

UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS
DEPARTAMENTO DE ENGENHARIA DE PRODUÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

João Vítor Silva Robazzi

**Novos limitantes inferiores para o
flowshop com *buffer zero***

São Carlos - SP
2018

João Vítor Silva Robazzi

**Novos limitantes inferiores para o
flowshop com *buffer zero***

Dissertação apresentada ao Programa de Engenharia de Produção da Escola de Engenharia de São Carlos como parte dos requisitos para a obtenção do título de Mestre em Engenharia de Produção.

Área de concentração: Processos e Gestão de Operações

Orientador: Prof. Dr. Marcelo Seido Nagano

São Carlos - SP

2018

AUTORIZO A REPRODUÇÃO TOTAL OU PARCIAL DESTE TRABALHO, POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

Ficha catalográfica elaborada pela Biblioteca Prof. Dr. Sérgio Rodrigues Fontes da EESC/USP com os dados inseridos pelo(a) autor(a).

R628n	Robazzi, João Vítor Silva Novos limitantes inferiores para o flowshop com buffer zero / João Vítor Silva Robazzi; orientador Marcelo Seido Nagano. São Carlos, 2018. Dissertação (Mestrado) - Programa de Pós-Graduação em Engenharia de Produção e Área de Concentração em Processos e Gestão de Operações -- Escola de Engenharia de São Carlos da Universidade de São Paulo, 2018. 1. Programação da Produção. 2. Flowshop. 3. Bloqueio. 4. Setup Dependente da Sequência. 5. Branch and Bound. 6. Programação Linear Inteira Mista. 7. Tempo Total de Fluxo. 8. Tempo Total de Atraso. I. Título.
-------	--

FOLHA DE JULGAMENTO

Candidato: Engenheiro **JOÃO VITOR SILVA ROBAZZI**.

Título da dissertação: "Novos limitantes inferiores para flowshop com buffer zero".

Data da defesa: 08/08/2018.

Comissão Julgadora:

Resultado:

Prof. Dr. **Marcelo Seido Nagano**
(Orientador)
(Escola de Engenharia de São Carlos/EESC)

Aprovado

Prof. Dr. **Roberto Fernandes Tavares Neto**
(Universidade Federal de São Carlos/UFSCar)

Aprovado

Prof. Associado **José Francisco Ferreira Ribeiro**
(Faculdade de Economia, Administração e Contabilidade/FEA-RP/USP)

aprovado

Coordenador do Programa de Pós-Graduação em Engenharia de Produção:

Prof. Associado **Aldo Roberto Ometto**

Presidente da Comissão de Pós-Graduação:

Prof. Associado **Luís Fernando Costa Alberto**

Dedico este trabalho aos meus pais e a todos meus familiares.

Agradecimentos

Agradeço à minha mãe Célia e ao meu pai Luís, por me ensinarem os princípios mais valiosos que levo nesta vida. Aos meus irmãos Ana Luísa e Guilherme, por me apoiarem e me incentivarem nas mais difíceis situações. À minha futura e amada esposa Amanda, por ser minha companheira em todas as situações e por fazer meu cotidiano tão feliz. E a todos meus familiares.

Agradeço aos meus amigos, pela confiança e por agregarem tanto em meu desenvolvimento pessoal.

Agradeço ao meu orientador Prof. Dr. Marcelo Seido Nagano, por me motivar desde nosso primeiro contato a ter garra e seguir em frente. Agradeço também por seus conselhos e sua enorme paciência durante a orientação.

Agradeço à Escola de Engenharia de São Carlos e ao Instituto Federal de São Paulo por todo o suporte disponibilizado.

“Se você expressar o que habita em você, isso irá salvá-lo. Se você não expressar o que habita em você, isso irá lhe destruir.” (Jesus Cristo)

Resumo

Robazzi, João Vítor Silva **Novos limitantes inferiores para o *flowshop* com *buffer zero***. 95 p. Dissertação de mestrado – Escola de Engenharia de São Carlos, Universidade de São Paulo, 2018.

Resumo: O sequenciamento e a programação da produção trazem grandes benefícios financeiros às empresas se realizados de forma adequada. Atualmente, soluções generalizadas apresentam resultados aceitáveis, porém têm como consequência benefícios inferiores quando comparados a estudos específicos. O ramo da otimização de resultados possui dois tipos de soluções: as exatas para problemas de menores dimensões e não exatas, ou heurísticas, para problemas de médias e grandes dimensões. Este trabalho apresenta algoritmos exatos do tipo *Branch & Bound* e Modelos de Programação Linear Inteira Mista para solucionar quatro variações de problemas de *scheduling*: $F_m|block|\sum C_{jm}$, $F_m|block|\sum T_j$, $F_m|block, S_{ijk}|\sum C_{jm}$ e $F_m|block, S_{ijk}|\sum T_j$. As abordagens utilizadas são inéditas na literatura e apresentaram resultados animadores para a maioria dos cenários. O limitante para o tempo total de fluxo obteve resposta ótima em 100% dos casos para problemas de até 20 tarefas e 4 máquinas em menos de uma hora. Para o tempo total de atraso, o limitante se mostrou mais eficiente quando os valores das *due dates* apresentam alta taxa de dispersão. Para os casos com *setup*, foram elaboradas três variações de limitantes para cada problema. O limitante com *setup* que apresentou o melhor desempenho foi o que obteve a melhor relação entre o seu valor numérico e seu custo computacional. Os modelos MILP solucionaram 100% dos problemas sem *setup* para até 20 tarefas e 4 máquinas e para os casos com *setup*, foram solucionados problemas de até 14 tarefas e 4 máquinas no tempo limite de uma hora. Os testes computacionais mostram a eficiência na redução do número de nós e, conseqüentemente, no tempo de execução. Portanto, o estudo realizado indica que, para problemas de pequeno porte e médio, os métodos em questão possuem grande potencial para aplicações práticas.

Palavras-chave: Programação da Produção. *Flowshop*. Bloqueio. *Setup* Dependente da Sequência. *Branch & Bound*. Programação Linear Inteira Mista. Tempo Total de Fluxo. Tempo Total de Atraso.

Abstract

Robazzi, João Vítor Silva **New lower bounds for the zero buffer flowshop.**
95 p. Master Thesis – São Carlos School of Engineering, University of São Paulo, 2018.

Abstract: Job Sequence and Programming give benefits both financial and organizational to any company when performed properly. Nowadays, there is still a gap between theory and practice due to solutions that are short in specification. The analyzed problems differ in type and dimension thus modifying its complexity. The results optimization field is divided into two types of solution: the exact solution for minor problems and the non-exact solution for greater dimension problems. The present paper presents exact algorithms to solve the problems $F_m|block|\sum C_{jm}$, $F_m|block|\sum T_j$, $F_m|block, S_{ijk}|\sum C_{jm}$ by the Branch & Bounds and Mixed Integer Linear Program models. The approaches are new and presented good results for most cases. Bounds for the no-setup total flow time scenario solved 100% of the 20 jobs and 4 machines cases. High dispersion range due dates contributed for the effectiveness of the no-setup total tardiness bound's effectiveness. Three different approaches were developed for the setup cases. The best approach aimed to optimize the value/effort factor for the B&B. The Mixed Integer Linear Program models solved 100% of the no-setup cases for 20 jobs and 4 machines. The MILPs setup cases solved optimally 14 jobs and 4 machines cases. Computational tests were executed and analyzed and they highlighted the node count reduction and, consequently, the execution time. The present study points out that the exact methods can be applied to small and medium scheduling problems in practice.

Keywords: Scheduling. Flowshop. Blocking. Sequence Dependent setup. Branch & Bound. Mixed Integer Linear Program. Total Flow Time. Total Tardiness.

Lista de Ilustrações

Figura 1	Relação entre os ambientes de produção	26
Figura 2	Representação dos termos - <i>FS</i> com <i>setup</i> e bloqueio	29
Figura 3	Representação <i>B&B</i>	45
Figura 4	Limitação do ótimo	47
Figura 5	Avaliação dos nós candidatos ao ótimo	48
Figura 6	Representação das estimativas de início e liberação dos trabalhos	55
Figura 7	Representação de $I_{t,k} + q_{jkm}$	56
Figura 8	Representação das estimativas de início e liberação dos trabalhos com <i>setup</i>	59

Lista de Tabelas

Tabela 1	Número de soluções e número de nós em processo de ramificação sem limitação	47
Tabela 2	Comparação dos limitantes para <i>Total Flow Time</i>	71
Tabela 3	Desvio Relativo Médio <i>Total Flow Time</i>	71
Tabela 4	Comparação dos limitantes para <i>Total Tardiness</i> - Cenário 1	73
Tabela 5	Desvio Relativo Médio <i>Total Tardiness</i> - Cenário 1	73
Tabela 6	Comparação dos limitantes para <i>Total Tardiness</i> - Cenário 2	74
Tabela 7	Desvio Relativo Médio <i>Total Tardiness</i> - Cenário 2	74
Tabela 8	Comparação dos limitantes para <i>Total Tardiness</i> - Cenário 3	75
Tabela 9	Desvio Relativo Médio <i>Total Tardiness</i> - Cenário 3	75
Tabela 10	Comparação dos limitantes para <i>Total Tardiness</i> - Cenário 4	76
Tabela 11	Desvio Relativo Médio <i>Total Tardiness</i> - Cenário 4	76
Tabela 12	Comparação dos limitantes para <i>Total Flow Time</i> com <i>setup</i> - Cenário 1	78
Tabela 13	Desvio Relativo Médio <i>Total Flow Time</i> com <i>setup</i> - Cenário 1	78
Tabela 14	Comparação dos limitantes para <i>Total Flow Time</i> com <i>setup</i> - Cenário 2	79
Tabela 15	Desvio Relativo Médio <i>Total Flow Time</i> com <i>setup</i> - Cenário 2	79
Tabela 16	Comparação dos limitantes para <i>Total Flow Time</i> com <i>setup</i> - Cenário 3	80
Tabela 17	Desvio Relativo Médio <i>Total Flow Time</i> com <i>setup</i> - Cenário 3	80
Tabela 18	Comparação dos limitantes para <i>Total Flow Time</i> com <i>setup</i> - Cenário 4	81
Tabela 19	Desvio Relativo Médio <i>Total Flow Time</i> com <i>setup</i> - Cenário 4	81
Tabela 20	Comparação dos limitantes para <i>Total Tardiness</i> com <i>setup</i> - Cenário 1	82
Tabela 21	Desvio Relativo Médio <i>Total Tardiness</i> com <i>setup</i> - Cenário 1	83
Tabela 22	Comparação dos limitantes para <i>Total Tardiness</i> com <i>setup</i> - Cenário 2	83
Tabela 23	Desvio Relativo Médio <i>Total Tardiness</i> com <i>setup</i> - Cenário 2	83
Tabela 24	Comparação dos limitantes para <i>Total Tardiness</i> com <i>setup</i> - Cenário 3	84
Tabela 25	Desvio Relativo Médio <i>Total Tardiness</i> com <i>setup</i> - Cenário 3	84
Tabela 26	Comparação dos limitantes para <i>Total Tardiness</i> com <i>setup</i> - Cenário 4	85

Tabela 27	Desvio Relativo Médio <i>Total Tardiness</i> com <i>setup</i> - Cenário 4	85
Tabela 28	Tempo (s) - Modelos <i>MILP</i>	87

Sumário

1	Introdução	19
1.1	Objetivo da Pesquisa	21
1.1.1	Objetivos Específicos	22
1.2	Justificativa	22
1.3	Estrutura do trabalho	24
2	Programação da Produção	25
2.1	Ambientes de Produção: α	25
2.2	Restrições Tecnológicas: β	26
2.3	Objetivos do Sequenciamento: γ	28
2.4	Definição do Problema	28
2.5	Terminologia	29
2.6	Cálculo das Funções Objetivo	30
3	Revisão Bibliográfica	33
3.1	Ignall e Schrage (1965)	33
3.2	Bansal (1977)	34
3.3	Rios-Mercado e Bard (1999)	34
3.4	Ronconi e Armentano (2001)	35
3.5	Chung, Flynn e Kirca (2002)	36
3.6	Ronconi (2005)	37
3.7	Chung, Flynn e Kirca (2006)	37
3.8	Companys e Mateo (2007)	38
3.9	Moslehi e Khorasanian (2013)	39
3.10	Jesus e Rodrigues (2014)	40
3.11	Sanches, Takano e Nagano (2016)	40
3.12	Takano e Nagano (2017)	41
3.13	Toumi et al. (2017a)	42

3.14	Toumi et al. (2017b)	43
3.15	Lacunas encontradas durante a revisão bibliográfica	43
4	O Método <i>Branch & Bound</i>	45
4.1	Ramificação	46
4.2	Limitantes	47
4.2.1	Limitantes Superiores	48
4.2.2	Limitantes Inferiores	48
4.3	Regras de Exploração	49
4.3.1	Melhor Limitante	49
4.3.2	Profundidade	49
4.4	O Pseudocódigo	49
5	Propostas de Novos Limitantes Inferiores	53
5.1	O Limitante Inferior para <i>Total Flow Time</i>	54
5.2	O Limitante Inferior para <i>Total Tardiness</i>	56
5.3	O Limitante Inferior para <i>Total Flow Time</i> com <i>setup</i>	57
5.4	O Limitante Inferior para <i>Total Tardiness</i> com <i>setup</i>	59
6	Propostas de Novas Modelagens <i>MILP</i>	61
6.1	Funções Objetivo	61
6.2	Equações de Restrição	61
6.2.1	Modelagem sem <i>setup</i>	63
6.2.2	Modelagem com <i>setup</i>	65
7	Experimentação Computacional e Análise dos Resultados	69
7.1	Resultados do método <i>Branch & Bound</i>	69
7.1.1	Resultados para <i>Total Flow Time</i>	70
7.1.2	Resultados para <i>Total Tardiness</i>	72
7.1.3	Resultados para <i>Total Flow Time</i> com <i>setup</i>	77
7.1.4	Resultados para <i>Total Tardiness</i> com <i>setup</i>	82
7.2	Resultados da Modelagem <i>MILP</i>	86
8	Conclusão	89
9	Trabalhos Futuros	91
	Referências Bibliográficas	93

Introdução

Nas últimas décadas as indústrias vêm enfrentando grandes mudanças e maior competitividade no cenário mundial dos negócios devido aos avanços tecnológicos, da globalização do mercado e das mudanças políticas e econômicas. Este aumento de competitividade exige que as empresas e seus departamentos de planejamento e controle da produção (PCP) aprimorem seus processos organizacionais e utilizem os recursos de forma mais eficiente para que possam se manter no mercado (SLACK, 2009). A partir desta situação, entende-se que o gerenciamento da programação da produção se torna cada vez mais significativo, e visa atingir objetivos cruciais como redução nos custos de produção e a satisfação do cliente (BLANCHARD, 2009).

Em sistemas manufatureiros as tarefas são alocadas em ambientes de produção primeiramente como matéria-prima ou produtos não acabados que se tornarão produtos prontos para o consumo do cliente. Para que o produto final seja alcançado, o mesmo necessita atravessar um ou mais processos de fabricação realizados por máquinas ou estações de trabalho, que são dispostos de forma lógica, sempre visando maximizar a eficiência da produção (GROOVER, 2008). A programação da produção realizada pelos PCPs das indústrias de manufatura é considerada substancialmente impactante, uma vez que levam à redução de custos, fator que acentua a prosperidade do empreendimento (PINEDO, 2008).

Na prática, as empresas que utilizam as ferramentas desenvolvidas para sequenciamento da produção possuem interesses e ambientes fabris diferentes e, apesar do grande avanço na teoria de *scheduling*, esta área possui uma grande variedade de problemas não resolvidos, ou problemas que se renovam com o surgimento de novas técnicas de produção (GRAVES, 1981). O termo *scheduling* pode ser definido como o estudo de decisões de sequenciamento e programação de tarefas. Este estudo visa otimizar as sequências de produção e tem como objetivo minimizar desperdícios nos processamentos e maximizar a utilização de recursos juntamente com o lucro (PINEDO, 2008).

Por conta da grande variedade e mutabilidade no setor industrial, seria praticamente impossível definir uma solução geral para os problemas de sequenciamento que fosse tão eficiente quanto uma solução estudada para cada caso em específico. Geralmente, para

aproximar os ambientes reais dos modelos teóricos, o número de variáveis consideradas é aumentado, fazendo com que o problema adquira maior complexidade de resolução (GRAVES, 1981).

Um ambiente amplamente visto na atualidade industrial é o *flowshop* (*FS*), onde os produtos possuem as mesmas rotas de processamento dentro de uma fábrica. Um exemplo deste ambiente pode ser visto em uma empresa que fabrica sacos de papel para embalar diversos tipos de produtos, como cimento em pó, ração animal e carvão (PINEDO, 2008). Todos os produtos possuem a mesma lógica de processamento, iniciando pelo corte do papel, seguido pela impressão do logotipo do produto e finalizando com a costura parcial do saco. O que difere cada tarefa é o tempo em que cada processo é realizado, por exemplo, o molde do saco de carvão é menor que o molde do saco de ração, fazendo com que o processo seja menos demorado. Estas considerações se estendem a todos os outros processos de diversos produtos. Outro fator a ser considerado é o tamanho do lote que será processado, o que pode influenciar nos estoques finais e intermediários da fábrica (PINEDO, 2008).

Uma característica muito comum em ambientes *FS* é a inexistência (*buffer zero*) ou a limitação de estoques intermediários ocasionando o bloqueio de determinados equipamentos ou estágios. Outro fator que também tende a causar bloqueio entre as máquinas ou estágios de processamento é a característica física do material, assim como as tecnologias processuais, principalmente, quando envolvem temperatura e viscosidade. Por exemplo, em uma indústria metalúrgica os metais fundidos têm a necessidade de permanecer no estágio de aquecimento até que o estágio subsequente esteja pronto para recebê-lo, ou seja, não possua outra tarefa sendo processada e seu *setup* tenha sido realizado (HALL; SRISKANDARAJAH, 1996).

Dentre as diversas características de um ambiente de produção, uma delas é o tempo de *setup*, ou tempo de preparação da máquina, que deve ser realizada para que o processo possa ser devidamente realizado. Tal preparação consiste em trocar ferramentas de corte, moldes, fluidos de corte, matérias primas, programar máquinas automatizadas e muitos outros tipos de mudanças. Em muitos estudos, o tempo de *setup* é incluído diretamente no tempo de processamento, simplificando o sequenciamento, porém não considerando critérios importantes para a obtenção de uma boa resposta final (LEE; DICESARE, 1994).

Para se obter melhores resultados e aproximar a teoria da prática, alguns estudiosos passaram a utilizar o tempo de *setup* separadamente do tempo de processamento. Este tempo necessário para a preparação da máquina pode ser considerado dependente ou independente da sequência de processamento das tarefas, fator que pode otimizar a resposta final obtida por um algoritmo específico (GUPTA; HARIRI; POTTS, 1997).

Além das características específicas de um determinado ambiente, é viável a análise do objetivo principal da programação da produção. Por muitos anos este critério foi a redução do *makespan*, ou tempo total de conclusão de todas as tarefas de um lote,

fator que visa maximizar o aproveitamento dos recursos (FRAMINAN; GUPTA; LEISTEN, 2004). No decorrer dos anos, os focos vem se voltando para outros objetivos além do *makespan*. O *tardiness* ou atraso representa o tempo excedido no término de uma tarefa em relação a sua data de entrega. Quando não obedecido, este critério pode resultar em multas contratuais, aumentando o custo total, além da perda de confiabilidade, fator que resulta em perda de clientes e danos à reputação da empresa. Por ser um critério tanto financeiro quanto reputacional, a minimização do atraso total, ou *total tardiness* (*TT*) vem ganhando cada vez mais espaço nos ambientes de *scheduling* (RONCONI; KAWAMURA, 2010).

O tempo total de fluxo, ou *total flow time* (*TFT*), é um fator que também vem sendo alvo de novas pesquisas. O tempo de fluxo de um produto pode ser entendido como o intervalo de tempo em que este é disponibilizado para o processamento e sua finalização na última máquina. A minimização deste objetivo implica em redução de custos ligados aos estoques de um sistema de produção, além de minimizar o dispêndio causado por um produto que se encontre no fluxo de processamento (WANG; SUN; SUN, 2011).

Os problemas de *FS* são considerados *NP-Hard* ou NP-Difícil para quantidades de máquinas ou estágios maiores que 3 (PINEDO, 2008), ou seja, não existem algoritmos conhecidos que solucionem o problema de forma ótima em tempo polinomial. Por conta da sua alta complexidade, é comum a utilização de métodos heurísticos, ou não-exatos, que são capazes de chegar a uma solução aceitável, porém, não garantem que seja ótima. Entretanto, para pequenos problemas, alguns métodos conseguem atingir a solução ótima em tempos aceitáveis.

O presente trabalho apresenta abordagens exatas no ambiente de *FS* com bloqueio visando a minimização do *TFT* e do *TT* separadamente. Posteriormente, para cada cenário citado, foi considerada a inclusão do tempo de *setup* dependente da sequência e das máquinas, totalizando quatro tipos diferentes de problemas. Os procedimentos de resolução propostos foram o *Branch&Bound* (*B&B*) e modelagens de programação linear inteira mista (*MILP*). Para os métodos escolhidos se vê necessário o profundo conhecimento das características mais fundamentais de cada cenário abordado. Os cálculos dos limitantes e as modelagens realizadas exploram particularidades matemáticas de cada problema em específico, a fim de buscar a solução ótima no menor tempo possível.

1.1 Objetivo da Pesquisa

Este trabalho visou minimizar o *TT* e *TFT* no ambiente de *FS* separadamente, com as restrições de bloqueio e tempo de *setup*. A fim de alcançar respostas ótimas em tempos cada vez menores, foram adaptados dois tipos de procedimentos. O primeiro deles foi o *B&B*, método de solução utilizado em problemas de otimização combinatória que seleciona possíveis sub-sequências candidatas ao ótimo e exclui as não promissoras. A eficiência

deste método é baseada em três fatores principais: Limitantes, Regra de Ramificação e Regra de Exploração. O estudo realizado focou em desenvolver novos limitantes inferiores (LB) para o método em questão, visto que a influência de bons limitantes inferiores reduz significativamente o tempo computacional do algoritmo. O segundo método consistiu em elaborar as modelagens $MILP$ dos quatro cenários e possibilitar resoluções exatas a partir de um *software* de otimização, no caso deste trabalho foi utilizado o pacote *IBM ILOG CPLEX*.

1.1.1 Objetivos Específicos

Os pontos necessários para alcançar os objetivos foram:

- ❑ Revisar trabalhos da literatura relacionados ao $B\&B$ utilizados para solucionar problemas de FS ;
- ❑ Desenvolver algoritmos para o cálculo do TT e TFT com e sem *setup* em ambientes de FS com bloqueio;
- ❑ Utilizar formulações confiáveis na literatura para gerar as *due dates* dos produtos;
- ❑ Desenvolver LBs para direcionar a busca da solução ótima realizada pelo $B\&B$ em todos os cenários mencionados anteriormente;
- ❑ Implementar o procedimento $B\&B$ utilizando a regra de exploração de profundidade;
- ❑ Utilizar os LB desenvolvidos no procedimento implementado;
- ❑ Comparar os resultados obtidos entre todas as variações de resolução apresentadas;
- ❑ Comparar os resultados da melhor resolução desenvolvida neste trabalho com as mais eficientes da literatura;
- ❑ Construir modelos $MILP$ e testar sua eficiência perante ao *software CPLEX*.

1.2 Justificativa

A teoria do *scheduling* possui grande potencial econômico nas aplicações industriais. Porém, um dos motivos que atenua a inovação proveniente desta teoria é a falta de conhecimento da área pelos responsáveis nos setores de programação da produção (MACCARTHY; LIU, 1993). Um dos fatores que diminui o interesse nas aplicações práticas é a falta de similaridade da teoria com os ambientes industriais, tornando alguns resultados teóricos impraticáveis na realidade. Visto que a lógica de produção FS aparece em abundância nos sistemas fabris atuais, pode-se considerar o seu estudo justificável (PINEDO, 2008).

Para aproximar os sistemas ideais dos reais, o presente trabalho considerou duas das restrições mais comuns vistas nos sistemas produtivos na atualidade. A primeira foi a redução do espaço físico das indústrias, que exige que todos os estoques sejam cada vez menores, inclusive os intermediários. Na teoria de sequenciamento da produção este tipo de limitação é chamado de bloqueio (HALL; SRISKANDARAJAH, 1996). A segunda restrição foi o tempo de *setup*, que geralmente é considerado como parte do tempo de processamento.

