neste capítulo será usado o indicador *PP_TM_RL*, que apresenta três objetivos para medir o resultado do escalonamento. Este indicador significa obter *Prazo Pontual para as Tarefas (PP, primeiro objetivo)*, *Tempo Mínimo de Produção (TM, segundo objetivo)* e *Redução do Lead Time dos Jobs (RL, terceiro objetivo)*.

O método proposto e a aplicação desenvolvida neste capítulo foram apresentados em formato de um artigo científico aprovado no XVIII Congresso Brasileiro de Automática (CBA-2010).

3.1 Autômatos TTGs dos recursos

Uma diferença singular deste autômato para o de Brandin e Wonham (1992) é que este apresenta somente um evento de início de operação independente do número de produtos que o recurso possa executar, será mostrado que para fins de escalonamento esta modificação é permitida. Os nomes dos eventos dos produtos seguem a mesma estrutura do capítulo anterior e os nomes dos eventos de início de operação ficam assim estruturados: I + Nome da Máquina, como exemplo, o evento de início de operação da máquina M1 é o IM1, o da máquina M2 é IM2 e esta lógica é seguida para todas as outras máquinas.

O autômato *ATG* da figura 18 contém o evento IM4 (Início de tarefa na máquina M4) com intervalo [*lower time* , *upper time*] igual $[1, \infty]$ e os eventos FA4, FB4, FC4, FD4 e FE4 com $[7,7]$, $[2,2]$, $[1,1]$, $[1,1]$ e $[1,1]$ respectivamente. Com estes valores, a tradução temporizada do autômato da figura 18 através do TTCT é o autômato *TTG* mostrado na figura 19. Pode se notar, que no estado 3 do autômato da figura 19, os eventos FC4, FD4 e FE4 estão habilitados e não é mais permitida qualquer ocorrência do evento do tempo (t) a partir deste estado. Nesta situação, esses eventos ocorrerem em exatamente uma unidade de tempo depois da ocorrência do evento IM4. Para os eventos que tinham *lower time* maiores que 1 tick, como era previsto para FA4 e FB4, não é mais possível a sua ocorrência, pois os valores de *upper times* dos eventos FC4, FD4 e FE4 limitam o maior tempo possível (*deadlines*) para finalizar uma tarefa depois da ocorrência do evento IM4. Pela definição de Brandin e Wonham (1992), essa situação é correta, pois os autores trabalham com o conceito de *deadlines* para os eventos e por isso o resultado.

Diante desse conceito, não é possível construir os autômatos *TTGs* dos recursos deste método a partir de um *ATG*. Na figura 20 é mostrado o autômato *TTG* desta abordagem da máquina M4 para exemplificar a questão.

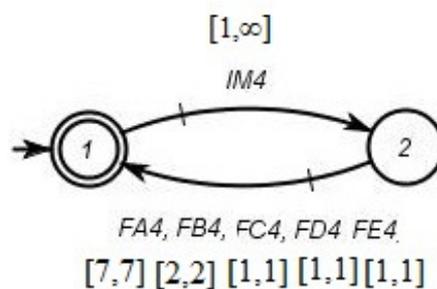


Figura 18: ATG M4

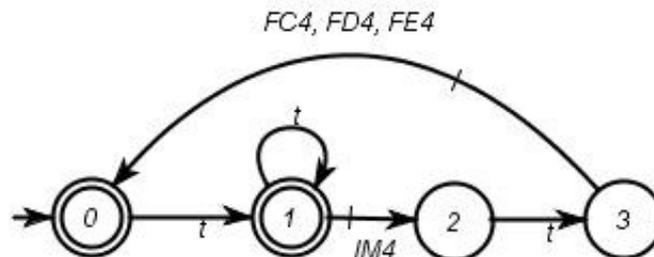


Figura 19: TTG M4 com problema

O autômato *TTG* da figura 20 abaixo tem o mesmo objetivo do autômato da figura 11, ou seja, modelar o comportamento da máquina M4.

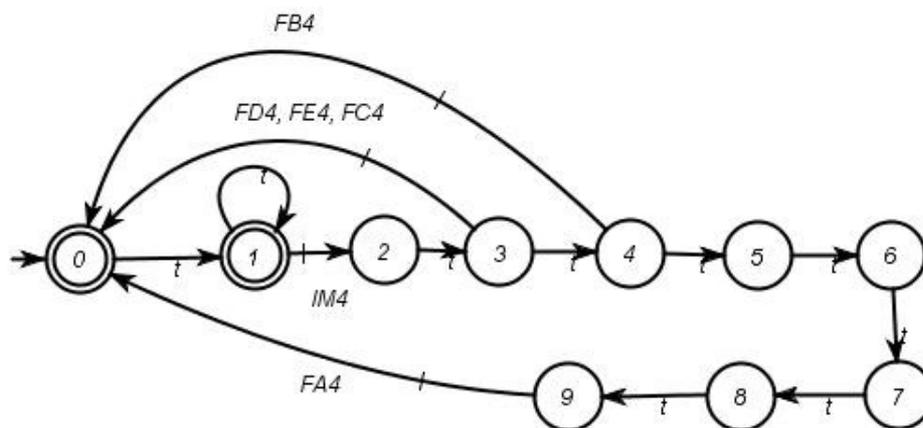


Figura 20: Autômato TTG da máquina M4

A apresentação do *ATG* da figura 18 tem o objetivo de mostrar que não é possível gerar um *TTG* correspondente com os aspectos desejados do autômato da figura 20, ou seja, não existe nenhum autômato *ATG*, através da metodologia de Brandin e Wonham (1992) capaz de gerar o autômato *TTG* deste método. No estado 3, com o conceito de *deadline* por evento em Brandin e Wonham (1992), um dos eventos deste estado (FC4, FD4 ou FE4) tem **obrigação** de ocorrer, porque foi alcançado o seu respectivo *deadline*. Com isso, não há a possibilidade da ocorrência de outros eventos diferentes daqueles e as transições dos eventos FA4 e FB4 não seriam construídas, pois os estados com números de 4 a 9 não existiriam.

Com a substituição dos vários eventos de início de operação para somente um, os eventos de início e fim de operação precisam ser controláveis e forçáveis para este modelo. Essa alteração se torna necessária, para que seja possível encontrar um supervisor com a prerrogativa de preemptar o evento do relógio e conseqüentemente finalizar as operações. Com estas características do modelo proposto, fica mantida a noção de controlabilidade da TCS para SEDT (BW 1992). Diferentemente de Brandin e Wonham (1992), que a proposta é o controle dos eventos em tempo real, esta pesquisa tem o objetivo de obter um ordenamento que satisfaça as especificações, então neste contexto, os eventos de fim de operação podem ser os eventos que determinam quais os produtos ou tarefas são processados.

No capítulo 2, na seção 2.1 mostrou-se que o produto síncrono entre *TTGs* com eventos em comum pode resultar num comportamento não real do sistema. Como os conjuntos de eventos dos recursos para este método são disjuntos, pode-se construir a planta através do produto síncrono dos *TTGs* e conseqüentemente, a modelagem através dos *ATGs* não é mais necessária. Essa modificação não gera nenhuma perda no modelo e é justificável em virtude da característica desse problema. Faz sentido colocar eventos com nomes diferentes para produtos diferentes em máquinas diferentes.

Para manter as características do autômato da figura 20 e as vantagens proporcionadas com sua modelagem para fins de escalonamento, optou-se por excluir os modelos *ATGs* e trabalhar diretamente com os *TTGs*. Com a exclusão destes modelos, é modificada parte da semântica anterior que tinha relação com os *deadlines* para os eventos. Ao invés disso, assume-se com esta alteração, que não existe mais *deadlines* para os eventos e sim *deadlines* para os recursos, ou seja, existe nesta nova proposta, um limite superior de tempo para que um recurso finalize uma tarefa, desde que tenha sido inicializado.

A construção da planta diretamente com os *TTGs* é então permitida e necessária, o que possibilita iniciar a síntese com a planta mais enxuta e manter o comportamento real do sistema, sem inconsistências. A título de exemplo, o autômato temporizado da figura 20 tem 10 estados e 15 transições (10,15), enquanto o autômato que representara este mesmo recurso na abordagem de Brandin e Wonham (1992) tem (19,24), como mostrado na figura 11. A planta *TTG* obtida com este método tem 1.575 estados, enquanto que em BW (1992)

tem 36.960. Na seção 3.7 serão apresentados com detalhes os resultados obtidos com esta abordagem.

Em função da utilização de apenas um evento de início de operação, os eventos de fim de operação ocorrem a partir de uma única cadeia de eventos *tick*, limitada pelo maior tempo de processamento das tarefas. Por exemplo, no caso da figura 20, a tarefa com tempo 7 define o tamanho desta cadeia. Nas figuras 21, 22 e 23 dos autômatos *TTGs* das máquinas M1, M2 e M3 também podem ser observadas estas questões.

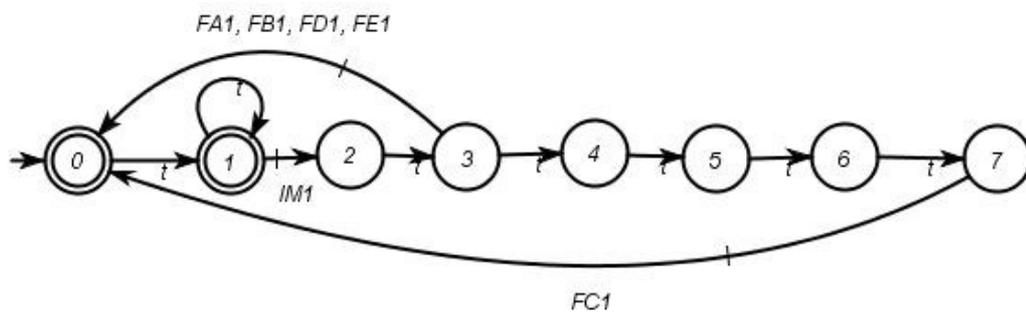


Figura 21: Autômato *TTG* da máquina M1

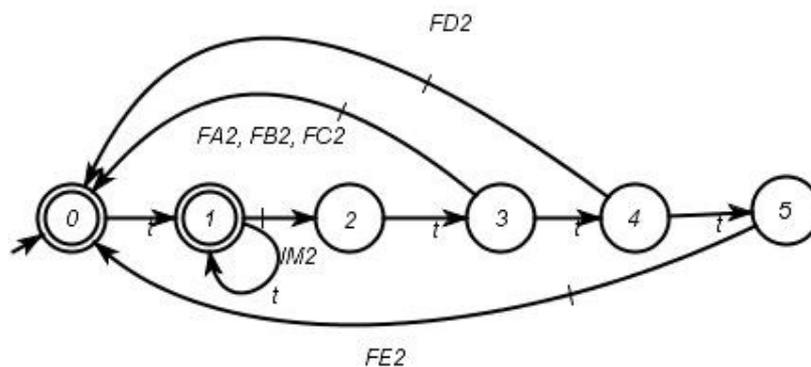


Figura 22: Autômato *TTG* da máquina M2

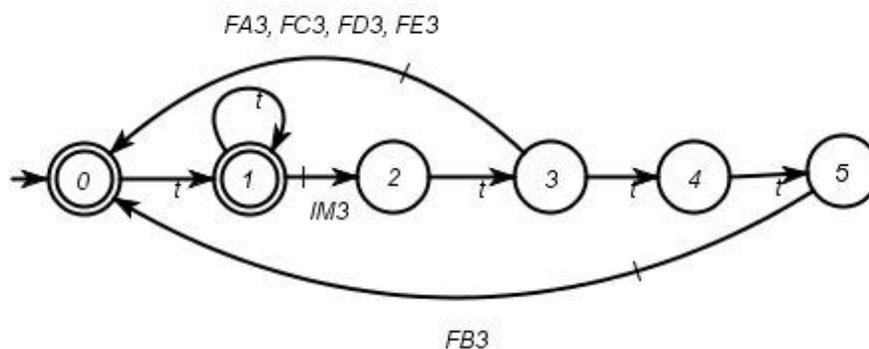


Figura 23: Autômato *TTG* da máquina M3

Assim, diferentemente de Brandin e Wonham (1992), o número de estados dos *TTGs* dos recursos da planta depende apenas do maior tempo de execução e não depende mais da quantidade de produtos que o recurso pode executar. Essa diferença é muito relevante, pois os recursos atuais de manufatura são muito flexíveis, produzem variados produtos e com a alteração proposta pode-se atenuar o crescimento dos modelos. Esta modificação no modelo do recurso, impõe a limitação de que só é possível saber qual o produto está sendo processado, quando o evento de fim de processamento ocorre. Este fato, além da consideração da natureza dos eventos, impede a utilização do resultado do modelo como um supervisor, já que a ação de controle associada a um evento de início de operação se mantém indeterminada até o final da operação. No entanto, ao gerar um escalonamento *offline*, escolhendo uma das cadeias reconhecidas pelo supervisor, pode-se sempre deduzir o significado do evento de início a partir do próximo evento de fim de operação.

3.2 Autômatos *TTGs* das especificações dos roteiros dos produtos

Os modelos das especificações de roteiro devem ser redefinidos a partir dos modelos modificados. Como existe somente um evento de início de operação em cada recurso, é necessário que ocorra para cada estado alcançado com um evento de fim de operação, o evento de início para o recurso seguinte.

A introdução do evento de início de operação em cada etapa assegura que, enquanto não for finalizada a primeira operação de um produto, não poderá ser iniciada a sua segunda. Este cuidado está vinculado à duração das operações, questão central em sistemas a eventos discretos temporizados. Se por exemplo, o tempo de duração da primeira operação fosse menor que o tempo da segunda e, antes de finalizar a primeira, tivesse ocorrido pelo menos uma execução do evento de início da segunda, o sistema poderia finalizar esta última com duração menor e não com o seu tempo completo. Nesta situação, o tempo teria corrido em paralelo para as duas operações, o que não representaria um comportamento real.

Por exemplo, a figura 24 representa o autômato do roteiro do produto A. A ordem dos eventos segue a engenharia de processo para os produtos apresentada na tabela 1. Neste autômato, em todos os estados, todos os eventos de início de operação estão habilitados. Isso permite que outro produto possa ser processado nos recursos compartilhados em momentos diferentes.

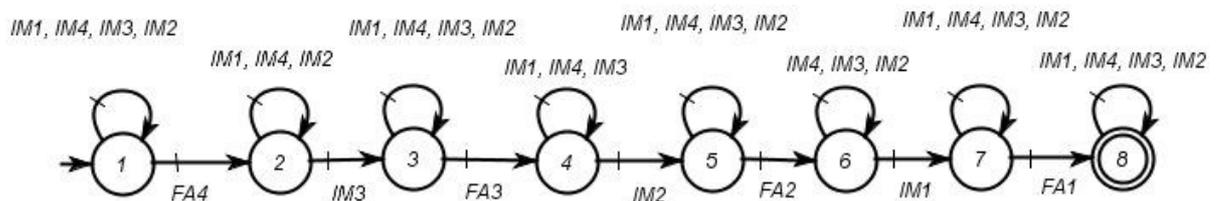


Figura 24: SA: Autômato TTG da especificação de roteiro de A

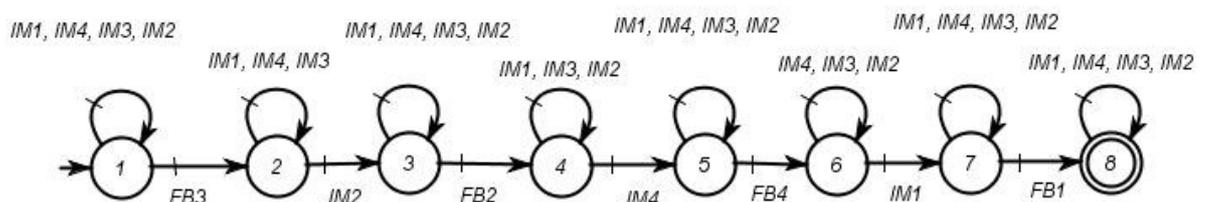


Figura 25: SB: Autômato TTG da especificação de roteiro de B

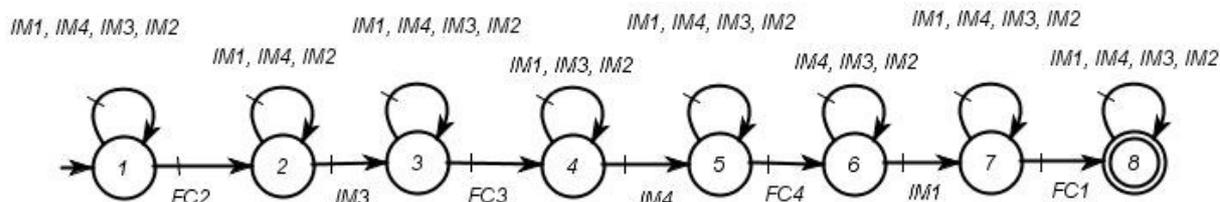


Figura 26: SC: Autômato TTG da especificação de roteiro de C

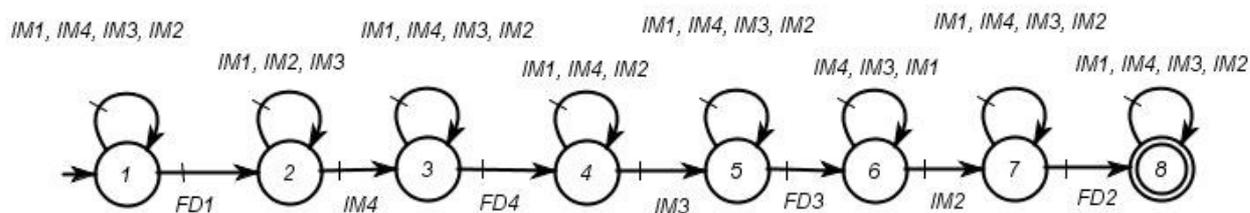


Figura 27: SD: Autômato TTG da especificação de roteiro de D

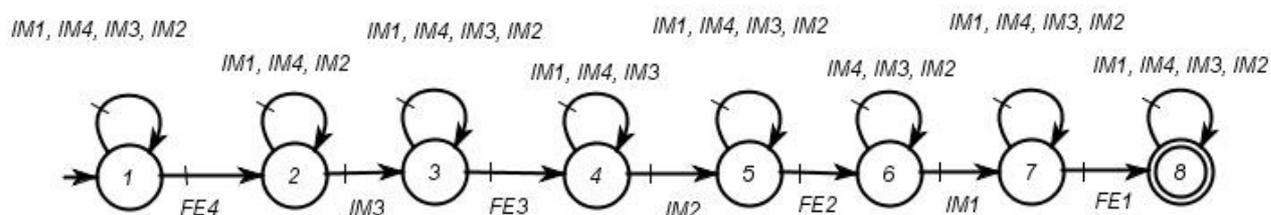


Figura 28: SE: Autômato TTG da especificação de roteiro de E

3.3 Autômatos TTGs das especificações dos prazos de entregas

A fim de generalizar a restrição de tempo usada por Brandin e Wonham (1992), utiliza-se aqui o conceito de prazo de entrega. Definem-se então dois tipos de especificações temporais: Prazo global e prazo individual por produto.

O autômato da especificação de prazo global impõe a restrição de prazo máximo para terminar todas as operações para todos os produtos envolvidos. Esta especificação segue o mesmo modelo do apresentado por Brandin e Wonham (1992) para o tempo máximo de execução das tarefas. Este autômato está apresentado na seção 2.3.3 desta dissertação.

Como introduzido no início desta seção, a restrição de tempo usada por BW (1992) será estendida para o conceito de prazo de entrega por produto. Este tipo de especificação

se tornou muito relevante na pesquisa, porque além de provocar vantagens computacionais, prazos de entrega são informações rotineiras em ambientes produtivos. Prazos de entrega acordados entre as empresas e clientes fazem parte da regra do negócio. Na tabela 4 são apresentados novamente os dados do problema anterior, mas agora adicionada dos prazos de entrega dos produtos.

