

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE ENGENHARIA DE PRODUÇÃO
ENGENHARIA DE PRODUÇÃO**

CAROLINA MEIRA COSTA

**AVALIAÇÃO E ADAPTAÇÃO DE MÉTODOS HEURÍSTICOS PARA O
PROBLEMA DE PROGRAMAÇÃO *NO-IDLE FLOW SHOP* COM
FUNÇÃO OBJETIVO BI-CRITÉRIO**

TRABALHO DE CONCLUSÃO DE CURSO

PONTA GROSSA

2017

CAROLINA MEIRA COSTA

**AVALIAÇÃO E ADAPTAÇÃO DE MÉTODOS HEURÍSTICOS PARA O
PROBLEMA DE PROGRAMAÇÃO *NO-IDLE FLOW SHOP* COM
FUNÇÃO OBJETIVO BI-CRITÉRIO**


Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do título de Bacharel, em Engenharia de Produção, do Departamento Acadêmico de Engenharia de Produção, da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Dr. Fábio José Ceron Branco

PONTA GROSSA

2017

Dedico este trabalho ao meu amado pai,
aonde quer que esteja, e a minha família,
pelos momentos de preocupação e
ausência.

	<p style="text-align: center;">Ministério da Educação</p> <p style="text-align: center;">UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ</p> <p style="text-align: center;">CÂMPUS PONTA GROSSA</p> <p style="text-align: center;"><i>Departamento Acadêmico de Engenharia de Produção</i></p>	
---	--	---

TERMO DE APROVAÇÃO DE TCC

Avaliação e adaptação de métodos heurísticos para o problema de programação *no-idle flow shop* com função objetivo bi-critério

por

Carolina Meira Costa

Este Trabalho de Conclusão de Curso (TCC) foi apresentado em 29 de junho de 2017 como requisito parcial para a obtenção do título de Bacharel em Engenharia de Produção. O candidato foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Prof. Fábio José Ceron Branco

Prof. Orientador

Prof. Juan Carlos Claros Garcia

Membro titular

Prof. Everton Luiz de Melo

Membro titular

“A Folha de Aprovação assinada encontra-se na Coordenação do Curso”.

RESUMO

COSTA, M. C. **Avaliação e adaptação de métodos heurísticos para o problema de programação *no-idle flow shop* com função objetivo bi-critério.** 2017. 55. Trabalho de Conclusão de Curso (Bacharelado em Engenharia de Produção) - Universidade Tecnológica Federal do Paraná. Ponta Grossa, 2017.

A exigência do uso eficiente dos recursos de um sistema produtivo torna as atividades do Planejamento e Controle da produção fatores de desempenho competitivos dentro das organizações. Este trabalho trata da programação da produção, conhecida como *scheduling*, como ferramenta útil na tomada de decisão das empresas, na gestão de operações. Serão utilizados métodos de ordenação inicial de destaque na literatura para o problema de programação *flow shop* permutacional, e outros métodos adaptados, avaliados através de ferramentas estatísticas. A pesquisa trata do problema *flow shop*, com restrição *no-idle* e função objetivo bi critério composta pela ponderação entre o tempo total da programação – *makespan* – e o tempo médio de fluxo – *mean flowtime*. Como resultado da experimentação computacional desenvolvida, obtiveram-se os melhores desempenhos nas soluções do problema testado através das heurísticas SPT e Triangular.

Palavras-chave: Programação da produção. *Scheduling*. *Flow shop* permutacional. *No-idle*. *Makespan*. *Mean Flowtime*.

ABSTRACT

COSTA, M. C. Evaluation and adaptation of heuristics methods for the scheduling problem of no-idle flow shop with bi-criterion objective function. 2017. 55. Trabalho de Conclusão de Curso (Bacharelado em Engenharia de Produção) - Universidade Tecnológica Federal do Paraná. Ponta Grossa, 2017.

A requirement of the efficient use of resources in a production system makes the Production Planning and Control an important and competitive factor between organizations. This paper deals with production scheduling as a useful tool in the decision making of companies in operations management. Will be used initial ordering methods highlighted in the literature for the problem of flow shop sequencing, and other adapted methods, attributes through statistical tools. The research deals with the flow shop problem, with no-idle constraint and objective function bi criterion composed by the weighting between the total scheduling time - makespan - and the mean flow time - mean flowtime. As a result of the computational experimentation developed, we obtained the best performances in the solutions of the problem tested through the SPT and Triangular heuristics.

Keywords: *Production scheduling. Flow shop sequencing. No-idle. Makespan. Mean Flowtime.*

LISTA DE FIGURAS

Figura 1 – Processo de produção	11
Figura 2 – Horizontes de planejamento.....	15
Figura 3 – Relação entre os ambientes produtivos	21
Figura 4 – Gráfico de Gantt para <i>Flow shop</i>	23
Figura 5 – Programação de tarefas <i>Flow shop</i>	24
Figura 6 – Gráfico de Gantt para problema <i>No-wait Flow Shop</i>	26
Figura 7 – Gráfico de Gantt para problema <i>No-idle Flow Shop</i>	28
Figura 8 – Gráfico de Gantt para heurística SPT	29
Figura 9 – Gráfico de Gantt para heurística SPT com restrição <i>no-idle</i>	29
Figura 10 – Gráfico de Gantt para heurística LPT	30
Figura 11 – Gráfico de Gantt para heurística LPT com restrição <i>no-idle</i>	30
Figura 12 – Princípio de construção Heurística Triangular.....	32
Figura 13 – Gráfico de Gantt para heurística Triangular	32
Figura 14 – Gráfico de Gantt para heurística Triangular com restrição <i>no-idle</i>	32
Figura 15 – Princípio de construção heurística Triangular Invertida.....	33
Figura 16 – Gráfico de Gantt para heurística Triangular Invertida	34
Figura 17 – Gráfico de Gantt para heurística Triangular Invertida com restrição <i>no-idle</i>	34
Figura 18 – Gráficos de comparação entre as heurísticas para PS e DRM com o parâmetro $\alpha = 0,0$ para número de tarefas e máquinas	39
Figura 19 – Gráficos de comparação entre as heurísticas para PS e DRM com o parâmetro $\alpha = 0,2$ para número de tarefas e máquinas.....	40
Figura 20 – Gráficos de comparação entre as heurísticas para PS e DRM com o parâmetro $\alpha = 0,4$ para número de tarefas e máquinas.....	40
Figura 21 – Gráficos de comparação entre as heurísticas para PS e DRM com o parâmetro $\alpha = 0,6$ para número de tarefas e máquinas.....	41
Figura 22 – Gráficos de comparação entre as heurísticas para PS e DRM com o parâmetro $\alpha = 0,8$ para número de tarefas e máquinas.....	42
Figura 23 – Gráficos de comparação entre as heurísticas para PS e DRM com o parâmetro $\alpha = 1,0$ para número de tarefas e máquinas.....	43
Figura 24 – Gráficos de comparação entre as heurísticas para TMC com o parâmetro $\alpha = 0,6$ para número de tarefas e máquinas	43
Figura 25 – Gráfico de superfície da PS (%) da heurística SPT para as variações do parâmetro α e do número de tarefas (n)	44
Figura 26 – Gráfico de superfície do DRM (%) da heurística SPT para as variações do parâmetro α e do número de tarefas (n)	44
Figura 27 – Gráfico de superfície da PS (%) da heurística Triangular para as variações do parâmetro α e do número de tarefas (n)	45
Figura 28 – Gráfico de superfície do DRM (%) da heurística Triangular para as variações do parâmetro α e do número de tarefas (n).....	45