Nos ambientes reais, o *setup* pode variar de acordo com a semelhança entre dois produtos. Foi demonstrado por Gupta, Hariri e Potts (1997) que a utilização do tempo de *setup* dependente da sequência separado do tempo de processamento aumenta a eficiência da solução final obtida. Portanto, é possível perceber que quanto mais próxima a modelagem teórica está do ambiente real mais eficiente será a resolução final do estudo de sequenciamento.

O objetivo mais estudado no âmbito do *scheduling* é a redução do tempo total de processamento de um conjunto de tarefas, também conhecido como *makespan*. Tendo em vista que poucos trabalhos abordam outras FOs, que também possuem grande importância, o foco deste trabalho foi abordar dois outros critérios de tomada de decisão que também trazem benefícios econômicos a um sistema produtivo.

O primeiro deles é a redução no atraso dos processamentos dos produtos. O não cumprimento de prazos é um dos principais fatores que causa a desorganização da logística entre setores ou empresas (RONCONI; KAWAMURA, 2010). O outro fator, também relacionado ao cliente, é o *TFT* de um conjunto de tarefas. A redução deste resulta em um ambiente de produção mais dinâmico, além de diminuir o estoque de entrada de um sistema produtivo (WANG; SUN; SUN, 2011).

O estudo do sequenciamento da produção proporciona redução de custo e aumento na velocidade da produção e, conseqüentemente, acréscimo de confiabilidade das empresas quando a programação é feita de maneira adequada. Logo, por conta das aplicações práticas explicitadas anteriormente, pode-se perceber que o estudo de *scheduling* é importante para que seu desenvolvimento seja contínuo e, principalmente, para que as empresas possam se manter competitivas no mercado.

O problema em questão é muito pouco estudado na literatura. A revisão realizada mostra que poucos são os trabalhos que exploram os casos de *TFT* e *TT* com bloqueio solucionados por *BEB*. Para os casos com *setup*, não foram encontrados trabalhos relacionados. O presente trabalho colabora com novas abordagens para os ambientes em específico, tornando este original e contribuindo para o desenvolvimento de possíveis novos estudos.

1.3 Estrutura do trabalho

Este trabalho é dividido em oito capítulos, dispostos de forma sequenciada em relação ao desenvolvimento do problema e sua solução. O capítulo 1 apresenta a introdução ao problema, focando em fornecer uma visão geral do tema estudado, apresentando os objetivos e os passos que serão dados para atingi-los. Ainda neste capítulo, são descritas as motivações e justificativas para a execução da pesquisa.

O capítulo 2 apresenta os diversos ambientes que podem ser classificados nos problemas de sequenciamento da produção. Nesta parte do trabalho, são apontadas as variações dos três termos de classificação de Graham et al. (1979). Ainda nesta parte são evidenciados os problemas tratados neste trabalho.

No terceiro capítulo (3) são evidenciados os problemas encontrados na literatura relacionados diretamente a este trabalho. O capítulo 4 apresenta as particularidades do método *B&B*. O capítulo 5 explica como foram elaborados os limitantes propostos neste trabalho. No capítulo 6, são apresentadas as modelagens *MILP* para os quatro ambientes em questão.

Por fim, o capítulo 7 apresenta e comenta os resultados obtidos a partir dos estudos realizados e dos testes computacionais executados. E finalmente, no capítulo 8 são apresentadas as conclusões obtidas a partir do trabalho realizado.

Programação da Produção

O termo programação da produção pode ser entendido como a alocação de um conjunto de tarefas que serão processadas em um determinado ambiente. Tal alocação se baseia em dois parâmetros principais: a sequência em que as tarefas serão processadas e o momento de início de cada uma delas em cada máquina (PINEDO, 2008). A complexidade do cálculo do sequenciamento pode variar de acordo com o tamanho do conjunto de tarefas e também com as restrições tecnológicas do ambiente em que esta ordenação é aplicada. A padronização da descrição do ambiente de produção foi proposta por Graham et al. (1979) se baseando em três termos identificadores: $\alpha|\beta|\gamma$, que significam respectivamente, ambiente de produção, restrições tecnológicas e objetivo a ser alcançado. Todos eles serão definidos a seguir.

2.1 Ambientes de Produção: α

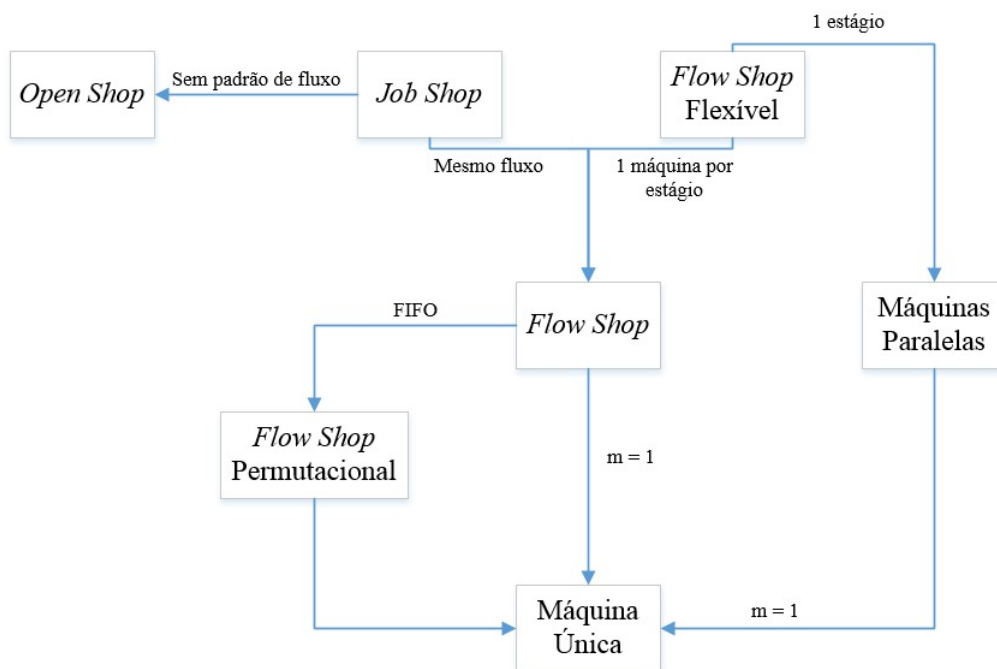
O primeiro dos termos, α , representa o ambiente em que a produção ocorrerá. Este identificador define a lógica de fluxo dos produtos em relação às máquinas ou estágios de produção. Pode-se denominar este conjunto como operações de ambientes de máquinas. Para a melhor compreensão, a seguir foram definidos os principais ambientes encontrados na literatura (MACCARTHY; LIU, 1993).

- **Máquina Única** (1): Ambiente composto por apenas uma máquina, pode ser considerado o tipo mais simples entre todos os ambientes.
- **Máquinas Paralelas** (P_m): Existem m máquinas que realizam o mesmo tipo de processo alocadas em paralelo. Cada tarefa precisa ser processada por apenas uma das máquinas, apenas uma vez.
- **Flow Shop** (F_m): Existem m máquinas em série neste ambiente. Todas as tarefas precisam ser processadas em todas as máquinas em uma mesma sequência pré estipulada.

- ❑ **Flow Shop Flexível (FF_c)**: Semelhante ao Flow Shop, porém existem m máquinas em paralelo em cada estágio de produção c .
- ❑ **Job Shop (J_m)**: Existem m máquinas e as tarefas possuem rotas particulares de produção. Não existe máquina inicial, nem máquina final.
- ❑ **Open Shop (O_m)**: Ambiente composto por m máquinas. Não há nenhuma restrição em relação à rota que a tarefa percorrerá. O programador deve definir a rota de cada tarefa.

Os tipos de ambientes de produção podem possuir várias semelhanças, sendo assim, simples de se relacionar. A Figura 1 mostra quais as diferenças e o quão distante estes ambientes se apresentam em termos de disposição das máquinas e do fluxo das tarefas.

Figura 1 – Relação entre os ambientes de produção



Fonte: Adaptado de Maccarthy e Liu (1993).

2.2 Restrições Tecnológicas: β

O segundo termo do padrão proposto por Graham et al. (1979), β , representa as restrições tecnológicas em que o ambiente de produção se encontra. Tal campo pode possuir quantidades ilimitadas de restrições, e sendo particular ao sistema produtivo estudado. A seguir estão indicadas as restrições comumente encontradas nos ambientes de produção (PINEDO, 2008).

- **Setup dependente da sequência** (S_j, S_{ij}, S_{ijk}): Representa o tempo de *Setup* quando ocorre a transição entre as tarefas **i** e **j**. Ocorre devido à troca de ferramentas, limpeza, ajustes, entre outros. Se o tempo de *Setup* também depende das máquinas, é adicionado o índice **k** à variável.
- **Preempção** ($Prmp$): O processamento da tarefa pode ser interrompido antes de seu término naquela máquina. A tarefa pode ser armazenada para que seja finalizada posteriormente. Tal tipo de restrição ocorre geralmente em processamentos computacionais.
- **Precedência** ($Prcc$): Uma tarefa **j** qualquer necessita que uma ou mais tarefas estejam prontas para que seu processamento possa ocorrer.
- **Data de Liberação** (r_j): O processamento da tarefa **j** não pode ser iniciado antes de sua data de liberação r_j .
- **Permutação** ($Prmu$): Restrição exclusiva dos ambientes *FS*. O sistema produtivo segue obrigatoriamente a filosofia *First In First Out (FIFO)*, ou seja, a sequência das tarefas se mantém a mesma até o fim.
- **Bloqueio** ($Block$): Estoque intermediário limitado ou nulo (*zero buffer*) entre as máquinas. Se o estoque estiver cheio ou não existir estoque e a máquina seguinte estiver ocupada, a máquina anterior a este estoque não pode liberar uma tarefa já processada. Tal bloqueio faz com que a máquina anterior ao estoque seja impossibilitada de ser preparada (*setup*) ou de processar a próxima tarefa.
- **Sem Espera** ($No - wait$): O fluxo de uma tarefa deve ser contínuo, não podendo haver intervalo de tempos entre os processos de cada máquina. As metalúrgicas são um exemplo clássico deste tipo de restrição, pois os metais tem suas propriedades físicas alteradas caso fiquem ociosos.
- **Espera Máxima** (a_{ijt}): A tarefa possui tempo máximo de espera para inicializar seu processamento. É conhecido também como tempo de deterioração.
- **Indisponibilidade de máquina** ($brkdw$): As máquinas estão suscetíveis a quebras aleatórias, ocasionando indisponibilidade.
- **Recirculação** ($Rcrc$): Pode ocorrer em *Job Shops* quando uma tarefa passa por uma máquina mais de uma vez.

A partir das informações obtidas nesta seção, é possível concluir que quanto maior a complexidade do espaço fabril, mais restrições serão consideradas em seus cálculos. A adição excessivas destas limitações aumenta o custo computacional na resolução do problema, tornando inviável em termos de tempo de processamento. Portanto, é interessante que apenas as restrições mais importantes sejam consideradas na formulação do problema.

2.3 Objetivos do Sequenciamento: γ

O último termo da padronização de Graham et al. (1979), γ , representa o objetivo que se pretende alcançar. Geralmente esses objetivos são baseados em redução de custo ou tempo. Os objetivos mais utilizados estão descritos na sequência.

- ❑ **Makespan** (C_{max}): Representa o tempo de término da última tarefa na última máquina, ou seja, o tempo total de finalização do conjunto de tarefas.
- ❑ **Tempo total de fluxo** ($\sum C_{jm}$): Soma do tempo de finalização de todas as tarefas.
- ❑ **Tempo total de fluxo ponderado** ($\sum C_{jm}w_j$): Soma do tempo de finalização de todas as tarefas com pesos atribuídos. Geralmente este o peso é relacionado ao custo para manter a tarefa no ambiente de produção.
- ❑ **Atraso total** ($\sum T_j$): Soma dos tempos computados de atraso de todas as tarefas.
- ❑ **Atraso total ponderado** ($\sum T_jw_j$): Soma dos tempos computados de atraso de todas as tarefas com pesos atribuídos. A atribuição dos pesos está relacionada à multas e perda de confiabilidade do cliente.
- ❑ **Atraso Máximo** (T_{max}): Representa o maior atraso computado entre todas as tarefas.
- ❑ **Número de Tarefas Atrasadas** ($\sum U_j$): Corresponde à soma do número de tarefas que extrapolaram o prazo de finalização.
- ❑ **Adiantamento Total** ($\sum E_j$): Corresponde à soma dos tempos remanescentes da finalização de uma tarefas em relação ao seu prazo de finalização.
- ❑ **Adiantamento Máximo** (E_{max}): Representa o máximo adiantamento computado entre as tarefas em relação aos seus prazos de finalização.

Muitos outros objetivos menos comuns podem ser encontrados na literatura (por exemplo, número total de tarefas atrasadas ponderadas) e até mesmo estudos que visam focar em múltiplos objetivos (por exemplo, minimizar o *TFT* juntamente com o tempo total de atraso). Quando este tipo de FO é explorada, cada parte desta é ponderada e somada.

2.4 Definição do Problema

De acordo com a notação de Graham et al. (1979) os quatro problemas abordados neste trabalho foram:

- ❑ $F_m|block|\sum C_{jm}$: *Flowshop* com bloqueio visando minimizar o tempo total de fluxo;

- $F_m|block|\sum T_j$: *Flowshop* com bloqueio visando minimizar o tempo total de atraso;
- $F_m|block, S_{ijk}|\sum C_{jm}$: *Flowshop* com bloqueio e *setup* visando minimizar o tempo total de fluxo;
- $F_m|block, S_{ijk}|\sum T_j$: *Flowshop* com bloqueio e *setup* visando minimizar o tempo total de atraso.

2.5 Terminologia

A seguir, para a definição matemática de cada um dos problemas, foram apresentados os termos necessários para o cálculo das funções objetivo.

n : Número de tarefas;

m : Número de máquinas;

p_{jk} : Tempo de processamento da tarefa j na máquina k ;

s_{ijk} : Tempo de *setup* entre as tarefas i e j na máquina k ;

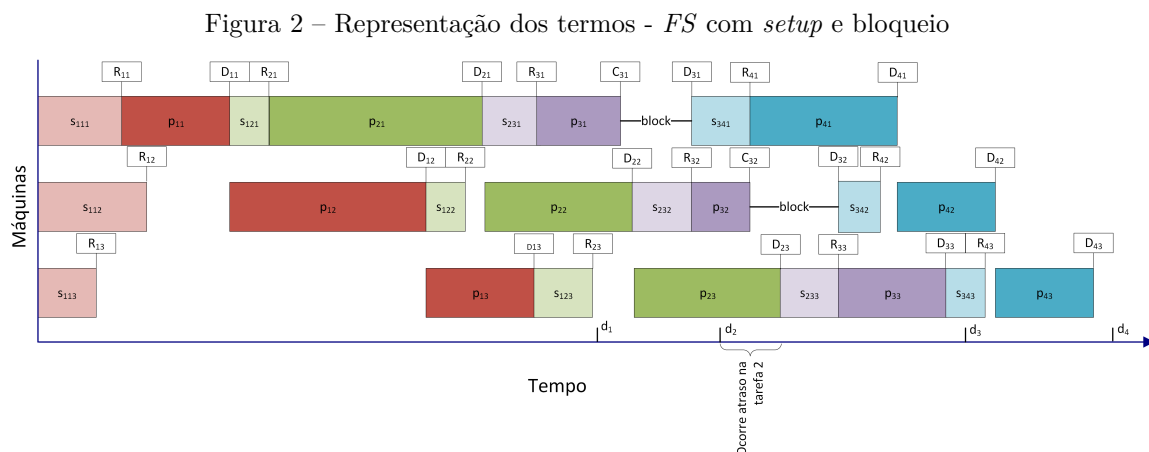
C_{jk} : Momento de finalização da tarefa j na máquina k ;

D_{jk} : Momento de saída da tarefa j na máquina k (Variável utilizada na ocorrência de bloqueio);

R_{jk} : Término do *setup* da máquina k para execução da tarefa j ;

d_j : *Due Date* da tarefa j ;

A figura 2 demonstra cada termo apresentado num exemplo prático de *FS* com bloqueio e *setup*, no qual $n = 4$ e $m = 3$.



Fonte: Figura elaborada pelo autor.

É possível perceber que ocorre atraso na tarefa 2, pois o seu tempo de liberação na última máquina, D_{23} , é maior que sua *due date*, d_2 . Para os casos onde há a ocorrência de bloqueio, representado pela palavra *block*, é possível perceber que o tempo de finalização de uma tarefa, C_{jk} , é menor que seu tempo de liberação, D_{jk} . Na imagem, são representados apenas os tempos de finalização que não coincidem com os tempos de liberação.

2.6 Cálculo das Funções Objetivo

O primeiro passo para a obtenção das duas FOs, TFT e TT , é calcular quais são os tempos de liberação de cada tarefa na última máquina (D_{jm}). As equações a seguir indicam como obter os valores destas variáveis para o caso sem *setup*.

$$D_{10} = 0 \quad (1)$$

$$D_{1k} = \sum_{q=1}^k p_{1q} \quad (k = 1, \dots, m-1) \quad (2)$$

$$D_{j0} = D_{j-1,1} \quad (j = 2, \dots, n) \quad (3)$$

$$D_{jk} = \max(D_{j,k-1} + p_{jk}, D_{j-1,k+1}) \quad (j = 2, \dots, n), (k = 1, \dots, m-1) \quad (4)$$

$$D_{jm} = D_{j,m-1} + p_{jm} \quad (j = 1, \dots, n) \quad (5)$$

A Equação 1 indica que a primeira tarefa se iniciará no instante 0. A Equação 2 representa a data de liberação da primeira tarefa em todas as máquinas, salvo a última. A Equação 3 indica o instante em que as tarefas $j = 2, \dots, n$ poderão ser iniciadas na primeira máquina. A Equação 4 indica que a tarefa j na máquina k só será liberada no instante em que este estiver pronto e a próxima máquina já estiver liberada. A Equação 5 calcula o tempo de liberação das tarefas na última máquina (RONCONI; ARMENTANO, 2001).

Para se obter os tempos de liberação das tarefas na última máquina nos casos com tempo de *setup*, é necessário o cálculo da variável que indica o momento de término da preparação das máquinas, R_{jk} . As equações a seguir mostram como este cálculo é realizado.

$$R_{1k} = s_{01k} \quad (k = 1, \dots, m) \quad (6)$$

$$D_{j1} = \max(R_{j2}, R_{j1} + p_{j1}) \quad (j = 1, \dots, n) \quad (7)$$

$$R_{jk} = D_{ik} + s_{ijk} \quad (j = 2, \dots, n) \quad (k = 1, \dots, m) \quad (8)$$

$$D_{jk} = \max(R_{j,k+1}, D_{j,k-1} + p_{jk}) \quad (j = 1, \dots, n) \quad (k = 2, \dots, m-1) \quad (9)$$

$$D_{jm} = D_{j,m-1} + p_{jm} \quad (j = 1, \dots, n) \quad (10)$$

Inicialmente, a partir da equação 6, a data de término da primeira tarefa em todas as máquinas é calculada. Em seguida, a equação 7 indica o cálculo do tempo de término do processamento das tarefas na primeira máquina. A equação 8 calcula os tempos de preparação das máquinas para as demais tarefas. A equação 9 indica o cálculo dos instantes de liberação das tarefas a partir da segunda até a penúltima máquina. E, finalmente, a partir da equação 10 é possível se obter o tempo de liberação de cada tarefa na última máquina (TAKANO; NAGANO, 2017).

Após a realização dos cálculos das datas de liberação, é possível calcular o *TFT* de uma sequência, com e sem *setup*, a partir da equação 11.

$$TFT = \sum_{j=1}^n D_{jm} \quad (11)$$

De forma semelhante, pode-se calcular o *TT* a partir da equação 12.

$$TT = \sum_{j=1}^n \max(D_{jm} - d_j, 0) \quad (12)$$

É importante ressaltar que o resultado do *TT* é estritamente positivo, por isto é necessário o uso da função $\max(x, y)$ que retorna o maior valor entre os parâmetros x e y . Com a terminologia e todos os cálculos fundamentais dos problemas definidos, se tornam possíveis aplicações de métodos de resolução.

Revisão Bibliográfica

A teoria do *scheduling* vem recebendo muita atenção desde a década de 1950. Muitos livros abordam este assunto por conta de sua importância no aumento do desempenho da produção de bens e serviços (MACCARTHY; LIU, 1993). Este capítulo descreve limitantes encontrados na literatura que foram elaborados para ambientes semelhantes aos tratados neste trabalho.

3.1 Ignall e Schrage (1965)

Ignall e Schrage (1965) foram os primeiros a utilizar o método *B&B* para problemas de sequenciamento de tarefas em um *FS* com três máquinas e n tarefas. Para diminuir a quantidade de nós criados e tomar um caminho mais lógico, os autores desenvolveram um *LB* baseado nas máquinas como gargalo do processo. Este limitante faz uma relaxação em que todas as máquinas possuem capacidade infinita para os trabalhos da sequência não parcial (*NPS*), com exceção da máquina k . Para cada máquina é calculado o tempo total de processamento da *NPS* e é somado ao menor tempo de processamento para ser processadas nas máquinas seguintes. A equação 13 descreve o cálculo do *LB*.

$$LB_{Ignall} = \max_{1 \leq k \leq m} \left(C_{|PS|,k} + \sum_{j \in NPS} p_{jk} + \min_{j \in NPS} \sum_{i=k+1}^m p_{ji} \right) \quad (13)$$

Onde $C_{|PS|,k}$ é o fim da sequência parcial (*PS*) na máquina k e $|PS|$ é a cardinalidade da *PS*. Os autores também propuseram uma regra de dominância na qual caso os nós possuírem as mesmas tarefas em permutações diferentes nas suas sequências parciais, (σ_1 e σ_2) e atendam $C_{|PS|,k}(\sigma_1) \leq C_{|PS|,k}(\sigma_2) \forall k$, então o nó 1 domina o nó 2 e este último pode ser excluído. A regra de exploração utilizada neste caso foi a de melhor limitante.

3.2 Bansal (1977)

Bansal (1977) aborda um $B\mathcal{E}B$ num FS permutacional com o objetivo de minimizar o TFT para m máquinas. Neste artigo é proposto um LB em que as tarefas ainda não sequenciadas são indexadas pela *shortest processing time first* (SPT), pois esta minimiza o TFT em máquina única. A ideia é somar o tempo estimado em que cada tarefa pode ter seu processamento iniciado na máquina k , com a soma dos tempos de processamento em SPT também em k , e por fim, adicionar o tempo de processamento dos trabalhos não sequenciados da máquina $k + 1$ até a última máquina. O LB é descrito pela equação 14.

$$LB_{Bansal} = Q(\sigma) + \max_{1 \leq k \leq m} \left((n - |PS|) \max_{1 \leq r \leq k} \left(q_r + \min_{j \in NPS} \sum_{l=r}^{k-1} p_{jl} \right) + \sum_{j=|PS|+1}^n (n - j + 1) p_{jk} + \sum_{j=|PS|+1}^n \sum_{r=k+1}^m p_{jr} \right) \quad (14)$$

Onde $Q(\sigma)$ é o TFT da sequência parcial do nó. O autor realiza os experimentos computacionais para pequenos problemas apenas para testar os conceitos matemáticos. Não são realizadas análises de esforço computacional.

3.3 Rios-Mercado e Bard (1999)

Rios-Mercado e Bard (1999) fazem uma adaptação do LB para o $B\mathcal{E}B$ de Ignall e Schrage (1965) incluindo tempos de *setup* separadamente dos tempos de processamento e dependentes da sequência. Os autores consideram os tempos de *setup* como sendo os mínimos para cada máquina, que são obtidos através da resolução de um Problema do Caixeiro Viajante Assimétrico. Como este problema é de implementação complexa e aumenta consideravelmente o tempo de processamento, é necessário o uso de um limitante forte para que o cálculo seja viável. O LB pode ser descrito pela equação 15.

$$LB_{RiosMercado} = \max_{1 \leq k \leq m} \left(C_{|PS|,k} + \sum_{j \in NPS} p_{jk} + \min_{j \in NPS} \sum_{i=k+1}^m p_{ji} + L_k(NPS) \right) \quad (15)$$

Onde $L_k(NPS)$ é a soma dos valores dos mínimos *setups* do conjunto de tarefas na sequência não parcial na máquina k . O autor utiliza uma regra de dominância dos nós em que, caso as sequências parciais dos nós possuam as mesma tarefas em permutações diferentes, porém, com o mesmo trabalho na última posição e $C_{|PS|,k}(\sigma_1) \leq C_{|PS|,k}(\sigma_2) \forall k$, então σ_1 domina σ_2 e este último é deletado. A regra de exploração utilizada é a de melhor limitante.