Tabela 4: Prazos de entrega dos produtos

Produto	Ordem				Prazo (tick)
	Tempo de operação em ticks (tick = 10 minutos)				
A	M4 7	M3 1	M2 1	M1 1	18
B	M3 3	M2 1	M4 2	M1 1	11
C	M2 1	M3 1	M4 1	M1 5	16
D	M1 1	M4 1	M3 1	M2 2	12
E	M4 1	M3 1	M2 3	M1 1	10

Define-se então, uma especificação temporal para cada produto. O autômato da figura 29 representa o prazo de entrega de 100 minutos (10 ticks) para o produto E. A imposição da restrição temporal neste caso está associada apenas a este produto.

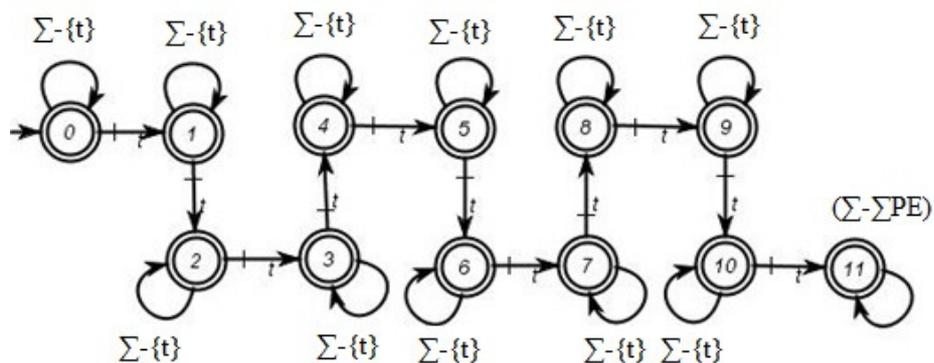


Figura 29: *STempE*: Exemplo do autômato *TTG* da especificação de prazo de entrega do produto E

Pode-se notar que até o estado 10 estão habilitados todos os eventos e no estado 11, o conjunto de eventos específicos do produto E ($\sum PE$) é desabilitado. Isso impõe a restrição de 10 ticks (100 minutos) para a entrega do produto E. Para a resolução do problema com este método, cada produto tem sua especificação de prazo obtida de modo semelhante ao da figura 29.

3.4 Algoritmo incremental para a síntese

Para o cálculo do escalonamento, que respeite os prazos de entregas dos produtos, propõe-se executar os passos da síntese de forma incremental aplicando-se alternadamente as restrições temporais S_{Temp_i} e de roteiro S_i para cada produto $i = A, B, \dots, P$. Segue abaixo, o algoritmo para encontrar o supervisor que impõe a restrição dos prazos de entregas e dos roteiros de produção para todos os produtos.

1	Planta = Sync (Recurso 1, Recurso 2, ..Recurso N)
2	Para $i = A, B, \dots, P$
3	Supervisor $T_i = \text{SupCon (Planta, } S_{Temp_i})$
4	Planta = Supervisor T_i
5	Supervisor $R_i = \text{SupCon (Planta, } S_i)$
6	Planta = Supervisor R_i
7	Fim Para
8	Supervisor = Planta

Algoritmo 1: Prazos e Roteiros

A escolha da sequência de índices, ou seja, a ordem em que os produtos participarão da síntese pode ser feita de forma aleatória. A variação entre diferentes ordens de processamento dos produtos no algoritmo gera apenas supervisores intermediários diferentes, mas o supervisor final terá o mesmo significado. A síntese proposta é realizada com os índices ordenados de forma crescente dos prazos de entrega dos produtos. Essa ordenação é uma sugestão adicional para amenizar o crescimento dos autômatos intermediários.

Quando não for possível encontrar um supervisor, significa que não existe uma solução de escalonamento, que atenda a todos os prazos dos produtos simultaneamente. Nesse caso, uma especificação de prazo de entrega de um produto, cuja restrição temporizada já foi considerada na síntese, deve ser aumentada (*relaxamento do prazo*). Este aumento é feito até que seja possível encontrar novamente um supervisor e o algoritmo continua para

os produtos restantes. Sob essas condições, a ordem dos índices altera o resultado e nesse caso a ordem deveria ser do produto mais prioritário para o menos prioritário.

Na *Linha 1* do algoritmo é construída a planta através do produto síncrono das máquinas envolvidas (M1,M2,M3 e M4). Na *linha 3* é feita a primeira síntese para encontrar o supervisor que garante a especificação do prazo de entrega do produto *i*. Na *linha 5*, executa-se a síntese do supervisor encontrado anteriormente com a especificação de roteiro de produção do produto de índice *i*. Esses passos se repetem, até que todos os produtos da planta tenham participado da síntese. O supervisor encontrado da *linha 8* contém todos os sequenciamentos de eventos que permite gerar o escalonamento da produção.

3.5 Otimização do supervisor com tempo mínimo global

A partir do resultado do algoritmo 1 da seção 3.4, faz-se a síntese como em BW (1992). São aplicadas as imposições de especificações temporais globais $STempG(T)$ (prazo único para terminar todas as tarefas dos produtos) (*Linha 3*). As especificações temporais sofrem decréscimos unitários sucessivos no valor do seu tempo (*Linha 4*) e o algoritmo continua enquanto existe um supervisor. O supervisor final encontrado é o escalonamento com o menor tempo de execução para todas as tarefas, que respeita o prazo de entrega dos produtos.

1	Enquanto Supervisor \neq Vazio
2	Planta = Supervisor
3	Supervisor = SupCon (Planta, STempG(T))
4	T = T - 1
5	Fim Enquanto
6	Supervisor = Planta

Algoritmo 2:Redução do prazo máximo das tarefas por BW (1992)

Nesta pesquisa, diferentemente de BW (1992), será utilizado um algoritmo inspirado no método da bissecção, usualmente utilizado para cálculo numérico de zeros de funções contínuas. Para este algoritmo define-se o valor inicial do limite superior do intervalo do tempo (T_{max}) e o valor inicial do limite inferior do intervalo (T_{min}). O valor médio (T_{med}) é definido como a parte inteira da média aritmética entre T_{max} e T_{min} . Considera $T_{med} = (T_{max} + T_{min} / 2)$ e calcula o supervisor com T_{med} através da função

$SupCon(Planta,STempG(Tmed))$. Se existir um supervisor, então $Tmax$ recebe o valor de $Tmed$ calculado e $Tmin$ continua com o mesmo valor, caso contrário, $Tmin$ recebe o valor de $Tmed$ calculado e $Tmax$ continua com o mesmo valor. Como a cada passo descarta-se a metade do intervalo, esse algoritmo converge para o prazo ótimo em tempo logarítmico no tamanho do intervalo.

Para o algoritmo de síntese foi delineado o intervalo inicial dos prazos da seguinte maneira: o valor inicial do limite superior do intervalo ($Tmax$) é o maior prazo realizável, ou seja, o maior prazo estipulado no algoritmo 1 da seção 3.4 e que foi possível encontrar um supervisor. O valor inicial do limite inferior do intervalo ($Tmin$) foi definido como a maior soma dos tempos de processamento entre os produtos nas máquinas. Com os valores do limite superior e inferior do intervalo definidos pode-se iniciar o algoritmo.

O algoritmo começa através da síntese do supervisor encontrado na seção 3.4 com a especificação temporal de prazo igual ao tempo médio ($Tmed$) (*Linha 4*).

1	Planta = Supervisor
2	Enquanto ($Tmax > Tmin$)
3	$Tmed = \text{Quociente}(Tmax + Tmin) / 2$
4	SupervisorOt = SupCon (Planta, STempG(Tmed))
5	Se Existe(SupervisorOt)
6	$Tmax = Tmed$
7	Planta = SupervisorOt
8	Senão
9	$Tmin = Tmed$
10	Fim Se
11	Fim Enquanto
12	Supervisor = Planta

Algoritmo 3: Redução do prazo máximo das tarefas pelo método da bissecção

As especificações temporais, os limites do intervalo, sofrem alterações sucessivas no valor do seu tempo conforme *Linhas 5 a 9* do algoritmo, ou seja, se for possível encontrar um supervisor com uma especificação temporal de valor $Tmed$, então o algoritmo deve continuar e $Tmax$ recebe o valor do último $Tmed$ calculado, caso contrário, se não foi encontrado um supervisor para a especificação temporal com valor de $Tmed$, então $Tmin$ recebe o valor de $Tmed$. O algoritmo continua enquanto $Tmax$ for maior que $Tmin$. O supervisor final encontrado é o escalonamento com o menor tempo de execução para todas as tarefas e que respeita os prazos individuais dos produtos.

3.6 Otimização do supervisor para tempo mínimo por produto

Após obtermos o supervisor que respeite os prazos de entrega de todos os produtos na seção 3.4 através do algoritmo 1 e em seguida, um supervisor que, além disso, impõe a restrição de tempo mínimo para todas as tarefas na seção 3.5 (algoritmo 3), o objetivo específico deste passo, é minimizar o *lead time* de produtos prioritários. Para esse fim, a planta inicial deste algoritmo é igual ao último supervisor encontrado na seção 3.5 com o algoritmo 3 (*linha 1*). Entende-se que minimizar o *lead time* dos produtos, significa reduzir o período de tempo entre o instante em que o produto pode iniciar suas operações até efetivamente o instante final de suas operações, ou seja, até o instante para ele ficar pronto.

Os modelos dos autômatos das especificações para esta etapa seguem a lógica da figura 29, ou seja, são especificações de prazo de entrega por produto. Para este algoritmo são encontrados supervisores a partir das especificações temporais de prazo reduzido para produtos prioritários.

1	Planta = Supervisor
2	Para $i = A, B, \dots, P$
3	Enquanto $(T_{max_i} > T_{min_i})$
4	$T_{med_i} = \text{Quociente}(T_{max_i} + T_{min_i}) / 2$
5	Supervisor $T_i = \text{SupCon}(\text{Planta}, S_{Temp_i}(T_{med_i}))$
6	Se Existe(Supervisor T_i)
7	$T_{max_i} = T_{med_i}$
8	Planta = Supervisor T_i
9	Senão
10	$T_{min_i} = T_{med_i}$
11	Fim Se
12	Fim Enquanto
13	Fim Para
14	Supervisor = Planta

Algoritmo 4: Redução do *lead time* para produtos prioritários

Este algoritmo é inspirado também no método da bissecção como em 3.5. O valor do limite superior do tempo inicial para o produto prioritário de índice i (T_{max_i}) é o seu *lead time* encontrado no supervisor do algoritmo 3 da seção 3.5. Esse valor poder ser o seu próprio prazo de entrega ou pode ser um valor menor, isso depende, se a redução do tempo global provocada pelo algoritmo 3 da seção 3.5 tenha ou não afetado diretamente o produto prioritário em questão.

O valor do limite inferior inicial para o produto prioritário de índice i ($Tmin_i$) será igual ao somatório dos tempos de produção deste produto nos recursos. Considera-se para isso, que todos os recursos estão dedicados para este produto somente. Caso o valor de $Tmax_i$ inicial e $Tmin_i$ inicial sejam iguais, o algoritmo não precisa ser executado para este produto, pois não há como melhorar o seu *lead time*. Para encontrar este supervisor, propõe-se aplicar as restrições temporais de prazo reduzido $Stemp_i$ para cada produto prioritário de índice $i = A, B, \dots, P$ (linha 5).

Na linha 3 é verificado se o $Tmax_i$ é maior que $Tmin_i$, se não for, então não precisa calcular $Tmed_i$ e não entra no cálculo. Caso contrário, as especificações temporais de prazo por produto sofrem alterações sucessivas no valor do seu tempo conforme Linhas 3 a 12. Nessas linhas do algoritmo, é verificada a possibilidade de encontrar um supervisor com uma especificação temporal de valor $Tmed_i$ para o produto prioritário de índice i . Se existir um supervisor $SupervisorT_i$, então o algoritmo continua e $Tmax_i$ recebe o valor do último $Tmed_i$ calculado, caso contrário, $Tmin_i$ recebe o valor de $Tmed_i$ e o algoritmo continua para este produto enquanto $Tmax_i$ for maior que $Tmin_i$. O supervisor final encontrado é o escalonamento com o menor tempo de execução para todas as tarefas e também com o menor *lead time* para os produtos prioritários.

3.7 Escolha de uma cadeia de eventos no supervisor

Como o supervisor, por definição, é o controle menos restritivo possível, podem existir diferentes sequências de eventos no supervisor que levariam a um estado marcado. Desta forma, diferentes escalonamentos de tarefas, que respeitem todas as restrições impostas, ainda seriam possíveis, deve-se então, escolher um desses caminhos para gerar o escalonamento. A escolha deste é feita da seguinte maneira: a partir do estado de origem inicial, verificam-se quais são as possíveis transições de saída, se existir mais de uma, escolhe-se a primeira. O estado de destino alcançado com a transição escolhida passa a ser o estado de origem e este processo continua até que não seja mais possível, encontrar um estado de origem, a partir do último estado de destino encontrado.

Quando há opção de escolha de diferentes transições durante este processo, regras heurísticas podem ser desenvolvidas para escolher qual evento é o mais indicado, por exemplo, regras para produtos prioritários, produtos com setups reduzidos, produtos com

tempos de produção mais curtos, de menores datas de entrega, maiores valores de venda, etc. Muitas vezes, como todos os objetivos para o escalonamento já são alcançados a priori com o indicador escolhido, o primeiro evento é suficiente e a escolha por outro não é relevante.

3.8 Algoritmo para geração do escalonamento

A partir da escolha de uma cadeia de eventos explicada na seção anterior 3.7, é necessário traduzir esta cadeia de eventos, em tarefas distribuídas no tempo. Para este fim foi desenvolvido o algoritmo desta seção.

Como descrito anteriormente, esta abordagem modelou apenas um evento de início de tarefa em cada recurso. Diante dessa premissa é necessário fazer a relação entre a ocorrência de um evento de fim de tarefa com seu respectivo início. Esta informação não está diretamente disponível na cadeia de eventos escolhida, mas seguramente pode ser obtida e se torna necessária para registrar os momentos de início e fim dos eventos. Para cada evento da cadeia escolhida, existe à informação de qual é o seu respectivo evento de início de operação. Por exemplo, se um evento qualquer já é um evento de início, então o nome do evento e de seu respectivo evento de início de operação são iguais. Tomamos como exemplo, o autômato da figura 20 (máquina M4). O evento IM4 é o evento de início de operação para qualquer produto nesta máquina, então para qualquer evento da máquina M4 está associado à informação do evento de início de operação igual a IM4. A partir desta associação, será possível identificar o exato momento de início e fim das operações.

No vetor chamado de *Escalonamento* está armazenado a cadeia escolhida do supervisor encontrado no algoritmo 4, ou seja, é um vetor que contém os eventos e estados de apenas uma cadeia, que parte do estado inicial e alcança um estado marcado no supervisor final. Como o vetor de *Escalonamento* é originário de um supervisor, por definição, ele está ordenado de maneira que respeita a sequência lógica da ocorrência dos eventos. Quando tomamos, por exemplo, um evento de fim de operação no vetor *Escalonamento* e, fazemos uma busca deste índice do vetor para trás em direção ao índice igual a 1 (*linha 2*) e, encontramos o primeiro evento de início de operação que se relaciona com este evento de fim, anota-se em uma variável do vetor *Escalonamento* de índice (*i*), o índice (*j*) do evento de início de operação. Da *linha 0* até a *linha 8* do algoritmo 5 é feita a indexação entre os

eventos de fim de operação com seus respectivos eventos de início de operação através cadeia de eventos escolhida Essa etapa (*linha 0 até a linha 8*) foi desenvolvida com objetivo de facilitar a atribuição dos valores da variável Relógio para as variáveis de Início e Fim das operações no vetor Escalonamento.

A variável Relógio tem a responsabilidade de assumir valores dos dias e horas que são acumuladas a partir de uma data escolhida inicial (*Linha 10*), essa data corresponde ao início de execução das tarefas, geralmente em ambientes reais de produção é o início de um turno de trabalho. O Relógio muda de valor a cada ocorrência do evento tick no supervisor (*Linha 14*). No algoritmo, a cada ocorrência deste evento, é somado o valor de 10 minutos ao Relógio. Esse acréscimo, no caso de 10 minutos, é o valor escolhido para a duração do evento tick deste problema (5/4/J/PP_TM_RL). O tick pode assumir qualquer valor, mas deve guardar relação com o ambiente produtivo do escalonamento, esse parâmetro é uma decisão de modelagem.

```

0      Para  $i = 1, \dots, n$ 
1          Se Escalonamento (i) Evento  $\neq$  Eventos Iniciais de operação e Tick
2              Para  $j = i, \dots, 1$  Passo = -1
3                  Se Escalonamento (i) Evento Inicial de Op = Escalonamento (j) Evento
4                      Escalonamento (i) Índice de Início Op. = j
5                  Fim Se
6              Fim Para j
7          Fim Se
8      Fim Para i
9
10     Relógio = Data qualquer escolhida para início
11
12     Para  $i = 1, \dots, n$ 
13         Se Escalonamento (i) Evento = Tick
14             Relógio = Relógio + Valor da duração do evento tick (No caso = 10 min)
15         Caso Contrário
16             Se Escalonamento (i) Evento = Evento de Início de Operação
17                 Escalonamento (i) Hora de Início de Operação = Relógio
18             Caso Contrário
19                 Escalonamento (i) Hora de Fim de Operação = Relógio
20             Fim Se
21         Fim Se
22     Fim Para i

```

Algoritmo 5: Geração do escalonamento

Entre as *linhas 12 e 22*, é verificado qual o tipo de evento (tick (*linha 13*), Evento de

Início (*linha 16*) ou Evento de Fim de Operação (*linha 19*) da ocorrência e, para cada um deles é executada uma atribuição específica de valor.

Quando o evento tick ocorre, como já descrito, o valor da variável Relógio é aumentado de uma unidade de tempo. Quando ocorrem eventos do tipo Início de Operação, a variável Hora de Início de Operação recebe o valor atual do Relógio e por último, para eventos do tipo Fim de Operação, a variável Hora de Fim de Operação recebe o valor atual do Relógio. Com as atribuições de horários aos eventos, é possível montar as tarefas ao longo do tempo e mostrar o resultado na forma do gráfico de Gantt.

3.9 Resultados alcançados

3.9.1 Gráfico de Gantt

O gráfico de Gantt é utilizado como uma ferramenta de planejamento e controle da produção, ele foi desenvolvido pelo engenheiro Henry Gantt em 1917. Neste gráfico, cada atividade ou operação é representada por uma barra de comprimento correspondente a sua duração. No eixo horizontal está a linha do tempo e no eixo vertical estão os recursos envolvidos.

A figura 30 apresenta o gráfico de Gantt obtido a partir da escolha de uma sequência de eventos do supervisor final encontrado. Esta figura representa um escalonamento dentre os vários possíveis encontrados no supervisor final. Para a construção do Gantt, a informação da data de início das atividades deve ser informada pelo planejador da produção, esse valor é um dado de entrada, um parâmetro que deve ser informado. Podemos notar no Gantt, todas as operações dos produtos distribuídas ao longo do tempo. No eixo vertical estão os recursos utilizados: M1, M2, M3 e M4 e no eixo horizontal está a escala do tempo. Os nomes P1, P2, P3, P4 e P5 (pedidos dos produtos) representam os produtos A, B, C, D e E respectivamente.