LISTA DE TABELAS

Tabela 1 – Tempos de processamento de 5 tarefas (J) em 4 máquinas.....	29
Tabela 2 – Valores em Unidade de Tempo (u.t.) para <i>makespan</i> e <i>mean flowtime</i> - STP	30
Tabela 3 – Valores em Unidade de Tempo (u.t.) para <i>makespan</i> e <i>mean flowtime</i> - LPT	31
Tabela 4 – Valores em Unidade de Tempo (u.t.) para <i>makespan</i> e <i>mean flowtime</i> da heurística Triangular.....	33

LISTA DE SIGLAS

JIT	<i>Just in Time</i>
NIFS	<i>No-idle Flow Shop</i>
NWFS	<i>No-wait Flow Shop</i>
PCP	Planejamento e Controle da Produção
PO	Pesquisa Operacional
STP	Sistema Toyota de Produção
FO	Função Objetivo

SUMÁRIO

1 INTRODUÇÃO	8
1.1 OBJETIVOS	9
1.1.1 Objetivo Geral	9
1.1.2 Objetivos Específicos.....	9
1.2 JUSTIFICATIVA	10
2 REFERENCIAL TEÓRICO	13
2.1 SISTEMAS DE PRODUÇÃO	13
2.1.2 Flexibilidade como Critério Competitivo	14
2.2 PLANEJAMENTO E CONTROLE DA PRODUÇÃO	15
2.2.1 Produção <i>Just in Time</i>	18
2.3 PESQUISA OPERACIONAL	19
2.4 PROGRAMAÇÃO DA PRODUÇÃO	20
2.4.1 Sequenciamento e Programação	23
2.5 <i>FLOW SHOP</i> PERMUTACIONAL.....	25
2.5.1 <i>No-wait Flow Shop</i> (NWFS).....	27
2.5.2 <i>No-idle Flow Shop</i> (NIFS)	28
2.6 HEURÍSTICAS	29
2.6.1 Heurística <i>Shortest Processing Time</i> (SPT)	30
2.6.2 Heurística <i>Longest Processing Time</i> (LPT)	32
2.6.3 Heurística Triangular.....	33
2.6.4 Heurística Triangular Invertida.....	35
2.6.5 Heurística Randômica.....	37
3 METODOLOGIA	39
3.1 DESENVOLVIMENTO	40
3.2 ANÁLISE DOS RESULTADOS	41
4 CONCLUSÃO	48
REFERÊNCIAS	52
APÊNDICE A	56
APÊNDICE B	58

1 INTRODUÇÃO

Um sistema produtivo pode ser definido, segundo Branco (2011, p.1), como um processo, ou um conjunto de processos, pelo qual os produtos e serviços são gerados. O mesmo tem, portanto, como principal finalidade, atender de forma satisfatória determinada demanda proveniente do mercado consumidor.

Desta forma, cada vez mais, os sistemas produtivos precisam ser capazes de proporcionar aos seus clientes, produtos ou serviços que atendam os requisitos de elevada qualidade, variedade, baixo custo e rapidez de entrega.

Deste modo, o mercado consumidor competitivo apresenta um cenário em que, as prioridades competitivas são dadas pelas decisões tomadas pela empresa em determinadas situações de mercado, a fim de atender as necessidades do mesmo.

A flexibilidade do sistema produtivo demandada, bem como o uso dos métodos de produção *Just-in-Time* (JIT) (TAIICHI OHNO, 1950), exigem das organizações o uso eficiente de todos os seus recursos produtivos. Neste contexto, independente do tipo de sistema de produção, é necessário tomar uma série de decisões que dependem diretamente do horizonte de tempo considerado (FUCHIGAMI, 2010, p.27).

Para tanto, o presente trabalho expõe a programação da produção como ferramenta útil no processo de tomada de decisão, uma vez que busca resolver problemas reais e complexos de uma forma não intuitiva, e identifica uma melhor alternativa através de métodos heurísticos, com alta qualidade de solução e eficiência computacional.

A atividade de programar tarefas, conhecida na literatura como *scheduling*, envolve a alocação (associação das tarefas nas máquinas), o sequenciamento (ordem das tarefas em cada máquina) e a posterior programação de tais tarefas (obtenção dos tempos de início e término de cada operação) (FUCHIGAMI, 2010, p.28). Desta forma, a atual pesquisa trata o *scheduling* na área da Pesquisa Operacional como um método de tomada de decisão dos problemas operacionais de qualquer sistema produtivo.

Apresenta-se, portanto, o *scheduling* em um sistema de produção *flow shop* permutacional, em que o fluxo de cada tarefa de uma determinada linha deve passar obrigatoriamente por todos os estágios, sem saltos ou inversões. O problema tradicional de programação possui a finalidade de obter um sequenciamento de tarefas que otimize uma determinada medida de desempenho. Dessa forma, um conjunto de n tarefas deve ser processado por um conjunto de m máquinas.

É válido ressaltar que o número de programações possíveis cresce à medida que o número de tarefas e operações aumenta. Para um conjunto de n tarefas a serem processadas em m máquinas existem $n!$ (n fatorial) sequências possíveis na programação (FUCHIGAMI, 2010, p.31). Este tipo de problema é classificado na literatura como um problema *NP-hard*, evidenciando a complexidade do trabalho desenvolvido na presente pesquisa.

1.1 OBJETIVOS

São apresentados, nas próximas duas subseções, os objetivos do presente trabalho, separados em objetivo geral e objetivos específicos.

1.1.1 Objetivo Geral

O presente trabalho pretende avaliar alguns métodos de ordenação inicial, já conhecidos na literatura, para solução do problema de programação de operações *flow shop*, com restrição *no-idle*, bi-critério entre *makespan* e *mean flowtime* e, comparar os mesmos mediante experimentação computacional e ferramentas estatísticas.