3.4 Ronconi e Armentano (2001)

Ronconi e Armentano (2001) concluíram um estudo que utilizou um $B\mathcal{E}B$ para solucionar um problema de FS com bloqueio, além de desenvolver quatro novos LB baseados nas datas de término das tarefas. O método utilizado inverte a lógica convencional do $B\mathcal{E}B$ e calcula a NPS isolada. Em seguida, com valores obtidos a partir da sequência não parcial, estima-se os tempos de término das tarefas na sequência parcial. O LB é representado pela equação 16.

$$LB_{NPS}^i = \max_{1 \leq k \leq m} \left(\sum_{q=1}^k p_{1q} + \sum_{r=2}^i \max(p_{rk}, p_{r-1, k+1}) + \sum_{q=k+1}^m p_{iq} \right) \quad \forall i \in NPS \quad (16)$$

Então, o atraso total das tarefas nunca será menor que: $\sum_{i=1}^{|NPS|} (LB_{NPS}^i - d_{s1(i)})$

Onde $s1$ é o conjunto de *due dates* ordenados pela regra *Earliest Due Date First (EDD)*.

Para a estimativa das datas de finalização na última máquina na sequência parcial, é necessário obter a estimativa da última tarefa da NPS em cada máquina. As equações 17, 18 e 19 realizam este cálculo.

$$LD_{q1} = \sum_{q=1}^k p_q^1 + \max_{1 \leq q \leq k-1} \sum_{g=2}^{|NPS|} \max(p_q^g, p_{q+1}^g) \quad (17)$$

$$LD_{q2} = \sum_{q=1}^k p_q^1 + \sum_{g=1}^{|NPS|-1} \max(p_k^g + 1, p_{k+1}^g) \quad (18)$$

$$LD_k = \max(LD_{q1}, LD_{q2}) \quad \forall k \quad (19)$$

Onde, p_k^g é o g -ésimo menor tempo de processamento na máquina k entre todas as tarefas da NPS . É interessante notar que a equação 19 pode ser utilizada para calcular o LB para o *makespan*, pois estima a data de liberação do último trabalho na NPS . Com a equação 19 realizada, é possível calcular a estimativa do tempo de liberação da última tarefa da NPS na máquina k .

A partir das estimativas realizadas pela 19, é possível calcular da data da liberação das tarefas na PS a partir de:

$$D_{j0} = LD_1 \quad (j = |NPS| + 1) \quad (20)$$

$$D_{jk} = \max(D_{j, k-1} + p_{jk}, LD_{k+1}) \quad (j = |NPS| + 1) \quad (k = 1, \dots, m-1) \quad (21)$$

$$D_{j0} = D_{j-1,1} \quad (j = |NPS| + 2, \dots, n) \quad (22)$$

$$D_{jk} = \max(D_{j,k-1} + p_{jk}, D_{j-1,k+1}) \quad (j = |NPS| + 2, \dots, n) \quad (k = 1, \dots, m-1) \quad (23)$$

$$D_{jm} = D_{j,m-1} + p_{jm} \quad (j = |NPS| + 1, \dots, n) \quad (24)$$

$$D_{jm} = D_{j,m-1} + p_{jm} \quad (j = |NPS| + 1, \dots, n) \quad (25)$$

Então, para definir as *due dates* na *PS*:

$$LB_{PS} = \sum_{j=|NPS|+1}^n (D_{jm} - d_j) \quad (26)$$

$$LB_{RonconiTT} = LB_{PS} + LB_{NPS} \quad (27)$$

Os autores utilizam a regra de exploração por profundidade, na qual o próximo nó a ser ramificado é o que possui o maior nível. Em caso de empates, o nó com o menor *LB* é o escolhido. Uma heurística desenvolvida por Nawaz, Jr e Ham (1983) foi utilizada para a geração do *UB* inicial.

3.5 Chung, Flynn e Kirca (2002)

Chung, Flynn e Kirca (2002) propuseram uns novos *LBs* para *BEB* visando diminuir o *TFT* com e sem peso de um *FS* permutacional. Para o desenvolvimento do *LB* foi considerada que uma das máquinas possui capacidade unitária (um trabalho por vez) e todas as outras têm capacidades infinitas (n trabalhos por vez, ou seja, trabalhos não necessitam esperar por máquinas vagas). Logo, é possível se obter um *LB* do *TFT* solucionando o *TFT* de um problema de máquina única. Para a resolução, utiliza-se a conhecida *SPT*, que dará a melhor sequência visando minimizar o *TFT*. O cálculo do *LB* é representado pela equação 28.

$$LB_{ChungTFT} = Q(\sigma) + \max_{1 \leq k \leq m} \sum_{t=s+1}^n \left(\max \left(q_k + \sum_{j=|PS|+1}^t p_{jk}; q_{k-1} + \sum_{j=|PS|+1}^t p_{j,k-1} + p_{tk} \right) + \sum_{r=k+1}^m p_{jr} \right)$$

(28)

Onde $C_{|PS|,k}$ é o tempo de finalização do último trabalho da PS na máquina k . Os autores utilizam a regra de exploração que prioriza os nós de maior nível com *backtracking*, que após ramificar o primeiro nó a partir de um nó pai, este já é imediatamente ramificado, encontrando um UB em pelo menos n recursões. Esta técnica é eficiente pela sua economia de memória, pois no máximo $n - 1$ nós estarão na lista de nós ativos.

3.6 Ronconi (2005)

Ronconi (2005) voltou a utilizar o método $B\acute{e}B$ para encontrar a solução ótima em FS com bloqueio. Este estudo visa a minimização do *makespan* a partir do cálculo de um novo LB que explora a ocorrência de bloqueio entre as tarefas. A ideia foi iniciada na publicação de Ronconi e Armentano (2001), na qual os autores desenvolveram uma formulação para o cálculo do tempo de liberação da última tarefa em cada uma das máquinas para o cálculo do TT e sugerem a adaptação para *makespan*. O LB foi adaptado e é representado pela equação 29.

$$LB_{RonconiC_{max}} = \max_{1 \leq k \leq m} \left(D_{|PS|,k} + \sum_{g=1}^{|NPS|} \max(a_k^g, b_{k+1}^g) + \sum_{q=k+1}^m p_{min_q} \right) \quad (29)$$

Onde a_k^g é o g -ésimo menor tempo de processamento dos trabalhos na NPS na máquina k e b_{k+1}^g é o g -ésimo menor valor de $D_{|PS|,k+1} - D_{|PS|,k}$. O trabalho apresentou melhores resultados do que o proposto por Ronconi e Armentano (2001), atingindo a resposta ótima mais rapidamente em 85,2% dos problemas propostos. A regra de exploração utilizada foi a de profundidade e a solução inicial foi adquirida através de uma heurística de inserção.

3.7 Chung, Flynn e Kirca (2006)

Chung, Flynn e Kirca (2006) estenderam o LB para TFT publicado em Chung, Flynn e Kirca (2002) para a minimização do TT . É utilizada a mesma formulação para encontrar as subestimativas das datas de liberação de cada trabalho na NPS em cada máquina. A equação 30 mostra como as datas de liberação são obtidas para cada trabalho e para cada máquina.

$$E_{tk} = \max \left(C_{|PS|,k} + \sum_{j=|PS|+1}^{t-1} p_{jk}, \sum_{j=|PS|+1}^t p_{j,k-1} \right) \quad (30)$$

Para estimar o mínimo atraso que pode ocorrer no nó considerado, utilizam-se as equações 31, 32 e 33.

$$h_{jk} = d_j - \sum_{r=k}^m p_{jr} \quad \forall j \in NPS, \quad \forall k \in m \quad (31)$$

Onde h_{jk} é o mínimo atraso que pode ocorrer em um trabalho j caso este parta da máquina k .

$$T_{jk} = \max(E_{tk} - h_{[t]k}, 0) \quad \forall j \in NPS, \quad \forall k \in m \quad (32)$$

Onde $h_{[t]k}$ é o t -ésimo menor valor de h_{tk} .

$$LB_{ChungTT} = TT_{PS} + \max_{1 \leq k \leq m} \sum_{t=|PS|+1}^n T_{tk} \quad (33)$$

Onde TT_{PS} é o atraso total da sequência parcial do nó em específico. Os autores utilizam a regra de exploração de profundidade com *backtracking*, a mesma utilizada por Chung, Flynn e Kirca (2002). Os resultados obtidos mostram que o LB para $B\mathcal{E}B$ pode solucionar problemas com $n \leq 20$ na contagem limite de nós de 4×10^6 , caso utilizadas regras de dominância.

3.8 Companys e Mateo (2007)

Companys e Mateo (2007) adaptaram a heurística NEH de Nawaz, Jr e Ham (1983) para construir a solução inicial e aplicaram a ferramenta de busca local para obter melhorias na resposta para o *makespan*. O resultado desta combinação foi utilizado como UB inicial para o problema de $B\mathcal{E}B$. O método foi aplicado em FS com e sem bloqueio. A contribuição interessante deste estudo é a aplicação do método *LOMPEN* que cria uma nova instância espelho para o problema onde os tempos de processamento das máquinas são invertidos. Por exemplo, a máquina $p_{jm} = p_{j1}$, $p_{j(m-1)} = p_{j2}$ e assim sucessivamente para todas as máquinas e todas tarefas. A regra de exploração utilizada foi elaborada pelo autor e depende do número de nós ativos que o problema possui. Dado o número de nós ativos N e dois números pré-fixados N_0 e N_1 :

- Se $0 < N < N_1$: O nó com o menor LB no menor nível é escolhido;
- Se $N_0 \leq N < N_1$: O nó com menor LB é escolhido e os desempates são realizados pelo maior nível;
- Se $N_1 \leq N$: O nó de maior nível é o escolhido e o desempate é feito pelo nó de menor LB .

Para o cálculo do LB são escolhidas duas máquinas entre todas as m e é aplicada a regra de Johnson (1954) que calcula a solução ótima no FS com duas máquinas. O cálculo é realizado para todas as máquinas u e v onde $u < v$ e para todos os trabalho da sequência não parcial. O maior valor encontrado entre as combinações de u e v é utilizado como LB .

3.9 Moslehi e Khorasanian (2013)

Moslehi e Khorasanian (2013) utilizaram o $B\mathcal{E}B$ para encontrar soluções ótimas em ambientes de FS com bloqueio visando minimizar o TFT das tarefas. Para o procedimento foram desenvolvidos três LBs e foi utilizado um método heurístico NEH-MK, que é uma modificação do clássico NEH de Nawaz, Jr e Ham (1983), para o UB . Os LBs desenvolvidos são derivados uns dos outros. A equação 34 apresenta o cálculo do primeiro LB , que é base para o segundo e para o terceiro.

$$LB_{Moslehi} = TFT_{PS} + \max_{1 \leq k \leq m} \left(\sum_{t=|PS|+1}^n ED_{tk} + \sum_{t \in NPS} \sum_{r=k+1}^m p_{jr} \right) \quad (34)$$

Onde os valores de ED_{tk} são as subestimativas de liberação do trabalho t na máquina k e são definidos a partir de 35 e 36.

$$ED_{tm}^{LB1} = D_{|PS|,m} + \sum_{r=1}^{t-|PS|} p_{[r]m} \quad \forall t \in NPS \quad (35)$$

$$ED_{tk}^{LB1} = D_{|PS|,k} + \sum_{r=1}^{t-|PS|} \max(p_{[r]k}, Q_{t,k+1}(r)) \quad \forall t \in NPS \quad (36)$$

Onde $Q(r)$ é o r -ésimo menor termo do conjunto Q onde:

$$Q_{k,t} = \{D_{s,k} - Ds, k - 1\} \cup \{p[r]k | 1 \leq r \leq t - |PS| - 1\} \quad \forall t \geq |PS| + 2 \quad (37)$$

O segundo LB é calculado de forma semelhante, porém o termo ED é atualizado de forma a considerar o bloqueio na máquina seguinte. A equação 38 descreve a alteração do segundo LB .

$$ED_{tk}^{LB2} = \max(ED_{t-1,k+1}^{LB2}, ED_{tk}^{LB1}) \quad \forall t \geq |PS| + 2; \quad \forall k \leq m - 1. \quad (38)$$

A terceira atualização do LB é semelhante à utilizada por Bansal (1977) e considera a partida do trabalho da máquina y somada com seus tempos de processamento até a

máquina k como sendo o valor mínimo de início do trabalho t na máquina k . A equação 39 descreve o terceiro e último LB apresentado por Moslehi e Khorasanian (2013).

$$ED_{tk}^{LB3} = \max\left(\max_{1 \leq y \leq k-1} \left(ED_{ty} + \min_{r=y+1}^k p_{tr}\right), ED_{tk}^{LB2}\right) \quad \forall t \geq s+2; \quad \forall 2 \leq k. \quad (39)$$

De acordo com as equações apresentadas é possível perceber que o valor de $ED_{tk}^{LB1} \leq ED_{tk}^{LB2} \leq ED_{tk}^{LB3}$ para qualquer situação. Por outro lado, a complexidade computacional também obedece $O(ED_{tk}^{LB1}) \leq O(ED_{tk}^{LB2}) \leq O(ED_{tk}^{LB3})$. Então os autores buscam encontrar o melhor custo benefício entre os três LBs . Os autores concluíram que ED_{tk}^{LB2} possui o melhor desempenho em termos de tempo e ED_{tk}^{LB2} em número de nós. Os métodos solucionaram problemas de até 18 tarefas e 10 máquinas no tempo limite de 3600 segundos.

3.10 Jesus e Rodrigues (2014)

Jesus e Rodrigues (2014) propuseram um LB para FS permutacional com tempos de *setup* dependentes apenas das máquinas visando a minimização do *makespan*. A equação 40 apresenta o cálculo do LB elaborado.

$$b_j^* = D_{qk,j} + (\theta_T^{qk} - sc_{qk}^j) + R_j \quad \forall j \leq m \quad (40)$$

Onde, $D_{qk,j}$ é a data de liberação do último trabalho da sequência parcial do nó na máquina j . θ_T^{qk} é a soma dos menores *setups* da máquina j . sc_{qk}^j são os tempos *setups* das tarefas da sequência parcial na máquina j e R_j é a soma dos tempos de processamento das tarefas ainda não sequenciadas no nó em específico. O valor final do LB é dado pela equação 41.

$$LB_{Jesus} = \max_{1 \leq j \leq m} (b_j^*) \quad (41)$$

Os autores alcançaram as respostas ótimas até quatro tarefas e duas máquinas, logo não foram testados problemas suficiente grandes para verificar a eficiência do algoritmo.

3.11 Sanches, Takano e Nagano (2016)

Sanches, Takano e Nagano (2016), propuseram um trabalho onde as mais eficientes heurísticas desenvolvidas para realizar a redução do *makespan* em FS com bloqueio foram utilizadas para gerar uma resposta inicial, ou UB , para um $B\&B$. Entre elas estão:

□ *MinMax (MM)* de Ronconi (2004);

- *Profile Fitting (PF)* de McCormick et al. (1989);
- *Weighted Profile Fitting (wPF)* de Pan e Wang (2012).
- *Pan and Wang (PW)* de Pan e Wang (2012).

O *LB* utilizado foi o desenvolvido por Ronconi (2005) e a regra de exploração foi a de profundidade. Os resultados mostraram que o uso de um *UB* reduz consideravelmente o tempo de processamento de cada um dos problemas. Para poucas máquinas a heurística *MM* foi a que obteve melhores resultados como limitante superior e conforme a quantidade de máquinas aumenta, *PW* se mostrou mais eficiente.

3.12 Takano e Nagano (2017)

Takano e Nagano (2017), foram os primeiros a abordarem o método *B^ℓB* para *makespan* com as restrições de *setup* e bloqueio. Os autores utilizam uma nova abordagem capaz de estimar o tempo mínimo de bloqueio e o tempo máximo de ociosidade entre duas tarefas. A equação 42 mostra como o tempo mínimo de bloqueio pode ser calculado.

$$LBB_{j+1}^k = \max\left(0, (p_{j,k+1} + s_{j,j+1,k+1}) - (UBO_{j+1}^k + s_{j,j+1,k} + p_{j+1,k})\right) \quad (42)$$

Com $LBB_{j+1}^m = 0$ e UBO_{j+1}^k é o tempo máximo de ociosidade da máquina k e é calculado a partir da equação 43.

$$UBO_{j+1}^k = \max\left(0, (UBO_{j+1}^{k-1} + s_{j,j+1,k-1} + p_{j+1,k-1}) - (p_{jk} + s_{j,j+1,k})\right) \quad (43)$$

Onde $UBO_{j+1}^1 = 0$. Considere que $s_{-}s_{ijk}$ seja a soma dos menores *setups* entre os trabalhos na *NPS* e $s_{-}LBB_{ijk}$ seja a soma dos menores tempos mínimos dos bloqueios entre os trabalhos da *NPS* na máquina k . Os termos são equacionados em 44 e 45.

$$s_{-}s_{ijk} = \sum_{i \in NPS \cup |PS|} \left(\min(s_{i,NPS,k}) \right) - \max\left(\min(s_{NPS,NPS,k}) \right) \quad \forall k \leq m - 1 \quad (44)$$

$$s_{-}LBB_{ijk} = \sum_{i \in NPS \cup |PS|} \left(\min(LBB_{i,NPS}^k) \right) - \max\left(\min(LBB_{NPS,NPS}^k) \right) \quad (45)$$

A equação 46 calcula o a soma dos tempos de processamento somados com os possíveis bloqueios estimados na equação 42 da última tarefa em todas as máquinas a partir de $k + 1$.

$$LW_k = \min\left(\sum_{q=k+1}^m (p_{|NPS|,q}) + \min\left(\sum_{q=k+1}^m (LBB_{|NPS|,|NPS|}^q) \right) \right) \quad \forall k \leq m - 1 \quad (46)$$

O último trabalho da *NPS* é retirado e dois novos conjuntos são definidos a partir dos mesmos cálculos realizados em 44 e 45. Estes novos conjuntos são chamados de s_sn_{ijk} e s_LBBn_{ijk} .

A partir das propriedades definidas, os autores desenvolveram quatro variações do *LB* para *setup* que utilizam o mesmo cálculo para a última máquina. Este cálculo é representado pela equação 46.

$$LB_m = C_{PS,m} + s_s_{ijm} + \sum_{h \in NPS} p_{hm} \quad (47)$$

Para as máquinas restantes os cálculos são representados a seguir.

$$LB_k^{TN1} = C_{PS,k} + s_s_{ijk} + \sum_{h \in NPS} p_{hk} + s_LBB_{ij}^k + \min_{g \in NPS} \left(\sum_{q=k+1}^m (p_{gq} + \min(LBB_{gg}^q)) \right) \quad \forall k \leq m-1 \quad (48)$$

$$LB_k^{TN2} = C_{PS,k} + s_s_{ijk} + \sum_{h \in NPS} p_{hk} + s_LBB_{ij}^k + \min_{g \in NPS} \left(\sum_{q=k+1}^m (p_{gq}) + \min \sum_{q=k+1}^m (LBB_{gg}^q) \right) \quad \forall k \leq m-1 \quad (49)$$

$$LB_k^{TN3} = C_{PS,k} + s_s_{ijk} + \sum_{h \in NPS} p_{hk} + s_LBB_{ij}^k + s_LBBn_{ij}^k + LW_k \quad \forall k \leq m-1 \quad (50)$$

$$LB_k^{TN4} = C_{PS,k} + s_s_{ijk} + \sum_{h \in NPS} p_{hk} \quad \forall k \leq m-1 \quad (51)$$

Os autores afirmam que o melhor *LB* encontrado foi LB_k^{TN2} , seguido por LB_k^{TN1} , LB_k^{TN4} e LB_k^{TN3} . O melhor *LB* atingiu o ótimo em 75% menos tempo que o pior dos limitantes. Este trabalho também apresenta modelagens de programação linear inteira mista para o problema abordado.

3.13 Toumi et al. (2017a)

Toumi et al. (2017a), aumentaram em termos matemáticos o valor do *LB* para *makespan* desenvolvido por Ronconi (2005). Os autores provam que o último termo do *LB*

pode ter seu valor aumentado e continuar sendo um limitante. A equação 52 mostra como fica o novo termo.

$$\sum_{q=k+1}^m p_{min_q} \leq p_{min,k+1} + \min_{i \in NPS} \sum_{j=k+2}^m p_{ji} \quad (52)$$

Todo o restante da estrutura foi utilizado como o de Ronconi (2005), salvo o UB que foi definido como infinito. Os autores mostraram que o limitante reduz o número de nós porém o custo computacional do cálculo se mostrou maior que o de Ronconi (2005), logo, não conseguiram superá-la no tempo de execução.

3.14 Toumi et al. (2017b)

Toumi et al. (2017b), adaptaram o LB de Ronconi (2005) para o FS com bloqueio a fim de diminuir o atraso total e o atraso total ponderado das tarefas. Desta vez a formulação foi adaptada para estimar o tempo de liberação na última máquina para cada uma das tarefas. A equação 53 apresenta o cálculo.

$$LB_{ToumiTT} = TT_{PS+} \max_{1 \leq k \leq m} \left(\sum_{i=|PS|+1}^n \max \left(D_{k,|PS|+} + \sum_{j=|PS|+1}^i \max(a_{k[h]}, b_{k+1,[h-1]}) - f_{k[i]}, 0 \right) \right) \quad (53)$$

Onde o termo $f_{k[i]}$ é descrito pela equação 54.

$$f_{ki} = d_i - \sum_{r=k}^m p_{ir} \quad \forall i \in NPS, \quad \forall k \in m \quad (54)$$

Os autores utilizam a mesma estrutura de $B\mathcal{E}B$ e apresentam resultados melhores que a abordagem de Ronconi e Armentano (2001), apesar de não serem feitas comparações na publicação.

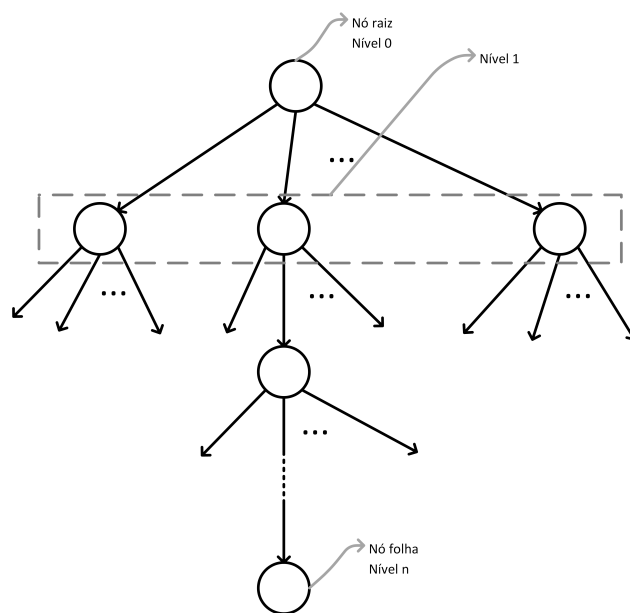
3.15 Lacunas encontradas durante a revisão bibliográfica

Após a realização da revisão da literatura relacionada diretamente ao problema abordado neste trabalho, é possível perceber que poucas publicações apresentam soluções para os casos de TFT e TT com bloqueio solucionados por $B\mathcal{E}B$. Portanto, o presente trabalho visou abordar especificamente estes casos. Para os casos com $setup$ não foram encontrados trabalhos na literatura. Logo, o intuito foi contribuir para o preenchimento desta lacuna encontrada durante a revisão bibliográfica.

O Método *Branch & Bound*

A partir de uma árvore de busca, o *B&B* é conhecido por solucionar uma variedade de problemas combinatoriais. O propósito deste método é encurtar o caminho da enumeração realizada e, conseqüentemente, reduzir o tempo computacional. Para isto, o algoritmo utiliza como estratégia reduzir o problema principal de complexidade alta, em dois ou mais subproblemas mais simples de serem solucionados. Este processo ocorre de forma recursiva, ou seja, os nós poderão ser novamente divididos de acordo com a necessidade do problema. Cada recursão realizada gera um novo nível de profundidade, com o máximo de n níveis, onde n é o número de trabalhos a serem sequenciados no problema original (BAKER, 2009). Para fins didáticos, e para abstrair o algoritmo do tipo árvore, o problema principal será definido como “nó raiz” e os subproblemas provenientes de sua divisão serão definidos simplesmente como “nó”. A Figura 3 ilustra a lógica de formação da árvore.