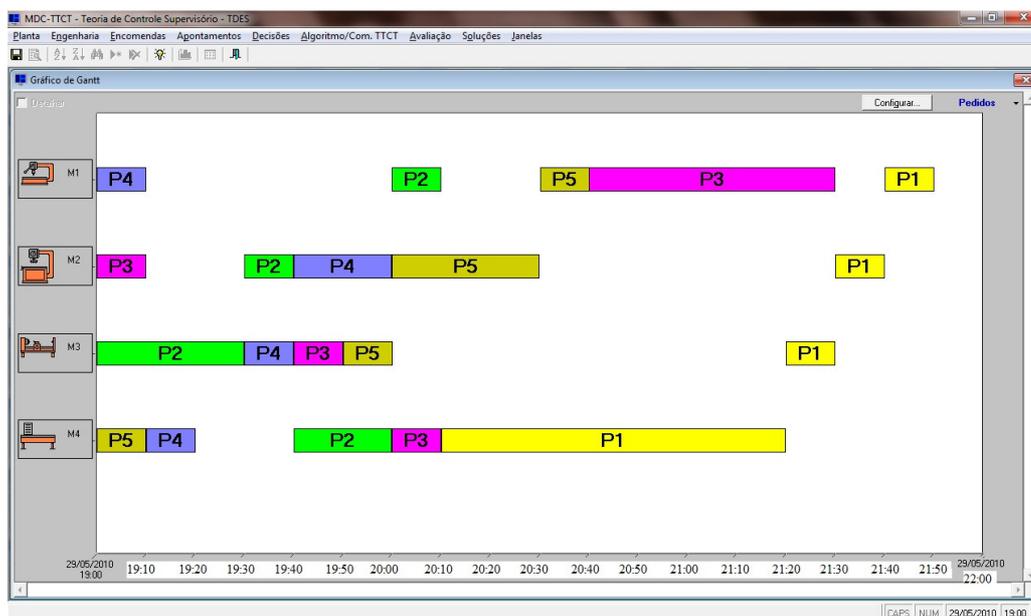


Figura 30: Gráfico de Gantt

O resultado representa o comportamento ótimo do escalonamento com relação ao indicador PP_TM_RL. Foi encontrado com este escalonamento, o menor tempo possível de execução para completar todas as tarefas nos seus respectivos prazos e com os menores *lead times* para produtos prioritários.

3.9.2 Comparação entre os métodos de síntese

Os números de estados dos supervisores intermediários podem ser vistos na tabela 5. Como descrito na seção 2.3.4, não foi possível encontrar uma solução para o problema 5/4/J/F com a proposta de síntese de BW (1992). Com o objetivo então, de comparar as duas abordagens pelo aspecto de modelagem dos autômatos, é utilizada a síntese do algoritmo 1 da seção 3.4 em ambas as modelagens. Os supervisores intermediários mostrados na tabela 5 são expressos da seguinte maneira: O supervisor A garante a especificação do prazo de entrega do produto A e do roteiro de produção de A, o supervisor B garante a especificação do prazo de entrega do produto B e do roteiro de produção B, desta maneira, o Supervisor i garante a especificação do prazo de entrega do produto i e do roteiro de produção de i ($i = A, B, \dots, P$).

Tabela 5: Quantidades de estados

Autômato	Número de estados com Abordagem da pesquisa	Número de estados com Abordagem de Brandin e Wonham
Planta temporizada	1.575	36.960
Supervisor E	44.029	662.067
Supervisor B	24.515	1.245.475
Supervisor D	31.913	<i>out of memory</i>
Supervisor C	44.822	X
Supervisor A	9.809	X
Resultado do Algoritmo 1	9.809	X
Resultado do Algoritmo 3	2.165	X
Resultado do Algoritmo 4	363	X

Com os modelos dos autômatos de Brandin e Wonham (1992), o sistema TTCT não foi capaz de executar todas as sínteses necessárias. A partir do resultado do Supervisor B, foi verificado que o sistema não pôde seguir adiante em função do tamanho dos autômatos envolvidos. Para evitar esse problema, propõem-se as modificações apresentadas nos modelos das máquinas e das especificações. O objetivo de trabalhar com 5/4/J/F foi comprovar que para esta dimensão de problema não é mais possível obter resultados com a abordagem de Brandin e Wonham (1992).

A síntese incremental, além de ser um diferencial do método proposto para inibir o crescimento dos supervisores, facilita o entendimento dos resultados obtidos para o problema de escalonamento, pois trata os produtos e seus respectivos prazos de forma isolada e sequencial, possibilitando assim, avaliações intermediárias durante o processamento do algoritmo sem alteração do resultado global desejado. Esta análise sugere também que a abordagem desenvolvida é mais eficiente para tratar o problema de escalonamento com a TCS. O tamanho da planta e o crescimento do número de estado se mantiveram expressivamente mais comportados, constatando que é compensador utilizar os algoritmos e a modelagem proposta dos autômatos para o problema de escalonamento. No próximo capítulo, este método será aplicado a um estaleiro de reparos navais.

CAPÍTULO 4

Escalonamento das atividades de um estaleiro de reparo naval

Com o objetivo de aprofundar o método proposto, este capítulo apresenta uma aplicação real da metodologia através de um estaleiro de reparo naval. Será mostrada passo a passo a modelagem dos recursos, das operações e das obras do estaleiro e ao final do capítulo serão apresentados os resultados obtidos.

Para esta aplicação, o estaleiro tem dez pedidos em carteira. Cada pedido contém uma lista de serviços que deve ser realizada nas embarcações. O objetivo desta aplicação é executar os serviços para todas as embarcações nos prazos combinados com os clientes, ou seja, ser pontual. Além disso, devem-se executar também todas as atividades no menor tempo possível e tentar reduzir o *lead time* dos pedidos. Seguem nas próximas seções, a descrição do estaleiro, os recursos e as obras do estaleiro que a pesquisa aborda para gerar o escalonamento.

4.1 Descrição do estaleiro

O estaleiro da pesquisa é o maior estaleiro de reparo naval do hemisfério sul. O estaleiro tem cinco diques: três flutuantes e dois secos. Como os diques são os recursos mais nobres, pois são os mais caros, os mais restritos de um estaleiro de reparo e porta de entrada de receitas, quanto menor for o tempo de permanência de um navio no dique, maior será a rotatividade dos serviços e, conseqüentemente, maior o lucro para a empresa.

Na figura 31, é apresentado o layout do estaleiro e através dele é possível identificar a posição dos três diques flutuantes, dois diques secos e as diversas oficinas. Cada dique tem seus guindastes de apoio e estes podem estar localizados em sua própria estrutura ou nos cais. Os guindastes apóiam diretamente quase todas as equipes de trabalho, que se deslocam pelo estaleiro. Cada cor da legenda da figura 31 representa a capacidade de içagem de carga dos guindastes. O estaleiro, além dos diques, guindastes e

equipes de trabalho, tem uma série de outros recursos produtivos que estão distribuídos ao longo do seu terreno. Alguns recursos estão alocados e agrupados em centros de custos. O agrupamento é feito, como na maioria dos casos de produção sob encomenda, de acordo com o processo. Os recursos que fazem operações de usinagem estão no centro de usinagem; os recursos referentes à tubulação e válvulas estão próximos entre si; as máquinas de corte de plasma têm um centro específico, e assim por diante.

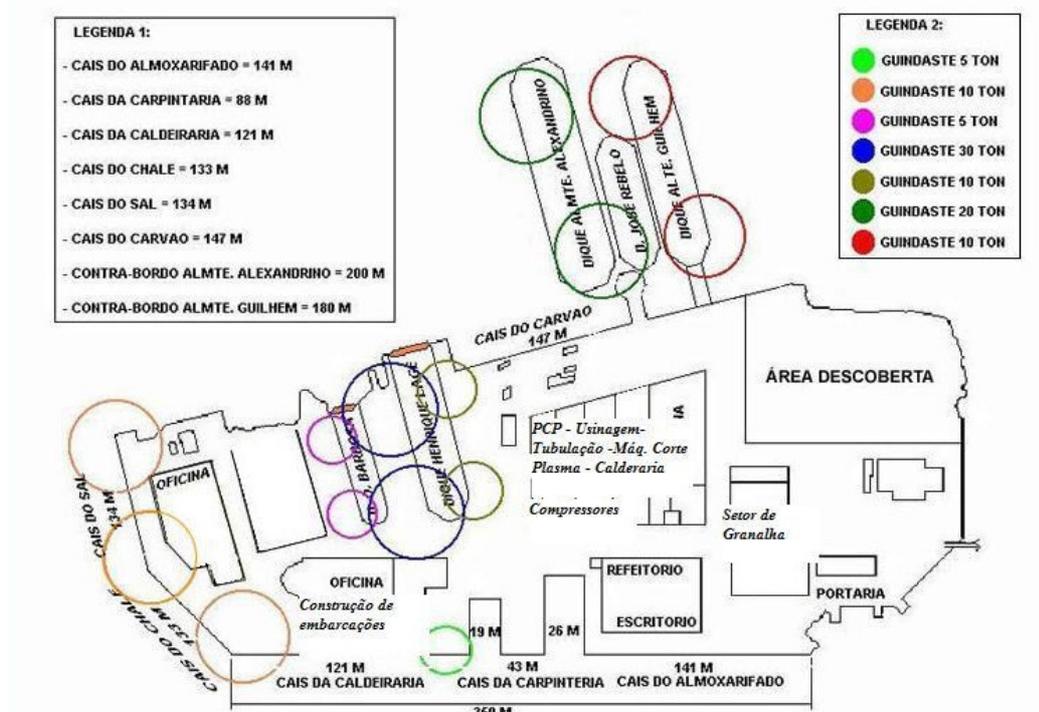


Figura 31: Layout do estaleiro

Para o estaleiro, diferentemente do centro de trabalho apresentado nos capítulos 2 e 3, utiliza-se o conceito de obra ao invés de produto. Os prazos de entregas estão vinculados às obras e cada uma delas é identificada através de um código de pedido.

4.1.1 Recursos

Embora um setor de atividades de grande porte, como é o caso do estaleiro, pode ter parte de seu arranjo físico agrupado por processo (caso das oficinas), o arranjo físico posicional domina. Os recursos são deslocados até os produtos - navios e estruturas -

durante o processo de fabricação, e não o contrário. Assim, os principais recursos em uma obra de reparo naval são as equipes de trabalho e suas ferramentas: os soldadores, os mecânicos, os pintores, o pessoal de controle de qualidade, entre outros

Os critérios para a escolha dos recursos modelados foram: i) os que executam serviços críticos, ii) os de maior demanda, iii) os que apresentam capacidade limitada e iv) os que são compartilhados entre as obras. Recursos que são dedicados não são necessários para a modelagem, já que não há disputa por alocação de tarefas.

Foi percebido durante a pesquisa, que as equipes de trabalho são os principais recursos para o problema no estaleiro. O grande trabalho de modelagem estaria na alocação de equipes, com suas respectivas operações para as obras. Todas as oficinas presentes no estaleiro (usinagem, mecânica, tubulação, etc.) atenderiam bem a necessidade para todos os diques, esses recursos têm capacidade suficiente para executar as demandas que são solicitadas (essa afirmação é fundamentada através de duas reuniões realizadas no estaleiro com os próprios responsáveis pelo escalonamento - Novembro de 2009 e Janeiro de 2010). O *gargalo* é o trabalho feito pelas equipes no navio, são elas que determinam o tempo de permanência das embarcações nos diques.

Como há disponibilidade de máquinas e ferramentas para as equipes de acordo com o número de operadores, ou seja, não existe operador sem máquina ou ferramenta, então, pode-se constatar que as equipes são as restrições. Caso existisse um operador sem máquina ou ferramenta, dever-se-ia modelar também estes recursos, pois se transformariam em restrições, já que haveria disputa dos operadores por sua disponibilidade. Do ponto de vista de modelagem, nada se alteraria do que será apresentado, somente existiria a adição de mais recursos. Para o estaleiro, o número de obras e o número de recursos envolvidos no problema, segundo a notação de French (1982) serão de 10/5/J/PP_TM_RL. Desta maneira têm-se $6,3 \times 10^{32}$ possíveis soluções de escalonamento. Abaixo, na tabela 6, são apresentadas as equipes e operações que foram modeladas para o estaleiro.

Tabela 6: Equipes e operações do estaleiro

Recursos	Operações
Equipe de Hidrojato	Hidrojato
Equipe de Pintura	Pintura
Equipe de Reparo Estrutural	Reparo Estrutural
Equipe de Controle de qualidade	Controle de qualidade
Equipe de Mecânica	Mecânica

Todas as equipes mencionadas trabalham com suas respectivas ferramentas, são elas: bombas de hidrojatos, bombas de jato de granalha, máquinas de solda, bombas de pintura, refletores, exaustores, macacos, bombas hidráulicas, compressores, entre outras. Também são encontrados nos estaleiros de reparo os recursos de transporte tais como: empilhadeiras e carretas.

Para as operações (hidrojato, pintura, controle de qualidade, mecânica e reparo estrutural), a modelagem foi feita de forma agregada. Cada uma delas tem uma quantidade grande de micro atividades (serviços) e não é objetivo da modelagem trabalhar de forma detalhada, mesmo porque, existe muita flexibilidade operacional no “chão de fábrica” do estaleiro e o importante é organizar as operações, deixando com as equipes as decisões detalhadas no dia a dia. A quantidade atual de serviços do estaleiro está em torno de 15.000, esse número mostra a grande flexibilidade deste setor. Segue na tabela 7, um exemplo de serviço para cada operação.

Tabela 7: Exemplo de serviços

Reparo Estrutural	Substituir chapeamento externo em aço ASTM - 131- A ou similar por peça.
Mecânica	Madre do Leme - Retirar e montar a madre do Leme.
Pintura	Costado - Pintura com tinta (exceto as do tipo autopolimento) e solvente fornecidos pelo Armador. Área contínua, com intervalos entre demãos até 6 horas.
Hidrojato	Fundo - Jatear com água doce alta pressão com 200 bar.
Controle de	Chapeamento - Inspeccionar com teste de líquido penetrante

Qualidade	quando solicitado pelo Armador.
-----------	---------------------------------

O objetivo da agregação foi organizar e agrupar a grande quantidade de serviços em macro operações para que fosse possível executar o escalonamento em um nível gerencial. Com o escalonamento de mais alto nível gerado, cada operação pode ser desmembrada em serviços, e os responsáveis pela execução operacional podem ter flexibilidade na organização destes. Entretanto, deverão executar suas tarefas de maneira a atender o escalonamento, respeitando os horários de início e fim de trabalho das equipes conforme o plano do escalonamento.

4.1.2 Obras do estaleiro

Nesta seção serão apresentadas as demandas de serviços inspirados em casos típicos do estaleiro. São dez obras diferentes contratadas, uma para cada navio. Seguem a seguir na tabela 8, as dez obras numeradas (P1, P2,..., P10) conforme a chegada de pedidos, com seus respectivos prazos de entregas.

Tabela 8: Obras do estaleiro

Obra	Navio	Cliente	Descrição do serviço	Prazo em semanas
P1	Navio Petroleiro Barao de Maua O.S1	PETROBRÁS	ATRACAÇÃO PARA REPARO ESTRUTURAL DE AVARIA DE COSTADO, HIDROJATO E PINTURA E MECÂNICA	12
P2	Navio Petroleiro Presidente Juscelino O.S1	PETROBRÁS	REPARO ESTRUTURAL DE ROTINA, HIDROJATO, PINTURA E MECÂNICA	8
P3	Navio Supply SC Lancer O.S1	SCHAHIN PETRÓLEO	ATRACAÇÃO PARA REPARO ESTRUTURAL DE AVARIA DE BULBO, HIDROJATO, PINTURA E PINTURA	10
P4	Navio Petroleiro Jose do Patrocinio O.S2	PETROBRÁS	REPARO ESTRUTURAL NOS TANQUES DE LASTRO, HIDROJATO, PINTURA E MECÂNICA	14
P5	Navio Petroleiro Lambari O.S1	PETROBRÁS	MECÂNICA, REPARO ESTRUTURAL, HIDROJATO E PINTURA	8
P6	Navio de Pesquisa	BRAX	MECÂNICA, REPARO	24

	Scan Nordic O.S1	SHIPHOLDING	ESTRUTURAL, HIDROJATO E PINTURA	
P7	Navio Graneleiro Mv Sky L O.S1	ANANGEL BRIGHT COMPANIA NAVIERA S.A.	REPARO ESTRUTURAL DE TANQUES DE LASTRO, MECÂNICA, HIDROJATO E PINTURA	16
P8	Navio Graneleiro Navision Bulker O.S1	NAVISION BULKER COMPANY SA	MECÂNICA, REPARO ESTRUTURAL DE AVARIA DE FUNDO, HIDROJATO E PINTURA	20
P9	Navio de Pesquisa	BRAX SHIPHOLDING	MECÂNICA, REPARO ESTRUTURAL NA POPA, HIDROJATO E PINTURA	26
P10	Navio Supply	BRAX SHIPHOLDING	REPARO ESTRUTURAL NO DECK, HIDROJATO, PINTURA E TROCA DE EIXO	23

Na tabela 8, por exemplo, o prazo de entrega do navio petroleiro Barão de Mauá (P1) que sofreu avaria no costado é de doze semanas e necessita de serviços de reparo estrutural, de pintura, hidrojato, controle de qualidade e de mecânica.

No problema apresentado, todas as dez obras desta aplicação, pelas suas características, têm a necessidade de se posicionar em lugar seco para os serviços contratados. Como o estaleiro tem cinco diques, alguns deles estão com mais de um navio ao mesmo tempo, essa situação não é rara e é boa para a lucratividade da empresa, em compensação, o gerenciamento das operações se torna ainda mais complicado e as operações de docagem e desdocagem se tornam mais complexas.

Em função dos tempos reais de execução dos serviços no estaleiro, o valor escolhido para a modelagem do evento tick é igual a uma semana (7 dias). Na tabela 9 são apresentadas as durações das atividades para as dez obras. Os valores apresentados estão em semanas ou em ticks, ou seja, o valor 1 na tabela significa uma semana de trabalho, o valor 2, duas semanas e assim por diante.

Tabela 9: tempos de operações das obras

Recursos/Obras	Tempos das obras em semanas										TT
	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	
Hidrojato	1	1	1	2	1	3	1	1	2	3	16
Mecânica	2	3	5	1	1	1	1	3	2	3	22
Cont.Qualidade	1	1	1	1	1	1	1	1	1	1	10
Pintura	1	1	1	3	2	2	1	1	2	3	17
Rep. Estrutural	1	1	1	2	1	7	1	1	3	4	22
Total	6	7	9	9	6	14	5	7	10	14	87

Como exemplo, a obra P6 realiza a operação de hidrojato em três semanas, a de mecânica em uma semana, controle de qualidade também em uma semana, pintura em duas semanas e o reparo estrutural em sete semanas. É importante ressaltar, que as ordens de execução de cada obra (roteiro das obras) estão descritas na tabela 10 mais abaixo. Na tabela 9 são apresentadas somente as durações das atividades, o tempo total das atividades por operação e totais por obra para exemplificar a ordem de grandeza do tempo necessário para o escalonamento do estaleiro.

Tabela 10: Roteiro das Obras

Obra	Ordem				
	Tarefa 1	Tarefa 2	Tarefa 3	Tarefa 4	Tarefa 5
P1	Rep. Estrutural	Hidrojato	Pintura	Cont.Qualidade	Mecânica
P2	Mecânica	Hidrojato	Pintura	Cont.Qualidade	Rep. Estrutural
P3	Rep. Estrutural	Hidrojato	Pintura	Cont.Qualidade	Mecânica
P4	Hidrojato	Pintura	Cont.Qualidade	Mecânica	Rep. Estrutural
P5	Rep. Estrutural	Mecânica	Hidrojato	Pintura	Cont.Qualidade
P6	Hidrojato	Pintura	Mecânica	Rep. Estrutural	Cont.Qualidade
P7	Hidrojato	Pintura	Cont.Qualidade	Mecânica	Rep. Estrutural
P8	Rep. Estrutural	Hidrojato	Pintura	Cont.Qualidade	Mecânica
P9	Mecânica	Hidrojato	Pintura	Cont.Qualidade	Rep. Estrutural
P10	Rep. Estrutural	Hidrojato	Pintura	Cont.Qualidade	Mecânica

Na seção seguinte serão descritos os autômatos dos recursos e dos roteiros das obras do estaleiro.

4.2 Aplicação do método ao estaleiro

Os nomes dos eventos para os recursos são definidos da seguinte forma: os eventos de início de operação começam com a letra I (início) e são adicionados pela letra correspondente aos recursos. O evento de início de operação no reparo estrutural é o **IR**, I + R (reparo estrutural), para o controle de qualidade **IC**, para o hidrojato **IH**, para a equipe de mecânica **IM** e para a pintura **IP**. Para os eventos de fim de operação, deve ser feita a diferenciação entre as obras e por isso, os nomes dos eventos de fim ficam concatenados da seguinte maneira: F (Fim) + Código da Obra + Recurso, por exemplo, o fim da operação da obra P10 no reparo estrutural é o **FP10R**, no controle de qualidade **FP10C**, no hidrojato **FP10H** e assim por diante para todas as obras.