1.1.2 Objetivos Específicos

São apresentados os objetivos específicos a serem desenvolvidos no presente trabalho, com a finalidade de atingir o objetivo geral do mesmo:

- i) Efetuar a revisão bibliográfica dos conceitos de destaque na literatura para solução problema de programação de operações *flow shop*;
- ii) Implementar computacionalmente métodos de ordenação inicial para solução problema de programação de operações *flow shop* com a restrição *no-idle* e,

iii) Adaptar os métodos utilizados para outros problemas a fim de verificar a eficiência computacional e a qualidade de solução para o problema específico apresentado no presente trabalho.

1.2 JUSTIFICATIVA

Em qualquer sistema produtivo atual, o foco principal está envolvido no atendimento das necessidades do cliente. Este fato exige das empresas uma gestão voltada à redução dos custos e a melhora da qualidade de forma geral. Para tanto, é necessário que hajam as atividades de Planejamento e Controle da Produção (PCP) eficientes, as quais demandam constantes tomadas de decisões, que se adotadas de forma intuitiva e equivocada podem acarretar em desperdícios que representam em custos elevados de produção.

Considerando o ambiente extremamente instável e exigente de flexibilidade proveniente do mercado consumidor, a programação da produção, no que se refere ao melhor sequenciamento das tarefas de um sistema produtivo, pode oferecer bons resultados no PCP. Atualmente, um número cada vez maior de tarefas (n) deve ser processado por uma quantidade de máquinas (m), em que a variação destes parâmetros interfere diretamente na complexidade da solução.

Este contexto torna propícia a utilização de *scheduling*, para as tomadas de decisão estratégicas do nível operacional, uma vez que, oferece soluções não intuitivas e de alta qualidade para os problemas do PCP (FUCHIGAMI, 2010, p.32). Justifica-se assim a avaliação de métodos para solução problema de programação de operações *flow shop*, desenvolvida neste trabalho.

Ainda é importante ressaltar que o problema em questão é classificado como *Np-hard*, em que o número de programações possíveis cresce fatorialmente à medida que o número de tarefas e operações aumenta. Tal característica torna o problema passível de solução ótima (solução exata) somente em casos de pequeno porte, evidenciando novamente a complexidade do trabalho e justificando o desenvolvimento da pesquisa através da avaliação dos métodos heurísticos e meta-heurísticos para soluções de alta qualidade. Nesta pesquisa apenas métodos heurísticos de ordenação inicial são estudados.

Na capítulo 2 e subseções, de referencial teórico, serão expostos os conteúdos necessários para o atingimento dos objetivos apresentados para o presente trabalho, bem como das heurísticas implementadas na experimentação computacional. Na Seção 3, é caracterizada e descrita a metodologia adotada no desenvolvimento do trabalho, e da análise dos resultados obtidos. Por fim, na Seção 4, expõe-se as conclusões provenientes dos resultados alcançados.

2 REFERENCIAL TEÓRICO

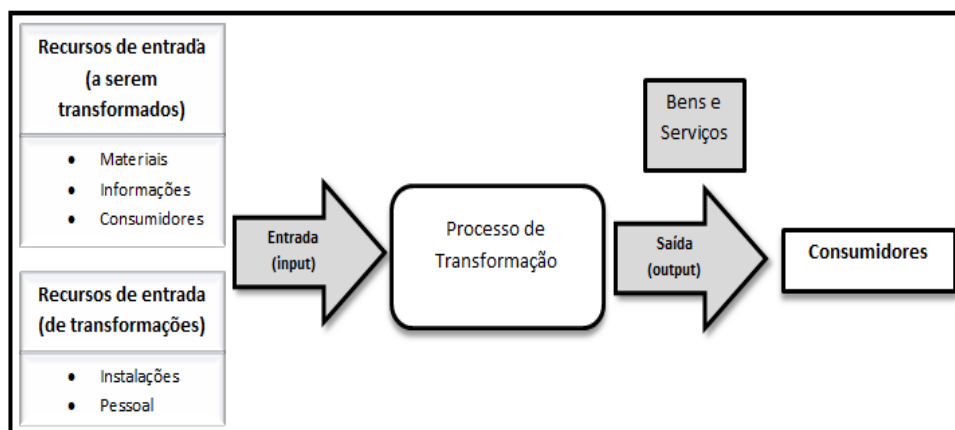
Neste capítulo são abordados os principais trabalhos da literatura para instrumento de referencial teórico desta proposta de pesquisa. O mesmo apresenta-se subdivido em seis partes: sistemas de produção, planejamento e controle da produção, pesquisa operacional, programação da produção, *flow shop* permutacional e heurísticas.

2.1 SISTEMAS DE PRODUÇÃO

As organizações existem, necessariamente, com a função de produzir algum tipo de bem (produto) ou serviço a seus clientes. A natureza da produção acontece através de um processo de transformação de entradas em saídas. As entradas (*input*) podem ser traduzidas nos recursos necessários a serem transformados, ou nos recursos de transformação. As saídas (*output*) são os bens ou serviços resultantes do processo de transformação. (SLACK et al, 2009)

A Figura 1 esquematiza o processo produtivo da transformação de recursos de entradas (*input*) em saídas (*output*).

Figura 1 – Processo de produção



Fonte: adaptado de Slack et al. (2009)

O processo básico de produção descrito na Figura 1 é comum na maioria das organizações, independente do tipo de bem ou produto a que ela se destine a produzir. A diferenciação pode surgir somente, do modo em que as organizações definem a sua estratégia e a partir dela, administram a sua produção no controle de suas operações.

Neste contexto, a forma como a unidade de negócios age em resposta às pressões do mercado e orienta sua gestão de operações, é um dos fatores que definem a sua competência distintiva. Tal capacidade pode ser definida como o diferencial decisivo para a competitividade no mercado consumidor (PAIVA et al., 2009).

Para tanto, é necessário primeiramente que se estabeleçam os critérios competitivos da produção. Tal entendimento pode partir da observação do desenvolvimento da manufatura, sendo possível perceber que, conforme o critério competitivo da produção modifica de acordo com o mercado consumidor, muda-se a base de organização da produção.

O início da produção de bens manufaturados surgiu com o artesanato, passou pela produção em massa e a administração científica, que originaram o Fordismo de Henry Ford (início por volta de 1903) em que a palavra de ordem era a padronização. Finalmente chegou-se ao Sistema Toyota de Produção e a produção enxuta, que introduziu a quebra do *trade-off* entre custo e qualidade, e inseriu um melhor sistema de controle e planejamento da produção.

No contexto atual, segundo Paiva (2009, p.40-41), as indústrias enfrentam mudanças rápidas em seus ambientes competitivos. A consequência disto é o elevado fator de **imprevisibilidade da demanda**. Cada vez mais, os sistemas produtivos precisam ser capazes de proporcionar aos seus clientes, uma variedade de bens ou serviços de forma rápida e prática.