Figura 3 – Representação *B&B*



Fonte: Figura elaborada pelo autor.

Assim como seu nome indica, o algoritmo possui dois procedimentos principais: *Branch* (ramificar) e *Bound* (limitar) (BAKER, 2009). Segundo Ignall e Schrage (1965), um terceiro procedimento também deve ser levado em consideração durante o algoritmo: a regra de exploração dos nós. Os três componentes foram definidos nas seções 4.1, 4.2 e 4.3.

4.1 Ramificação

Segundo Baker (2009), o procedimento de ramificação tem como objetivo dividir o nó raiz em dois ou mais nós, substituindo o problema principal em novos subproblemas que possuem as seguintes características:

- (i) Mutuamente exclusivos;
- (ii) Menor tamanho e complexidade que o nó que o originou;
- (iii) Possui uma solução parcial do problema original;

Cada nó criado a partir do processo de ramificação é composto por duas partes principais: Sequência parcial (*PS*) e Sequência não parcial (*NPS*).

O número de trabalhos sequenciados de um nó é igual ao valor do seu nível de recursão. Logo, o nó raiz possui $|PS| = 0$ e $|NPS| = n$, onde $|PS|$ e $|NPS|$ representam as cardinalidades da sequência parcial e da sequência não parcial, respectivamente. O primeiro nível de ramificação terá nós com $|PS| = 1$ e $|NPS| = n - 1$, o segundo nível terá $|PS| = 2$ e $|NPS| = n - 2$ e assim sucessivamente até que $|PS| = n$ e $|NPS| = 0$. Quando $|PS| = n$, o nó respectivo será denominado como “nó folha”, pois se encontra na última ramificação da árvore, e este indica que uma solução factível para o problema original foi encontrada (CHUNG; FLYNN; KIRCA, 2002).

Caso o procedimento de ramificação ocorresse para todos os nós possíveis, $n!$ soluções factíveis seriam encontradas tornando o processo de busca demasiadamente exaustivo. Além disso, $\frac{n!}{|PS|!}$ nós seriam criados por nível. A Tabela 1 explicita quantos nós seriam criados e quantas soluções factíveis seriam encontradas.

Tabela 1 – Número de soluções e número de nós em processo de ramificação sem limitação

n	$n!$	$\sum_{ PS =1}^n \frac{n!}{ PS !}$
10	3628800	6235301
12	4.79E+08	8.23E+08
14	8.72E+10	1.5E+11
16	2.09E+13	3.6E+13
18	6.4E+15	1.1E+16
20	2.43E+18	4.18E+18

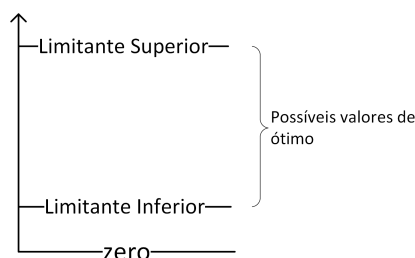
Fonte: Tabela elaborada pelo autor.

Então, pode-se perceber que o procedimento de ramificação isolado não é suficiente para diminuir a complexidade do problema original. A seção 4.2 apresenta como os limitantes são capazes de encontrar caminhos mais promissores para se chegar à solução ótima mais rapidamente.

4.2 Limitantes

Visando selecionar os ramos que possuem chance de encontrar a solução ótima, os limitantes restringem a gama de respostas que o problema pode alcançar. Caso estes não fossem utilizados, o resultado da função objetivo poderia variar entre 0 e $+\infty$. O limitante superior representa o valor máximo que a solução ótima do problema original pode alcançar e, de forma complementar, o limitante inferior define o valor mínimo que a solução parcial do nó específico pode atingir (LOMNICKI, 1965). A Figura 4 mostra como estes limitantes estão dispostos em relação ao ótimo e como a gama de respostas é reduzida, facilitando a busca ao ótimo.

Figura 4 – Limitação do ótimo



Fonte: Figura elaborada pelo autor.

Ao se diminuir o conjunto de respostas possíveis em que o ótimo está inserido, nós não promissores passam a ser retirados do problema, evitando cálculos desnecessários.

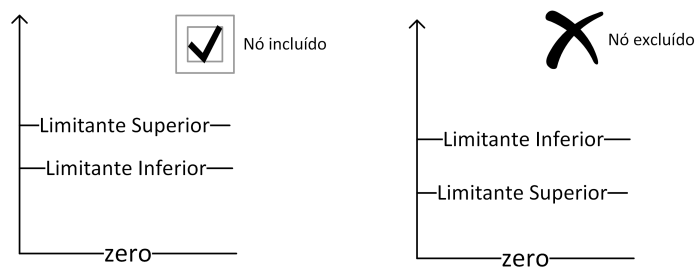
4.2.1 Limitantes Superiores

Ao se inicializar a resolução de um problema, o limitante superior é representado por valores infinitos e é atualizado sempre que soluções factíveis para o problema original forem encontradas. Como alternativa, pode-se utilizar heurísticas para se obter uma resposta inicial para o problema, tornando o limitante superior menor que infinito logo no início da resolução (SANCHES; TAKANO; NAGANO, 2016). É possível perceber que quanto menor o valor deste limitante, menor será o conjunto de candidatas ao ótimo. Tendo em vista que o foco principal deste trabalho são os limitantes inferiores, o valor do UB foi definido como infinito no início de cada problema e atualizado sempre que se encontrou um nó folha.

4.2.2 Limitantes Inferiores

Ao se criar um novo nó, seu limitante inferior é calculado e avaliado de forma que, caso $LB < UB$, este será incluído em uma lista de nós ativos, caso contrário, este nó será imediatamente excluído. Tal procedimento ocorre pois, como já mencionado, o propósito de se calcular o LB de um nó específico é verificar o mínimo que o valor de uma solução factível proveniente deste mesmo nó pode atingir. Logo, se o mínimo for maior que uma resposta já existente (UB), é impossível que a partir deste nó se alcance a solução ótima (BAKER, 2009). A Figura 5 mostra como estes nós são avaliados.

Figura 5 – Avaliação dos nós candidatos ao ótimo



Fonte: Figura elaborada pelo autor.

A não continuação no procedimento de ramificação a partir deste nó excluído encurta a busca pelo ótimo, tornando o tempo total do algoritmo significativamente menor. Os cálculos dos limitantes inferiores foram exibidos no capítulo 5 deste trabalho.

4.3 Regras de Exploração

Em uma lista de nós ativos, é necessário optar qual será o próximo nó a ser ramificado. A partir de parâmetros contidos nos nós, a regra de exploração tem a função de selecionar o próximo elemento da lista a ser dividido. Tal regra também define o critério de parada do algoritmo $B\&B$ (IGNALL; SCHRAGE, 1965). Nas subseções, estão apresentadas duas das regras mais utilizadas na resolução de problemas no ambiente FS .

4.3.1 Melhor Limitante

Ao se calcular os LB s dos nós ramificados e caso sejam definidos como ativos, estes são alocados de forma ordenada na lista de nós ativos. Para a regra de melhor limitante, esta ordenação é feita de forma não-decrescente em relação ao seu LB e o próximo nó a ser ramificado é o primeiro da lista. Em caso de empates, é selecionado o nó de maior $|PS|$. Caso o primeiro nó da lista seja um nó folha, o algoritmo $B\&B$ para, pois o resultado da PS encontrada é factível e ótimo para o problema original (BAKER, 2009). Para este caso, é necessário o cálculo de um UB inicial, pois este nunca é atualizado com uma solução factível que não seja ótima, logo, esta regra é fortemente dependente da heurística inicial.

4.3.2 Profundidade

Este método de exploração tem como prioridade de ordenação da lista de nós ativos o nível de recursão de forma não-crescente. Em caso de empates, será selecionado o nó de menor LB . Uma das vantagens desta regra é que seu número de nós na lista ativa é de no máximo n caso seja utilizada a estratégia de backtracking (CHUNG; FLYNN; KIRCA, 2002), gerando economias em termos de memória computacional para seus cálculos. Neste caso, o algoritmo não para ao se atingir um nó folha na primeira posição da lista, apenas atualiza o UB do problema original. Esta é considerada outra vantagem, pois o UB é atualizado com frequência, aumentando a exclusão de nós não promissores (CHUNG; FLYNN; KIRCA, 2006). Tendo em vista as vantagens desta regra, este será o método de exploração utilizado neste trabalho.

4.4 O Pseudocódigo

Para a apresentação do pseudocódigo, é necessária a definição da terminologia de algumas variáveis. Terminologia global:

UB : Limitante superior, ou melhor FO vigente;

S : Melhor sequência vigente;

$LB_{função}$: Cálculo do limitante inferior;

L : Lista de nós ativos $L = \{N_1, N_2, N_3, \dots\}$;

NR : Nó raiz;

NP : Nó pai, ou nó de origem de outros nós;

NN : Novo nó criado;

Terminologia intrínseca aos nós:

N_{PS} : Sequência parcial

$N_{|PS|}$: Cardinalidade da sequência parcial

N_{NPS} : Sequência não parcial

$N_{|NPS|}$: Cardinalidade da sequência não parcial

N_{LB} : Valor do LB

N_{LB} : Valor da FO, em caso de nó folha

O algoritmo 1 apresenta os passos para a obtenção da solução ótima a partir do $B\&B$.

Algoritmo 1 *Branch & Bound*

```

1: Início;
2:  $UB \leftarrow \infty$ ;
3:  $S \leftarrow \emptyset$ ;
4:  $L \leftarrow \emptyset$ ;
5: Crie  $NR$  e adicione em  $L$ ;
6: enquanto  $L \neq \{\emptyset\}$  faça
7:     Remova  $N_1$  de  $L$ ;
8:      $NP \leftarrow N_1$ ;
9:     #Procedimento de Ramificação
10:    para todo  $j$  em  $NP_{NPS}$  faça
11:         $NN_{PS} \leftarrow NP_{PS} + j$ ;
12:         $NN_{NPS} \leftarrow NP_{NPS} - j$ ;
13:        #Procedimento de Limitação
14:         $NN_{LB} \leftarrow LB_{função}$ ;
15:        se  $NN_{LB} < UB$  então
16:             $NN$  não é viável;
17:        senão
18:            se  $NN_{|PS|} = n$  então
19:                Calcule  $NN_{FO}$ ;
20:                se  $NN_{FO} < UB$  então
21:                     $UB \leftarrow NN_{FO}$ ;
22:                     $S \leftarrow NN_{PS}$ ;
23:            fim se
24:        senão
25:            #Procedimento de Exploração
26:            Adicione  $NN$  de forma ordenada em  $L$ ;
27:        fim se
28:    fim se
29: fim para
30: fim enquanto
31: retorna  $UB$  e  $S$ ;
32: Fim;

```

Propostas de Novos Limitantes Inferiores

Um limitante inferior de um problema é uma estimativa do mínimo valor que a FO deste pode atingir. Para um mesmo problema existem infinitos *LBs*, porém, caso válidos, o maior *LB* encontrado é sempre o de maior eficiência em termos matemáticos, pois limita de forma mais restrigente o conjunto de valores candidatos ao ótimo.

Este trabalho buscou elaborar, de forma não dispendiosa em termos computacionais, os valores dos *LBs* para as quatro variações dos problemas tratados. Então, é importante ressaltar que o objetivo do B&B é reduzir o tempo de busca do ótimo de um problema combinatório. Logo, caso um *LB* possua complexidade de cálculo demasiadamente trabalhoso, o tempo computacional utilizado ao final do problema pode não compensar se comparado a um *LB* de menor valor e de cálculo mais simples. Para o cálculo dos limitantes, são necessárias as seguintes definições:

$p_{[j]k}$: ($j - |PS|$)-ésimo menor tempo de processamento entre as tarefas da NPS na máquina \mathbf{k} ;

$s_{i[1]k}$: menor tempo de *setup* após tarefas \mathbf{i} entre todas as tarefas da NPS na máquina \mathbf{k} ;

$s_{[1]jk}$: menor tempo de *setup* que antecede a tarefa \mathbf{j} entre todas as tarefas da NPS na máquina \mathbf{k} ;

q_{jkr} : Soma dos tempos de processamento da tarefa \mathbf{j} da máquina \mathbf{k} até a máquina \mathbf{r} .

I_{jk} : Subestimativa da data de início de processamento da tarefa \mathbf{j} na máquina \mathbf{k} .

H_{jk} : Subestimativa da data de liberação da tarefa \mathbf{j} na máquina \mathbf{k} .

h_{jk} : $d_j - q_{jkm}$.

As seções 5.1, 5.2, 5.3 e 5.4 apresentarão os cálculos e as particularidades de cada *LB* elaborado.

5.1 O Limitante Inferior para *Total Flow Time*

O cálculo do *LB* para o problema de *TFT* sem *setup* tem como objetivo estimar os tempos de liberação na última máquina das tarefas na *NPS* e somá-los com o *TFT* já calculado da *PS*. As equações recursivas a seguir apresentam como esta estimativa é realizada.

$$I_{|PS|+1,1} = H_{|PS|,1} = D_{|PS|,1} \quad (55)$$

$$H_{t,1} = H_{t-1,1} + p_{[t],1} \quad (t = |PS| + 1, \dots, n) \quad (56)$$

$$I_{t,1} = \max(H_{t-1,1}, H_{t-2,2}) \quad (t = |PS| + 2, \dots, n) \quad (57)$$

$$I_{|PS|+1,k} = \max(H_{|PS|,k}, I_{|PS|+1,k-1} + p_{[1],k-1}) \quad (k = 2, \dots, m) \quad (58)$$

$$H_{|PS|,k} = I_{|PS|+1,k} \quad (k = 2, \dots, m) \quad (59)$$

$$H_{t,k} = H_{t-1,k} + p_{[t],k} \quad (t = |PS| + 1, \dots, n) \quad (k = 2, \dots, m - 1) \quad (60)$$

$$I_{t,k} = \max(H_{t-1,k}, H_{t,k-1}, H_{t-2,k+1}) \quad (t = |PS| + 2, \dots, n) \quad (k = 2, \dots, m - 1) \quad (61)$$

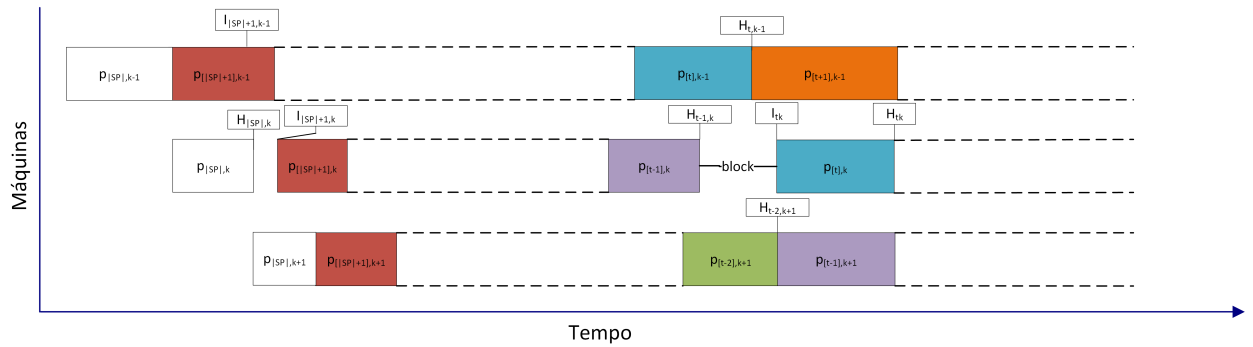
$$I_{t,m} = \max(H_{t-1,m}, H_{t,m-1}) \quad (t = |PS| + 2, \dots, n) \quad (62)$$

A equação 55 subestima o tempo em que o primeiro trabalho da *NPS* tem seu início na primeira máquina. Em seguida, na equação 56 são realizados os cálculos para se encontrar os possíveis tempos de liberação na primeira máquina. O próximo passo, equação 57, atualiza estes tempos a partir da propriedade de bloqueio. A equação 58 subestima qual o momento que a primeira tarefa da *NPS* poderá ter seu início para a máquina k . Este momento será quando a máquina estiver liberada e seu processamento finalizar na máquina anterior. Como não se sabe qual será o primeiro trabalho a ser processado, se utiliza o menor tempo de processamento na máquina $k - 1$ entre todas as tarefas. As equações 59 e 60 subestimam os tempos de liberação de cada tarefa da segunda até a penúltima máquina. A equação 61 define que o tempo mínimo para que a tarefa t possa ser processada na máquina k é quando:

- Esta mesma tarefa t tenha sido processada na máquina $k - 1$;
- A tarefa $t - 1$ tenha seu processamento finalizado em k ;
- A máquina $k + 1$ estiver desocupada para que a tarefa $t - 1$ possa ser removida de k .

Como não existem máquinas a jusante de m , esta máquina nunca terá a ocorrência de bloqueio, logo, a equação 62 considera que o momento mais cedo que a tarefa t pode ser processada é quando esta já tiver sido processada na máquina $m - 1$ e quando a máquina m estiver disponível. A Figura 6 ilustra como os termos estão dispostos em um Gráfico de Gantt.

Figura 6 – Representação das estimativas de início e liberação dos trabalhos

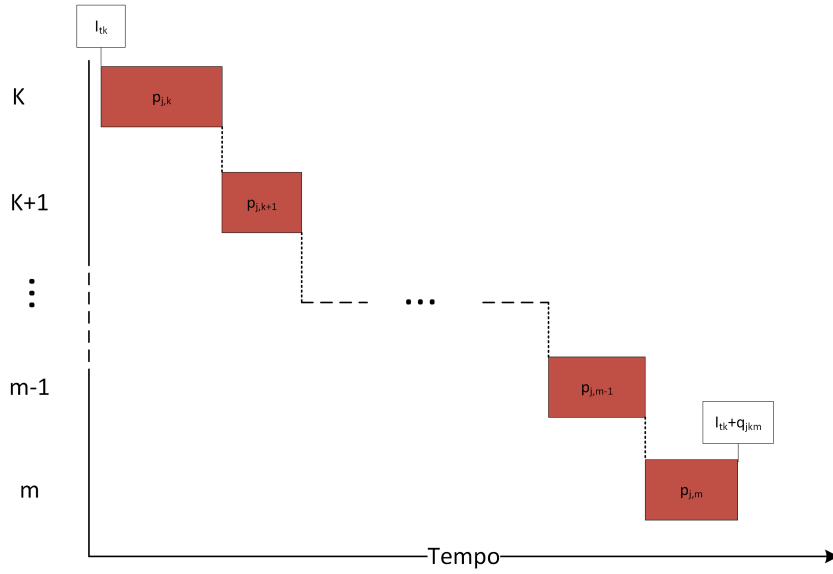


Fonte: Figura elaborada pelo autor.

A partir das subestimativas dos tempos de início de cada tarefa da NPS em cada máquina, é possível calcular um LB de TFT para cada máquina separadamente. A equação 63 apresenta como este cálculo é realizado.

$$LB_k^{TFT} = \sum_{t=|PS|+1}^n I_{t,k} + \sum_{j=|PS|+1}^n q_{jkm} \quad (k = 1, \dots, m) \quad (63)$$

A Figura 7 representa como a partir dos $I_{t,k}$ calculados, é possível se estimar o tempo de liberação na última máquina.

Figura 7 – Representação de $I_{t,k} + q_{jkm}$ 

Fonte: Figura elaborada pelo autor.

Calculados os LBs para cada máquina, a escolha do maior é feita e somada com o TFT da sequência parcial do nó.

$$LB_{TFT} = TFT_{PS} + \max_{1 \leq k \leq m} LB_k^{TFT} \quad (64)$$

A equação 78 faz a representação do cálculo final do LB para TFT , que é denominado como LB_{TFT} .

Parte do cálculo deste LB é utilizado nos próximos limitantes apresentados, por este motivo, este foi o primeiro a ser apresentado.

5.2 O Limitante Inferior para *Total Tardiness*

O cálculo do limitante para TT utiliza as equações 55, 56, 57, 58, 59, 60, 61 e 62 apresentadas na seção anterior. Para se chegar ao limitante inferior para TT , a idéia de Chung, Flynn e Kirca (2006) para subestimar o atraso foi utilizada.

$$T_{t,k} = \max(I_{tk} - h_{[t-|PS|]k}, 0) \quad (t = |PS| + 1, \dots, n) \quad (k = 1, \dots, m) \quad (65)$$

A equação 65 calcula o atraso de cada trabalho na NPS para as estimativas de início de processamento feitas em cada máquina.

$$LB_k^{TT} = \sum_{t=|PS|+1}^n T_{t,k} \quad (k = 1, \dots, m) \quad (66)$$

A equação 66 realiza a soma dos atrasos para as estimativas feitas em cada máquina.

$$LB_{TT} = TT_{PS} + \max_{1 \leq k \leq m} LB_k^{TT} \quad (67)$$

Por fim, a equação 67 seleciona a máquina que estima a maior somas dos atrasos e soma o atraso da sequência parcial deste nó.

5.3 O Limitante Inferior para *Total Flow Time* com *setup*

O caso com *setup* para *TFT* realiza as subestimativas dos tempos de liberação das tarefas na última máquina. Como a ideia de um *LB* é sempre buscar o mínimo valor que uma FO pode atingir, será somada a inclusão do menor tempo de *setup* possível antes de cada tarefa.

Para definir os mínimos *setups*, considere que:

- (i) $Setup_k(j) = \{s_{[1],|PS+1|,k}, s_{[1],|PS+2|,k}, \dots, s_{[1],j,k}, \dots, s_{[1],n,k}\}$ ($k = 2, \dots, m - 1$)
- (ii) $\dot{Setup}_k(j)$ seja $Setup_k(j)$ com seus elementos ordenados de forma não-decrescente.

As equações a seguir representam o passo a passo para a obtenção da subestimativa dos tempos de início de processamento.

$$H_{|PS|,k} = D_{|PS|,k} \quad (k = 1, \dots, m) \quad (68)$$

$$I_{|PS|+1,1} = D_{|PS|,1} + s_{|PS|,[1],1} \quad (69)$$

$$H_{|PS|+1,1} = I_{|PS|+1,1} + p_{[1],1} \quad (70)$$

$$I_{t,1} = \max(H_{t-1,1}, H_{t-2,2} + \dot{Setup}_2(t-1)) + \dot{Setup}_1(t) \quad (t = |PS| + 2, \dots, n) \quad (71)$$

$$H_{t,1} = I_{t,1} + p_{[t],1} \quad (t = |PS| + 2, \dots, n) \quad (72)$$

$$I_{|PS|+1,k} = \max(H_{|PS|,k} + s_{|PS|,[1],k}, I_{|PS|+1,k-1} + p_{[1],k-1}) \quad (k = 2, \dots, m) \quad (73)$$

$$I_{t,k} = \max\left(\max\left(H_{t-1,k}, H_{t-2,k+1} + \dot{S}etup_{k+1}(t-1)\right) + \dot{S}etup_k(t), H_{t,k-1}\right) \quad (74)$$

$$(t = |PS| + 2, \dots, n)$$

$$(k = 2, \dots, m-1)$$

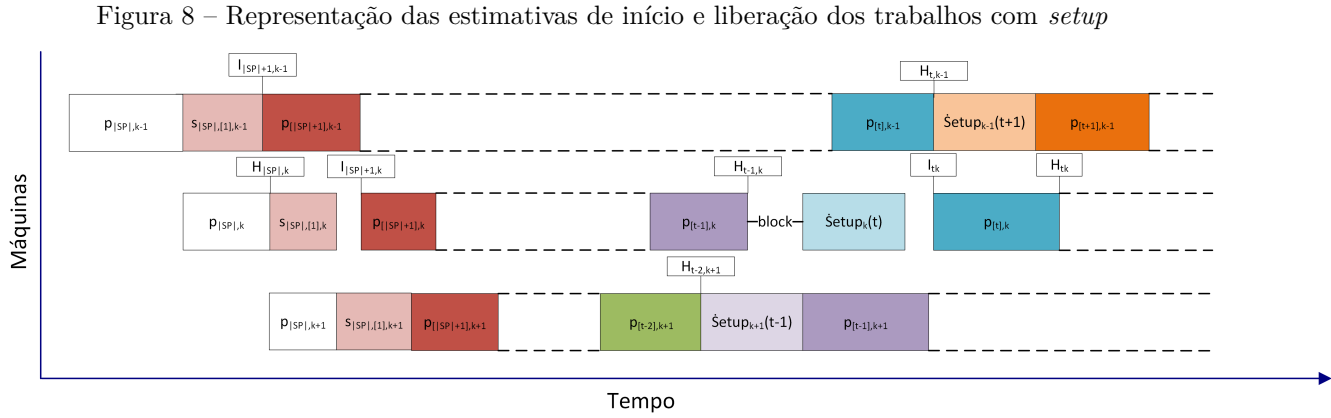
$$H_{t,k} = I_{t,k} + p_{[t],k} \quad (t = |PS| + 1, \dots, n) \quad (k = 2, \dots, m-1) \quad (75)$$

$$I_{t,m} = \max\left(H_{t-1,m} + \dot{S}etup_m(t), H_{t,m-1}\right) \quad (t = |PS| + 2, \dots, n) \quad (76)$$

A equação 68 apenas atribui valores às variáveis para seu uso posterior. As equações 69 e 70 determinam a subestimativa do tempo de início e de término do primeiro trabalho na primeira máquina. A equação 71 aplica a propriedade de bloqueio causado pela segunda na primeira máquina e estima o mínimo tempo que cada trabalho t pode começar a ser processado. Além do processamento, a preparação do trabalho também é bloqueada. A equação 72 subestima o tempo de término da tarefa t na primeira máquina. A equação 73 determina que o primeiro trabalho em cada máquina só poderá ser processado quando seu *setup* estiver devidamente finalizado e este mesmo trabalho tenha sido processado na máquina anterior. Como não é considerado bloqueio neste caso, o *setup* pode ser adiantado. A equação 74 determina que para todas as outras t tarefas terem seu processamento iniciado são necessárias as seguintes condições:

- Esta mesma tarefa t tenha sido processada na máquina $k-1$;
- A tarefa $t-1$ tenha seu processamento finalizado em k e seu *setup* realizado;
- A máquina $k+1$ esteja desocupada para que a tarefa $t-1$ possa ser removida de k .
Perceba que neste caso o bloqueio não permite que o *setup* seja adiantado e, ainda, o *setup* do trabalho $t-1$ na máquina $k+1$ também gera bloqueio.