4.2.1 Autômatos dos recursos

O autômato da figura 32 representa a equipe de reparo estrutural (R), o evento **IR** corresponde ao início do trabalho e os outros eventos, com exceção do tick (t), estão relacionados ao fim de trabalho de reparo estrutural para cada obra.

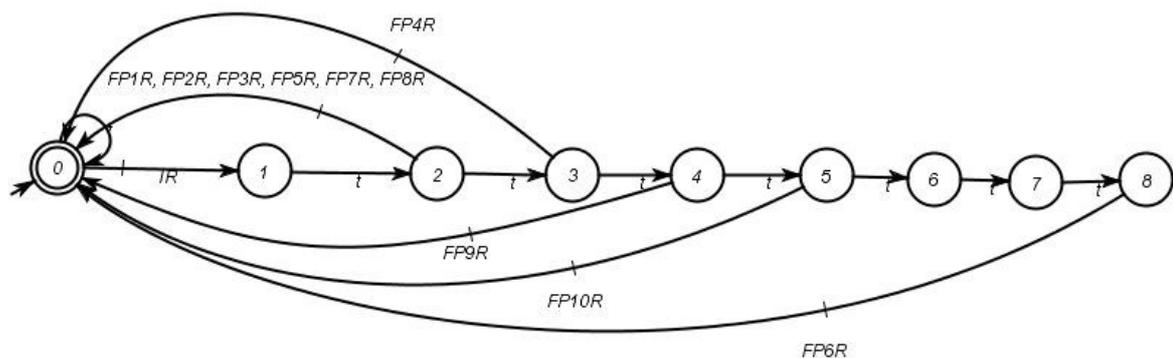


Figura 32: Equipe de Reparo Estrutural

Como pode ser visto no autômato da figura 32, para as obras contratadas, o máximo de tempo em uma operação de reparo estrutural será de sete semanas. Desta forma, este período define o tamanho de estados do autômato. Se adicionarmos, por exemplo, quaisquer outros pedidos que necessitem de reparo estrutural e que seus respectivos tempos

estimados nesta operação não excedam sete semanas, o número de estados deste autômato não se altera e, conseqüentemente, o tamanho da planta permanece o mesmo.

A equipe de reparo estrutural, dentre as equipes modeladas, é geralmente a que apresenta a maior utilização no estaleiro. Existem muitos serviços no costado, convés, tanques, superestrutura e nas inúmeras estruturas internas do navio que consomem bastante tempo. O navio é docado somente quando há necessidade de retirá-lo de dentro d'água e isso acontece, quando o trabalho exige algum reparo na estrutura externa ou por manutenção preventiva determinada por órgãos classificadores.

Para a equipe de controle de qualidade, que possui um tempo médio de inspeção de uma semana para cada obra, tem-se o modelo de autômato da figura 33.

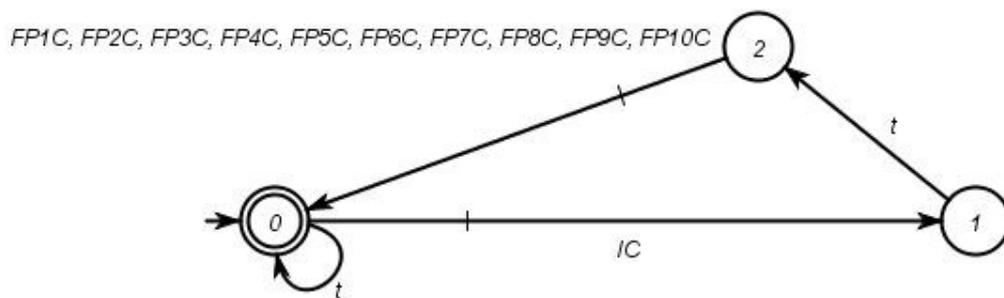


Figura 33: Equipe de Controle de Qualidade

O controle de qualidade é iniciado através da ocorrência do evento IC, e é finalizado, dependendo da obra, com os eventos FP1C, FP2C, FP3C, FP4C, FP5C, FP6C, FP7C, FP8C, FP9C, FP10C.

A operação de hidrojato (H) prepara a área que receberá a pintura. As chapas de aço dos navios devem estar limpas e adequadas antes de receberem a tinta durante o processo de pintura. Para a equipe de hidrojato foi modelado o autômato da figura 34.

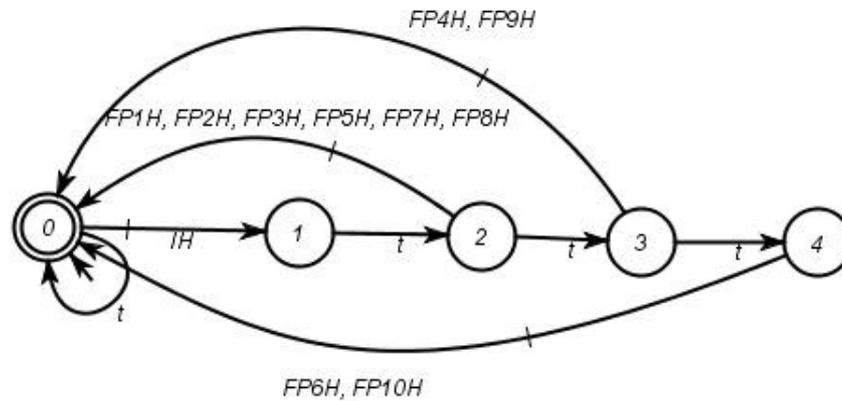


Figura 34: Equipe de Hidrojato

A equipe de mecânica executa serviços, em sua grande maioria, nas peças de equipamentos mecânicos, na remoção do hélice, nos sistemas de propulsão, nos impelidores laterais e leme. O autômato da figura 35 modela a equipe de mecânica.

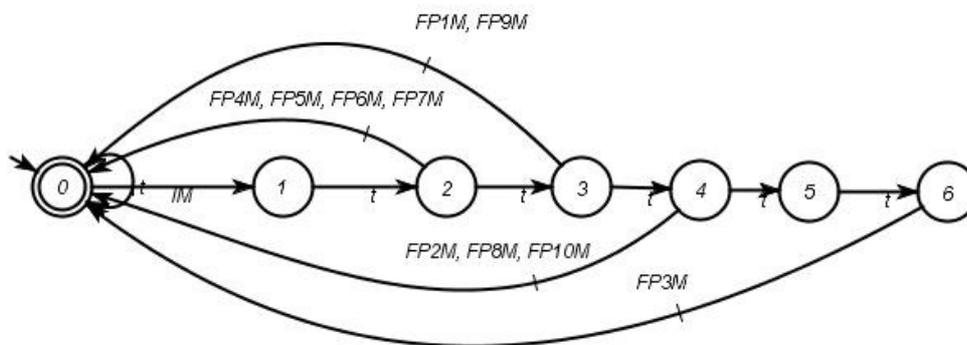


Figura 35: Equipe de Mecânica

Por último, a figura 36 modela a equipe de pintura. O maior percentual do tempo de trabalho desta equipe é concentrado no costado do navio. A tinta da pintura na maioria dos casos serve como proteção contra corrosão e acúmulo de impurezas.

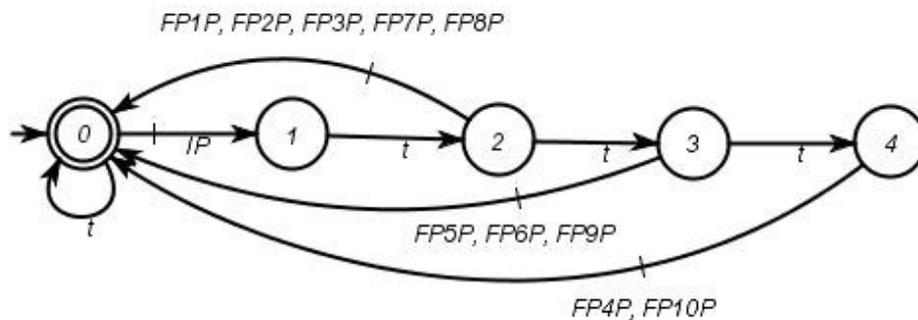


Figura 36: Equipe de Pintura

4.2.2 Autômatos dos roteiros das obras

A tabela 10 na seção 4.1.2 descreve os roteiros das obras e os autômatos desta seção foram construídos com as informações contidas nesta tabela. A figura 37 mostra o autômato do roteiro da obra P1.

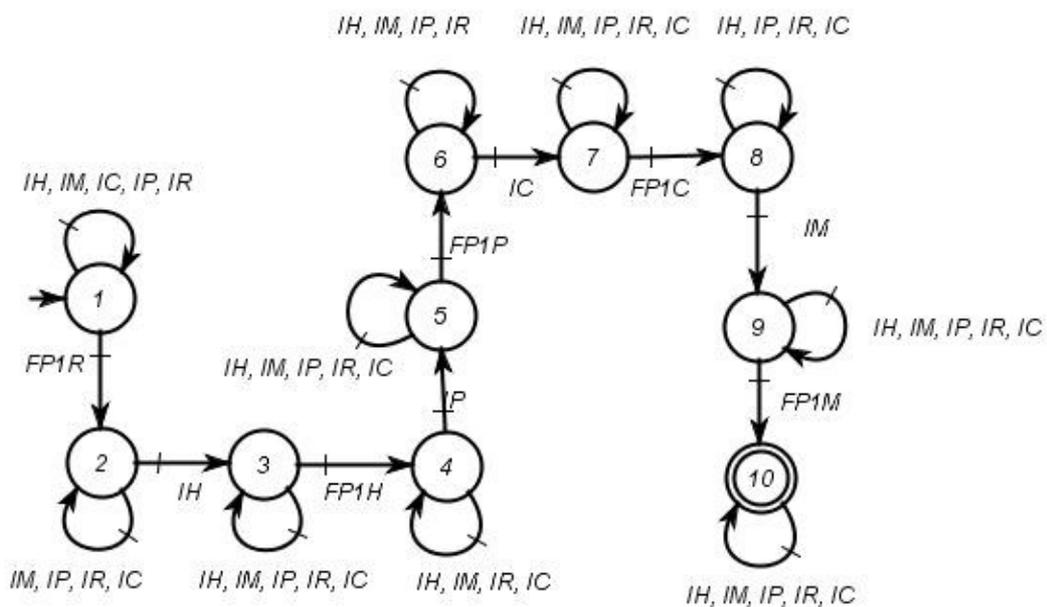


Figura 37: Obra P1- Navio Barão de Mauá

Uma característica do autômato da figura 37 é que em todos os estados, os eventos de início de tarefas estão habilitados (IR, IH, IP, IC e IM). Esta modelagem possibilita que

outra obra possa usar os recursos compartilhados em momentos diferentes. O autômato acima representa a sequência de operações necessárias para fazer a obra do Navio Petroleiro Barão de Mauá mencionada na seção 4.1.1. A sequência de tarefas é:

Reparo Estrutural → Hidrojato → Pintura → Controle de Qualidade → Mecânica.

Seguem abaixo os autômatos para as outras nove obras. A modelagem destes segue a mesma lógica do autômato apresentado na figura 37.

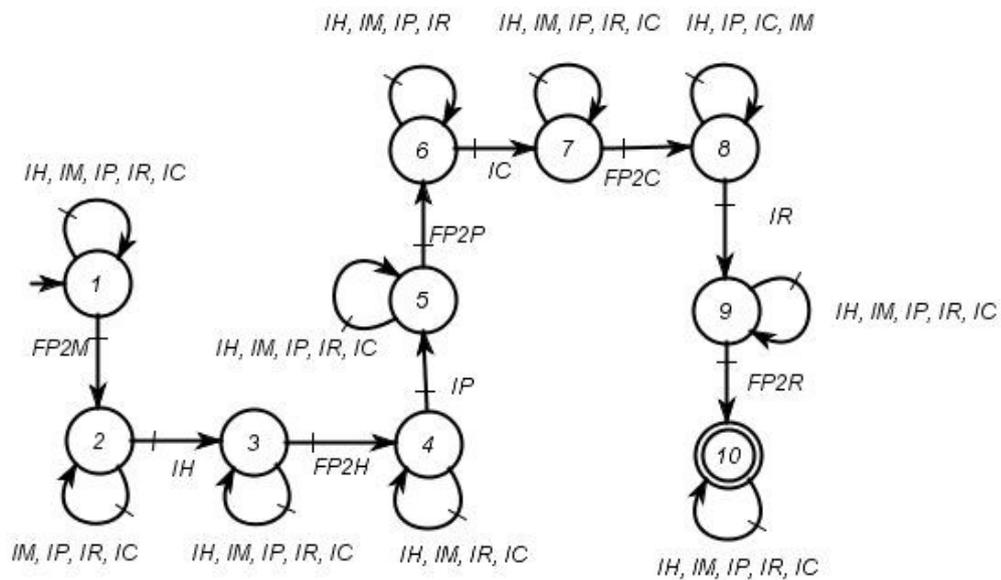


Figura 38: Obra P2- Navio Presidente Juscelino

P2: Mecânica → Hidrojato → Pintura → Controle de Qualidade → Reparo Estrutural.