Desta forma, surge a necessidade das organizações em se tornarem ágeis nas respostas dadas às mudanças de mercado, serem capazes de gerar os bens e serviços de acordo com os requisitos dos seus consumidores e ainda assim, se manterem competitivas.

2.1.2 Flexibilidade como Critério Competitivo

Paiva et al. (2009, p.73) conceitua a flexibilidade na abordagem estratégica como “a capacidade que um determinado sistema produtivo tem de responder a variáveis externas e internas”. Considera-se que na abordagem operacional, o conceito de flexibilidade apresentado em nada serve se não levar em conta a que custo, e em que tempo o sistema responde a essas variáveis (PAIVA et al., 2009).

A flexibilidade como critério competitivo é válida quando se considera as novas necessidades dos consumidores geradas pelo dinamismo do mercado e se pretende atendê-las de forma eficiente, de acordo com o conjunto de restrições do sistema produtivo considerado.

Neste contexto, a flexibilidade está alinhada com a intenção estratégica da organização como critério competitivo, e pode servir como um indicador da avaliação do desempenho operacional da mesma. Para que sirva como uma medida de desempenho adequada, a flexibilidade deve ser derivada da estratégia, estando de acordo com as prioridades competitivas da operação, e definida de forma clara com foco no melhoramento (CORRÊA E CORRÊA, 2013, p. 113).

Segundo Slack et al. (2009) a flexibilidade deve ser capaz de realizar alterações nas operações de alguma forma. As mudanças alcançadas devem estar relacionadas com o critério competitivo em que se busque otimizar, e devem conforme o autor, atender a quatro tipos de exigências (SLACK et al, 2009) :

- **Flexibilidade de produto/serviço** – a habilidade da operação em introduzir ou modificar produtos e serviços;
- **Flexibilidade de *mix*** – a habilidade da operação em produzir uma ampla variedade ou composto de produtos e serviços;
- **Flexibilidade de volume** – a habilidade da operação em alterar seus níveis de saída (*outputs*) para produzir diferentes quantidades ou volumes de produtos e serviços ao longo do tempo;
- **Flexibilidade de entrega** – a habilidade da operação em alterar os tempos de entrega de seus serviços ou produtos.

2.2 PLANEJAMENTO E CONTROLE DA PRODUÇÃO

O mercado consumidor surge com uma demanda específica qualquer e o sistema produtivo relacionado a ela deve ser capaz de produzir e entregar os seus produtos e serviços de acordo com o prazo, a quantidade e a qualidade requeridos. Para tanto, é preciso garantir as melhores condições para que os processos produtivos ocorram de forma eficaz e eficiente e, sejam capazes de garantir o atendimento a demanda do consumidor.

Segundo Slack et al. (2009, p.283), “o planejamento e controle diz respeito à conciliação entre o que o mercado requer e o que as operações podem fornecer”, ou seja, é do PCP que surge a interação entre os processos produtivos de bens e serviços e o gerenciamento dos recursos escassos (materiais, mão de obra, informações e consumidores) necessários para atender os aspectos da oferta e demanda (Slack et al., 2009, p.283). Corrêa e Corrêa (2013, p. 373) ressaltam o fato de que qualquer tipo de organização possui dentro de si uma **função de operações** que gere algum **valor agregado** a seus clientes. Sendo assim, qualquer tipo de organização tem a necessidade de planejar e controlar as suas operações.

Os conceitos de planejamento e controle, mesmo que distintos, podem ser considerados complementares. Slack et al. (2009, p.283) conceitua o **planejamento** como “uma formalização (plano) do que se pretende que aconteça em determinado momento no futuro”. A atividade de planejar porém, não garante que os eventos previstos realmente ocorram, já que diversos fatores externos e internos podem contribuir para situações não esperadas (Slack et al., 2009, p.283). Deste modo, é necessário que haja a atividade de **controle**, definida por Slack et al. (2009, p.283) como o processo de lidar com as variações. Ou seja, é o controle das operações que realiza os ajustes e intervenções necessários para que os propósitos do planejamento sejam atingidos. A atividade de controle pode significar a revisão e alteração dos planos em curto prazo, e depende da capacidade de ação e resposta rápida do sistema produtivo.

Fuchigami (2005, p. 17) constitui um sistema produtivo de três subsistemas interativos, sendo eles (adaptado de FUCHIGAMI, 2005, p.17):

- i) **Estrutural:** constituído da parte física do sistema, como os recursos materiais, instalações, etc.;
- ii) **Social:** formado pelas relações sociais dentro da organização;
- iii) **Organizacional:** composto pela hierarquia de decisão, departamentos, etc.

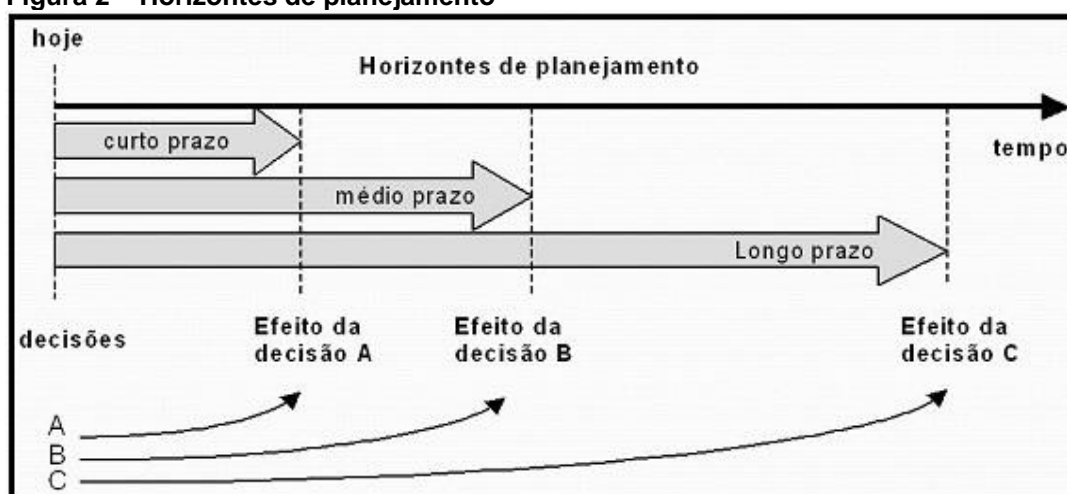
Ainda segundo Fuchigami (2005, p. 17), o PCP está contido no subsistema organizacional, e é responsável por comandar e gerenciar o processo produtivo. Em outra perspectiva, é possível dizer que o PCP atua nos três níveis hierárquicos de um sistema produtivo. No nível estratégico, o PCP é responsável por determinar o

plano de produção, no nível tático define-se o plano-mestre da produção e no nível operacional determina a **programação da produção** e faz o acompanhamento e o controle da mesma.