Simultaneamente, a 75 estima os tempos de liberação das tarefas até a máquina $m-1$. Por fim, a equação 76 subestima os tempos de liberação na última máquina. Neste caso, como não há bloqueio na máquina m o *setup* pode ser adiantado. A Figura 8 apresenta os termos definidos.



Fonte: Figura elaborada pelo autor.

De forma análoga ao caso sem *setup*, o cálculo do *LB* para cada máquina pode ser realizado a partir da equação 77.

$$LB_k^{TFT+ST} = \sum_{t=|PS|+1}^n I_{t,k} + \sum_{j=|PS|+1}^n q_{jkm} \quad (k = 1, \dots, m) \quad (77)$$

Então, a escolha do maior *LB* é feita e somada com o *TFT* da sequência parcial do nó.

$$LB_{TFT+ST} = TFT_{PS} + \max_{1 \leq k \leq m} LB_k^{TFT+ST} \quad (78)$$

Para este caso, a busca dos mínimos *setup* é considerada de complexidade alta, pois são buscados em matrizes tridimensionais de ordem mn^2 .

5.4 O Limitante Inferior para *Total Tardiness* com *setup*

O último limitante desenvolvido agrega as equações elaboradas para *TFT* com *setup* e *TT* sem *setup*. Utilizam-se as equações 68, 69, 70, 71, 72, 73, 74, 75 e 76 estimam os tempos de início das tarefas e a equação 65 estima os atrasos.

$$LB_k^{TT+ST} = \sum_{t=|PS|+1}^n T_{t,k} \quad (k = 1, \dots, m) \quad (79)$$

A equação 79 realiza a soma dos atrasos para as estimativas feitas em cada máquina.

$$LB_{TT+ST} = TT_{PS} + \max_{1 \leq k \leq m} LB_k^{TT+ST} \quad (80)$$

Por fim, a equação 80 seleciona a máquina que estima a maior somas dos atrasos e soma o atraso da sequência parcial deste nó.

Propostas de Novas Modelagens *MILP*

Os modelos de programação linear inteira mista são utilizados em algoritmos matemáticos de soluções exatas, como por exemplo o *Simplex*.

Os *MILP* propostos a seguir são compostos por três partes principais. A primeira corresponde à FO, também vista como o foco principal da resolução do problema. A segunda representa de restrições que a busca da solução deve obedecer. A terceira restringe as variáveis a certos valores, por exemplo, não negatividade.

6.1 Funções Objetivo

As duas funções objetivo já foram apresentadas neste trabalho na seção 2.3 deste documento e são as mesmas para o caso com *setup* e sem *setup*. A primeira delas é o *TFT* e é representada pela equação 81.

$$\text{Minimizar : } \sum D_{jm} \quad (j = 1, \dots, n) \quad (81)$$

Neste caso, como há interferência de bloqueio, os tempos de término são representados pelos tempos de liberação D_{jk} e m representa a última máquina. E a segunda, que representa o *TT* é, pode ser calculada a partir da equação 82.

$$\text{Minimizar : } \sum T_j \quad (j = 1, \dots, n) \quad (82)$$

6.2 Equações de Restrição

As equações de restrição limitam a região factível e estão diretamente ligadas à complexidade do problema. Para o caso sem *setup* as funções que limitam as soluções são idênticas tanto para *TFT* quanto para *TT*. De forma complementar, para o caso com *setup*, a modelagem das restrições são as mesmas para as duas FOs apresentadas (WOLSEY, 1998).

Algumas restrições são comuns aos problemas de sequenciamento apresentadas neste trabalho. Duas variáveis booleanas serão definidas.

$$x_{j\sigma} = \begin{cases} 1, & \text{se a tarefa } \mathbf{j} \text{ pertencer à posição } \sigma \\ 0, & \text{caso contrário} \end{cases}$$

$$y_{ij\sigma} = \begin{cases} 1, & \text{se a tarefa } \mathbf{i} \text{ precede diretamente a tarefa } \mathbf{j} \text{ na posição } \sigma \\ 0, & \text{caso contrário} \end{cases}$$

As restrições a seguir são comuns às quatro variações de problemas tratados neste trabalho.

$$\sum_{j=1}^n x_{j\sigma} = 1 \quad (\sigma = 1, \dots, n) \quad (83)$$

$$\sum_{\sigma=1}^n x_{j\sigma} = 1 \quad (j = 1, \dots, n) \quad (84)$$

$$y_{ij\sigma} \geq x_{j\sigma} + x_{j\sigma-1} \quad (i = 1, \dots, n) \quad (j = 1, \dots, n) \quad (\sigma = 2, \dots, n) \quad (i \neq j) \quad (85)$$

$$\sum_{i=1}^n \sum_{j=1}^n y_{ij\sigma} = 1 \quad (\sigma = 2, \dots, n) \quad (86)$$

$$\sum_{i=1}^n \sum_{j=1}^n y_{ij1} = 0 \quad (87)$$

$$x_{j\sigma}, y_{ij\sigma} \in \{0, 1\} \quad (i = 1, \dots, n) \quad (j = 1, \dots, n) \quad (\sigma = 1, \dots, n) \quad (88)$$

A restrição 83 garante que para cada posição da sequência será atribuída apenas uma tarefa. De forma complementar, a restrição 84 garante que cada tarefa será alocada em apenas uma posição da sequência. A restrição 85 juntamente com a restrição 86 garantem que nenhuma tarefa será antecedida diretamente por mais de uma outra tarefa. A restrição 87 garante que nenhuma tarefa antecederá a tarefa que ocupa a primeira posição da sequência. A restrição 88 garante que as variáveis de decisão possuam valores booleanos.

Com as variáveis que restringem a composição da sequência definidas, o próximo passo foi restringir os tempos de liberação das tarefas nas máquinas e os instantes de término de preparação, esta última apenas para o caso com *setup*.

6.2.1 Modelagem sem *setup*

Para o caso sem *setup*, os cálculos dos tempos de liberação apresentados por Ronconi e Armentano (2001) foram utilizados como base, as equações necessárias para restringir o problema estão apresentadas a seguir.

$$D_{10} = 0 \quad (89)$$

$$D_{j0} = D_{j-1,1} \quad (j = 2, \dots, n) \quad (90)$$

$$D_{1k} = D_{1,k-1} + \sum_{j=1}^n p_{jk} * x_{j1} \quad (k = 1, \dots, m) \quad (91)$$

$$D_{jk} \geq D_{j,k-1} + \sum_{j=1}^n p_{jk} * x_{jk} \quad (k = 1, \dots, m) \quad (j = 2, \dots, n) \quad (92)$$

$$D_{jk} \geq D_{j-1,k+1} \quad (k = 1, \dots, m-1) \quad (j = 2, \dots, n) \quad (93)$$

$$D_{jk} \geq 0 \text{ e inteiro} \quad (k = 0, \dots, m) \quad (j = 1, \dots, n) \quad (94)$$

As equações 89 e 90 simulam a liberação das tarefas de uma máquina anterior a máquina 1 para facilitar os cálculos subsequentes. A equação 89 define o tempo de início da primeira tarefa igual a zero. A equação 90 diz que o tempo de início das demais tarefas é igual ao tempo de término da tarefa anterior somado com seu tempo de processamento. A restrição 91 calcula o tempo de partida da primeira tarefa em todas as máquinas. Neste caso, o tempo de liberação da tarefa é exatamente o tempo de liberação da máquina anterior somado com seu tempo de processamento. A equação 92 define que o tempo de liberação de uma tarefa é igual ao tempo de liberação desta mesma tarefa na máquina anterior somado com seu tempo de processamento e juntamente com a equação A equação 93 que define que a máquina seguinte tem que estar liberada da tarefa anterior para que não ocorra bloqueio. A restrição 94 restringe os valores dos tempos de liberação à não-negativos e inteiros. A modelagem para o caso de *TFT* sem *setup* é apresentada a seguir.

$$\begin{aligned}
\min \quad & \sum D_{jm} && (j = 1, \dots, n) \\
\text{s.a.} \quad & \sum_{j=1}^n x_{j\sigma} = 1 && (\sigma = 1, \dots, n) \\
& \sum_{\sigma=1}^n x_{j\sigma} = 1 && (j = 1, \dots, n) \\
& D_{10} = 0 \\
& D_{j0} = D_{j-1,1} && (j = 2, \dots, n) \\
& D_{1k} = D_{1,k-1} + \sum_{j=1}^n p_{jk} * x_{j1} && (k = 1, \dots, m) \\
& D_{jk} \geq D_{j,k-1} + \sum_{j=1}^n p_{jk} * x_{jk} && (j = 2, \dots, n) (k = 1, \dots, m) \\
& D_{jk} \geq D_{j-1,k+1} && (k = 0, \dots, m) (j = 2, \dots, n) \\
& D_{jk} \geq 0 \text{ e inteiro} && (j = 1, \dots, n) (k = 1, \dots, m) \\
& x_{j\sigma} \in \{0, 1\} \text{ e inteiro} && (j = 1, \dots, n) (\sigma = 1, \dots, n)
\end{aligned}$$

A modelagem para o caso de *TT* sem *setup* é apresentada a seguir.

$$\begin{aligned}
\min \quad & \sum T_j && (j = 1, \dots, n) \\
\text{s.a.} \quad & \sum_{j=1}^n x_{j\sigma} = 1 && (\sigma = 1, \dots, n) \\
& \sum_{\sigma=1}^n x_{j\sigma} = 1 && (j = 1, \dots, n) \\
& T_j \geq D_{jm} - \sum_{i=1}^n d_i * x_{ji} && (j = 1, \dots, n) \\
& D_{10} = 0 \\
& D_{j0} = D_{j-1,1} && (j = 2, \dots, n) \\
& D_{1k} = D_{1,k-1} + \sum_{j=1}^n p_{jk} * x_{j1} && (k = 1, \dots, m) \\
& D_{jk} \geq D_{j,k-1} + \sum_{j=1}^n p_{jk} * x_{jk} && (j = 2, \dots, n) (k = 1, \dots, m) \\
& D_{jk} \geq D_{j-1,k+1} && (k = 0, \dots, m) (j = 2, \dots, n) \\
& D_{jk} \geq 0 \text{ e inteiro} && (j = 1, \dots, n) (k = 1, \dots, m) \\
& T_j \geq 0 \text{ e inteiro} && (j = 1, \dots, n) \\
& x_{j\sigma} \in \{0, 1\} && (j = 1, \dots, n) (\sigma = 1, \dots, n)
\end{aligned}$$

6.2.2 Modelagem com *setup*

Neste caso a variável que representa o tempo de término da preparação do *setup* é incluída na modelagem.

$$R_{1k} = \sum_{i=1}^n \sum_{j=i}^n s_{ijk} * x_{j1} \quad (k = 1, \dots, m) \quad (95)$$

$$R_{\sigma k} = D_{\sigma-1,k} + \sum_{i=1}^n \sum_{j \neq i}^n s_{ijk} * y_{ij\sigma} \quad (\sigma = 2, \dots, n) \quad (k = 1, \dots, m) \quad (96)$$

$$D_{\sigma k} \geq R_{\sigma,k+1} \quad (\sigma = 1, \dots, n) \quad (k = 1, \dots, m-1) \quad (97)$$

$$D_{\sigma 1} \geq R_{\sigma,1} + \sum_{j=1}^n p_{j1} * x_{j\sigma} \quad (\sigma = 1, \dots, n) \quad (98)$$

$$D_{\sigma k} \geq D_{\sigma,k-1} + \sum_{j=1}^n p_{jk} * x_{j\sigma} \quad (\sigma = 1, \dots, n) \quad (k = 2, \dots, m) \quad (99)$$

$$D_{\sigma k}, R_{\sigma k} \geq 0 \text{ e inteiro } (\sigma = 1, \dots, n) \quad (k = 1, \dots, m) \quad (100)$$

A restrição 95 é utilizada para definir o tempo de término dos *setups* da primeira tarefa. Como o *setup* pode ser adiantado, esta depende apenas da data de liberação da mesma, que no caso é $r_j = 0$. A equação 96 calcula o tempo de término dos *setups* para todas as outras tarefas. Este término pode ser definido como a soma da data de liberação da tarefa anterior nesta mesma máquina e o tempo de *setup* entre estas tarefas. A equação 97 é utilizada para verificar se há ocorrência de bloqueio na primeira até a penúltima máquina. A equação 98 define que o tempo de liberação da tarefa na posição σ será obrigatoriamente maior ou igual que o término de seu *setup* somado com seu tempo de processamento para a primeira máquina. Caso seja maior, a equação 97 entra em vigor e ocorre bloqueio para esta tarefa. A equação 98 determina as tarefas na primeira máquina poderão ser liberadas logo ao fim de seu tempo de *setup* somado com seu tempo de processamento. Tal situação não ocorrerá caso haja bloqueio ocasionado pela segunda máquina. A equação 99 indica que a liberação da tarefa na posição σ só poderá ocorrer após sua liberação na máquina anterior somado com seu tempo de processamento. A última restrição garante que as variáveis serão inteiras e não negativas. A modelagem para o *TFT* com *setup* é apresentada a seguir.

$$\begin{aligned}
\min \quad & \sum D_{jm} && (j = 1, \dots, n) \\
\text{s.a.} \quad & \sum_{j=1}^n x_{j\sigma} = 1 && (\sigma = 1, \dots, n) \\
& \sum_{\sigma=1}^n x_{j\sigma} = 1 && (j = 1, \dots, n) \\
& y_{ij\sigma} \geq x_{j\sigma} + x_{j\sigma-1} && (i = 1, \dots, n) (j = 1, \dots, n) \\
& && (\sigma = 2, \dots, n) (i \neq j) \\
& \sum_{i=1}^n \sum_{j=1}^n y_{ij\sigma} = 1 && (\sigma = 2, \dots, n) \\
& \sum_{i=1}^n \sum_{j=1}^n y_{ij1} = 0 \\
& R_{1k} = \sum_{i=1}^n \sum_{j=i}^n s_{ijk} * x_{j1} && (k = 1, \dots, m) \\
& R_{\sigma k} = D_{\sigma-1,k} + \sum_{i=1}^n \sum_{j \neq i}^n s_{ijk} * y_{ij\sigma} && (\sigma = 2, \dots, n) (k = 1, \dots, m) \\
& D_{\sigma k} \geq R_{\sigma, k+1} && (\sigma = 1, \dots, n) (k = 1, \dots, m-1) \\
& D_{\sigma 1} \geq R_{\sigma, 1} + \sum_{j=1}^n p_{j1} * x_{j\sigma} && (\sigma = 1, \dots, n) \\
& D_{\sigma k} \geq D_{\sigma, k-1} + \sum_{j=1}^n p_{jk} * x_{j\sigma} && (\sigma = 1, \dots, n) (k = 2, \dots, m) \\
& D_{\sigma k}, R_{\sigma k} \geq 0 \text{ e inteiro} && (\sigma = 1, \dots, n) (k = 1, \dots, m) \\
& x_{j\sigma}, y_{ij\sigma} \in \{0, 1\} && (i = 1, \dots, n) (j = 1, \dots, n) \\
& && (\sigma = 1, \dots, n)
\end{aligned}$$

A modelagem para o *TT* com *setup* é apresentada a seguir.

$$\begin{aligned}
\min \quad & \sum T_j && (j = 1, \dots, n) \\
\text{s.a.} \quad & \sum_{j=1}^n x_{j\sigma} = 1 && (\sigma = 1, \dots, n) \\
& \sum_{\sigma=1}^n x_{j\sigma} = 1 && (j = 1, \dots, n) \\
& y_{ij\sigma} \geq x_{j\sigma} + x_{j\sigma-1} && (i = 1, \dots, n) (j = 1, \dots, n) \\
& && (\sigma = 2, \dots, n) (i \neq j) \\
& \sum_{i=1}^n \sum_{j=1}^n y_{ij\sigma} = 1 && (\sigma = 2, \dots, n) \\
& \sum_{i=1}^n \sum_{j=1}^n y_{ij1} = 0 \\
& T_j \geq D_{jm} - \sum_{i=1}^n d_i * x_{ji} && (j = 1, \dots, n) \\
& R_{1k} = \sum_{i=1}^n \sum_{j=i}^n s_{ijk} * x_{j1} && (k = 1, \dots, m) \\
& R_{\sigma k} = D_{\sigma-1,k} + \sum_{i=1}^n \sum_{j \neq i}^n s_{ijk} * y_{ij\sigma} && (\sigma = 2, \dots, n) (k = 1, \dots, m) \\
& D_{\sigma k} \geq R_{\sigma, k+1} && (\sigma = 1, \dots, n) (k = 1, \dots, m-1) \\
& D_{\sigma 1} \geq R_{\sigma, 1} + \sum_{j=1}^n p_{j1} * x_{j\sigma} && (\sigma = 1, \dots, n) \\
& D_{\sigma k} \geq D_{\sigma, k-1} + \sum_{j=1}^n p_{jk} * x_{j\sigma} && (\sigma = 1, \dots, n) (k = 2, \dots, m) \\
& D_{\sigma k}, R_{\sigma k} \geq 0 \text{ e inteiro} && (\sigma = 1, \dots, n) (k = 1, \dots, m) \\
& T_j \geq 0 \text{ e inteiro} && (j = 1, \dots, n) \\
& x_{j\sigma}, y_{ij\sigma} \in \{0, 1\} && (i = 1, \dots, n) (j = 1, \dots, n) \\
& && (\sigma = 1, \dots, n)
\end{aligned}$$

Com os modelos definidos, foi possível utilizar o *solver Cplex* para encontrar a melhor resposta possível para cada caso. É importante salientar que o número de equações aumenta a complexidade do problema, logo, é esperado que os casos sem *setup* sejam menos complexos e os casos de *TFT* mais simples que os de *TT*.

Experimentação Computacional e Análise dos Resultados

Este capítulo apresenta os resultados computacionais obtidos para cada método separadamente. Todos os testes realizados utilizaram a mesma base de dados de tempos de processamento fornecidos por Ronconi e Armentano (2001). Foram utilizadas 27 combinações a partir de uma distribuição uniforme que variou de 1 até 99. As combinações de (n, m) são:

$$(n, m) = \{(10, 2), (10, 3), (10, 4), (10, 5), (10, 7), (10, 10), (12, 2), (12, 3), (12, 4), (12, 5), (12, 7), (12, 10), (14, 2), (14, 3), (14, 4), (14, 5), (14, 7), (14, 10), (16, 2), (16, 3), (16, 4), (18, 2), (18, 3), (18, 4), (20, 2), (20, 3), (20, 4)\}$$
(101)

Cada combinação possui 20 variações de tempos de processamento, totalizando 540 problemas. Para determinar o melhor limitante, foi utilizado como parâmetro o desvio relativo médio, representado pela equação 102.

$$DR = \frac{DM - DM^*}{DM^*}$$
(102)

Onde é medida a distância relativa entre o limitante analisado (DM) e o melhor limitante encontrado (DM^*).

7.1 Resultados do método *Branch & Bound*

Os testes de *B&B* foram realizados em uma máquina virtual fornecida pela Escola de Engenharia de São Carlos da Universidade de São Paulo (EESC - UPS) com processador *Intel(R) Xeon(R) CPU E5-2660 0 2.20GHz 2.20GHz*, com 16GB de memória RAM DDR3

disponíveis e sistema operacional *Windows Server 2012 R2 Datacenter*. A máquina possui 8 núcleos, porém apenas 1 foi utilizado por problema. Todos os códigos do *B&B* foram implementados de forma estruturada na linguagem *C/C++* na *IDE Microsoft Visual Studio Community 2017*, também fornecido pela EESC - UPS. Para nenhum caso foram utilizados *UBs* iniciais e estes foram definidos como ∞ .

7.1.1 Resultados para *Total Flow Time*

Além do método elaborado LB_{TFT} , para este caso foram programados também dois dos métodos com melhor desempenho para $F_m|block|\sum C_{jm}$ encontrados na literatura. O primeiro é de Chung, Flynn e Kirca (2002), que utiliza uma ideia muito parecida com o LB_{TFT} , porém sem a adição do fator que calcula o bloqueio estimado ocasionado na máquina $k + 1$. O segundo, elaborado por Moslehi e Khorasanian (2013) utiliza este fator de bloqueio, porém o termo utilizado para estimar o tempo de ociosidade da máquina k é o mesmo que o utilizado por Bansal (1977) que possui seu método dominado por Chung, Flynn e Kirca (2002) e um custo computacional elevado. A ideia do LB_{TFT} foi unir os pontos fortes de cada um dos algoritmos e avaliar o resultado. Além disso, um dos termos utilizado por Chung, Flynn e Kirca (2002) para calcular o tempo de ociosidade de k ocasionado pela primeira tarefa da *NPS*, possui custo computacional elevado e pouco ganho no valor numérico de LB , logo, este foi retirado do cálculo de LB_{TFT} . A Tabela 2 apresenta as médias dos 20 problemas de cada combinação (n, m) . São apresentadas as médias do número de nós criados, do tempo gasto e do número de problemas não solucionados em 3600 segundos. Ao final da tabela, é apresentada a média de todas as combinações para cada comparação.