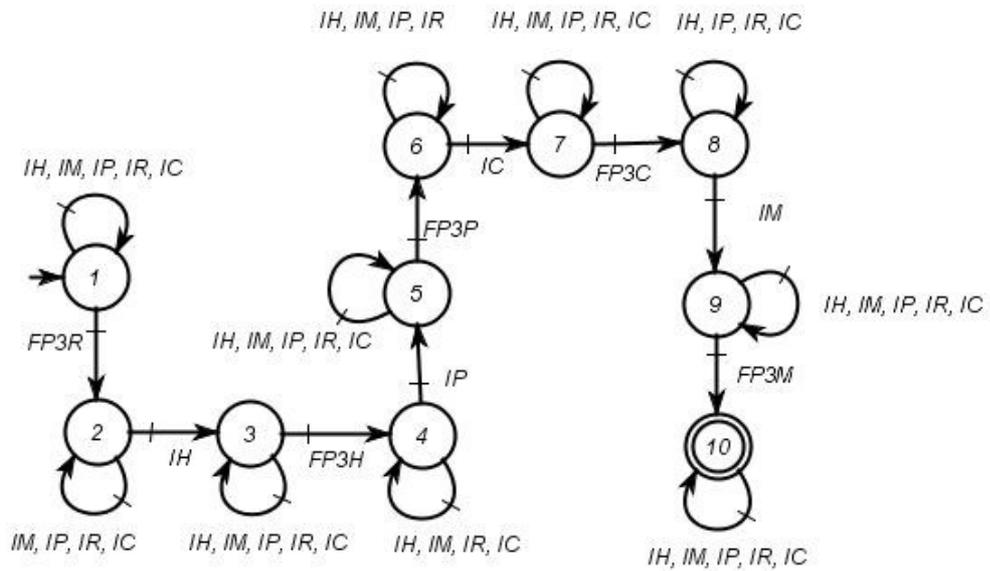


Figura 39: Obra P3- Navio Supply SC Lancer

P3: Reparo Estrutural → Hidrojato → Pintura → Controle de Qualidade → Mecânica

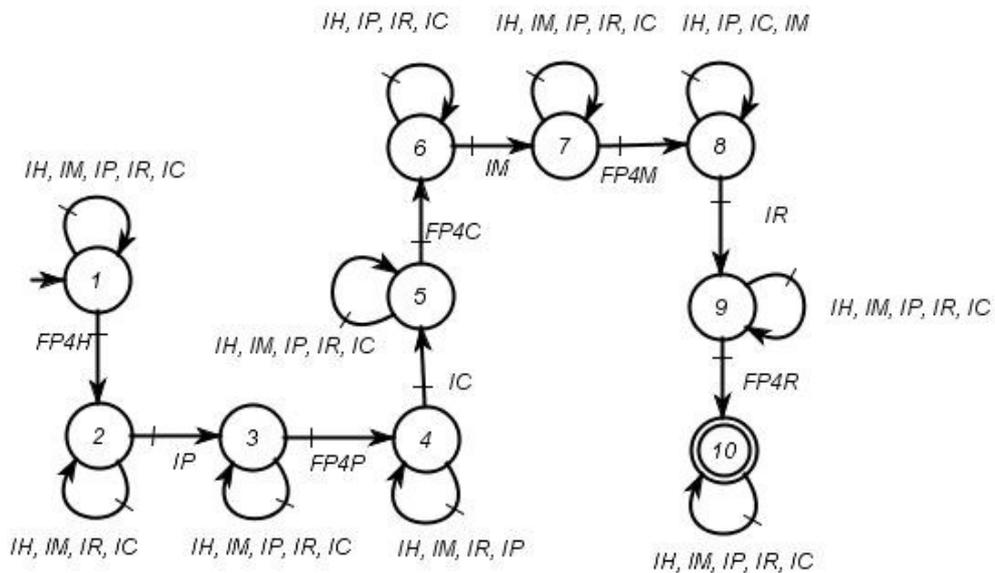


Figura 40: Obra P4 - Navio Petroleiro Jose do Patrocínio

P4: Hidrojato → Pintura → Controle de Qualidade → Mecânica → Reparo Estrutural

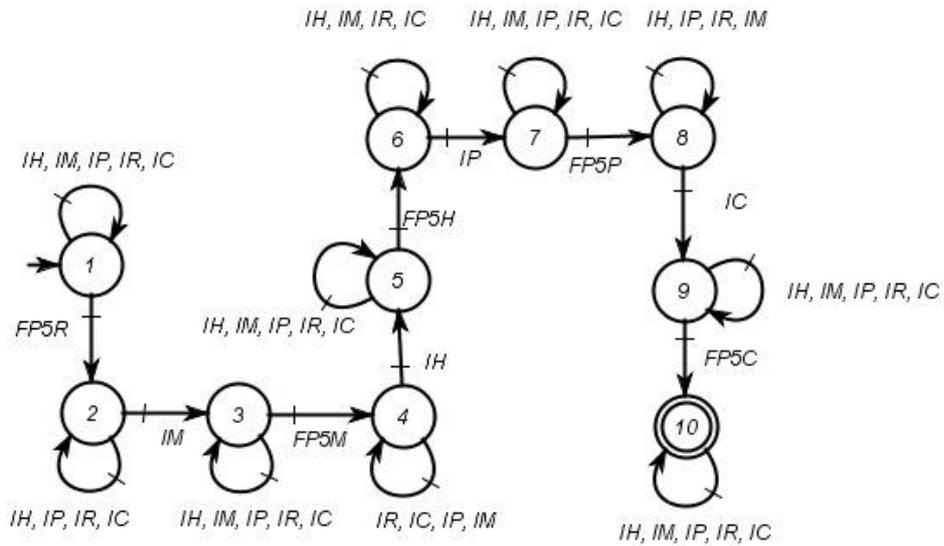


Figura 41: Obra P5 - Navio Petroleiro Lambari

P5: Mecânica → Reparo Estrutural → Hidrojato → Pintura → Controle de Qualidade

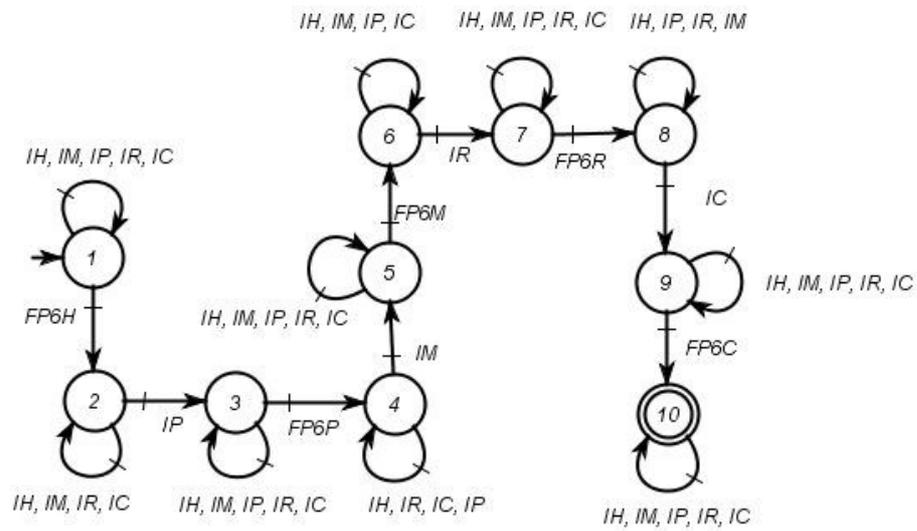


Figura 42: Obra P6 - Navio de Pesquisa Scan Nordic

P6: Hidrojato → Pintura → Mecânica → Reparo Estrutural → Controle de Qualidade

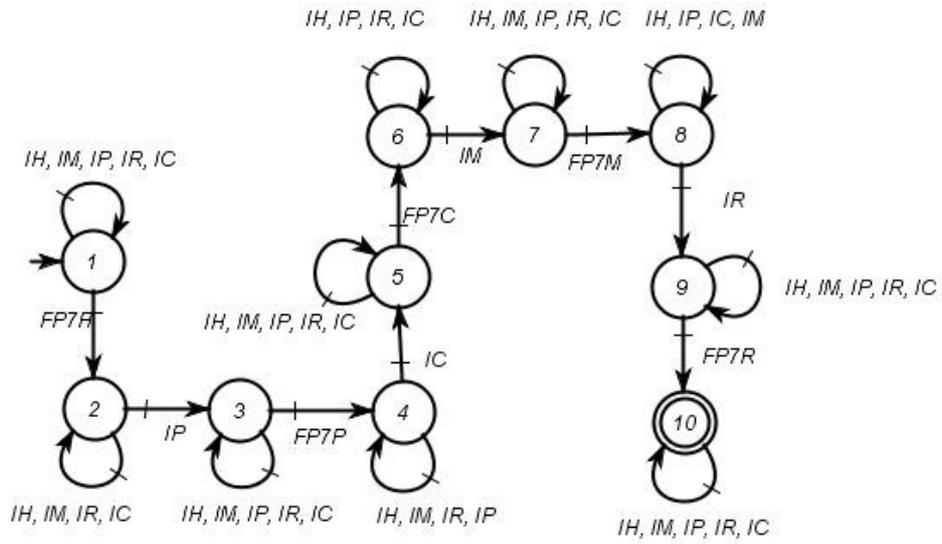


Figura 43: Obra P7 - Navio Graneleiro Mv Sky L O.S1

P7: Hidrojato → Pintura → Controle de Qualidade → Mecânica → Reparo Estrutural

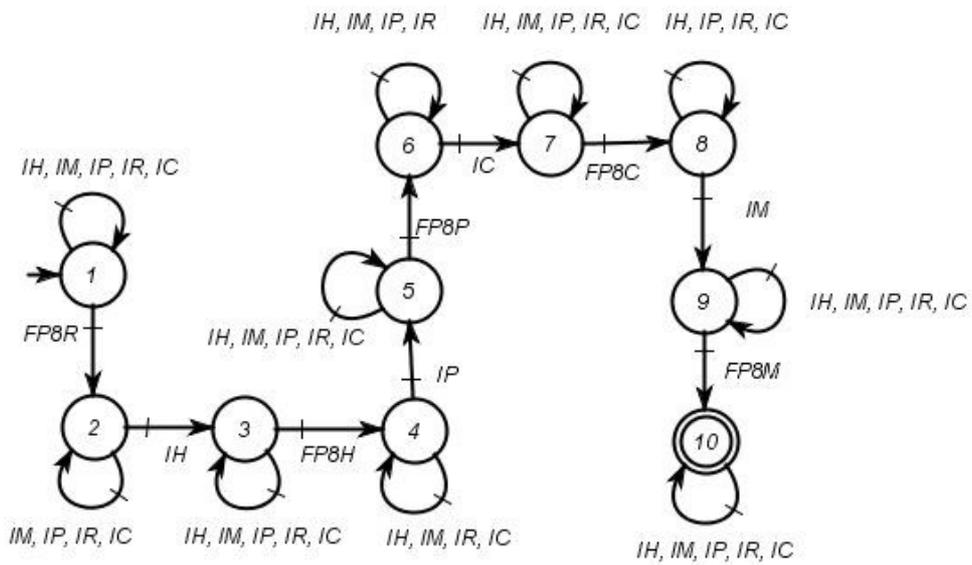


Figura 44: Obra P8 - Navio Graneleiro Navision Bulker

P8: Reparo Estrutural → Hidrojato → Pintura → Controle de Qualidade → Mecânica

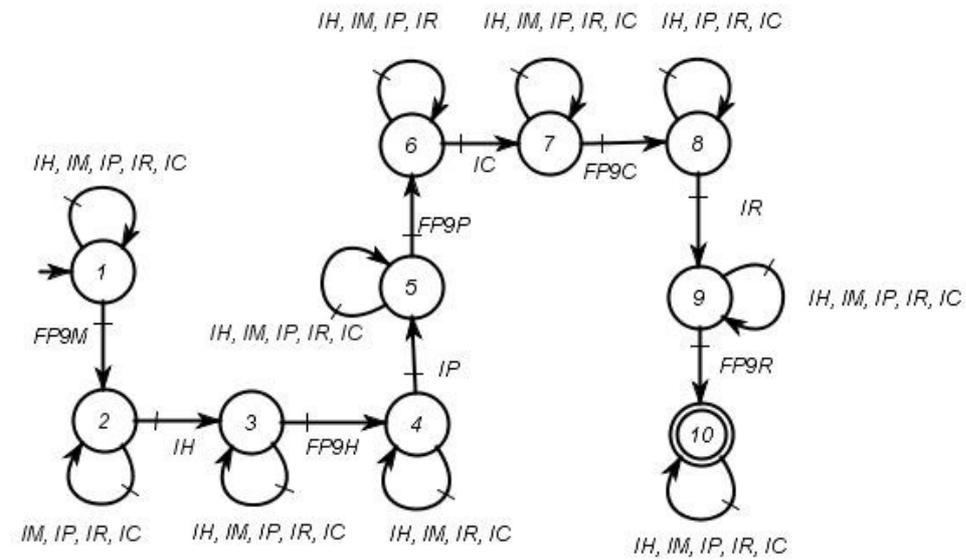


Figura 45: Obra P9 - Navio de Pesquisa

P9: Mecânica → Hidrojato → Pintura → Controle de Qualidade → Reparo Estrutural.

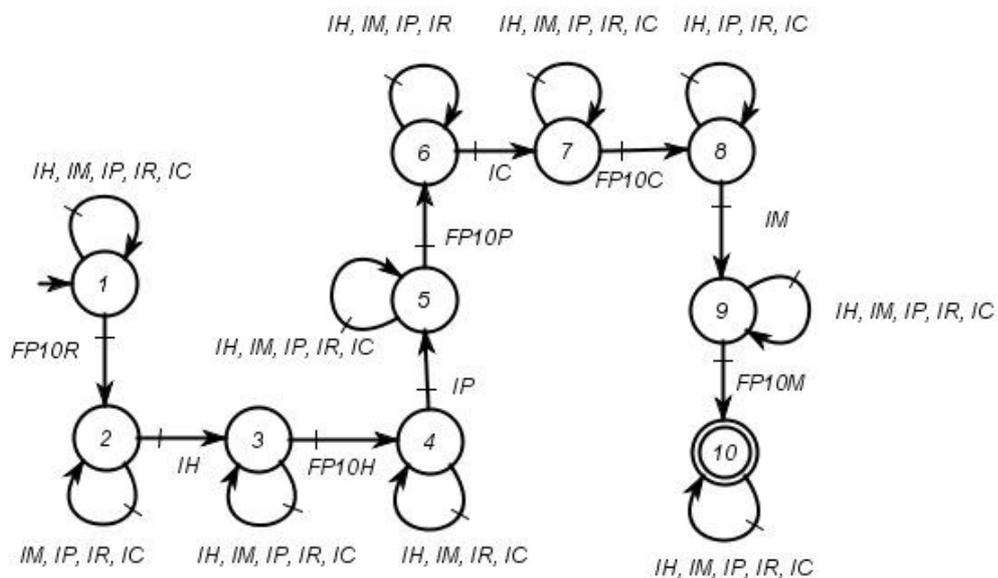


Figura 46: Obra P10 - Navio Supply

P10: Reparo Estrutural → Hidrojato → Pintura → Controle de Qualidade → Mecânica

Com todos os autômatos dos roteiros apresentados, é possível visualizar que os eventos IC, IH, IM, IP e IR (início de trabalho nos recursos) são comuns a todos os modelos e, devido a isso, surge a necessidade de introduzir neste tipo de especificação, alguns cuidados. Tomamos como exemplo a obra P10, ela executa primeiro a operação de

reparo estrutural e a de hidrojato logo em seguida. A introdução do evento de início de operação em cada etapa assegura que, enquanto não for finalizada a operação de reparo estrutural de P10, não poderá ser iniciada a operação de hidrojato, mesmo que já tenha ocorrido o evento de início de operação de hidrojato para outra obra. Assim, é garantida a ordem das obras, impossibilitando a ocorrência paralela de operações que são feitas em sequência. Se no caso da obra P10, o tempo de duração do reparo estrutural fosse menor que o tempo de operação do hidrojato e, antes de finalizar a operação de reparo estrutural, tivesse ocorrido pelo menos uma execução do evento de início de hidrojato, o sistema poderia finalizar esta última, com duração menor e não com o seu tempo completo para P10. Com esta modelagem, a contagem do tempo é iniciada a partir do fim da operação de reparo estrutural. A lógica é a mesma para as outras operações. Seguem abaixo na tabela 11 os nomes dos conjuntos dos eventos das obras.

Tabela 11: Conjuntos dos eventos das obras

Descrição do	Conjunto	Eventos
Eventos da obra P1	$\sum P1$	{FP1R, FP1C, FP1H, FP1M, FP1P}
Eventos da obra P2	$\sum P2$	{FP2R, FP2C, FP2H, FP2M, FP2P}
Eventos da obra P3	$\sum P3$	{FP3R, FP3C, FP3H, FP3M, FP3P}
Eventos da obra P4	$\sum P4$	{FP4R, FP4C, FP4H, FP4M, FP4P}
Eventos da obra P5	$\sum P5$	{FP5R, FP5C, FP5H, FP5M, FP5P}
Eventos da obra P6	$\sum P6$	{FP6R, FP6C, FP6H, FP6M, FP6P}
Eventos da obra P7	$\sum P7$	{FP7R, FP7C, FP7H, FP7M, FP7P}
Eventos da obra P8	$\sum P8$	{FP8R, FP8C, FP8H, FP8M, FP8P}
Eventos da obra P9	$\sum P9$	{FP9R, FP9C, FP9H, FP9M, FP9P}
Eventos da obra P10	$\sum P10$	{FP10R, FP10C, FP10H, FP10M, FP10P}
Eventos da planta	\sum	$\sum P1 \cup \sum P2 \cup \sum P3 \cup \sum P4 \cup \sum P5 \cup \sum P6 \cup \sum P7 \cup \sum P8 \cup \sum P9 \cup \sum P10 \cup \{tick\}$

4.2.3 Autômatos de prazos para as obras

Como descrito no capítulo 3, utiliza-se o conceito de prazo de entrega por obra. Este conceito, além de ser um dos fatores relevantes que permitiu executar o escalonamento das operações no estaleiro com a TCS, é um dado que faz parte do dia-a-dia nos estaleiros. Tentar cumprir prazos estabelecidos se torna fundamental para aumentar o nível de competitividade frente aos concorrentes. Os autômatos de prazos das dez obras foram construídos com as informações do campo “Prazo de Entrega” da tabela 8. Define-se

então, uma especificação temporal, um prazo para cada obra. O autômato da figura 47, por exemplo, representa o prazo de entrega de 12 semanas para a obra P1.

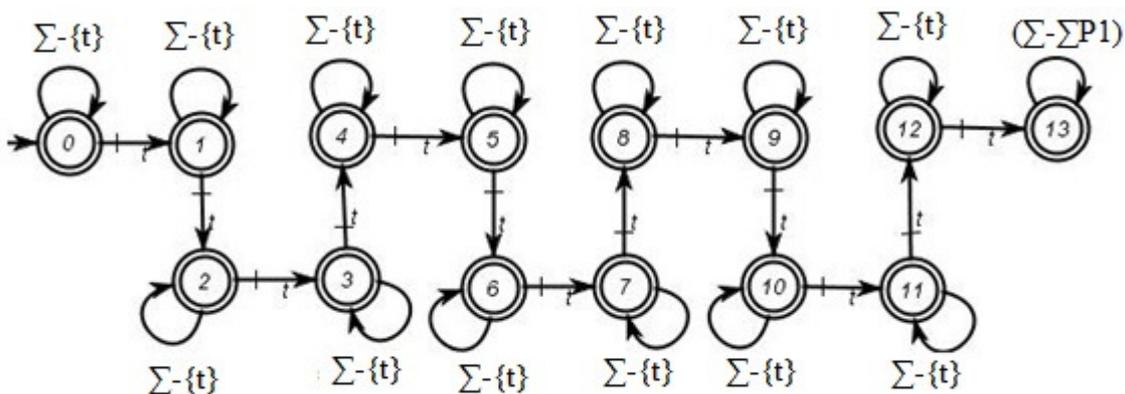


Figura 47: Prazo para a obra P1 (12 semanas)

Pode-se notar que até o estado 12 estão habilitados todos os eventos e, no estado 13, o conjunto de eventos específicos da obra P1 ($\Sigma P1$) é desabilitado. Isso impõe a restrição de 12 semanas para entrega da obra P1. Para gerar o escalonamento segundo a proposta desta pesquisa, cada obra deve ter sua especificação de prazo de entrega, que é obtida de modo semelhante ao da figura 47. Seguem a seguir os autômatos de prazos para o restante das obras.

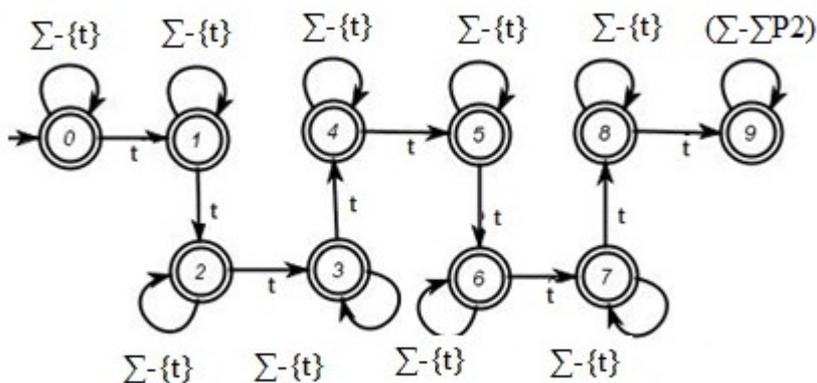


Figura 48: Prazo para a Obra P2 (8 semanas)

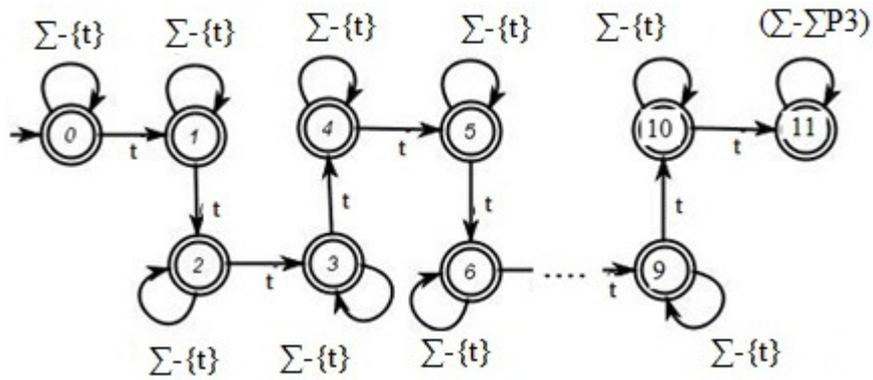


Figura 49: Prazo para a Obra P3 (10 semanas)

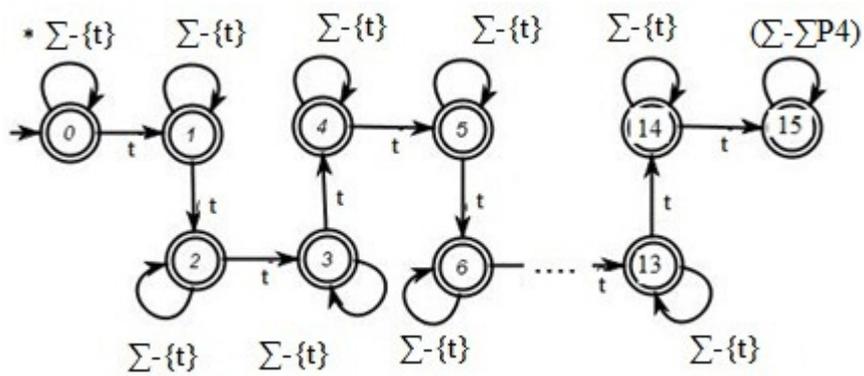


Figura 50: Prazo para a Obra P4 (14 semanas)

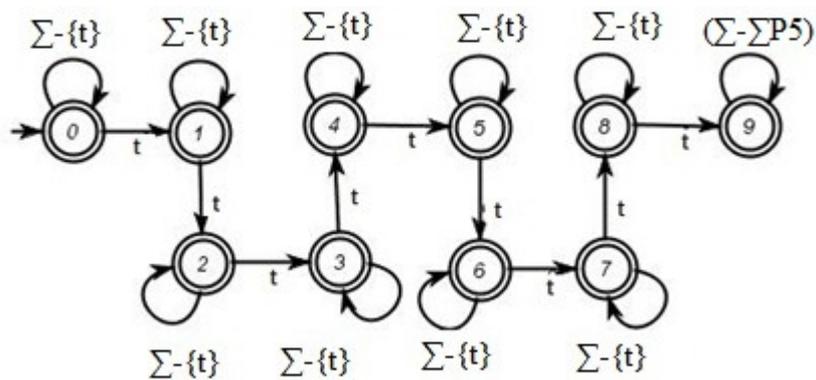


Figura 51: Prazo para a Obra P5 (8 semanas)

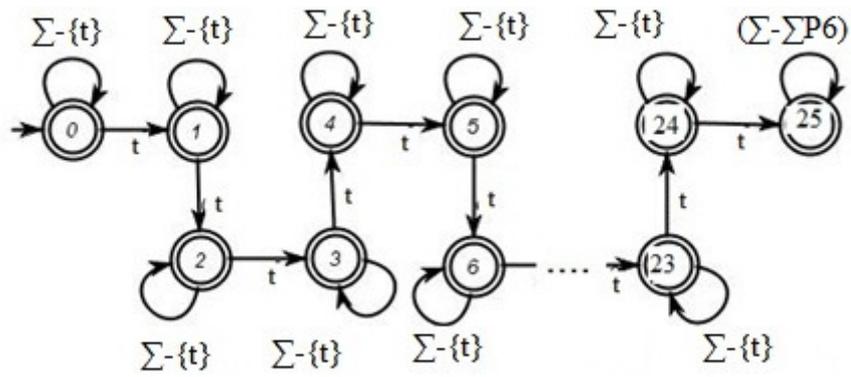


Figura 52: Prazo para a Obra P6 (24 semanas)

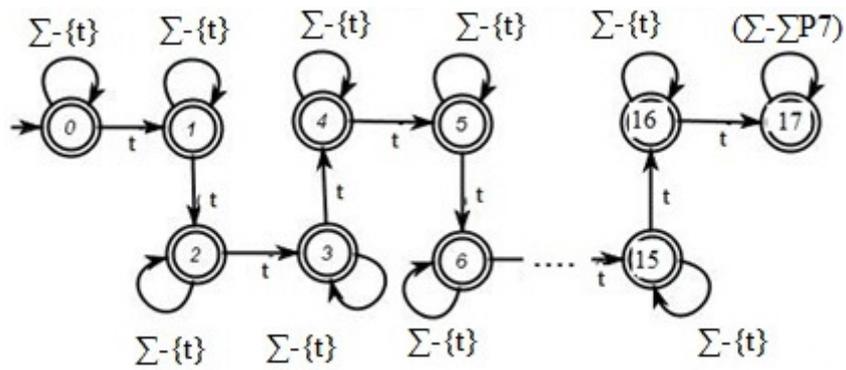


Figura 53: Prazo para a Obra P7 (16 semanas)