Neste sentido, o processo de planejamento, para que seja bem realizado, deve estabelecer o horizonte de planejamento coerente com as decisões envolvidas no mesmo (CORRÊA E CORRÊA, 2013, p. 373). Para cada nível hierárquico de planejamento apresentado envolvem-se determinados tipos de decisões a serem tomadas e, portanto, cada um possui um horizonte de planejamento adequado. O **horizonte de planejamento** é definido por Corrêa e Corrêa (2013, p. 372) pelo “período de tempo futuro sobre o qual se tenha interesse em desenvolver uma visão”.

A Figura 2 apresenta o esquema do horizonte de planejamento para o PCP a longo, médio e curto prazo.

Figura 2 – Horizontes de planejamento



Fonte: Corrêa e Corrêa (2013).

Desta forma, o PCP, especificamente na **programação da produção** no nível operacional é o foco em que o presente trabalho pretende desenvolver seus estudos. Conforme Rangel e Fuchigami (2014, p.160), a programação da produção, conhecida na literatura como *scheduling*, tem levado os pesquisadores a cada vez mais desenvolverem pesquisas nesta área, nas últimas décadas, por representar uma das atividades mais importantes de uma organização no nível operacional. Segundo os autores, *scheduling* está relacionada ao bom funcionamento do sistema produtivo através da otimização de diversos critérios de desempenho, baseada na eficiente utilização dos recursos, na entrega dos produtos e serviços nos prazos

estipulados e na redução dos custos de produção (RANGEL E FUCHIGAMI, 2014, p.160).

Os sistemas produtivos atuais apresentam-se cada vez mais interessados na introdução de operações que estejam incluídas de alguma forma na produção JIT. Este contexto se deve as mudanças das características exigidas das organizações provenientes do mercado consumidor. Demandas de novos produtos com a necessidade de entrega rápida e pontual surgem a todo o momento. Para tanto, é possível que se relacione *scheduling* com os critérios de desempenho exigidos na produção JIT a fim de torná-los critérios competitivos.

2.2.1 Produção *Just in Time*

As mudanças ocorridas na organização da produção industrial, e decorrentes da evolução da gestão de operações acarretaram em um panorama industrial altamente competitivo e dinâmico. Existe a necessidade constante do planejamento e controle dos critérios competitivos de cada organização, e portanto, a busca incessante **pela excelência das operações**.

Neste sentido, surgiu no Japão, em meados da década de 1970, a filosofia JIT desenvolvida pela Toyota Motor Company na busca de um sistema de administração da produção que fosse capaz de coordenar a produção com a demanda específica de diferentes tipos de veículos, visando assim, o **mínimo de atraso** possível na produção (CORRÊA E GIANESI, 2011, p.56).

O objetivo fundamental da filosofia JIT é a melhoria contínua do processo produtivo. A introdução do sistema “puxado” a partir da demanda, que pretende produzir somente os itens necessários, nas quantidades necessárias e no momento necessário a cada estágio (CORRÊA E GIANESI, 2011, p.57). Segundo et al. (2009, p.36), o Sistema Toyota de Produção (STP), em que o sistema JIT encontra-se inserido, condiciona as organizações á competirem globalmente, dentre outros fatores, pela busca de um melhor sistema de controle e planejamento da produção.

O *scheduling* pode ser associado ao JIT quando pretende obter uma solução que minimize critérios de desempenho associados aos adiantamentos e atrasos das tarefas. Em um sistema produtivo o adiantamento do término de uma tarefa pode

significar a criação de estoques no processo, bem como o atraso do término de uma tarefa pode estar associado a algum tipo de multa (RANGEL E FUCHIGAMI, 2015, p. 99). Ambos representam **custos desnecessários** de produção, e justificam o interesse cada vez maior das empresas na escolha de tais critérios de desempenho.

Entre os fatores que sustentam o STP é válido ressaltar para *scheduling* os seguintes (PAIVA, et al; 2009, p. 37):

- **Fluxo de produção:** deve buscar um ritmo uniforme de produção. A redução do tempo de preparação de cada máquina agiliza a mesma.
- **Flexibilidade:** capacidade de trocar o ferramental no mínimo de tempo possível (tempo de *setup*) e passar a produção de um produto para outro. Existem operações externas, que são as que podem ser realizadas com a máquina em operação, e internas que são executadas com a máquina parada.

2.3 PESQUISA OPERACIONAL

O desenvolvimento industrial acarretou em um contexto de alta complexidade organizacional, em que grandes inovações na administração da produção valorizaram o conhecimento organizacional integrado com os processos desenvolvidos entre todas as áreas funcionais da empresa.

Na contramão das vantagens obtidas por tal desenvolvimento, está a consideração de Hillier e Lieberman (2006, p. 1) de que “à medida que aumentam a complexidade e a especialização em uma organização, torna-se cada vez mais difícil alocar os recursos disponíveis para as diversas atividades de maneira mais eficiente para a organização como um todo”. Segundo Paiva (2009, p.41) “esta é a arma competitiva baseada na capacidade de manufaturar e montar os produtos certos, no local certo, no tempo certo e ao preço correto”.

Considerando o contexto descrito anteriormente, na necessidade da busca de soluções para problemas de condução e coordenação das operações de uma organização, surge a **Pesquisa Operacional** (PO) como a tentativa de uma abordagem científica na gestão das organizações. Tendo a sua origem atribuída ao período da Segunda Guerra Mundial (HILLIER E LIEBERMAN, 2006, p.1), a PO

crece “em razão do empreendimento de guerra, em que havia uma necessidade premente de se alocar de forma eficiente os escassos recursos para as diversas operações” (HILLIER E LIEBERMAN, 2006, p.1).

A natureza da PO está diretamente envolvida com a coordenação das operações e trata efetivamente da gestão prática de qualquer tipo de organização. Para ser bem-sucedida a PO precisa, de forma abrangente, tentar solucionar os conflitos de interesse entre as unidades de uma organização e oferecer a melhor solução para a empresa como um todo, apresentando soluções positivas e acessíveis para o processo de tomada de decisão (HILLIER E LIEBERMAN, 2006, p.3).

No planejamento e controle de curto prazo surgem, em quantidades diárias, necessidades da tomada de decisão sobre atividades estratégicas das operações. Corrêa e Corrêa (2013, p. 29) ressaltam a importância em perceber que “as numerosas decisões operacionais, que se tomam diariamente, somadas se tornam tão ou mais relevantes que as grandes decisões de longo prazo”. Segundo Paiva et al. (2009, p. 138), o processo de criação da estratégia é um processo contínuo ao longo do tempo e, deve ser capaz de criar as competências da empresa, que para ser bem conduzido, tal processo deve considerar as tomadas de decisões e orientá-las da melhor maneira possível.