Tabela 2 – Comparação dos limitantes para *Total Flow Time*

Tarefas	Máquinas	Contagem de Nós			Tempo (s)			Não Solucionados		
		Chung	Moslehi	LB_{TFT}	Chung	Moslehi	LB_{TFT}	Chung	Moslehi	LB_{TFT}
10	2	2.74E+02	2.62E+02	2.71E+02	0.003	0.005	0.004	0	0	0
10	3	3.53E+02	3.31E+02	2.63E+02	0.005	0.008	0.005	0	0	0
10	4	4.92E+02	4.93E+02	3.45E+02	0.007	0.016	0.008	0	0	0
10	5	1.20E+03	1.12E+03	1.01E+03	0.018	0.044	0.025	0	0	0
10	7	7.21E+02	6.79E+02	6.43E+02	0.019	0.046	0.021	0	0	0
10	10	1.10E+03	1.09E+03	9.98E+02	0.035	0.150	0.045	0	0	0
12	2	7.32E+02	6.55E+02	7.13E+02	0.016	0.024	0.016	0	0	0
12	3	9.31E+02	8.62E+02	7.04E+02	0.022	0.042	0.019	0	0	0
12	4	4.20E+03	4.10E+03	2.80E+03	0.113	0.208	0.089	0	0	0
12	5	3.18E+03	3.07E+03	1.81E+03	0.093	0.193	0.067	0	0	0
12	7	7.92E+03	7.07E+03	6.07E+03	0.286	0.718	0.264	0	0	0
12	10	1.21E+04	1.17E+04	9.33E+03	0.503	2.123	0.572	0	0	0
14	2	4.80E+03	4.25E+03	4.44E+03	0.130	0.142	0.124	0	0	0
14	3	1.30E+04	1.16E+04	1.01E+04	0.401	0.550	0.349	0	0	0
14	4	2.84E+04	2.76E+04	1.65E+04	1.007	1.847	0.719	0	0	0
14	5	8.29E+04	7.71E+04	5.31E+04	3.034	6.457	2.388	0	0	0
14	7	7.91E+04	7.64E+04	6.16E+04	3.750	11.582	4.029	0	0	0
14	10	1.91E+05	1.76E+05	1.52E+05	10.294	49.042	12.652	0	0	0
16	2	3.09E+04	2.83E+04	2.99E+04	1.024	0.997	1.076	0	0	0
16	3	1.52E+05	1.29E+05	8.33E+04	5.665	7.370	3.619	0	0	0
16	4	8.79E+04	8.06E+04	6.07E+04	4.078	7.839	3.352	0	0	0
18	2	4.66E+05	4.17E+05	4.49E+05	19.048	21.520	19.305	0	0	0
18	3	8.11E+05	7.46E+05	5.99E+05	38.780	58.617	31.610	0	0	0
18	4	1.57E+06	1.43E+06	1.04E+06	89.515	172.338	69.595	0	0	0
20	2	1.46E+06	1.35E+06	1.44E+06	72.862	84.608	78.805	0	0	0
20	3	5.37E+06	4.87E+06	4.14E+06	334.989	547.839	291.571	0	0	0
20	4	1.02E+07	8.74E+06	6.47E+06	750.827	1,464.279	559.391	0	0	0
Média		7.62E+05	6.74E+05	5.42E+05	49.50	90.32	39.99	0.00	0.00	0.00

A Tabela 3 apresenta os valores calculados dos desvios relativos médios a partir das médias finais da Tabela 2.

Tabela 3 – Desvio Relativo Médio *Total Flow Time*

	Contagem de nós	Tempo	Não Solucionados
Chung	0.41	0.24	0.00
Moslehi	0.24	1.26	0.00
LB_{TFT}	0.00	0.00	0.00

É possível perceber que o método elaborado LB_{TFT} reduziu o número médio de nós na maioria das combinações apresentadas. Além disso, o tempo médio necessário foi menor para os 540 problemas, mostrando a eficiência em termos de valor numérico e custo computacional. Todos os problemas foram solucionados em menos de 3600 segundos.

7.1.2 Resultados para *Total Tardiness*

O caso de *Total Tardiness* utilizou os mesmos tempos de processamento utilizados para o caso de *Total Flow Time*. Para a geração das *Due Dates* das tarefas foi utilizada a equação 103 elaborada por Ronconi e Armentano (2001) para definir os limites de uma distribuição uniforme que varia de $P(1 - TF - DR/2)$ e $P(1 - TF + DR/2)$. Onde, TF é o fator de atraso das tarefas e DR é a faixa de dispersão das *due dates*.

$$P = \max \left(\max_{1 \leq k \leq m} \sum_{j=1}^n p_{jk} + \min_{1 \leq j \leq n} \sum_{q=1}^{k-1} p_{jq} + \min_{1 \leq j \leq n} \sum_{q=k+1}^m p_{jq}, \max_{1 \leq j \leq n} \sum_{k=1}^m p_{jk} \right) \quad (103)$$

Os cenários a seguir representam as variações de problemas testados.

- Cenário 1: Baixo fator de atraso ($TF = 0.2$) e baixa dispersão da *due date* ($DR = 0.6$);
- Cenário 2: Baixo fator de atraso ($TF = 0.2$) e alta dispersão da *due date* ($DR = 1.2$);
- Cenário 3: Alto fator de atraso ($TF = 0.4$) e baixa dispersão da *due date* ($DR = 0.6$);
- Cenário 4: Alto fator de atraso ($TF = 0.4$) e alta dispersão da *due date* ($DR = 1.2$);

O LB_{TT} foi testado e comparado com os LB s desenvolvidos por Ronconi e Armentano (2001) e por Toumi et al. (2017a). O primeiro deles foi o pioneiro no assunto e desenvolveu um LB totalmente atípico que calcula primeiro o LB da NPS e em seguida “encaixa” a PS nos valores encontrados, encontrando um LB para o D_{jm} de todas as tarefas. Toumi et al. (2017a) utiliza a estimativa dos tempos de liberação das tarefas utilizando o fator bloqueio ocasionado por $k + 1$, porém não explora o tempo de ociosidade na máquina k . A Tabela 4 apresenta os resultados dos testes do cenário 1.

Tabela 4 – Comparação dos limitantes para *Total Tardiness* - Cenário 1

Tarefas	Máquinas	Contagem de Nós			Tempo (s)			Não Solucionados		
		Ronconi	Toumi	LB_{TT}	Ronconi	Toumi	LB_{TT}	Ronconi	Toumi	LB_{TT}
10	2	1.93E+03	3.14E+03	3.10E+03	0.008	0.024	0.026	0	0	0
10	3	5.43E+03	1.91E+03	1.55E+03	0.030	0.022	0.017	0	0	0
10	4	2.15E+04	2.00E+03	1.69E+03	0.127	0.024	0.022	0	0	0
10	5	9.66E+04	3.00E+03	2.72E+03	0.343	0.041	0.038	0	0	0
10	7	8.35E+04	1.65E+03	1.41E+03	0.519	0.024	0.028	0	0	0
10	10	1.85E+05	1.59E+03	1.40E+03	1.253	0.039	0.038	0	0	0
12	2	1.97E+04	2.89E+04	2.84E+04	0.043	0.287	0.319	0	0	0
12	3	1.24E+05	2.35E+04	2.10E+04	0.373	0.291	0.305	0	0	0
12	4	6.86E+05	2.15E+04	1.73E+04	2.405	0.321	0.324	0	0	0
12	5	1.23E+06	2.40E+04	1.79E+04	4.954	0.414	0.379	0	0	0
12	7	1.70E+06	1.95E+04	1.65E+04	8.075	0.371	0.422	0	0	0
12	10	5.36E+06	2.08E+04	1.73E+04	42.180	0.499	0.648	0	0	0
14	2	6.01E+05	9.05E+05	8.96E+05	0.978	10.018	11.890	0	0	0
14	3	1.19E+06	2.73E+05	2.33E+05	4.282	4.132	4.475	0	0	0
14	4	1.64E+07	4.06E+05	3.02E+05	70.406	7.053	6.636	0	0	0
14	5	3.08E+07	3.19E+05	2.52E+05	145.441	7.120	6.760	0	0	0
14	7	1.24E+08	3.03E+05	2.56E+05	751.031	7.869	8.840	0	0	0
14	10	1.92E+08	2.13E+05	1.76E+05	1606.439	7.065	9.134	1	0	0
16	2	7.70E+06	1.23E+07	1.22E+07	14.102	167.863	195.221	0	0	0
16	3	3.81E+07	9.20E+06	7.38E+06	149.309	163.833	161.114	0	0	0
16	4	1.46E+08	1.41E+07	6.93E+06	772.369	231.697	252.5786	2	0	0
18	2	1.79E+08	6.23E+07	4.37E+07	489.857	1120.922	1592.836	2	2	2
18	3	2.64E+08	8.20E+07	4.36E+07	1536.739	1990.959	2149.813	4	4	4
18	4	3.68E+08	7.77E+07	3.69E+07	2288.075	2165.974	2119.581	11	9	9
20	2	6.01E+08	8.84E+07	4.42E+07	1847.971	1831.553	1822.587	9	9	9
20	3	4.31E+08	1.13E+08	4.97E+07	2761.514	3422.843	3426.969	12	19	19
20	4	4.86E+08	1.02E+08	4.49E+07	3600.001	3600.001	3600.001	20	20	20
Média		1.07E+08	2.09E+07	1.08E+07	596.25	545.97	569.30	2.26	2.33	2.33

A Tabela 5 apresenta os desvios relativos médios obtidos a partir das médias da Tabela 4.

Tabela 5 – Desvio Relativo Médio *Total Tardiness* - Cenário 1

	Contagem de nós	Tempo	Não Solucionados
Ronconi	8.91	0.04	0.00
Toumi	0.94	0.00	0.03
LB_{TT}	0.00	0.04	0.03

É possível perceber que para problemas com poucas máquinas, o LB de Ronconi e Armentano (2001) solucionou mais problemas em média para o cenário 1. No geral, para o número de nós criados, o LB_{TT} foi superior. Para o tempo computacional, Toumi et al. (2017a) foi mais eficiente. A Tabela 6 apresenta os resultados para o cenário 2.

Tabela 6 – Comparação dos limitantes para *Total Tardiness* - Cenário 2

Tarefas	Máquinas	Contagem de Nós			Tempo (s)			Não Solucionados		
		Ronconi	Toumi	LB_{TT}	Ronconi	Toumi	LB_{TT}	Ronconi	Toumi	LB_{TT}
10	2	4.91E+03	2.48E+02	2.48E+02	0.009	0.002	0.002	0	0	0
10	3	1.07E+04	3.28E+02	3.16E+02	0.022	0.003	0.003	0	0	0
10	4	4.70E+04	6.09E+02	5.65E+02	0.128	0.006	0.006	0	0	0
10	5	1.73E+05	1.01E+03	9.33E+02	0.489	0.011	0.012	0	0	0
10	7	1.93E+05	5.04E+02	4.76E+02	0.702	0.008	0.008	0	0	0
10	10	3.43E+05	6.30E+02	5.97E+02	1.760	0.013	0.013	0	0	0
12	2	2.61E+05	2.97E+03	2.97E+03	0.480	0.031	0.031	0	0	0
12	3	8.86E+05	1.22E+03	1.20E+03	2.238	0.017	0.017	0	0	0
12	4	1.10E+06	2.36E+03	2.20E+03	3.216	0.038	0.035	0	0	0
12	5	2.74E+06	5.39E+03	5.25E+03	8.714	0.056	0.053	0	0	0
12	7	7.57E+06	2.40E+03	2.25E+03	24.729	0.054	0.056	0	0	0
12	10	1.40E+07	1.16E+04	1.07E+04	91.323	0.287	0.282	0	0	0
14	2	3.23E+06	9.40E+03	9.40E+03	5.683	0.132	0.144	0	0	0
14	3	4.50E+07	6.73E+03	5.95E+03	86.628	0.105	0.100	0	0	0
14	4	1.00E+08	5.71E+04	5.57E+04	281.971	0.396	0.446	0	0	0
14	5	1.67E+08	2.40E+04	2.23E+04	556.397	0.486	0.465	1	0	0
14	7	2.95E+08	3.73E+04	3.40E+04	1427.484	1.111	1.078	1	0	0
14	10	4.83E+08	4.14E+04	3.92E+04	3077.164	1.489	1.570	14	0	0
16	2	5.08E+07	5.36E+04	5.36E+04	447.649	0.377	0.454	1	0	0
16	3	6.08E+08	2.21E+04	2.19E+04	1345.859	0.500	0.550	3	0	0
16	4	5.67E+08	9.91E+04	8.64E+04	1674.226	2.711	2.365	7	0	0
18	2	1.85E+08	2.57E+05	2.57E+05	1526.183	1.958	2.033	6	0	0
18	3	1.27E+09	1.59E+07	1.59E+07	3054.475	55.566	69.114	16	0	0
18	4	1.22E+09	1.73E+06	1.13E+06	3420.896	44.074	28.891	19	0	0
20	2	1.43E+09	2.67E+06	2.67E+06	3232.714	65.010	62.870	17	0	0
20	3	1.52E+09	8.94E+05	8.77E+05	3495.908	38.857	35.571	19	0	0
20	4	1.30E+09	7.44E+06	4.49E+06	3600.001	191.464	126.963	20	0	0
Média		3.44E+08	1.09E+06	9.51E+05	1013.59	14.99	12.34	4.59	0.00	0.00

A Tabela 7 apresenta os desvios relativos médios obtidos a partir das médias finais da Tabela 6.

Tabela 7 – Desvio Relativo Médio *Total Tardiness* - Cenário 2

	Contagem de nós	Tempo	Não Solucionados
Ronconi	360.72	81.14	Inf
Toumi	0.15	0.21	0.00
LB_{TT}	0.00	0.00	0.00

Pode-se perceber que para o cenário 2, LB_{TT} e o LB apresentado por Toumi et al. (2017a) aumentaram muito sua eficiência se comparados com o cenário 1. Isso ocorre pois a taxa de dispersão das *due dates* faz com que seus valores se distanciem, tonando a sequência ótima relativamente trivial. Já o LB apresentado por Ronconi e Armentano (2001) se mostrou menos eficaz. LB_{TT} foi superior em termos de números de nós e tempo computacional, e empatou com Toumi et al. (2017a) em número de problemas resolvidos.

Nas Tabelas 8 e 10 estão apresentados os resultados dos dois últimos cenários. Nestes dois casos, LB_{TT} e Toumi et al. (2017a) apresentam resultados muito próximos, com leve vantagem do primeiro enquanto Ronconi e Armentano (2001) se mostra menos eficiente.

Tabela 8 – Comparação dos limitantes para *Total Tardiness* - Cenário 3

Tarefas	Máquinas	Contagem de Nós			Tempo (s)			Não Solucionados		
		Ronconi	Toumi	LB_{TT}	Ronconi	Toumi	LB_{TT}	Ronconi	Toumi	LB_{TT}
10	2	2.78E+03	1.06E+03	1.06E+03	0.006	0.009	0.010	0	0	0
10	3	7.77E+03	7.15E+02	6.10E+02	0.029	0.008	0.006	0	0	0
10	4	3.30E+04	9.40E+02	7.87E+02	0.132	0.011	0.009	0	0	0
10	5	1.31E+05	1.70E+03	1.37E+03	0.502	0.019	0.018	0	0	0
10	7	1.42E+05	1.25E+03	9.94E+02	0.708	0.017	0.015	0	0	0
10	10	3.46E+05	1.55E+03	1.43E+03	2.104	0.029	0.029	0	0	0
12	2	2.66E+04	1.16E+04	1.16E+04	0.061	0.124	0.145	0	0	0
12	3	2.13E+05	5.58E+03	5.50E+03	0.693	0.075	0.081	0	0	0
12	4	9.35E+05	6.79E+03	5.10E+03	3.927	0.132	0.087	0	0	0
12	5	1.40E+06	5.45E+03	4.21E+03	6.994	0.111	0.093	0	0	0
12	7	2.53E+06	1.08E+04	9.08E+03	17.011	0.251	0.203	0	0	0
12	10	1.15E+07	1.68E+04	1.36E+04	100.040	0.402	0.360	0	0	0
14	2	7.86E+05	1.19E+05	1.19E+05	1.684	1.712	1.855	0	0	0
14	3	3.01E+06	5.11E+04	4.53E+04	12.649	0.933	0.885	0	0	0
14	4	2.44E+07	6.99E+04	5.34E+04	114.825	1.434	1.210	0	0	0
14	5	3.71E+07	7.69E+04	5.54E+04	249.613	1.779	1.420	0	0	0
14	7	1.59E+08	8.14E+04	7.00E+04	1307.714	2.427	2.201	1	0	0
14	10	3.17E+08	2.00E+05	1.56E+05	3235.491	6.972	5.563	15	0	0
16	2	2.93E+06	1.47E+06	1.47E+06	6.612	24.964	27.066	0	0	0
16	3	3.60E+07	9.34E+05	6.85E+05	185.581	20.374	16.461	0	0	0
16	4	1.03E+08	4.44E+05	3.16E+05	727.241	12.313	9.295	2	0	0
18	2	1.16E+08	1.60E+07	1.60E+07	305.122	358.549	367.110	1	0	0
18	3	3.11E+08	5.74E+06	5.16E+06	1639.672	174.728	158.707	6	0	0
18	4	3.30E+08	8.40E+06	6.34E+06	2504.778	306.766	236.187	11	0	0
20	2	5.03E+08	9.91E+07	8.02E+07	1717.365	2693.220	1965.270	5	12	7
20	3	5.35E+08	5.17E+07	5.08E+07	3132.496	2080.564	1768.229	16	4	3
20	4	4.48E+08	5.06E+07	4.27E+07	3600.001	2329.206	1756.466	20	9	6
Média		1.09E+08	8.70E+06	7.56E+06	699.00	296.93	234.04	2.85	0.93	0.59

Na Tabela 9 são apresentados os desvios relativos médios obtidos a partir das médias da Tabela 8.

Tabela 9 – Desvio Relativo Médio *Total Tardiness* - Cenário 3

	Contagem de nós	Tempo	Não Solucionados
Ronconi	13.42	1.99	3.83
Toumi	0.94	0.27	0.58
LB_{TT}	0.00	0.00	0.00

Tabela 10 – Comparação dos limitantes para *Total Tardiness* - Cenário 4

Tarefas	Máquinas	Contagem de Nós			Tempo (s)			Não Solucionados		
		Ronconi	Toumi	LB_{TT}	Ronconi	Toumi	LB_{TT}	Ronconi	Toumi	LB_{TT}
10	2	1.74E+04	3.87E+02	3.87E+02	0.034	0.005	0.005	0	0	0
10	3	2.46E+04	4.37E+02	4.03E+02	0.070	0.006	0.006	0	0	0
10	4	6.94E+04	6.47E+02	5.79E+02	0.223	0.009	0.009	0	0	0
10	5	1.81E+05	1.27E+03	1.10E+03	0.683	0.018	0.018	0	0	0
10	7	2.21E+05	8.01E+02	7.32E+02	1.088	0.015	0.015	0	0	0
10	10	4.12E+05	1.25E+03	1.04E+03	2.155	0.034	0.028	0	0	0
12	2	7.00E+05	2.38E+03	2.38E+03	1.482	0.037	0.040	0	0	0
12	3	1.53E+06	3.11E+03	3.03E+03	4.919	0.048	0.053	0	0	0
12	4	3.13E+06	4.86E+03	4.09E+03	10.858	0.080	0.080	0	0	0
12	5	5.35E+06	7.77E+03	6.93E+03	20.935	0.135	0.150	0	0	0
12	7	7.67E+06	8.57E+03	7.05E+03	38.836	0.194	0.187	0	0	0
12	10	1.76E+07	1.40E+04	1.28E+04	129.645	0.362	0.394	0	0	0
14	2	1.85E+07	3.91E+04	3.91E+04	43.177	0.513	0.665	0	0	0
14	3	6.05E+07	1.92E+04	1.78E+04	215.004	0.355	0.400	0	0	0
14	4	1.33E+08	3.13E+04	2.64E+04	556.944	0.709	0.668	1	0	0
14	5	2.07E+08	4.62E+04	3.68E+04	1001.630	1.130	1.069	0	0	0
14	7	3.90E+08	5.66E+04	4.68E+04	2668.120	1.570	1.764	8	0	0
14	10	4.17E+08	1.49E+05	1.30E+05	3552.736	4.568	5.481	19	0	0
16	2	5.13E+08	2.13E+05	2.13E+05	1440.468	2.700	3.740	4	0	0
16	3	5.66E+08	1.56E+05	1.45E+05	2442.344	3.242	4.100	7	0	0
16	4	4.61E+08	1.64E+05	1.38E+05	2401.418	4.048	4.477	11	0	0
18	2	9.01E+08	1.75E+05	1.75E+05	2712.402	3.775	5.147	14	0	0
18	3	7.53E+08	6.59E+05	6.16E+05	3333.548	17.933	22.206	18	0	0
18	4	6.47E+08	2.08E+06	1.67E+06	3600.001	69.212	71.664	20	0	0
20	2	1.07E+09	1.23E+07	8.26E+06	3405.913	317.040	287.671	18	1	1
20	3	7.60E+08	5.87E+06	5.54E+06	3600.001	216.822	255.630	20	0	0
20	4	5.65E+08	1.72E+07	1.31E+07	3600.001	744.957	710.644	20	1	1
Média		2.78E+08	1.45E+06	1.12E+06	1288.32	51.46	50.97	5.93	0.07	0.07

A Tabela 11 apresenta os desvios relativos médios obtidos a partir das médias da Tabela 10.

Tabela 11 – Desvio Relativo Médio *Total Tardiness* - Cenário 4

	Contagem de nós	Tempo	Não Solucionados
Ronconi	247.21	24.28	83.71
Toumi	0.29	0.01	0.00
LB_{TT}	0.00	0.00	0.00

A partir dos resultados dos cenários 3 e 4, foi possível perceber que quanto maior o fator de atraso das *due dates*, maior a dificuldade de encontrar o ótimo. Isto ocorre pois quando um nó folha com o valor zero de atraso é encontrado, o algoritmo para e é encerrado. No geral, para o caso de TT , LB_{TT} e Toumi et al. (2017a) se apresentaram bem próximos em todos os cenários, enquanto Ronconi e Armentano (2001) mostrou superioridade no primeiro e se distanciou de forma negativa no outros três cenários.

7.1.3 Resultados para *Total Flow Time* com *setup*

A configuração de *TFT* com *setup* utilizou a mesma base de dados dos tempos de processamento e foram geradas quatro variações de tempos de *setup* a partir de distribuições uniformes com os seguintes mínimos e máximos:

- cenário 1: Mínimo de 1, máximo de 9;
- cenário 2: Mínimo de 1, máximo de 49;
- cenário 3: Mínimo de 1, máximo de 99;
- cenário 4: Mínimo de 1, máximo de 124;

Não foram encontrados limitantes inferiores para a configuração $F_m|block, S_{ijk}|\sum C_{jm}$, logo, para fins comparativos, foram realizados testes com três variações do LB_{TFT+ST} . O intuito foi avaliar quais cálculos de busca dos mínimos *setups* possuem custo computacional viável. As três variações são:

- LB_1^{TFT+ST} : Os termos $s_{i[1]k}$ e $s_{[1]jk}$ não tiveram suas buscas realizadas e foram definidos como zero;
- LB_2^{TFT+ST} : Neste caso, não foram realizadas as buscas do termo $\dot{S}etup_k(j)$ e estes foram definidos como zero;
- LB_3^{TFT+ST} : O LB foi calculado na íntegra.

A partir das Tabelas 12, 14, 16 e 18 apresentam os resultados obtidos para os casos de *TFT* com *setup*.

Tabela 12 – Comparação dos limitantes para *Total Flow Time* com *setup* - Cenário 1

Tarefas	Máquinas	Contagem de Nós				Tempo (s)			Não Solucionados		
		LB_1^{TFT+ST}	LB_2^{TFT+ST}	LB_3^{TFT+ST}	LB_1^{TFT+ST}	LB_2^{TFT+ST}	LB_3^{TFT+ST}	LB_1^{TFT+ST}	LB_2^{TFT+ST}	LB_3^{TFT+ST}	
10	2	5.81E+02	5.76E+02	5.30E+02	0.010	0.008	0.010	0	0	0	
10	3	8.76E+03	8.51E+03	8.39E+03	0.071	0.060	0.069	0	0	0	
10	4	4.43E+04	4.04E+04	4.02E+04	0.293	0.223	0.280	0	0	0	
10	5	3.82E+05	3.81E+05	3.82E+05	1.904	1.722	1.944	0	0	0	
10	7	6.96E+05	6.61E+05	6.72E+05	3.280	3.185	3.398	0	0	0	
10	10	1.56E+06	1.58E+06	1.60E+06	7.891	8.873	8.768	0	0	0	
12	2	2.17E+03	2.18E+03	1.93E+03	0.061	0.043	0.058	0	0	0	
12	3	9.95E+04	8.27E+04	8.13E+04	0.840	0.575	0.689	0	0	0	
12	4	5.51E+06	5.26E+06	5.21E+06	26.320	26.221	28.437	0	0	0	
12	5	1.35E+07	1.32E+07	1.33E+07	66.189	63.332	67.047	0	0	0	
12	7	4.16E+07	4.15E+07	4.09E+07	223.579	205.250	224.600	0	0	0	
12	10	9.09E+07	7.78E+07	7.94E+07	539.634	437.121	490.666	0	0	0	
14	2	1.69E+04	1.81E+04	1.51E+04	0.644	0.456	0.578	0	0	0	
14	3	8.36E+06	2.84E+06	8.13E+06	57.063	21.574	57.114	0	0	0	
14	4	7.31E+07	1.11E+08	1.09E+08	427.835	550.287	595.402	1	2	2	
14	5	2.05E+08	1.93E+08	1.93E+08	1191.026	1076.282	1129.617	3	2	3	
14	7	2.99E+08	3.07E+08	2.89E+08	2101.254	1976.018	2098.374	8	7	8	
14	10	5.89E+08	6.14E+08	5.75E+08	3510.736	3495.732	3511.206	19	19	19	
16	2	1.04E+05	1.17E+05	9.22E+04	5.135	3.696	4.657	0	0	0	
16	3	3.46E+06	3.53E+06	3.18E+06	129.559	89.907	119.861	0	0	0	
16	4	1.65E+08	1.41E+08	1.60E+08	1105.745	878.710	1065.442	4	3	4	
18	2	8.43E+05	1.03E+06	7.50E+05	55.860	40.276	51.584	0	0	0	
18	3	2.71E+07	3.78E+07	3.46E+07	840.463	711.829	873.607	0	1	1	
18	4	2.04E+08	2.70E+08	2.25E+08	2410.370	2313.624	2359.329	9	10	9	
20	2	6.18E+06	7.69E+06	5.38E+06	495.834	381.780	452.126	0	0	0	
20	3	7.24E+07	6.24E+07	5.10E+07	2653.069	2306.895	2488.527	12	9	10	
20	4	2.56E+08	2.81E+08	2.60E+08	3600.001	3600.001	3600.001	20	20	20	
Média		7.65E+07	8.05E+07	7.62E+07	720.54	673.84	712.35	2.81	2.70	2.81	

A Tabela 13 apresenta os desvios relativos médios obtidos a partir das médias da Tabela 12.