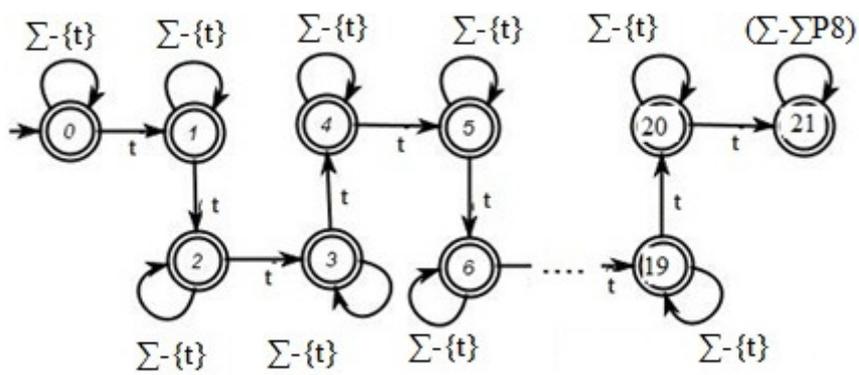


Figura 54: Prazo para a Obra P8 (20 semanas)

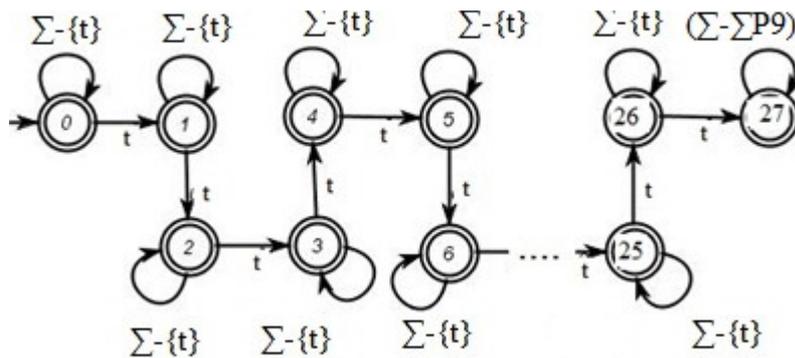


Figura 55: Prazo para a Obra P9 (26 semanas)

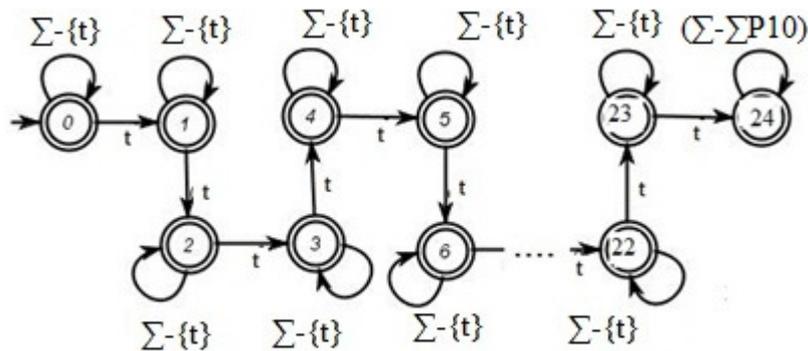


Figura 56: Prazo para a Obra P10 (23 semanas)

Com os autômatos da planta do estaleiro (as equipes), mais os autômatos das especificações (roteiros das obras e prazos) desenvolvidos, inicia-se a síntese conforme descrita no capítulo 3. Seguem na próxima seção, passo a passo, os resultados obtidos através dos algoritmos de síntese da pesquisa.

4.2.4 Síntese do supervisor

Com o objetivo de organizar e apresentar os resultados da síntese dos supervisores, conforme descrito no capítulo 3, os resultados serão divididos em fases. A fase 1 representa os resultados dos passos do algoritmo que executa a síntese incremental dos prazos e dos roteiros das obras (seção 3.4, algoritmo 1), a fase 2 representa os resultados dos passos do algoritmo da síntese entre o supervisor encontrado na fase 1 com os prazos globais reduzidos (seção 3.5, algoritmo 3) e por último, a fase 3 representa os passos do algoritmo da síntese entre o supervisor encontrado na fase 2 com os prazos reduzidos de

produtos prioritários (seção 3.6, algoritmo 4). Para a fase 3, participaram da síntese de redução de *lead time*, as obras P8, P6, P10, P9, P1 e P2. Esta fase se baseia na redução do prazo de obras prioritárias, que por definição já serão entregue no prazo estabelecido através das fases 1 e 2.

Tabela 12: Síntese para o estaleiro

P	Autômato	Número de estados	Tempo Computacional em segundos
Fase 1			
0	Planta temporizada	4.725	0
1	Supervisor T2 (8 semanas)	27.668	1
2	Supervisor R2	22.824	5
3	Supervisor T5 (8 semanas)	22.824	1
4	Supervisor R5	36.533	4
5	Supervisor T3 (10 semanas)	42.553	1
6	Supervisor R3	37.164	11
7	Supervisor T1 (12 semanas)	47.489	1
8	Supervisor R1	141.109	21
9	Supervisor T4 (14 semanas)	154.584	6
10	Supervisor R4	14.631	17
11	Supervisor T7 (16 semanas)	19.881	0
12	Supervisor R7	50.789	5
13	Supervisor T8 (20 semanas)	67.289	1
14	Supervisor R8	190.413	37
15	Supervisor T10 (23 semanas)	208.563	13
16	Supervisor R10	605.157	180
17	Supervisor T6 (24 semanas)	599.397	52
18	Supervisor R6	591.431	49
19	Supervisor T9 (26 semanas)	610.474	415
20	Supervisor R9	138.431	443
Total Fase 1 (26 semanas)			1263≈~21minutos
Fase 2			
21	Supervisor Ot 1 (20 semanas)	0	1
22	Supervisor Ot 2 (23 semanas)	0	2
23	Supervisor Ot 3 (25 semanas)	56.213	4
24	Supervisor Ot 4 (24 semanas)	0	1
Total Fase 2 (25 semanas)			5s
Fase 3			
Redução do lead time de P6 de 24 semanas para 23			
28	SupervisorTP6 (semanas 17)	0	0
29	SupervisorTP6 (semanas 20)	0	0
30	SupervisorTP6 (semanas 22)	0	0
31	SupervisorTP6 (semanas 23)	56.165	2
Redução do lead time de P8 de 20 semanas para 19			

25	SupervisorTP8 (semanas 15)	0	0
26	SupervisorTP8 (semanas 18)	0	0
27	SupervisorTP8(semanas 19)	30.328	2
Redução do lead time de P10 de 23 semanas para 22			
32	SupervisorTP10 (semanas 17)	0	0
	SupervisorTP10 (semanas 20)	0	0
33	SupervisorTP10 (semanas 22)	25.772	1
Redução do lead time de P9 de 26 semanas para 25			
34	SupervisorTP9 (semanas 18)	0	0
35	SupervisorTP9 (semanas 22)	0	0
36	SupervisorTP9 (semanas 24)	0	0
37	SupervisorTP9 (semanas 25)	25.772	0
Redução do lead time de P1 de 12 semanas para 11			
38	SupervisorTP1 (semanas 9)	0	0
39	SupervisorTP1 (semanas 10)	0	0
40	SupervisorTP1 (semanas 11)	25.772	0
Redução do lead time de P2 de 8 semanas para 7			
41	SupervisorTP2 (semanas 7)	16.071	0
Total Fase 3			5s
Total de todas as fases			1273=~21minutos

O supervisor final encontrado no passo 41 é ótimo porque cumpre com os critérios estabelecidos, ou seja, entrega todas as obras no prazo combinado com os clientes e no menor tempo possível para todas. Além desses objetivos alcançados, as obras P8, P6, P10, P9, P1 e P2 finalizam suas operações com os menores *lead times* possíveis. Para as outras obras (P3, P5, P7 e P4) não houve a possibilidade de reduzir esse tempo por dois motivos: i) algumas delas já estavam com o menor *lead time* possível, ou seja, o tempo de entrega já era exatamente igual a soma dos tempos de execução nos recursos, ii) a redução do *lead time* de algumas prejudicaria os prazos de entrega de outras.

Os resultados da tabela 12 foram obtidos através de um computador pessoal com a seguinte configuração: Processador Intel Core 2 Due, CPU T6670, 2.20 GHZ com memória RAM 4,00 GHZ e sistema operacional de 32 bits.

4.2.5 Gráfico de Gantt e escalonamento encontrado para o estaleiro

Com a síntese resolvida através dos algoritmos, serão mostrados os escalonamentos para cada fase (1, 2 e 3) a partir do gráfico de Gantt obtido em cada uma delas. Para a primeira fase, todas as obras foram entregues no prazo e a última operação de todas as obras foi finalizada em 26 semanas. A primeira operação das equipes foi iniciada em

01/06/2010, essa data é um parâmetro escolhido pelo planejador (capítulo 3, algoritmo 5), e a última operação foi finalizada em 31/11/2010. A figura 57 mostra o escalonamento do resultado da fase 1 através do gráfico de Gantt.

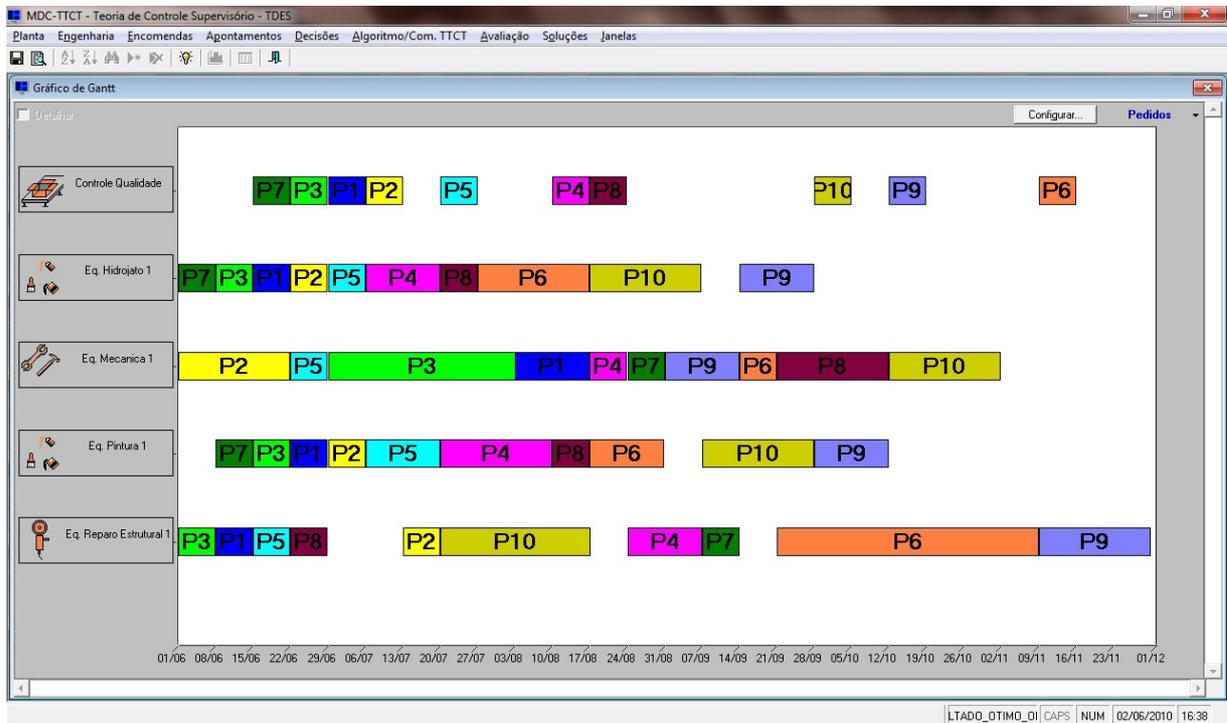


Figura 57: Escalonamento da fase 1

Na fase 2, todas as operações também foram entregues no prazo, sendo que houve uma redução em uma semana no tempo total das operações. A última operação de todas as obras foi finalizada em 25 semanas. A primeira operação das equipes foi iniciada em 01/06/2010 e a última operação foi finalizada em 24/11/2010. A figura 58 mostra o escalonamento do resultado da fase 2.



Figura 58: Escalonamento da fase 2

A redução do prazo global de 26 para 25 semanas foi possível em virtude do rearranjo da sequência da equipe de mecânica, que foi provocado pela síntese dos supervisores da fase 2. Na figura 57 podemos observar que P9 se posicionava antes do P6 na equipe de mecânica, entretanto, na figura 58, esta situação foi alterada, o escalonamento prevê P6 antes de P9. Com esta mudança, o seqüenciamento ou os instantes de início e fim de trabalho das equipes dependentes dessas atividades também foram alterados. Como exemplo, podemos verificar que na figura 57 havia um tempo ocioso entre os pedidos P7 e P6 na equipe de reparo estrutural, esse tempo era devido à espera de atividades antecessoras que ainda não estavam prontas para começar o reparo. O rearranjo de P6 e P9 na equipe de mecânica provocou uma antecipação de P6 na equipe de reparo estrutural, que eliminou a ociosidade entre P7 e P6. Podemos visualizar na figura 58 que a equipe de reparo estrutural não fica mais ociosa entre esses dois pedidos.

Por último, além de entregar todas as obras no prazo e executar todas as operações em 25 semanas, a fase 3, quando possível, reduz o *lead time* dos pedidos prioritários. Embora os resultados da síntese da fase 3, apresentados na tabela 20, indicavam uma redução dos *lead times* dos pedidos P1, P2, P6, P8, P9 e P10. Foi constatado que estes

pedidos, através das sínteses das fases 1 e 2 já foram escalonados com os menores prazos possíveis e por isso não houve mudança com relação aos *lead times* dessas obras.

Os resultados da síntese dos supervisores obtidos na fase 3 restringiu ainda mais o comportamento da planta, ou seja, o supervisor final obtido na fase 3 é mais restritivo do que o obtido ao final da fase 2. Para explicar então porque não houve uma redução dos *lead times* encontrado entre as fases 2 e 3, é necessário recorrer a interpretação do significado da especificação temporal, qualquer que seja ela, de prazo por produto ou prazo global. Na seção 2.3.3 foi apresentado o método de Brandin e Wonham (1992) e foi mostrada a especificação de prazo máximo para todas as tarefas, essa especificação estabelece um prazo máximo para todas as tarefas serem completadas. A especificação de prazo para o controle significa que as tarefas devem ser executadas em no máximo até o prazo especificado, mas dá a liberdade de serem completadas em menos tempo - *um supervisor ótimo garante o comportamento menos restritivo possível*. O método sugere que a fase 3 deva ser sempre executada, pois nem sempre as imposições de especificações de prazos de entrega e roteiro (fase 1) aliada as especificações de prazos globais (fase 2) são suficientes para garantir o escalonamento ótimo como foi neste caso. Além disso, o tempo computacional medido nas sínteses da fase 3 é muito compensador e favorece a geração do escalonamento.

Através do escalonamento final (Fase 3) é possível observar através das figuras 58 e 59, que o escalonamento obtido nesta fase é exatamente igual ao escalonamento da fase 2, isso pode acontecer, porque a cadeia de eventos escolhida na fase 2 para gerar o escalonamento já representava uma sequência de eventos que respeitava todas as restrições impostas e com o menor tempo possível para completar todas as tarefas.

O escalonamento final obtido tem a primeira operação das equipes iniciada em 01/06/2010 e a última operação finalizada em 24/11/2010, como na fase 2. A figura 59 mostra o escalonamento deste resultado.

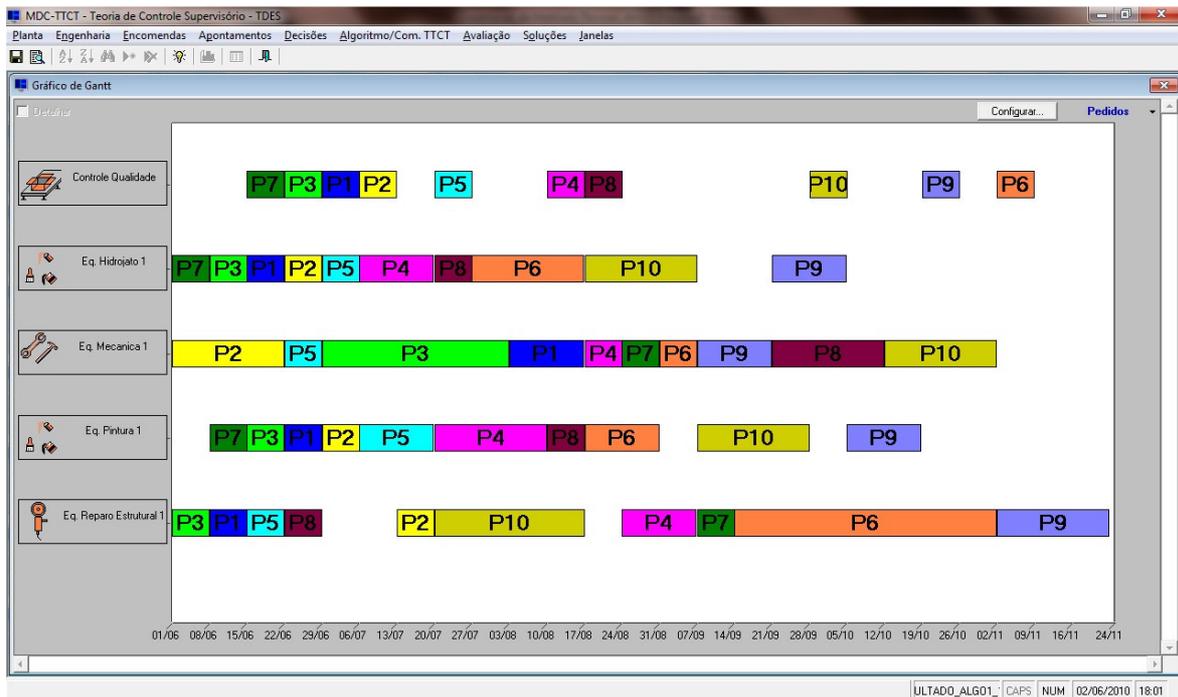
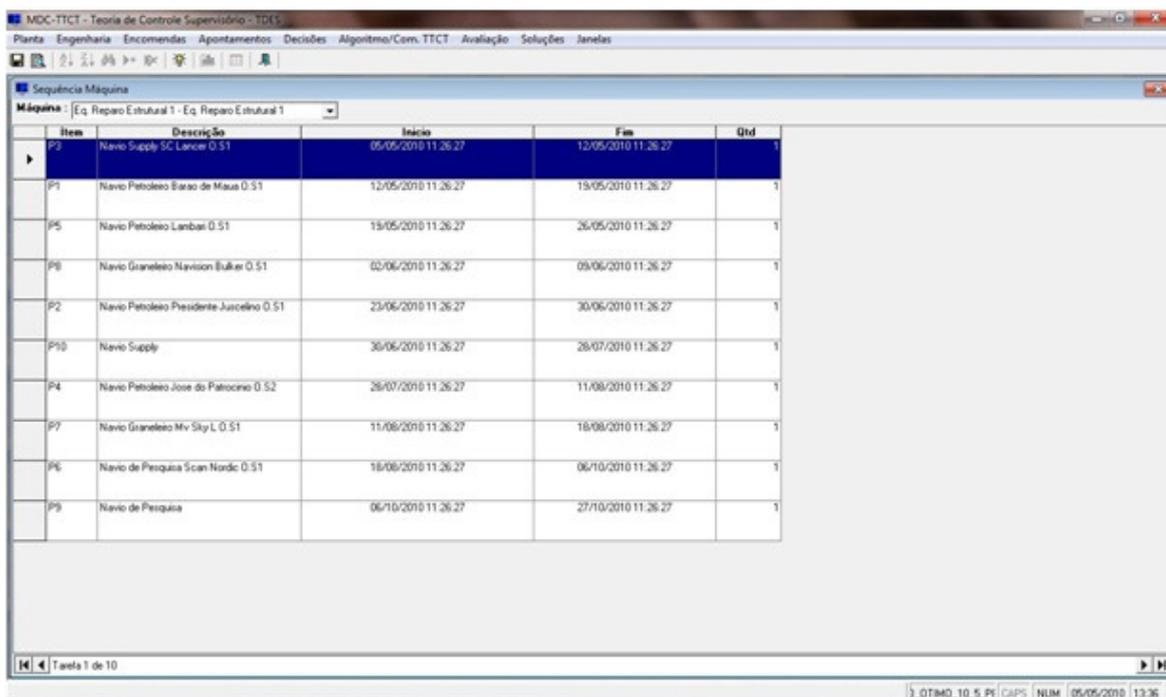


Figura 59: Escalonamento final do estaleiro

No estaleiro, este resultado deve ser enviado para os chefes das equipes para que eles possam executar as atividades operacionais com base no escalonamento. É enviado para cada equipe a sequência dos trabalhos, ou seja, quais são as prioridades e a ordem de execução das obras. Segue na figura 60, o exemplo do sequenciamento para a equipe de reparo estrutural.