Desta forma, a PO deve ser aplicada como ferramenta útil na busca pelo melhor caminho a ser seguido nos problemas organizacionais (HILLIER E LIEBERMAN, 2006, p.3). Segundo Fuchigami (2010, p. 27), a PO busca colaborar com a tomada de decisão intuitiva para problemas complexos e oferece a melhor alternativa através de métodos de solução inseridos em uma abordagem científica, que considera algum critério de análise.

2.4 PROGRAMAÇÃO DA PRODUÇÃO

O entendimento de *scheduling* pode partir mais facilmente da comparação com as atividades diárias do cotidiano das pessoas, em que é possível perceber a constante necessidade da programação de suas tarefas. Para o embarque em um meio de transporte qualquer, por exemplo, deseja-se primeiramente saber quando o

evento de partida do mesmo ocorrerá para então ser possível se chegar ao terminal para o embarque (BAKER; TRIETSCH, 2009, p.17).

Normalmente, a programação das tarefas é obtida em questão de tempo (horário de partida do ônibus às 08h). Contudo, uma resposta igualmente útil pode ser apresentada em termos do sequenciamento das tarefas ao invés de tempo, isto é, uma tarefa só ocorrerá após (ou logo após) a finalização da tarefa anterior a ela (BAKER; TRIETSCH, 2009, p.18). Em um exemplo prático, o ônibus partirá logo após a finalização das tarefas de limpeza e manutenção do mesmo.

Tais problemas de programação diários envolvem tarefas específicas a serem realizadas, e necessidades de recursos específicos a serem utilizados. Este contexto pode ser comparado ao ambiente de um sistema produtivo, quando em uma linha de produção existe um conjunto de n tarefas a serem processadas em m máquinas (recursos) disponíveis.

Dentro de um sistema produtivo, a quantidade de tarefas a serem executadas e a disponibilidade de recursos disponíveis pode ser suficientemente grande para que tal problema seja considerado de difícil resolução. De acordo com Lugo e Teixeira (2012, p. 12), a tarefa de *scheduling* em máquinas pode se tornar ainda mais complexa quando se considera os **fatores internos da produção**. Assim, é preciso que se considerem atrasos e interrupções das tarefas programadas decorrentes de, por exemplo, quebra e manutenção das máquinas, chegada de ordens de maior prioridade, bem como de qualquer tipo de evento inesperado no chão de fábrica (LUGO E TEIXEIRA, 2012, p.12).

Deste modo, o problema de programação das tarefas dentro de um processo produtivo constitui o *scheduling*, e pode ser definida, segundo Baker e Trietsch (2009, p. 4), como a alocação de recursos escassos para a execução de tarefas em uma base de tempo.

Em um sistema produtivo as tarefas constituem os produtos a serem gerados (tipos de peças, roupas, objetos, etc.), ou os serviços a serem efetuados (prestação de serviço à uma fila de pessoas, transporte de cargas, atendimento de pacientes em um hospital, etc.). Os recursos são basicamente definidos pelos recursos necessários para a execução das tarefas, e em uma fábrica geralmente

são representados pelas máquinas e equipamentos utilizados para a produção de qualquer produto (tarefa).

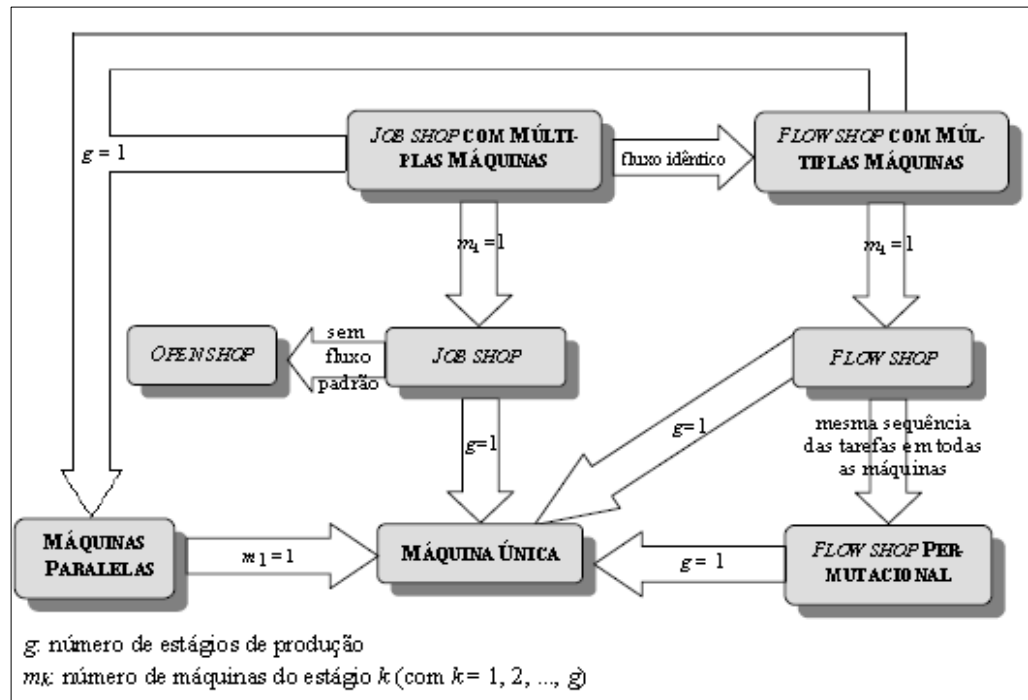
O *scheduling* envolve então, a resolução do problema que consiste em, designar as tarefas aos recursos, e obter uma sequência de processamento de tais tarefas que otimize um determinado critério. O critério de otimização utilizado depende geralmente do tipo de ambiente produtivo que esta sendo tratado. Em cada situação, dependendo dos objetivos do ambiente, se pode buscar otimizar: atraso, adiantamento, *makespan* (instante de término da última tarefa), *lateness* (afastamento da data de entrega), *flowtime* (tempo total de fluxo), entre outros.

Segundo Nagano et al. (2004, pg.1), os ambientes produtivos que, tradicionalmente, classificam os problemas de programação de tarefas, geralmente em função do fluxo das operações nas máquinas, são:

- **Job shop:** cada tarefa tem sua própria sequência de processamento de tarefas no conjunto de máquinas;
- **Flow Shop:** todas as tarefas têm a mesma sequência de processamento no conjunto de máquinas;
- **Flow shop permutacional:** ambiente *flow shop* onde, em cada máquina a sequência das tarefas é a mesma;
- **Open shop:** não há uma sequência específica ou preestabelecida para o processamento das tarefas nas máquinas;
- **Máquina única:** há somente uma única máquina disponível para o processamento das tarefas;
- **Máquinas paralelas:** existem diversas máquinas para o processamento das tarefas. Podendo ser máquinas idênticas, ou podem diferir entre si (máquinas paralelas uniformes ou não relacionadas);
- **Job Shop com Máquinas Múltiplas:** ambiente *Job shop* que possui um conjunto de máquinas paralelas em cada estágio de produção;
- **Flow Shop com Máquinas Múltiplas:** ambiente *Flow shop* com um conjunto de máquinas paralelas em cada estágio de produção.