Tabela 13 – Desvio Relativo Médio *Total Flow Time* com *setup* - Cenário 1

	Contagem de nós	Tempo	Não Solucionados
LB_1^{TFT+ST}	0.00	0.07	0.04
LB_2^{TFT+ST}	0.06	0.00	0.00
LB_3^{TFT+ST}	0.00	0.06	0.04

Tabela 14 – Comparação dos limitantes para *Total Flow Time* com *setup* - Cenário 2

Tarefas	Máquinas	Contagem de Nós					Tempo (s)	Não Solucionados			
		LB_1^{TFT+ST}	LB_2^{TFT+ST}	LB_3^{TFT+ST}	LB_1^{TFT+ST}	LB_2^{TFT+ST}		LB_3^{TFT+ST}	LB_1^{TFT+ST}	LB_2^{TFT+ST}	LB_3^{TFT+ST}
10	2	1.42E+03	1.31E+03	1.10E+03	0.028	0.018	0.020	0	0	0	
10	3	2.08E+04	1.96E+04	2.44E+04	0.191	0.152	0.187	0	0	0	
10	4	7.21E+04	4.88E+04	4.51E+04	0.544	0.387	0.432	0	0	0	
10	5	2.06E+05	2.42E+05	2.14E+05	1.382	1.355	1.333	0	0	0	
10	7	5.19E+05	4.96E+05	5.13E+05	3.305	2.867	3.095	0	0	0	
10	10	1.31E+06	1.20E+06	1.21E+06	6.753	6.917	7.048	0	0	0	
12	2	1.24E+04	1.29E+04	9.59E+03	0.183	0.248	0.277	0	0	0	
12	3	7.03E+05	8.21E+05	8.00E+05	5.097	4.963	4.076	0	0	0	
12	4	4.09E+06	2.43E+06	2.79E+06	29.105	17.793	21.131	0	0	0	
12	5	5.25E+06	6.08E+06	6.14E+06	39.324	37.569	41.504	0	0	0	
12	7	2.46E+07	2.55E+07	2.54E+07	147.206	142.323	153.871	0	0	0	
12	10	6.49E+07	5.11E+07	5.03E+07	447.156	335.646	359.675	0	0	0	
14	2	6.96E+04	7.41E+04	5.49E+04	2.657	1.658	2.177	0	0	0	
14	3	8.22E+06	6.02E+06	5.76E+06	97.614	70.584	72.341	0	0	0	
14	4	6.62E+07	3.26E+07	3.42E+07	550.658	300.324	369.081	1	0	0	
14	5	1.78E+08	1.35E+08	1.61E+08	1312.986	985.334	1193.124	4	2	4	
14	7	3.02E+08	3.94E+08	3.15E+08	2661.016	2828.815	2654.844	11	11	11	
14	10	4.84E+08	4.84E+08	4.47E+08	3365.358	3238.832	3308.020	18	16	17	
16	2	5.20E+05	5.95E+05	4.07E+05	27.855	19.407	22.540	0	0	0	
16	3	4.23E+07	3.39E+07	3.14E+07	724.443	539.128	646.615	2	1	1	
16	4	2.62E+08	1.98E+08	1.92E+08	2491.518	1731.481	1995.478	9	7	7	
18	2	4.53E+06	5.42E+06	3.55E+06	314.373	221.093	251.807	0	0	0	
18	3	1.22E+08	1.74E+08	1.13E+08	3171.886	2801.018	2947.936	15	12	14	
18	4	1.20E+08	1.50E+08	9.98E+07	3576.416	3407.678	3501.395	19	16	18	
20	2	2.18E+07	3.07E+07	1.86E+07	2053.918	1574.208	1736.984	3	2	2	
20	3	5.91E+07	8.60E+07	5.72E+07	3600.001	3600.001	3600.001	20	20	20	
20	4	2.97E+08	2.80E+08	2.75E+08	3600.001	3600.001	3600.001	20	20	20	
Média		7.66E+07	7.77E+07	6.82E+07	1045.59	943.33	981.30	4.52	3.96	4.22	

A Tabela 15 apresenta os desvios relativos médios obtidos a partir das médias da Tabela 14.

Tabela 15 – Desvio Relativo Médio *Total Flow Time* com *setup* - Cenário 2

	Contagem de nós	Tempo	Não Solucionados
LB_1^{TFT+ST}	0.12	0.11	0.14
LB_2^{TFT+ST}	0.14	0.00	0.00
LB_3^{TFT+ST}	0.00	0.04	0.07

Tabela 16 – Comparação dos limitantes para *Total Flow Time* com *setup* - Cenário 3

Tarefas	Máquinas	Contagem de Nós				Tempo (s)			Não Solucionados		
		LB_1^{TFT+ST}	LB_2^{TFT+ST}	LB_3^{TFT+ST}	LB_1^{TFT+ST}	LB_2^{TFT+ST}	LB_3^{TFT+ST}	LB_1^{TFT+ST}	LB_2^{TFT+ST}	LB_3^{TFT+ST}	
10	2	2.35E+03	2.05E+03	1.58E+03	0.039	0.032	0.030	0	0	0	
10	3	2.53E+04	1.72E+04	1.71E+04	0.274	0.177	0.205	0	0	0	
10	4	4.95E+04	4.93E+04	3.84E+04	0.542	0.421	0.446	0	0	0	
10	5	1.53E+05	1.24E+05	1.18E+05	1.284	0.973	1.068	0	0	0	
10	7	4.05E+05	2.96E+05	2.93E+05	2.272	2.124	2.431	0	0	0	
10	10	6.98E+05	8.08E+05	7.71E+05	4.728	5.416	5.950	0	0	0	
12	2	1.20E+04	1.15E+04	8.66E+03	0.317	0.218	0.260	0	0	0	
12	3	2.55E+05	2.00E+05	2.50E+05	3.771	2.310	2.953	0	0	0	
12	4	6.03E+05	7.48E+05	6.39E+05	10.668	8.469	9.621	0	0	0	
12	5	3.14E+06	2.33E+06	2.36E+06	35.531	22.606	27.407	0	0	0	
12	7	1.13E+07	1.11E+07	1.07E+07	97.695	82.675	90.788	0	0	0	
12	10	3.32E+07	2.06E+07	2.21E+07	266.644	171.832	218.601	0	0	0	
14	2	7.62E+04	7.62E+04	5.56E+04	2.665	1.983	2.378	0	0	0	
14	3	1.48E+06	2.29E+06	1.59E+06	36.153	37.268	40.787	0	0	0	
14	4	2.41E+07	1.05E+07	1.65E+07	328.047	162.731	239.095	0	0	0	
14	5	1.04E+08	4.65E+07	6.59E+07	1052.792	554.925	795.590	2	0	1	
14	7	2.16E+08	2.23E+08	2.19E+08	2088.795	1897.037	2051.695	7	7	7	
14	10	3.65E+08	3.58E+08	3.37E+08	3335.304	3050.372	3238.649	15	14	14	
16	2	1.27E+06	1.06E+06	7.28E+05	57.811	33.615	37.202	0	0	0	
16	3	3.15E+07	3.30E+07	2.81E+07	619.392	455.640	527.823	1	1	1	
16	4	1.04E+08	1.10E+08	9.28E+07	2197.687	1757.055	1922.422	6	6	7	
18	2	3.48E+06	4.19E+06	2.64E+06	253.645	167.221	196.373	0	0	0	
18	3	6.15E+07	8.90E+07	5.81E+07	3342.785	2933.593	3050.296	14	11	10	
18	4	1.40E+08	1.83E+08	1.40E+08	3600.001	3546.032	3592.311	20	19	19	
20	2	2.16E+07	3.05E+07	1.86E+07	2077.880	1618.644	1799.302	6	1	2	
20	3	6.22E+07	9.34E+07	6.41E+07	3600.001	3600.001	3600.001	20	20	20	
20	4	1.44E+08	1.47E+08	1.25E+08	3600.001	3600.001	3600.001	20	20	20	
Média		4.93E+07	5.06E+07	4.47E+07	985.80	878.27	927.91	4.11	3.67	3.74	

A Tabela 17 apresenta os desvios relativos médios obtidos a partir das médias da Tabela 16.

Tabela 17 – Desvio Relativo Médio *Total Flow Time* com *setup* - Cenário 3

	Contagem de nós	Tempo	Não Solucionados
LB_1^{TFT+ST}	0.10	0.12	0.12
LB_2^{TFT+ST}	0.13	0.00	0.00
LB_3^{TFT+ST}	0.00	0.06	0.02

Tabela 18 – Comparação dos limitantes para *Total Flow Time* com *setup* - Cenário 4

Tarefas	Máquinas	Contagem de Nós					Tempo (s)				Não Solucionados		
		LB_1^{TFT+ST}	LB_2^{TFT+ST}	LB_3^{TFT+ST}	LB_1^{TFT+ST}	LB_2^{TFT+ST}	LB_3^{TFT+ST}	LB_1^{TFT+ST}	LB_2^{TFT+ST}	LB_3^{TFT+ST}	LB_1^{TFT+ST}	LB_2^{TFT+ST}	LB_3^{TFT+ST}
10	2	1.86E+03	1.56E+03	1.20E+03	0.033	0.022	0.024	0	0	0			
10	3	1.32E+04	1.16E+04	1.79E+04	0.174	0.156	0.187	0	0	0			
10	4	5.19E+04	5.26E+04	4.01E+04	0.524	0.417	0.419	0	0	0			
10	5	1.11E+05	9.65E+04	9.19E+04	1.108	0.834	0.940	0	0	0			
10	7	3.05E+05	2.42E+05	2.49E+05	2.849	2.016	2.415	0	0	0			
10	10	6.91E+05	5.26E+05	5.54E+05	4.369	4.197	4.892	0	0	0			
12	2	1.70E+04	1.65E+04	1.19E+04	0.242	0.318	0.341	0	0	0			
12	3	1.68E+05	1.35E+05	1.21E+05	3.159	2.036	2.497	0	0	0			
12	4	8.59E+05	6.46E+05	5.93E+05	10.668	5.908	8.880	0	0	0			
12	5	1.06E+06	9.44E+05	1.26E+06	16.945	9.800	17.235	0	0	0			
12	7	6.85E+06	6.48E+06	6.53E+06	70.271	57.843	67.667	0	0	0			
12	10	2.11E+07	1.79E+07	1.84E+07	222.954	164.728	196.393	0	0	0			
14	2	9.03E+04	9.58E+04	6.59E+04	1.999	1.874	2.691	0	0	0			
14	3	4.12E+06	3.06E+06	2.00E+06	62.388	43.660	41.695	0	0	0			
14	4	2.67E+07	4.33E+07	3.12E+07	361.938	401.366	358.702	0	0	0			
14	5	1.68E+07	1.44E+07	1.44E+07	376.944	255.084	337.281	0	0	0			
14	7	2.08E+08	2.34E+08	2.15E+08	2362.521	2197.001	2275.122	8	9	8			
14	10	2.60E+08	2.59E+08	2.36E+08	3452.525	3085.414	3310.143	16	11	14			
16	2	4.64E+05	5.60E+05	3.65E+05	25.500	18.455	20.377	0	0	0			
16	3	1.09E+07	1.09E+07	7.58E+06	435.885	279.153	317.886	0	0	0			
16	4	8.14E+07	7.32E+07	7.46E+07	2695.172	1872.869	2437.614	9	4	6			
18	2	3.68E+06	4.57E+06	2.73E+06	265.789	178.755	203.287	0	0	0			
18	3	4.54E+07	6.71E+07	4.46E+07	2864.808	2692.193	2823.943	13	9	12			
18	4	1.32E+08	1.21E+08	1.30E+08	3531.263	3492.868	3507.656	19	19	19			
20	2	1.47E+07	2.27E+07	1.23E+07	1417.233	1157.732	1159.092	1	1	1			
20	3	6.49E+07	9.31E+07	6.23E+07	3600.001	3600.001	3600.001	20	20	20			
20	4	1.40E+08	1.49E+08	1.00E+08	3600.001	3600.001	3600.001	20	20	20			
Média		3.85E+07	4.16E+07	3.56E+07	940.27	856.47	899.90	3.93	3.44	3.70			

A Tabela 19 apresenta os desvios relativos médios obtidos a partir das médias da Tabela 18.

Tabela 19 – Desvio Relativo Médio *Total Flow Time* com *setup* - Cenário 4

	Contagem de nós	Tempo	Não Solucionados
LB_1^{TFT+ST}	0.08	0.10	0.14
LB_2^{TFT+ST}	0.17	0.00	0.00
LB_3^{TFT+ST}	0.00	0.05	0.08

O aumento dos limites de distribuição dos *setups* resultou na diminuição do número de nós para todas as variações de limitantes. Logo, pode-se perceber que este fator não é significativo para as comparações realizadas neste trabalho. Foi possível observar que para todos os cenários o limitante LB_3^{TFT+ST} foi o que mais reduziu o número de nós. Por outro lado, LB_2^{TFT+ST} solucionou mais problemas em um menor tempo de processamento. Os resultados de LB_1^{TFT+ST} se mostraram pouco eficientes para todos os cenários. Como o intuito final do $B\&B$ é a redução do tempo final de busca, pode-se considerar a busca de $\dot{Setup}_k(j)$ inviável para os casos estudados neste trabalho.

7.1.4 Resultados para *Total Tardiness* com *setup*

Como não se observou variação nos limites da distribuição dos *setups* para o caso de *TFT* com *setup*, o caso de *TT* com *setup* utilizou apenas a primeira distribuição de *setups*, 1 – 124, que foi a mais rápida. Neste caso, as *due dates* das tarefas tiveram a necessidade de ser recalculadas, pois com a adição dos tempos de *setup*, os prazos utilizados na seção 7.1.2 seriam curtos para a avaliação do método. A equação 104 foi utilizada para a geração do P e os quatro cenários de distribuição apresentados na seção 7.1.2 foram utilizados.

$$P = \max \left(\max_{1 \leq k \leq m} \sum_{j=1}^n p_{jk} + \min_{1 \leq j \leq n} \sum_{q=1}^{k-1} p_{jq} + \min_{1 \leq j \leq n} \sum_{q=k+1}^m p_{jq}, \max_{1 \leq j \leq n} \sum_{k=1}^m p_{jk} \right) + \frac{\sum_{k=1}^m \sum_{j=1}^n \sum_{i=1}^n s_{ijk}}{nm} \quad (104)$$

A ideia é adicionar a média dos tempos de *setup* de cada tarefas em cada máquina. As mesmas três variações dos limitantes, LB_1^{TT+ST} , LB_2^{TT+ST} e LB_3^{TT+ST} foram realizadas de forma análoga ao caso de *TFT* com *setup*. Os testes realizados para os quatro cenários de *TT* com *setup* estão apresentados nas Tabelas 20, 22, 24 e 26.

Tabela 20 – Comparação dos limitantes para *Total Tardiness* com *setup* - Cenário 1

Tarefas	Máquinas	Contagem de Nós			Tempo (s)			Não Solucionados		
n	m	LB_1^{TFT+ST}	LB_2^{TFT+ST}	LB_3^{TFT+ST}	LB_1^{TT+ST}	LB_2^{TT+ST}	LB_3^{TT+ST}	LB_1^{TFT+ST}	LB_2^{TFT+ST}	LB_3^{TFT+ST}
10	2	4.78E+03	5.24E+03	4.75E+03	0.046	0.045	0.070	0	0	0
10	3	1.35E+04	1.22E+04	1.11E+04	0.086	0.078	0.094	0	0	0
10	4	4.90E+04	3.69E+04	3.75E+04	0.212	0.158	0.207	0	0	0
10	5	1.26E+05	1.20E+05	1.33E+05	0.481	0.490	0.595	0	0	0
10	7	1.28E+05	1.25E+05	1.39E+05	0.532	0.485	0.640	0	0	0
10	10	2.20E+05	2.10E+05	2.20E+05	1.133	1.007	1.254	0	0	0
12	2	5.50E+04	6.14E+04	5.43E+04	0.736	0.614	0.757	0	0	0
12	3	1.61E+05	2.19E+05	1.60E+05	1.812	1.958	1.958	0	0	0
12	4	6.26E+05	5.14E+05	5.91E+05	3.756	3.156	3.736	0	0	0
12	5	1.55E+06	1.49E+06	1.60E+06	8.001	7.027	8.382	0	0	0
12	7	4.47E+06	4.31E+06	4.47E+06	21.713	17.322	21.061	0	0	0
12	10	1.70E+07	1.19E+07	1.70E+07	76.238	49.857	79.681	0	0	0
14	2	1.14E+06	1.24E+06	1.11E+06	20.106	15.745	20.261	0	0	0
14	3	3.10E+06	3.12E+06	3.04E+06	42.074	32.605	42.357	0	0	0
14	4	8.98E+07	8.87E+07	8.39E+07	296.074	280.339	307.695	1	1	1
14	5	2.96E+07	2.93E+07	2.96E+07	167.342	140.538	177.433	0	0	0
14	7	1.11E+08	1.11E+08	1.34E+08	493.997	465.257	613.687	1	1	1
14	10	2.18E+08	2.28E+08	2.19E+08	1333.918	1271.458	1387.241	3	1	3
16	2	2.29E+07	2.60E+07	2.25E+07	503.505	401.181	494.221	0	0	0
16	3	3.60E+07	4.25E+07	3.56E+07	959.381	775.627	990.134	2	0	2
16	4	1.92E+08	1.94E+08	1.87E+08	1749.466	1524.070	1797.742	3	3	5
18	2	1.04E+08	1.48E+08	1.01E+08	3207.501	3173.919	3227.694	14	12	14
18	3	1.09E+08	1.49E+08	1.06E+08	3462.082	3357.033	3463.336	18	17	18
18	4	2.57E+08	3.18E+08	2.49E+08	3600.001	3600.001	3600.001	20	20	20
20	2	9.12E+07	1.39E+08	8.74E+07	3438.036	3431.203	3434.358	19	19	19
20	3	8.40E+07	1.18E+08	8.11E+07	3600.001	3600.001	3600.001	20	20	20
20	4	2.80E+08	3.15E+08	2.65E+08	3600.001	3600.001	3600.001	20	20	20
Média		6.12E+07	7.15E+07	6.04E+07	984.75	953.75	995.36	4.48	4.22	4.56

A Tabela 21 apresenta os desvios relativos médios obtidos a partir das médias da Tabela 20.

Tabela 21 – Desvio Relativo Médio *Total Tardiness* com *setup* - Cenário 1

	Contagem de nós	Tempo	Não Solucionados
LB_1^{TT+ST}	0.01	0.03	0.06
LB_2^{TT+ST}	0.18	0.00	0.00
LB_3^{TT+ST}	0.00	0.04	0.08

Tabela 22 – Comparação dos limitantes para *Total Tardiness* com *setup* - Cenário 2

Tarefas		Máquinas		Contagem de Nós					Tempo (s)			Não Solucionados		
n	m	LB_1^{TFT+ST}	LB_2^{TFT+ST}	LB_3^{TFT+ST}	LB_1^{TFT+ST}	LB_2^{TFT+ST}	LB_3^{TFT+ST}	LB_1^{TFT+ST}	LB_2^{TFT+ST}	LB_3^{TFT+ST}	LB_1^{TFT+ST}	LB_2^{TFT+ST}	LB_3^{TFT+ST}	
10	2	4.84E+02	4.88E+02	4.80E+02	0.005	0.004	0.006	0	0	0				
10	3	2.59E+03	2.60E+03	2.58E+03	0.026	0.019	0.023	0	0	0				
10	4	3.36E+04	3.37E+04	3.46E+04	0.121	0.126	0.133	0	0	0				
10	5	9.57E+04	9.60E+04	9.56E+04	0.359	0.292	0.353	0	0	0				
10	7	9.53E+04	9.53E+04	9.53E+04	0.362	0.325	0.419	0	0	0				
10	10	3.21E+05	3.21E+05	3.21E+05	1.451	1.233	1.583	0	0	0				
12	2	4.33E+03	4.65E+03	4.29E+03	0.062	0.048	0.064	0	0	0				
12	3	2.17E+05	2.17E+05	2.17E+05	0.882	0.730	0.891	0	0	0				
12	4	3.40E+05	3.19E+05	3.20E+05	1.806	1.172	1.470	0	0	0				
12	5	1.60E+06	1.46E+06	1.58E+06	5.538	5.282	6.098	0	0	0				
12	7	1.29E+06	1.89E+06	1.29E+06	5.730	6.826	6.131	0	0	0				
12	10	1.49E+07	1.49E+07	1.49E+07	63.698	58.601	67.557	0	0	0				
14	2	2.01E+04	2.05E+04	1.99E+04	0.407	0.282	0.427	0	0	0				
14	3	4.48E+05	4.51E+05	4.48E+05	3.131	3.177	3.238	0	0	0				
14	4	2.85E+07	2.87E+07	2.87E+07	96.814	88.980	101.374	0	0	0				
14	5	6.71E+07	6.71E+07	6.71E+07	234.246	218.574	233.382	0	0	0				
14	7	1.42E+08	1.46E+08	1.40E+08	503.468	494.630	517.263	2	2	2				
14	10	1.97E+08	1.99E+08	1.96E+08	1023.596	981.500	1056.563	2	2	2				
16	2	6.96E+04	7.15E+04	6.93E+04	1.030	0.747	1.048	0	0	0				
16	3	4.15E+06	4.17E+06	4.15E+06	29.351	22.989	29.961	0	0	0				
16	4	1.25E+08	1.26E+08	1.21E+08	564.998	524.822	574.082	2	2	2				
18	2	2.04E+06	2.05E+06	2.03E+06	16.751	12.863	18.486	0	0	0				
18	3	1.54E+07	1.55E+07	1.54E+07	139.105	104.059	139.634	0	0	0				
18	4	2.88E+08	3.13E+08	2.90E+08	1946.608	1795.095	1999.944	9	7	9				
20	2	9.78E+06	1.20E+07	9.59E+06	384.269	321.468	389.539	1	1	1				
20	3	1.31E+08	1.41E+08	1.31E+08	1065.121	934.168	1069.224	4	4	4				
20	4	2.39E+08	2.79E+08	2.37E+08	2383.763	2246.778	2390.495	11	10	11				
Média		4.70E+07	5.01E+07	4.67E+07	313.80	289.81	318.87	1.15	1.04	1.15				

A Tabela 23 apresenta os desvios relativos médios obtidos a partir das médias da Tabela 22.