The screenshot shows a software window titled 'MDC-TTCT - Teoria de Controle Supervisório - TDI S'. The main window displays a table of activities for a structural repair team. The table has five columns: 'Item', 'Descrição', 'Início', 'Fim', and 'Qtd'. The data is as follows:

Item	Descrição	Início	Fim	Qtd
P3	Navio Supply SC Lanceol 0.51	05/05/2010 11:26:27	12/05/2010 11:26:27	1
P1	Navio Patroleo Barao de Maua 0.51	12/05/2010 11:26:27	19/05/2010 11:26:27	1
P5	Navio Patroleo Lambari 0.51	19/05/2010 11:26:27	26/05/2010 11:26:27	1
P8	Navio Granelero Navision Buker 0.51	02/06/2010 11:26:27	09/06/2010 11:26:27	1
P2	Navio Patroleo Presidente Juscelino 0.51	23/06/2010 11:26:27	30/06/2010 11:26:27	1
P10	Navio Supply	30/06/2010 11:26:27	26/07/2010 11:26:27	1
P4	Navio Patroleo Jose do Patrocin 0.52	26/07/2010 11:26:27	11/08/2010 11:26:27	1
P7	Navio Granelero Mv Sky L 0.51	11/08/2010 11:26:27	18/08/2010 11:26:27	1
P6	Navio de Pesquisa Scan Nordic 0.51	18/08/2010 11:26:27	06/10/2010 11:26:27	1
P9	Navio de Pesquisa	06/10/2010 11:26:27	27/10/2010 11:26:27	1

The interface also shows a status bar at the bottom with the text 'Tabela 1 de 10' and a system tray with the text 'OTIMD_10_5_PE CAPS NUM 05/05/2010 13:36'.

Figura 60: Sequência das atividades para a equipe de reparo estrutural

Todas as outras equipes têm este relatório de forma semelhante, naturalmente com horários e ordens diferentes para as obras. Em função dos bons resultados obtidos com esta metodologia, no próximo capítulo será apresentada a ferramenta desenvolvida que tem o objetivo de ser a interface entre TCS e área de planejamento da produção.

CAPÍTULO 5

A ferramenta MDC

O sistema MDC é uma ferramenta que foi desenvolvida em linguagem *Visual Basic*, no contexto deste trabalho, com o objetivo de gerar o escalonamento das operações através da Teoria de Controle Supervisório. Para esse fim, o sistema se integra com o software TTCT da Universidade de Toronto, contém algoritmos para construção dos modelos e geração dos resultados (capítulo 3) e representa a interface entre o planejador da produção e o TTCT.

5.1 Apresentação da ferramenta

5.1.1 Definição dos modelos dos recursos

Para a construção dos autômatos do problema de escalonamento, é necessário iniciar a modelagem através dos autômatos dos recursos envolvidos. Para esse fim, foi desenvolvida uma interface chamada *Recursos*, localizada no menu *Instalações* do MDC. A figura 61 apresenta esta tela.

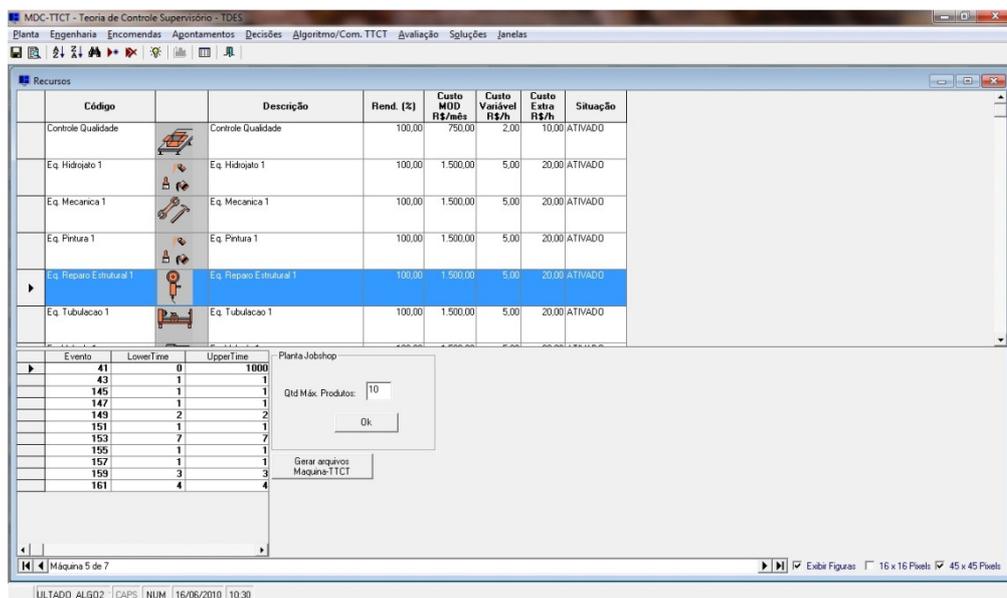


Figura 61: Interface de criação dos modelos dos recursos

Na figura 61 são apresentadas duas tabelas, a primeira tabela superior (tabela 1) contém o cadastro de recursos e a segunda (tabela 2) refere-se aos tempos de operações do recurso selecionado na tabela 1. No exemplo da figura 61, os recursos e os tempos cadastrados são referentes ao problema do estaleiro, que foi detalhado no capítulo anterior. Como exemplo apenas, os recursos cadastrados na tabela 1 da interface são: recurso de reparo estrutural, de mecânica, de pintura, de hidrojetado e de controle de qualidade. Na figura 61, o recurso de reparo estrutural está selecionado e em consequência, na tabela 2, são apresentados os tempos de operações para este recurso. Este recurso pode executar até 10 operações com tempos distintos. Os modelos dos autômatos dos recursos gerados através desta interface seguem a proposta apresentada no capítulo 4.

5.1.2 Definição dos modelos de prazos de entregas

A figura 62 apresenta a interface de cadastro dos pedidos e seus respectivos prazos. A partir desta interface, denominada *Pedidos*, localizada no menu *Encomendas*, são gerados os autômatos de especificações temporais dos prazos das tarefas. Os modelos, que foram apresentados no capítulo 3 e 4, para as especificações de prazo global e prazo de entrega por pedido são gerados através desta tela.

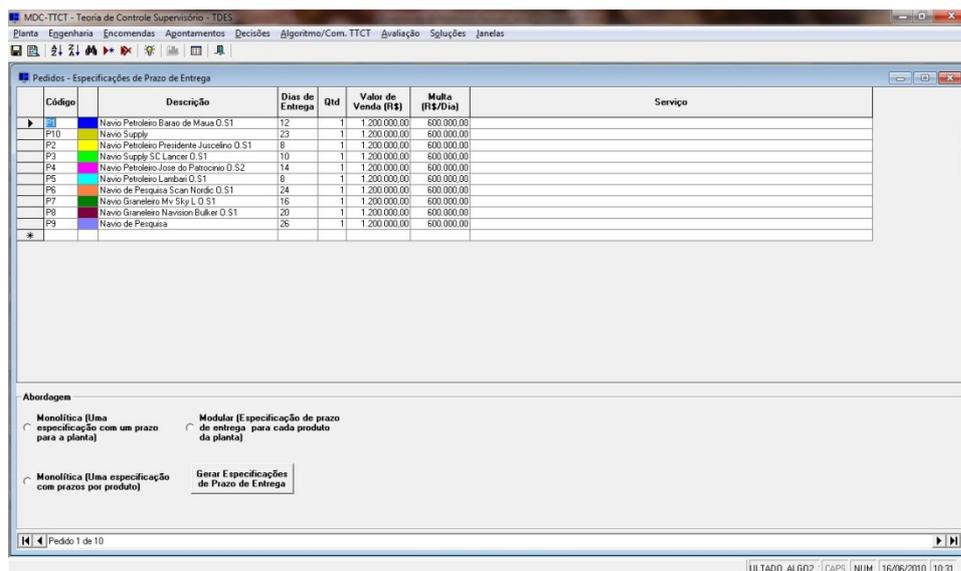


Figura 62: Interface de criação dos modelos das especificações de prazo de entrega

Cada pedido está representado por uma cor em uma linha da tabela. A partir das funções desenvolvidas nesta tela, são geradas as especificações temporais. O usuário pode escolher entre a especificação de prazo global ou prazo por pedido (apresentadas no capítulo 3). Com o cadastro dos pedidos feito, é possível gerar os autômatos dos prazos (global ou por pedido).

5.1.3 Definição das operações

Com os autômatos dos recursos e dos prazos gerados, faltam apenas os autômatos das operações dos pedidos. Após a criação destes modelos, o resultado é o pacote de informações necessárias para começar a síntese dos supervisores a fim de gerar o escalonamento. Esse pacote conceitualmente pode ser definido da seguinte maneira: os autômatos dos recursos dizem onde os pedidos podem ser feitos, os autômatos dos prazos dizem quando as operações devem ser entregues e os autômatos dos roteiros das operações (engenharia de processo) definem como os pedidos devem ser feitos. A figura 63 apresenta a interface “*Lista de serviços*” que fica localizada no menu *Engenharia*. O cadastro dos roteiros das operações é feito nesta interface.

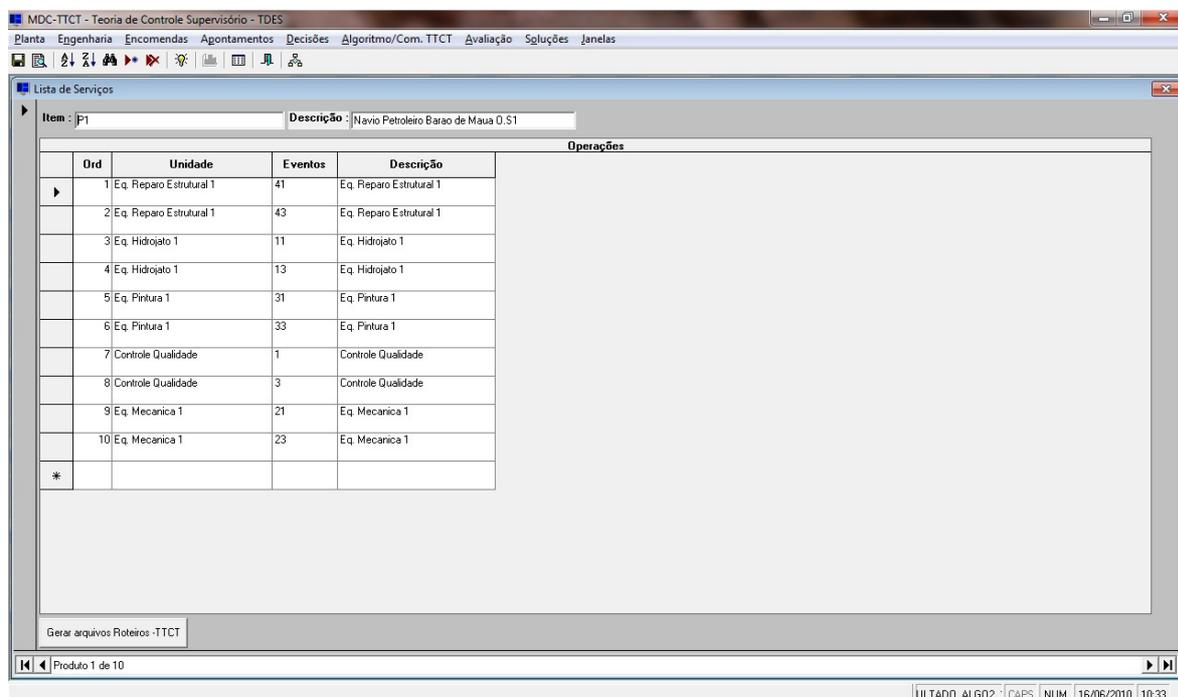


Figura 63: Interface de criação dos modelos das especificações dos roteiros

Tomamos como exemplo, um pedido P1, que deve executar cinco operações distintas para ficar pronto. O roteiro apresentado como exemplo é um pedido do estudo de caso (estaleiro). A quantidade de registros cadastrada, para o roteiro das operações de um pedido, é sempre o dobro do número de operações do pedido. No caso do pedido P1, que tem 5 operações serão 10 registros. Isso porque, para a primeira operação são feitos os cadastros do evento de início de operação e em seguida, do evento de fim de operação, para a segunda operação a mesma coisa, um registro de início de operação e um de fim de operação. Todas as operações são modeladas com 2 eventos, um de início e outro de fim.

Os possíveis eventos cadastráveis nesta tela estão contidos no conjunto de eventos dos autômatos dos recursos. O conjunto dos eventos dos roteiros é um subconjunto dos eventos da planta. Caso essa afirmação não fosse verdadeira, significaria dizer que existem roteiros de produção com fluxo de produção em outra planta, o que não faria sentido para resolver o problema de escalonamento para a planta cadastrada.

5.1.4 Integração com o software TTCT

O MDC tem a responsabilidade de construir os modelos dos autômatos (planta e todas as especificações) de forma automatizada e de executar os algoritmos para obter o escalonamento. Enquanto que a síntese dos supervisores, feita através dos passos dos algoritmos do capítulo 3, é realizada exclusivamente no software TTCT. O objetivo da integração é utilizar as funções implementadas no software TTCT, que são baseadas na teoria de controle de sistemas a eventos discretos temporizados a fim de gerar o escalonamento.

A comunicação entre o MDC e o TTCT é feita através de exportações e importações de arquivos em formato ATS. Este tipo de arquivo é nativo da linguagem do TTCT, e pode ser aberto através de formato texto. Ao final da síntese, o MDC importa o supervisor final, também em formato ATS e, é capaz de obter o escalonamento. A importação dos dados do supervisor é feita através de um módulo de integração desenvolvido. A figura 64 mostra a tela de importação do supervisor.



Figura 64: Interface de importação do supervisor

5.1.5 Gráfico de Gantt

A partir do gráfico de Gantt é possível visualizar de forma simples o comportamento das operações nos recursos e também possíveis atrasos. No MDC este gráfico é obtido através do menu *Algoritmo/Com.TTCT*.

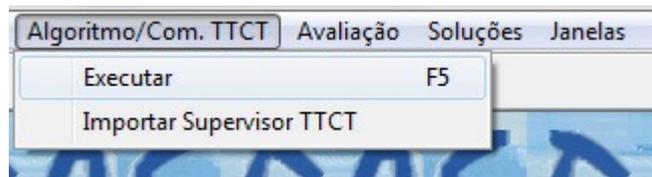


Figura 65: *Algoritmo/Com.TTCT*

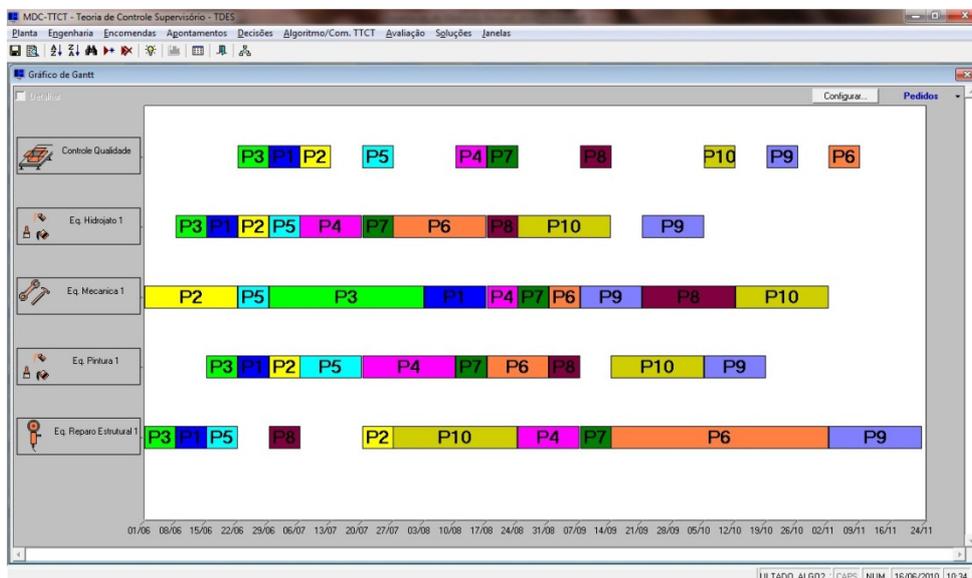


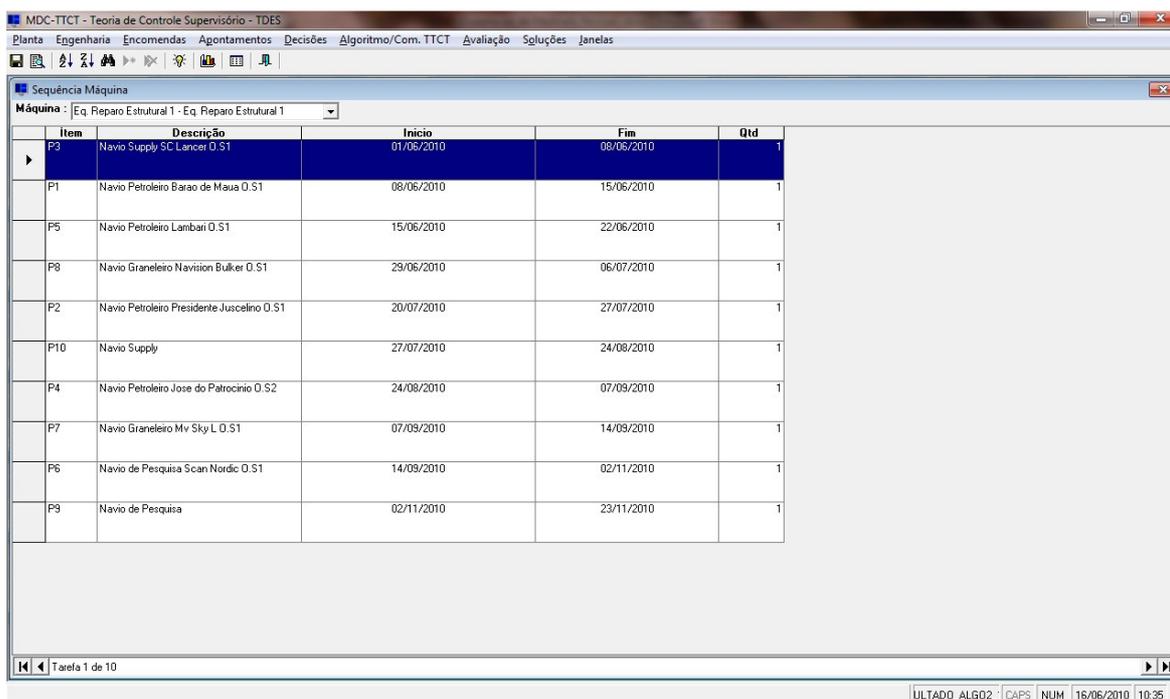
Figura 66: Gráfico de Gantt do sistema MDC

O Gantt também permite avaliar, se a fábrica ou determinados recursos estão sendo muito demandados e desta maneira, pode-se visualizar possíveis gargalos do processo. A figura 66 apresenta o gráfico de Gantt do MDC. Este gráfico fica localizado no menu *Avaliações*.