Fuchigami (2010) esquematiza o relacionamento entre os diferentes ambientes, conforme a Figura 3.

Figura 3 - Relação entre os ambientes produtivos



Fonte: Fuchigami (2010)

2.4.1 Sequenciamento e Programação

Dentre as atividades de planejamento e controle da produção, as variáveis de volume, tempo e qualidade devem ser consideradas. Na abordagem do tempo, quando em um sistema produtivo existe a chegada de ordens de produção (tarefas a serem processadas) é necessário que se tomem decisões sobre a ordem em que determinadas tarefas serão executadas. Conforme Slack et al. (2009), á essa tomada de decisão se dá o nome de **sequenciamento**.

O sequenciamento das tarefas deve ser tomado de acordo com fatores específicos do processo produtivo em questão. As prioridades de cada operação devem ser levadas em conta para orientar, os objetivos de desempenho a serem priorizados no sequenciamento. Dentro da programação da produção, esses fatores irão determinar o critério de desempenho a ser otimizado no problema.

Segundo Slack et al. (2009), os objetivos de confiabilidade, rapidez e custo são particularmente importantes, e podem ser explicados, de acordo com o autor como:

- Confiabilidade: atender a data prometida ao consumidor;

- Rapidez: minimizar o “tempo de fluxo” que o trabalho gasta no processo;
- Custo: Minimizar o estoque de trabalho em processo e o tempo ocioso dos centros de trabalho.

Corrêa e Corrêa (2013) relaciona o **sequenciamento** com a definição de prioridades no processo de decidir que tarefa fazer primeiro em determinado centro de trabalho. Para tanto, o autor enuncia as disciplinas de sequenciamento a serem utilizadas na definição de tais prioridades (CORRÊA E CORRÊA, 2013):

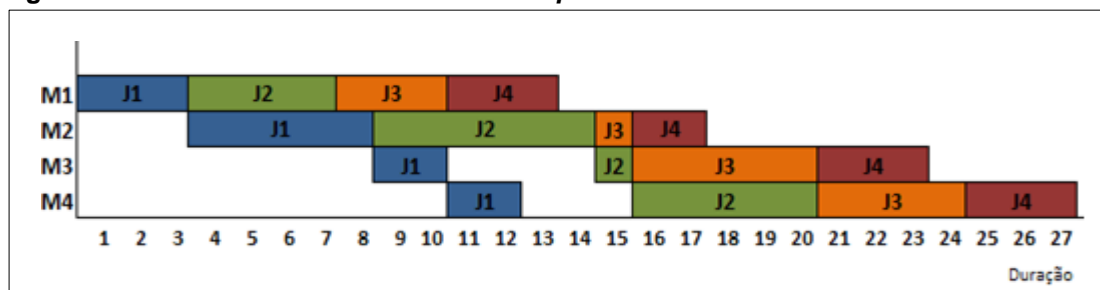
- Tempo de processamento da tarefa no centro de trabalho que está sendo sequenciado;
- Data prometida de entrega da tarefa (na ordem de produção);
- Momento de entrada da tarefa no centro de trabalho;
- Importância da tarefa solicitada;
- Tempo de operação restante – soma dos tempos de processamento das tarefas que ainda precisam ser executadas.

Sendo determinado o sequenciamento das tarefas a serem executadas, é útil que se desenvolva um cronograma detalhado em que seja possível apresentar os momentos de início e de término de cada tarefa. Slack et al. (2009) conceitua este cronograma como **programação**, e ressalta a importância de sua utilização em operações que requerem algum planejamento para certificar que a demanda do consumidor seja atendida. Para Corrêa e Corrêa (2013, p. 446), a programação das operações consiste em “alocar no tempo as atividades, obedecendo ao sequenciamento definido e ao conjunto de restrições considerado” (CORRÊA E CORRÊA, 2013, p. 446).

Para representação do programa das operações, o Gráfico de Gantt costuma ser a forma de representação mais tradicionalmente utilizada. No caso da programação de tarefas em máquinas, o Gráfico de Gantt é capaz de representar graficamente, através de barras, os tempos de início e fim de execução das tarefas, bem como indicar o tempo de processamento das operações em cada máquina. De forma prática, o Gráfico de Gantt serve de uma representação gráfica simples, do que deveria acontecer, e do que realmente está acontecendo em cada operação (SLACK et al. 2009, p. 299).

A Figura 4 apresenta a programação de um problema *flow shop* clássico para 4 tarefas e 4 máquinas.

Figura 4 – Gráfico de Gantt de um *Flow Shop* clássico



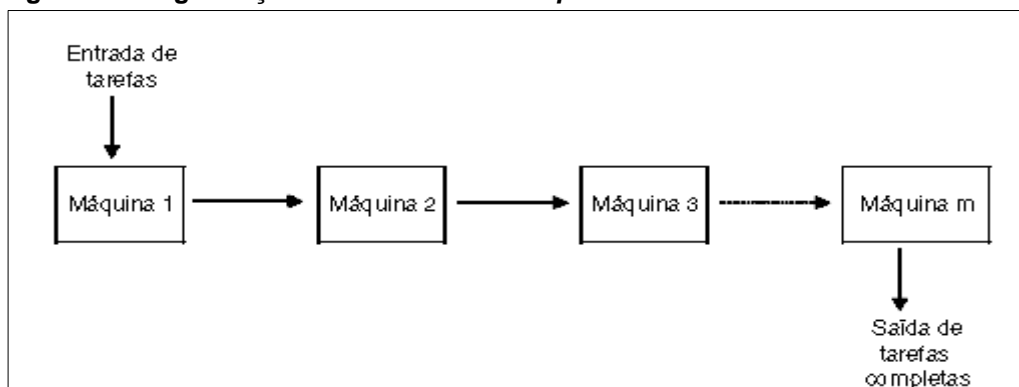
Fonte: adaptado de Branco (2011)

2.5 FLOW SHOP PERMUTACIONAL

O presente trabalho pretende tratar de *scheduling para um problema flow shop* permutacional. Segundo Branco (2011, p.19), o problema de *scheduling flow shop* clássico consiste num problema no qual um conjunto de n tarefas deve ser processado, **na mesma ordem**, por um conjunto de m máquinas (distintas entre si) contidas no sistema produtivo em questão. Quando a sequência de processamento é a mesma em todas as máquinas do conjunto de m máquinas distintas, têm-se o ambiente produtivo chamado *flow shop* permutacional.

A Figura 5 apresenta um esquema do fluxo da programação de tarefas em um problema *flow shop*.