Tabela 23 – Desvio Relativo Médio *Total Tardiness* com *setup* - Cenário 2

	Contagem de nós	Tempo	Não Solucionados
LB_1^{TT+ST}	0.01	0.08	0.11
LB_2^{TT+ST}	0.07	0.00	0.00
LB_3^{TT+ST}	0.00	0.10	0.11

Tabela 24 – Comparação dos limitantes para *Total Tardiness* com *setup* - Cenário 3

Tarefas	Máquinas	Contagem de Nós				Tempo (s)			Não Solucionados		
		LB_1^{FT+ST}	LB_2^{FT+ST}	LB_3^{FT+ST}	LB_1^{FT+ST}	LB_2^{FT+ST}	LB_3^{FT+ST}	LB_1^{FT+ST}	LB_2^{FT+ST}	LB_3^{FT+ST}	
10	2	1.16E+03	1.31E+03	1.17E+03	0.013	0.011	0.018	0	0	0	
10	3	4.03E+03	4.29E+03	4.16E+03	0.033	0.029	0.037	0	0	0	
10	4	7.48E+04	7.63E+04	7.48E+04	0.269	0.213	0.249	0	0	0	
10	5	1.92E+05	1.92E+05	1.90E+05	0.634	0.557	0.656	0	0	0	
10	7	2.95E+05	2.94E+05	2.95E+05	1.046	0.938	1.142	0	0	0	
10	10	5.88E+05	5.87E+05	5.88E+05	2.437	2.297	2.631	0	0	0	
12	2	1.58E+04	1.74E+04	1.55E+04	0.241	0.191	0.279	0	0	0	
12	3	9.80E+04	1.72E+05	1.19E+05	0.718	0.794	0.801	0	0	0	
12	4	2.89E+06	2.73E+06	2.81E+06	9.157	9.080	9.613	0	0	0	
12	5	3.51E+06	2.65E+06	2.36E+06	13.056	10.429	10.011	0	0	0	
12	7	1.00E+07	1.00E+07	9.99E+06	37.130	34.112	38.332	0	0	0	
12	10	3.60E+07	3.65E+07	3.50E+07	148.177	147.942	153.428	0	0	0	
14	2	1.46E+05	1.67E+05	1.43E+05	3.305	2.466	3.234	0	0	0	
14	3	4.04E+05	4.21E+05	4.03E+05	7.691	5.664	7.998	0	0	0	
14	4	5.00E+07	4.90E+07	5.00E+07	179.283	166.335	195.968	0	0	0	
14	5	1.51E+08	1.73E+08	1.46E+08	523.325	574.062	532.005	1	1	1	
14	7	4.20E+08	4.28E+08	4.02E+08	1437.280	1421.481	1439.722	7	7	7	
14	10	4.30E+08	4.55E+08	4.27E+08	2017.606	1945.543	2024.711	9	8	9	
16	2	1.58E+06	1.86E+06	1.56E+06	44.502	36.611	46.671	0	0	0	
16	3	5.92E+06	6.44E+06	5.77E+06	137.757	103.982	143.995	0	0	0	
16	4	8.86E+07	9.91E+07	1.01E+08	501.019	465.891	571.224	1	1	1	
18	2	1.90E+07	2.28E+07	1.87E+07	733.529	581.723	747.417	0	0	1	
18	3	6.60E+07	5.25E+07	6.46E+07	1527.824	1051.665	1590.114	2	1	3	
18	4	1.69E+08	2.17E+08	2.07E+08	2838.190	2423.270	2935.563	11	7	12	
20	2	7.13E+07	1.09E+08	6.80E+07	3274.353	3219.589	3319.156	17	16	17	
20	3	9.15E+07	1.42E+08	7.78E+07	3600.001	3474.409	3600.001	20	17	20	
20	4	1.19E+08	2.45E+08	1.11E+08	3600.001	3600.001	3600.001	20	20	20	
Média		6.43E+07	7.60E+07	6.41E+07	764.39	714.05	776.85	3.26	2.89	3.37	

A Tabela 25 apresenta os desvios relativos médios obtidos a partir das médias da Tabela 24.

Tabela 25 – Desvio Relativo Médio *Total Tardiness* com *setup* - Cenário 3

	Contagem de nós	Tempo	Não Solucionados
LB_1^{TT+ST}	0.00	0.07	0.13
LB_2^{TT+ST}	0.19	0.00	0.00
LB_3^{TT+ST}	0.00	0.09	0.17

Tabela 26 – Comparação dos limitantes para *Total Tardiness* com *setup* - Cenário 4

Tarefas	Máquinas	Contagem de Nós					Tempo (s)	Não Solucionados			
		LB_1^{FT+ST}	LB_2^{FT+ST}	LB_3^{FT+ST}	LB_1^{TT+ST}	LB_2^{TT+ST}		LB_3^{TT+ST}	LB_1^{FT+ST}	LB_2^{FT+ST}	LB_3^{FT+ST}
10	2	6.87E+02	7.17E+02	6.83E+02	0.008	0.006	0.010	0	0	0	
10	3	2.22E+03	2.29E+03	2.22E+03	0.021	0.017	0.023	0	0	0	
10	4	4.56E+04	4.79E+04	4.93E+04	0.167	0.152	0.185	0	0	0	
10	5	1.95E+05	1.92E+05	1.91E+05	0.610	0.568	0.671	0	0	0	
10	7	3.01E+05	3.24E+05	2.92E+05	1.052	0.994	1.068	0	0	0	
10	10	6.17E+05	6.21E+05	6.23E+05	2.462	2.695	2.689	0	0	0	
12	2	5.54E+03	5.97E+03	5.46E+03	0.083	0.065	0.085	0	0	0	
12	3	9.24E+04	9.23E+04	9.12E+04	0.565	0.529	0.802	0	0	0	
12	4	2.19E+06	2.20E+06	2.24E+06	7.658	6.699	7.270	0	0	0	
12	5	3.23E+06	3.14E+06	3.15E+06	12.627	10.821	12.719	0	0	0	
12	7	8.68E+06	8.68E+06	8.68E+06	33.344	29.248	32.616	0	0	0	
12	10	2.91E+07	3.05E+07	2.91E+07	122.747	125.320	132.387	0	0	0	
14	2	6.65E+04	7.01E+04	6.52E+04	1.373	0.984	1.393	0	0	0	
14	3	3.19E+05	3.26E+05	3.18E+05	4.966	3.436	5.343	0	0	0	
14	4	9.54E+07	9.59E+07	9.16E+07	263.491	262.853	268.077	1	1	1	
14	5	5.61E+07	5.61E+07	5.60E+07	212.443	207.403	223.198	0	0	0	
14	7	3.34E+08	3.31E+08	3.25E+08	1190.850	1144.726	1191.390	4	4	4	
14	10	5.24E+08	5.25E+08	5.04E+08	2233.821	2193.730	2261.978	10	10	10	
16	2	2.62E+05	2.78E+05	2.60E+05	7.124	5.130	7.644	0	0	0	
16	3	2.06E+07	2.10E+07	2.05E+07	137.631	121.731	147.729	0	0	0	
16	4	1.47E+08	1.48E+08	1.47E+08	694.447	664.442	722.311	1	1	1	
18	2	9.24E+05	9.85E+05	9.19E+05	38.099	26.665	38.699	0	0	0	
18	3	1.33E+07	1.41E+07	1.32E+07	463.026	331.322	476.436	1	0	1	
18	4	5.08E+08	4.50E+08	4.93E+08	2655.603	2319.306	2683.732	12	9	12	
20	2	2.36E+07	3.03E+07	2.30E+07	1067.561	906.618	1085.965	4	2	4	
20	3	1.71E+08	1.82E+08	1.57E+08	2502.549	2261.495	2522.729	12	9	12	
20	4	2.71E+08	2.74E+08	2.50E+08	3340.092	3162.652	3351.768	17	16	17	
Média		8.18E+07	8.05E+07	7.88E+07	555.35	510.73	562.18	2.30	1.93	2.30	

A Tabela 27 apresenta os desvios relativos médios obtidos da Tabela 26.

Tabela 27 – Desvio Relativo Médio *Total Tardiness* com *setup* - Cenário 4

	Contagem de nós	Tempo	Não Solucionados
LB_1^{TT+ST}	0.04	0.09	0.19
LB_2^{TT+ST}	0.02	0.00	0.00
LB_3^{TT+ST}	0.00	0.10	0.19

Para os casos de *TT* com *setup* foi possível perceber que taxas de dispersão maiores facilitam a chegada à solução ótima, assim como no caso sem *setup*. O fator de atraso aumentado dificulta a obtenção das respostas ótimas principalmente em casos com muitas tarefas. Não diferente dos resultados obtidos na seção 7.1.3, o limitante LB_2^{TT+ST} obteve o melhor desempenho em termos de tempo computacional e número de problemas solucionados. O limitante LB_3^{TT+ST} realizou a maior redução em termos de números de nós e LB_1^{TT+ST} se mostrou inviável em todos os quesitos.

7.2 Resultados da Modelagem *MILP*

Para a agilidade dos testes, os problemas utilizados nas modelagens *MILP* foram as de melhor desempenho observadas nos testes anteriores. A base de dados de tempos de processamento foi a mesma. Os tempos de *setup* foram os gerados a partir da distribuição uniforme que varia de 1 até 124. As *due dates* utilizadas foram as obtidas no cenário 2 da seção 7.1.2 para o caso de *TT* sem *setup* e para o caso de *TT* com *setup* foram utilizadas as *due dates* do cenário 2 da seção 7.1.4.

As modelagens foram solucionadas pela biblioteca *IBM ILOG CPLEX Optimization Studio v12.6.0* na linguagem *Python 3.6*. A máquina virtual cedida pela EESC - UPS apresentou instabilidades inesperadas na execução do programa, logo, foi utilizada uma *CPU* física na realização dos experimentos computacionais. O processador utilizado foi *Intel Core i7-8700K* com *overclock* de *4.7GHz*, *16GB RAM DDR4 3000MHz* no sistema operacional *Windows 10 Professional*. O código foi executado em paralelo com a utilização de *12 threads*. O tempo limite de cada problema foi definido em 3600 segundos. A Tabela 28 apresenta a média do tempo de processamento que cada tipo de problema levou para ser solucionado.

Tabela 28 – Tempo (s) - Modelos MILP

n	m	<i>TFT</i>	<i>TFT - setup</i>	<i>TT</i>	<i>TT - setup</i>
10	2	0.215	109.65	0.3165	161.42
10	3	0.1915	97.67	0.9255	472.01
10	4	0.176	89.76	0.412	210.12
10	5	0.2535	148.68	0.533	312.60
10	7	0.2875	168.62	0.5825	341.64
10	10	0.4365	256.01	0.934	547.79
12	2	0.191	128.82	1.547	1043.41
12	3	0.229	154.45	0.6565	442.79
12	4	0.458	308.91	1.2825	865.01
12	5	0.4725	366.49	1.1245	872.21
12	7	1.1035	855.93	2.0495	1589.69
12	10	2.2995	1783.60	3.335	2586.78
14	2	0.269	239.95	1.253	1117.67
14	3	0.6855	611.46	2.518	2246.04
14	4	1.436	1280.90	6.565	3600.00
14	5	3.034	3112.25	16.297	3600.00
14	7	8.5765	3600.00	15.5215	3600.00
14	10	37.0945	3600.00	46.6465	3600.00
16	2	0.6545	772.09	5.5915	3600.00
16	3	2.6775	3158.54	9.161	3600.00
16	4	4.5525	3600.00	15.077	3600.00
18	2	1.015	1376.96	12.111	3600.00
18	3	5.0815	3600.00	43.5085	3600.00
18	4	46.9845	3600.00	423.211	3600.00
20	2	3.381	3600.00	228.478	3600.00
20	3	51.888	3600.00	374.6125	3600.00
20	4	219.26	3600.00	1925.057	3600.00

Pode-se perceber que para o caso de *TFT* sem *setup*, o pacote *CPLEX* consegue encontrar a solução ótima para 20 tarefas e 4 máquinas em tempos relativamente baixos. A modelagem para *TT* sem *setup* difere pouco da modelagem para *TFT*, porém, esta se vê numa região mais restrita, por isso sua complexidade é aumentada. Quando há a adição dos tempos de *setup* o algoritmo encontra muito mais dificuldade para chegar na solução ótima. É possível perceber que para o caso de *TFT*, ocorre o estouro do tempo limite de execução para as 20 variações de tempo de processamento nos problemas de 14 tarefas e 7 máquinas. A redução no número de máquinas fez com que alguns problemas de 16 e 18 tarefas pudessem ser solucionados dentro do tempo limite. O caso de *TT* com *setup* apresentou o maior número de restrições e, conseqüentemente, a maior complexidade. O algoritmo foi capaz de solucionar problemas menores que 14 tarefas e quatro máquinas, logo, o aumento do número de tarefas foi significativo para a limitação da resolução dos problemas.

Conclusão

O presente trabalho propôs novos limitantes para o método *Branch & Bound* para solucionar o problema de *flowshop* com bloqueio, com e sem *setup* dependente da sequência e das máquinas, com o objetivo de minimizar o tempo total de fluxo e o tempo total de atraso, separadamente. Essa proposta foi incentivada pelo fato de que foram encontrados poucos trabalhos na literatura que abordam os mesmos problemas solucionados pelo método em específico, para os casos sem *setup*. Além disso, para os casos com *setup*, não foram encontradas abordagens dos ambientes tratados neste trabalho.

Para cada ambiente tratado, foi elaborado um limitante inferior diferente para o método *Branch & Bound* baseado nos melhores métodos encontrados na literatura. Foi possível abstrair e utilizar as partes mais eficientes de cada um dos métodos e integrá-los em uma solução que resulta na redução dos tempos de busca do algoritmo.

No ambiente que visa reduzir o *TFT* sem a consideração de *setup*, o limitante LB_{TFT} se apresentou com maior eficiência tanto no número de nós, quanto no tempo final de processamento na maioria dos casos.

Para o objetivo de *TT* sem *setup*, o limitante LB_{TT} obteve resultados superiores ao desenvolvido por Toumi et al. (2017a) em todos os cenários e mostrou desempenho superior a Ronconi e Armentano (2001) em praticamente todos os casos, com exceção às combinações onde $m = 2$ no primeiro cenário.

Como não foram encontrados métodos para comparação nos casos com *setup*, os limitantes LB^{TFT+ST} e LB^{TT+ST} para *TFT* e *TT*, respectivamente, foram modificados em busca da melhor eficiência. O caso que calcula o maior valor do LB , LB_3^{TFT+ST} e LB_3^{TT+ST} se mostrou mais eficiente em relação ao número de nós. Entretanto, os limitantes LB_2^{TFT+ST} e LB_2^{TT+ST} , apresentou melhor desempenho tanto em termos de tempo computacional quanto no número de problemas solucionados no tempo limite estipulado de 3600 segundos. O LB_1^{TFT+ST} e LB_1^{TT+ST} apresentaram alto custo computacional e baixa eficiência no limitante, logo, estes não são viáveis.

As modelagens *MILP* foram elaboradas e utilizadas para encontrar a capacidade de busca de resposta do pacote *Cplex* da empresa *IBM* em no máximo 3600 segundos. Para os

casos de *TFT* e *TT* sem *setup*, os resultados obtidos foram aceitáveis, uma vez que todas as variações de (n, m) conseguiram encontrar a resposta ótima no tempo máximo estipulado. Os casos com *setup* encontraram mais dificuldade, mas para as problemas menores que $(14, 7)$ para *TFT* e $(14, 4)$ *TT*, todos os problemas foram solucionados de forma ótima. Para fins de medida de desempenho, seria interessante comparar os resultados do *B&B* com os resultados da modelagem *MILP*, porém, por motivos de problemas estruturais, as comparações poderão ser feitas em trabalhos futuros.

Tomados em conjunto, os resultados obtidos indicam que o método *Branch & Bound* para pequenos e médios problemas possui grande potencial para aplicações práticas. Visto que sua tomada de decisão é obtida em tempos razoavelmente baixos para $n < 20$ nos casos sem *setup* e $n < 18$ nos casos com *setup*, é possível perceber um eventual fortalecimento entre a teoria e a prática do *scheduling*.

Trabalhos Futuros

Mais estudos são necessários para melhoria contínua da obtenção de boas respostas para o sequenciamento de tarefas. Em relação ao algoritmo *Branch & Bound*, a utilização da regra exploração de melhor limitante pode ser implementada juntamente com um *UB* eficiente e uma regra de dominância, visto que essa possui deficiências em termos de economia de memória e na atualização do limitante superior.

Para o caso de *TT*, visto que a ideia de Ronconi e Armentano (2001) possui peculiaridades, uma variação do algoritmo pode ser elaborado para buscar melhor eficiência no cenário 1 apresentado na seção 7.1.2.

As comparações realizadas nos problemas com *setup* foram realizadas a partir de variações dos limitantes elaborados neste trabalho. Para um futuro trabalho podem ser realizadas adaptações dos métodos que abordam ambientes semelhantes sem *setup*, como Toumi et al. (2017b) para *TT* e Moslehi e Khorasanian (2013) para *TFT*, para verificar seu desempenho nos casos com *setup*.

Como as configurações do computador utilizado para executar MILP eram diferentes do utilizado para o *B&B*, a comparação entre os métodos ficou inviabilizada. Seria interessante realizar novos testes em uma mesma máquina.

Em relação ao custo computacional, uma das sugestões para trabalhos futuros é a paralelização do código em um número maior de *threads*. Muitos dos cálculos não possuem precedentes, logo, podem ser executados de forma simultânea. Esta sugestão pode melhorar de forma significativa as buscas dos mínimos nos casos com *setup*, visto que várias delas não tem relação, além de possuírem complexidade elevada.

Referências Bibliográficas

BAKER, K. R. **Principles of Sequencing and Scheduling**. 3rd. ed. [S.l.]: John Wiley and Sons, Inc., 2009. ISBN 9780470391655, 9780470451793.

BANSAL, S. P. Minimizing the sum of completion times of n jobs over m machines in a flowshop: A branch and bound approach. **A I I E Transactions**, 1977. v. 9, n. 3, p. 306–311, 1977.

BLANCHARD, D. **Supply Chain Management**. 2rd. ed. [S.l.]: Best Practices, Wiley, 2009. ISBN 9780470531884.

CHUNG, C. S.; FLYNN, J.; KIRCA, O. A branch and bound algorithm to minimize the total flow time for m -machine permutation flowshop problems. **International Journal of Production Economics**, 2002. v. 79, n. 3, p. 185–196, 2002.

CHUNG, C. S.; FLYNN, J.; KIRCA, O. A branch and bound algorithm to minimize the total tardiness for m -machine permutation flowshop problems. **European Journal of Operational Research**, 2006. v. 174, n. 1, p. 1–10, 2006.

COMPANYS, R.; MATEO, M. Different behaviour of a double branch-and-bound algorithm on $fm|prmu|cmax$ and $fm|block|cmax$ problems. **Computers & Operations Research**, 2007. v. 34, n. 4, p. 938 – 953, 2007.

FRAMINAN, J. M.; GUPTA, J. N.; LEISTEN, R. A review and classification of heuristics for permutation flow-shop scheduling with makespan objective. **Journal of the Operational Research Society**, 2004. Springer, v. 55, n. 12, p. 1243–1255, 2004.

GRAHAM, R. et al. Optimization and approximation in deterministic sequencing and scheduling: a survey. **Annals of Discrete Mathematics**, 1979. v. 5, p. 287 – 326, 1979.

GRAVES, S. C. A review of production scheduling. **Operations research**, 1981. INFORMS, v. 29, n. 4, p. 646–675, 1981.

GROOVER, M. P. **Automação Industrial e Sistemas de Manufatura**. 3rd. ed. [S.l.]: Pearson, 2008. ISBN 8576058715, 9788576058717.

GUPTA, J. N. D.; HARIRI, A.; POTTS, C. Scheduling a two-stage hybrid flow shop with parallel machines at the first stage. **Annals of Operations Research**, 1997. v. 69, n. 0, p. 171–191, 1997.

- HALL, N. G.; SRISKANDARAJAH, C. A survey of machine scheduling problems with blocking and no-wait in process. **Operations research**, 1996. INFORMS, v. 44, n. 3, p. 510–525, 1996.
- IGNALL, E.; SCHRAGE, L. Application of the branch and bound technique to some flow-shop scheduling problems. **Operations research**, 1965. INFORMS, v. 13, n. 3, p. 400–412, 1965.
- JESUS, E.; RODRIGUES, M. T. Sequenciamento de tarefas com tempos de setup via branch and bound. **Revista GEINTEC - Gestão, Inovação e Tecnologias**, 2014. v. 4, n. 3, 2014.
- JOHNSON, S. M. Optimal two and three stage production schedules with set-up time included. **Naval Research Logistics Quarterly**, 1954. v. 1, n. 1, p. 61–68, 1954.
- LEE, D. Y.; DICESARE, F. Scheduling flexible manufacturing systems using petri nets and heuristic search. **IEEE Transactions on robotics and automation**, 1994. IEEE, v. 10, n. 2, p. 123–132, 1994.
- LOMNICKI, Z. A branch-and-bound algorithm for the exact solution of the three-machine scheduling problem. **Journal of the Operational Research Society**, 1965. Springer, v. 16, n. 1, p. 89–100, 1965.
- MACCARTHY, B. L.; LIU, J. Addressing the gap in scheduling research: a review of optimization and heuristic methods in production scheduling. **International Journal of Production Research**, 1993. v. 31, n. 1, p. 59–79, 1993.
- MCCORMICK, S. et al. Sequencing in an assembly line with blocking to minimize cycle time. 1989. v. 37, p. 925–935, 12 1989.
- MOSLEHI, G.; KHORASANIAN, D. Optimizing blocking flow shop scheduling problem with total completion time criterion. **Computers & Operations Research**, 2013. v. 40, n. 7, p. 1874 – 1883, 2013.
- NAWAZ, M.; JR, E. E. E.; HAM, I. A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. **Omega**, 1983. Elsevier, v. 11, n. 1, p. 91–95, 1983.
- PAN, Q.-K.; WANG, L. Effective heuristics for the blocking flowshop scheduling problem with makespan minimization. **Omega**, 2012. v. 40, n. 2, p. 218 – 229, 2012.
- PINEDO, M. L. **Scheduling: Theory, Algorithms, and Systems**. 3rd. ed. [S.l.]: Springer Publishing Company, Incorporated, 2008. ISBN 0387789340, 9780387789347.
- RIOS-MERCADO, R. Z.; BARD, J. A branch-and-bound algorithm for permutation flow shops with sequence-dependent setup times. **IIE Transactions**, 1999. v. 31, n. 8, p. 721–731, 1999.
- RONCONI, D. P. A note on constructive heuristics for the flowshop problem with blocking. **International Journal of Production Economics**, 2004. Elsevier, v. 87, n. 1, p. 39–48, 2004.
- RONCONI, D. P. A branch-and-bound algorithm to minimize the makespan in a flowshop with blocking. **Annals of Operations Research**, 2005. v. 138, n. 1, p. 53–65, 2005.

- RONCONI, D. P.; ARMENTANO, V. A. Lower bounding schemes for flowshops with blocking in-process. **Journal of the Operational Research Society**, 2001. v. 52, n. 11, p. 1289–1297, 2001.
- RONCONI, D. P.; KAWAMURA, M. S. The single machine earliness and tardiness scheduling problem: lower bounds and a branch-and-bound algorithm. **Computational and Applied Mathematics**, 2010. scielo, v. 29, p. 107 – 124, 06 2010.
- SANCHES, F. B.; TAKANO, M. I.; NAGANO, M. S. Evaluation of heuristics for a branch and bound algorithm to minimize the makespan in a flowshop with blocking. **Acta Scientiarum**, 2016. v. 38, n. 3, p. 321–326, 2016.
- SLACK, N. **Administração da Produção**. 3rd. ed. [S.l.]: Atlas, 2009. ISBN 8522453535,9788522453535.
- TAKANO, M. I.; NAGANO, M. S. A branch-and-bound method to minimize the makespan in a permutation flow shop with blocking and setup times. **Cogent Engineering**, 2017. Cogent, v. 4, n. 1, 2017.
- TOUMI, S. et al. Branch and bound algorithm for solving blocking flowshop scheduling problem with total tardiness and total weighted tardiness criteria. **International Journal of Operational Research**, 2017. Inderscience Publishers (IEL), v. 30, n. 4, p. 441–459, 2017.
- TOUMI, S. et al. Branch-and-bound algorithm for solving blocking flowshop scheduling problems with makespan criterion. **International Journal of Mathematics in Operational Research**, 2017. v. 10, n. 1, p. 34–48, 2017.
- WANG, J.-B.; SUN, L.-H.; SUN, L.-Y. Single-machine total completion time scheduling with a time-dependent deterioration. **Applied Mathematical Modelling**, 2011. Elsevier, v. 35, n. 3, p. 1506–1511, 2011.
- WOLSEY, L. A. **Integer programming**. New York, NY, USA: Wiley-Interscience, 1998.