5.1.6 Relatório de sequência dos recursos

O relatório de sequência dos recursos apresenta as informações contidas no gráfico de Gantt de forma tabular. Como escrito na seção 5.1.5, enquanto o Gantt permite avaliar a fábrica como um todo, se existe ou não, algum gargalo no processo, o relatório de sequência dos recursos permite avaliar a ordem em que as atividades devem ser feitas em um recurso específico.

Essa visualização é importante para a equipe operacional, porque é através destas listas, que o chão de fábrica pode executar o seu trabalho com mais facilidade. Os operadores sabem o que deve ser feito, porque há um relatório, separado por máquina, onde é apresentada a ordem de execução das operações. A figura 67 apresenta este relatório.



Item	Descrição	Início	Fim	Qtd
P3	Navio Supply SC Lancer O.S1	01/06/2010	08/06/2010	1
P1	Navio Petroleiro Barao de Maua O.S1	08/06/2010	15/06/2010	1
P5	Navio Petroleiro Lambari O.S1	15/06/2010	22/06/2010	1
P8	Navio Graneleiro Navision Bulker O.S1	29/06/2010	06/07/2010	1
P2	Navio Petroleiro Presidente Juscelino O.S1	20/07/2010	27/07/2010	1
P10	Navio Supply	27/07/2010	24/08/2010	1
P4	Navio Petroleiro Jose do Patrocinio O.S2	24/08/2010	07/09/2010	1
P7	Navio Graneleiro Mv Sky L O.S1	07/09/2010	14/09/2010	1
P6	Navio de Pesquisa Scan Nordic O.S1	14/09/2010	02/11/2010	1
P9	Navio de Pesquisa	02/11/2010	23/11/2010	1

Figura 67: Relatório de sequência dos recursos

A figura 67 mostra a ordem das operações através de um recurso. Neste exemplo, são 10 operações ordenadas de pedidos diferentes que devem ser feitas em um recurso. Estas operações têm seus horários de início e fim de trabalho e desta forma, os operadores sabem o que devem fazer, onde e quando.

CAPÍTULO 6

Conclusão e Trabalhos futuros

Este trabalho apresentou um novo método baseado na Teoria de Controle Supervisório para aplicações em escalonamento de operações. A modelagem proposta dos autômatos e as implementações dos algoritmos de síntese e de geração do escalonamento foram fundamentais para resolver o problema do estaleiro. Além da metodologia apresentada, outra contribuição desta pesquisa é o sistema MDC, que tem objetivo de ser a interface entre a TCS e o ambiente de Planejamento e Controle da Produção nas empresas e/ou no meio acadêmico.

Como vantagem do uso do método proposto, podemos citar a grande flexibilidade proporcionada com uso da TCS para modelar as restrições encontradas num ambiente produtivo em relação a outras técnicas. Outro benefício deste trabalho, gerado pelas modificações dos modelos, é que o tamanho dos autômatos dos recursos da planta depende apenas do maior tempo de execução das tarefas e não depende mais da quantidade de produtos que o recurso pode executar como em Brandin e Wonham (1992). A partir dos resultados alcançados, é mostrado que este método (modelos e algoritmos) é mais eficiente computacionalmente do que a abordagem de Brandin e Wonham (1992) e o tamanho da planta e dos supervisores se mantêm expressivamente menores. Isso indica que pode ser viável utilizar os algoritmos e a modelagem para problemas reais de escalonamento. Outra contribuição deste trabalho junto à comunidade científica foi à escrita de um artigo denominado “*Escalonamento da Produção com o uso da Teoria de Controle Supervisório*” aceito no XVIII Congresso Brasileiro de Automática (CBA 2010).

Como limitação do método, podemos citar que as modificações dos modelos não permitem a utilização do resultado como um supervisor *online*, já que a ação de controle associada a um evento de início de operação se mantém indeterminada até o final da operação.

Como perspectivas futuras deste trabalho podemos citar:

- a) Flexibilizar a modelagem através da TCS com outras especificações. Criar novos modelos de autômatos para diversas restrições encontradas no setor produtivo, tais como: máquinas simultâneas, ou seja, máquinas que só podem iniciar suas tarefas se outras máquinas também estiverem trabalhando (formação de pares de máquinas para trabalhos em conjunto); produtos conjugados, execução de um ou mais produtos diferentes numa mesma máquina ao mesmo tempo; alternativas de máquinas, máquinas opcionais para as etapas de produção; recursos secundários, necessidade de mais de um recurso para poder iniciar uma operação; tempos de setups diferentes por produto.
- b) Desenvolver algoritmos mais eficientes para realizar a síntese e para gerar o escalonamento. Uma abordagem modular apresentada em Queiroz e Cury (2000) e Queiroz (2004) seria muito apropriada. Como muitos recursos num centro de produção não são todos compartilhados entre os produtos ou pedidos, poder-se-ia subdividir o problema em várias plantas com suas respectivas especificações e executar uma síntese local. Deste modo, haveria a possibilidade de solucionar problemas de dimensões ainda maiores com a TCS. Outros algoritmos eficientes também poderiam ser usados, como sugestão, o BDD (Binary Decision Diagrams) (Saadatpoor, 2004).
- c) Expandir a ferramenta MDC: i) Desenvolvimento de novos modelos voltados para a área de planejamento da produção; ii) melhoria da usabilidade do sistema, melhorias nas telas, menus e funções; incorporação dos algoritmos da Teoria de Controle Supervisório.
- d) Desenvolver novas aplicações e fazer um *Benchmarking* para avaliar o desempenho do método proposto comparado com outras técnicas.
- e) Construir o modelo do supervisor online (tempo real), antes inviável, a partir do modelo do supervisor encontrado para o escalonamento, uma vez que este teve uma redução expressiva do seu número de estados em comparação com aquele.
- f) Integrar a metodologia desenvolvida com a TCS com outra tecnologia, como por exemplo, sistemas de capacidade finita com regras de filas (Costa 1996). O problema poderia ser dividido e tratado hierarquicamente, de forma que a TCS

resolveria o problema de escalonamento de mais alto nível, proporcionando resultados ótimos, enquanto o sistema de capacidade finita trabalharia com as informações em um nível mais operacional com o objetivo atender aos alvos de prazos estabelecidos no nível anterior.

- g) Aplicar esta tecnologia para estipular prazos das obras nos estaleiros ou para qualquer sistema produtivo durante a negociação com os clientes, ou seja, estimar prazos mais realistas e melhores frentes as demandas já contratadas. Atualmente, os estaleiros não têm o apoio de tecnologias mais adequadas para esta função e, os prazos definidos nos estaleiros, com seus respectivos clientes não são cumpridos em 100% das vezes (Informação obtida diretamente com os responsáveis pelo orçamento das obras). Claro, que muitos desses atrasos são provocados por fatores de difícil predição, tais como: quebra de recursos, falta de materiais, falta de operadores, clima e outros. Mas, na maioria dos casos, o prazo inicial combinado entre as partes, por si só, é gerador de problemas. A ferramenta desenvolvida com base na TCS poderia ser um apoio neste processo.

Referências Bibliográficas

- AKERS, S. B., (1956), "A Graphical Approach to Production Scheduling Problems", *Operations Research*, v. 5, pp. 244-245.
- APPLEGATE, D., COOK, W., (1991), "A Computational Study of the Job Shop Scheduling Problem", *ORSA Journal on Computing*, v. 3, n. 2, pp. 49-156.
- ASHOUR, S., (1967), "A Decomposition Approach for the Machine Scheduling Problem", *International Journal of Production Research*, v. 6, n. 2, pp. 109-122.
- ARINDAM CHAUDHURI, KAJAL DE., (2010) "Job Scheduling Problem Using Rough Fuzzy Multilayer Perception Neural Networks" *Journal of Artificial Intelligence: Theory and Application (Vol.1/Iss.1)* , February 2010.
- BAHASKARAN, K., PINEDO, M., (1991), "Dispatching", In : Salvendy, G., (editor), "Handbook of Industrial Engineering", Capítulo 83, John Wiley and Sons, Nova York.
- BAKER, K. R., (1974), "Introduction to Sequencing and Scheduling", New York , John Wiley.
- BAKER, K. R., SCHRAGE, L. E., (1978), "Finding an Optimal Sequence by Dynamic Programming : an Extension to Precedence-Related Tasks", *Operations Research*, v. 26, pp. 111-120.
- BENSANA, E., BEL., G., DUBOIS, D., (1988), "OPAL: A Multi-Knowledge-Based System for Industrial Job Shop Scheduling", *International Journal of Production Research*, v. 26, n. 5, pp. 795-819.
- BIEGEL, J. E., DAVEM, J. J., (1990), "Genetic Algorithms for Job Shop Scheduling", *Computers and Industrial Engineering*, v.19, n. 1-4, pp. 81-91.
- BLACKSTONE, J. H., PHILLIPS, D. T., HOGG, G. L., (1982), "A State of the Art Survey of Dispatching Rules for Manufacturing Job Shop Operations", *International Journal Of Production Research*, v. 20, n. 4, pp. 27- 37.
- BRANDIN, B.A.; WONHAM, W.M.; BENHABIB, B., (1992). "Manufacturing cell

- supervisory control-a timed discrete event system approach". Proceedings of the IEEE International Conference on Robotics and Automation, vol.2, 12-14, pp.931-936
- CASSANDRAS, C.G. e LAFORTUNE, S., (1999). "Introduction to Discrete Event Systems". Kluwer Academic Publishers, USA.
- CHENG, L.Z e HALL, G. N., (2008). "Maximum Profit Scheduling". Manufacturing & Service Operations Management, vol.10, No. 1, Winter 2008, pp.84-107
- CHEUNG, J. Y., (1994), "Scheduling", In : Dagli, C. H., (editor), *"Artificial Neural Networks for Intelligent Manufacturing"*, Capítulo 8, Chapman and Hall, Londres.,
- COSTA, R. S., (1996), "Pontualidade Total na Produção sob Encomenda : Conceito, Tecnologia e Uso da Simulação Computacional na Gestão do Chão-de-Fábrica". Tese de Doutorado, COPPE/Universidade Federal do Rio de Janeiro, Rio de Janeiro,Brasil.
- CURY, J.E.R., (2001). "Teoria de Controle Supervisório de Sistemas a Eventos Discretos". V Simpósio Brasileiro de Automação Inteligente, Canela-RS.
- DAGLI, C. H., HUGGAHALLI, R., (1995), "A Neural Net Architecture for Faster Dynamic Scheduling in Manufacturing Systems", In : *Proceedings of the IEEE International Conference in Robotics and Automation*", pp. 2408-2413.
- DAGLI, C. H., HUGGAHALLI, R., (1991), "A Neural Net Architecture for Faster Dynamic Scheduling in Manufacturing Systems", In : *Proceedings of the IEEE International Conference in Robotics and Automation*", pp. 2408-2413.
- DAVIS, L., (1991), *Handbook of Genetic Algorithms*, New York , Van Nostrand Reinhold.
- DORNDOF, U., PESCH, E., (1993), "Evolution Based Learning in a Job Shop Scheduling Environment", *Computers and Operations Research*, v.22, n. 1, pp. 25-40.
- ERHAN KESEN, SANCHOY K., AND ZÜLAL GÜNGÖR, (2010) "A genetic algorithm based heuristic for scheduling of virtual manufacturing cells" *Computers & Operations Research* Volume 37, Issue 6, June 2010, Pages 1148-115
- FISHER, M. L., (1973), "Optimal Solution of Scheduling Problems Using Lagrange

- Multipliers", *Operations Research*, v. 21, pp.1114-1127.
- FOX, M. S., (1987), "Constraint - Directed Search: A Case Study of Job Shop Scheduling", Pitman Publishing, London.
- FOX, M.S., SYCARA, K., (1990), "Overview of Cortes: A Constraint Based Approach to Production Planning Scheduling and Control", In : *Proceedings of the Fourth International Conference on Expert Systems. and Operations Management*, pp. 100-110, Hilton Head Island, EUA.
- FRENCH, S. (1982). "Sequencing and Scheduling: An Introduction to The Mathematics of the Job Shop", London, Ellis Horwood Limited
- GIGLIO, R., WAGNER, H., (1964), "Approximate Solutions for The Three-Machine Scheduling Problem", *Operations Research*, v. 12, pp. 305-324.
- GIFFLER, B., THOMPSON, G. L., (1960), "Algorithms for Solving Production Scheduling Problems", *Operations Research*", v.8, pp. 487-503.
- GOLDBERG, D. J. K., (2007), "Relevância dos Reparos Navais", Centro de Estudos em Gestão Naval, São Paulo, Outubro de 2007
- GOLDRATT, E. M., (1988), "Computerised Shop Floor Scheduling", *International Journal of Production Research*, v.26, n. 3, pp. 443-455.
- GRABOT, B., GENESTE, L., (1994), "Dispatching Rules in Scheduling: A Fuzzy Approach", *International Journal of Production Research*, v. 32, n. 4, pp. 903-915.
- HAUPT, R., (1989), "A Survey of Priority Rule Based Scheduling", *OR Spektrum*, v. 11, pp. 3-16.
- HELD, M., KARP, R. M., (1962), "A Dynamic Programming Approach to Sequencing Problems", *Journal of SIAM*, v.10, pp. 196-120.
- HOLLAND, J. H., (1975), "Adaptation in Natural and Artificial Systems", Cambridge Ma, MIT Press.
- HOPFIELD, J. J., TANK, D. W., (1985), "Neural Computational of Decisions in Optimisation Problems", *Biological Cybernetics*, v. 52, pp. 141-52.
- ISHIBUCHI H., YOSHIDA T., AND MURATA T., (2003) "Balance between Genetic

- Search and Local Search in Memetic Algorithms for Multiobjective Permutation Flowshop Scheduling," *IEEE Trans. on Evolutionary Computation*, Vol. 7, No. 2, pp.204-223, April 2003.
- JOHNSON, S. M., (1954), "Optimal Two- and Three-Stage Production Schedules with Set-Up Times Included", *Naval Research Logistics Quarterly*, v. 1, n. 3, pp.61-68.
- JACKSON, J. R., (1955), "*Scheduling a Production Line to Minimise Maximum Tardiness*", Research Report 43, Management Science Research Projects, University of California, Los Angeles, USA.
- LAMBERT, D. M.; STOCK, J. R.; ELLRAM, L. M., (1998) "*Fundamentals of logistics management*". Nova Iorque: McGraw-Hill
- LEPAPE, C., (1995), "SOJA: A Daily Workshop Scheduling System", *Expert System*, v. 85, pp. 95-211.
- LITTLE, D., JARVIS, P. C., (1993), "Survey of Current UK Shop Floor Scheduling Practice", *BPICS Control*, Dezembro 1992/Janeiro 1993, pp.35-39.
- LU H. L., HUANG GEORGE Q., YANG H. D. (2010) "Integrating order review/release and dispatching rules for assembly job shop scheduling using a simulation approach" *International Journal of Production Research*, February 2010
- MELETON, M. P., (1986), "OPT - Fantasy or Breakthrough", *Production and Inventory Management*, segundo quadrimeste, pp. 13-20.
- MONTAZER, M., VAN WASSENHOF, L., (1990), "Analysis of Scheduling Rules for FMS", *International Journal of Production Research*, v. 28, n. 2, 785-802.
- OSMAN, I. H., KELLY, J. P., (1996), (editores), *Meta-Heuristics: Theory and Applications*, Norwell, Massachusetts, Kluwer Academic Publishers.
- OW, P. S., SMITH, S. F., (1988), "Viewing Scheduling as an Opportunistic Problem-Solving Process", *Annals of Operations Research*, v. 12, pp. 85-108.
- O'YOUNG S.D., (1992) "On the synthesis of supervisors for timed discrete event processes" Revision submitted: *IEEE Trans. on Automatic Control*, July 1992
- PANWALKAR, S. S., ISKANDER, W., (1977), "A Survey of Scheduling Rules", *Operations Research*, v. 25, n. 1, pp. 45-61.
- PARK S. J. , YANG J. M. (2009) "Supervisory control for real-time scheduling of periodic

- and sporadic tasks with resource constraints” ScienceDirect , Automatica, August 2009.
- PEDROSO, M. C., CORREA, H.L., (1996), “Sistemas de Programação Finita: Uma Decisão Estratégica ?”, *RAE - Revista de Administração de Empresas*, v. 36, n. 4, pp. 60.
- PESCH, E., TETZLAFF, U. A. W., (1996), "Constraint Propagation Based Scheduling of Job Shops", *INFORMS Journal on Computing*, v. 8, n. 2, pp.144-157.
- QUEIROZ, M.H.; CURY, J.E.R. (2000). “Modular supervisory control of large scale discrete event system”. Proceeding of 5th International Workshop on Discrete Event System. Ghent, Belgium.
- QUEIROZ, M.H. (2004). “Controle Supervisório Modular e Multitarefa de Sistemas Compostos”. Tese de doutorado, UFSC, Florianópolis, Brasil.
- RABELO, R. J (1997) “Um Enquadramento para o Desenvolvimento de Sistemas de Escalonamento Ágil da Produção” Tese de doutorado. Universidade de Nova Lisboa, Portugal.
- RAMADGE, P.J.G. E WONHAM, W.M.; 1989, "The control of discrete event systems", Proceedings of the IEEE, Vol. 77, No. 1, pp. 81-98.
- ROSHANAEI V., BALAGH A. K. G., ESFAHANI M. S. AND VAHDANI B., (2010) “A mixed-integer linear programming model along with an electromagnetism-like algorithm for scheduling job shop production system with sequence-dependent set-up times” *The International Journal of Advanced Manufacturing Technology*, Volume 47, Numbers 5-8, March, 2010.
- ROY, R., MEIKLE, R., (1995), “The Role of Discrete Event Simulation Techniques in Finite Capacity Scheduling” *Journal of Operational Research Society*, v.46, pp. 1310 -1321.
- SAADATPOOR, A. (2004). “State Based Control of Timed Discrete Event Systems using Binary Decision Diagrams”. Master Thesis, University of Toronto, Toronto, Canada.

- SAÍSSE, M.C. (2001). “Inovação e flexibilidade na produção em massa: uma investigação sobre o uso de programação evolucionária aliada à simulação computacional para apoio à programação da produção no curto prazo”. Tese de doutorado. COPPE/UFRJ, Rio de Janeiro.
- SLACK, N., CHAMBERS, S., HARLAND, C., HARRISON, A., JOHNSTON, R., (1995), *Operations Management*, London, Pitman Publishing.
- SHAH, V. C., MADEY, G. R., MEHREZ, A.,(1992) "A Methodology for Knowledge Based Scheduling Decision Support", *OMEGA International Journal of Management Science*, v. 20, n. 5/6, pp. 679-703.
- SMITH, W. E., (1956), "Various Optimisers for Single Stage Production", *Naval Research Logistics Quarterly*, v. 3, pp. 59-66.
- TEIXEIRA R, , FERNANDES F., AND PEREIRA N. (2010)“Binary integer programming formulations for scheduling in market-driven foundries” *Computers & Operations Research*, May 2010.
- TERSINE, (1985) *Research Production / Operations Management: “Concepts, Structure & Analysis”* (2nd edition), North Holland, 1985.
- VAN DE VELDE, S., (1991), “*Machine Scheduling and Lagrangian Relaxation*”, Tese de Doutorado, CWI Amsterdam, Amsterdam, Holanda.
- VARELA, M.L.R. (2007). “Uma Contribuição para o Escalonamento da Produção baseado em Métodos Globalmente Distribuídos”. Tese de doutorado, Universidade do Minho, Azurém, Portugal.
- WONHAM, M.W. (2009) “Supervisory Control of Discrete-Event Systems”, ECE 1636F/1637S, University of Toronto.
- ZHOU N., XING K., NAGALINGAM S. AND LIN G., (2010) “Development of an Agent Based VCIM Resource Scheduling Process for Small and Medium Enterprises” *Proceedings of International MultiConference of Engineers and Computer Scientists 2010 Vol I, IMECS*, March 2010.

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)