Figura 5 - Programação de tarefas *Flow shop*



Fonte: Scardoelli (2006, p. 23)

O problema de programação *flow shop* permutacional pretende determinar uma sequência ótima de tarefas, dentre as sequências possíveis ($n!$), que busque

otimizar determinado critério de desempenho. O critério de desempenho geralmente é escolhido de acordo com as prioridades específicas de cada sistema produtivo, no caso do problema de programação *flow shop* permutacional um critério comum de otimização é a minimização do *makespan* que é o instante de término da última tarefa na última máquina (BRANCO, 2011).

Esse tipo de problema é classificado na literatura como um problema NP-*hard*, em que o número de programações possíveis cresce rapidamente à medida que o número de tarefas e operações aumenta. Isso significa que o problema de programação *flow shop* permutacional só é passível de solução ótima eficiente para problema de pequeno porte.

A busca de métodos de solução exata (solução ótima) dos problemas de programação de operações têm se desenvolvido nos últimos anos e, técnicas como Programação Linear Inteira (SELEN E HOTT, 1986) e (WILSON, 1989), e o método *Branch-and-Bound* de enumeração (IGNALL E SCHRAGE, 1965) e (POTTS, 1980) têm sido comumente utilizadas (FUCHIGAMI, 2010, p. 33). Ambas as técnicas são capazes de oferecer soluções ótimas de boa qualidade, entretanto, quando se trata de problemas de médio e grande porte, não se demonstram eficientes computacionalmente (FUCHIGAMI, 2010, p. 33).

Segundo Nagano e Moccellini (2004, p. 2), por exigir um alto esforço computacional, a busca por uma solução ótima neste tipo de problema apresenta uma importância mais teórica do que prática, e orienta as pesquisas no desenvolvimento de métodos heurísticos para a sua resolução. Os métodos heurísticos e meta-heurísticos geralmente se apresentam eficientes em problemas *flow shop* de médio e grande porte, pois são capazes de encontrarem soluções próximas do ótimo em tempos computacionais viáveis (BRANCO, 2006, p. 35), tornando possível a sua utilização no presente estudo.

Fuchigami (2010, p.33), define um **método heurístico** como “um processo de solução de problema apoiado em critérios racionais ou computacionais para escolher um caminho entre vários possíveis, sem a preocupação de percorrer todas as possibilidades ou atingir a melhor solução”. Desta forma, um método heurístico “provavelmente vai encontrar uma excelente solução viável, não necessariamente ótima, para o problema específico” (HILLIER E LIEBERMAN, 2006, p.599).

Uma das características dos métodos heurísticos é que eles tendem a ser específicos por natureza, significando que para cada problema específico um novo método heurístico precisa ser desenvolvido. Para tanto, foram criadas as **meta-heurísticas**, definidas por Hillier e Lieberman (2006, p. 599) como “um método de resolução geral, capaz de fornecer diretrizes estratégicas para desenvolver um método heurístico específico que se ajuste a um tipo de problema particular” (HILLIER E LIEBERMAN, 2006, p.599).

Nas próximas subseções são apresentadas variações do sistema clássico de produção, o primeiro com a restrição *no-wait* e, em seguida, com a restrição *no-idle*, sendo esta última o foco principal do presente estudo.

2.5.1 *No-wait Flow Shop* (NWFS)

O crescente interesse em pesquisas para os problemas *flow shop* tradicionais, nas últimas décadas, favoreceram também o desenvolvimento de pesquisas na área com foco em **restrições específicas** do problema, relacionadas à certos tipos de indústrias.

Uma dessas restrições se apresenta no problema ***no-wait flow shop***. Este tipo de problema, assim como um *flow shop* tradicional, consiste em encontrar uma programação ótima de um conjunto de n tarefas que deve ser processado por um conjunto de m máquinas distintas entre si (BRANCO, 2011, p.19). Sendo assim, o atributo que difere um problema NWFS de um problema *flow shop* tradicional é a restrição ***no-wait***, que assegura ao problema a condição de não permitir qualquer tempo de espera entre o processamento de uma tarefa de uma máquina para a outra (KALCZYNSKI e KAMBUROWSKI, 2008, p.678).

Deste modo, o processamento de uma tarefa em uma máquina, depois de iniciado, deve ser ininterrupto, da primeira máquina até a última. Ou seja, a operação $i + 1$ de uma tarefa tem que ser processada, pela próxima máquina da sequência, logo após o término da operação i , significando que $1 \leq i \leq m - 1$ (BRANCO, 2006, p. 36).

É válido ressaltar que, o único tempo de espera permitido no NWFS é o atraso, se necessário, do início do processamento da tarefa na primeira máquina. E ainda, que “o início do processamento de uma tarefa em uma máquina e o seu

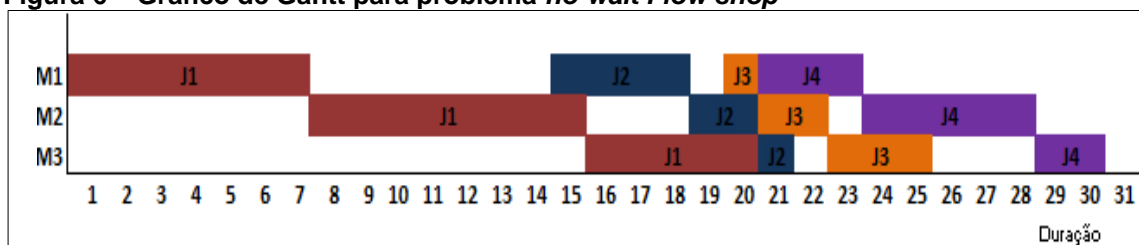
término na máquina anterior devem sempre coincidir” (SCARDOELLI *apud* ALLHVERDI ; ALDOWAISAN, 2006).

Um sistema *no-wait flow shop* pode se apresentar importante principalmente para as indústrias que possuam processos influenciados por fatores como o tempo, temperatura, alto custo de espera, etc. Indústrias químicas, farmacêuticas e de laminação de metais por possuírem características coerentes com os problemas *no-wait flow shop* podem ser exemplos disso (BRANCO; CRUZ, 2009, p. 2).

Segundo Moccellini et al. (2009, p. 26) diversos autores veem desenvolvendo heurísticas e meta-heurísticas para a resolução de problemas NWFS nos últimos anos, as quais estão se mostrando capazes de oferecer soluções próximas do sequenciamento ótimo, com tempos computacionais aceitáveis.

A Figura 6 apresenta um Gráfico de Gantt para um problema *no-wait flow shop* com a programação de um conjunto de 4 tarefas que devem ser processadas, na mesma sequência por um conjunto de 3 máquinas distintas.

Figura 6 – Gráfico de Gantt para problema *no-wait Flow shop*



Fonte: Próprio autor (2017)

2.5.2 No-idle Flow Shop (NIFS)

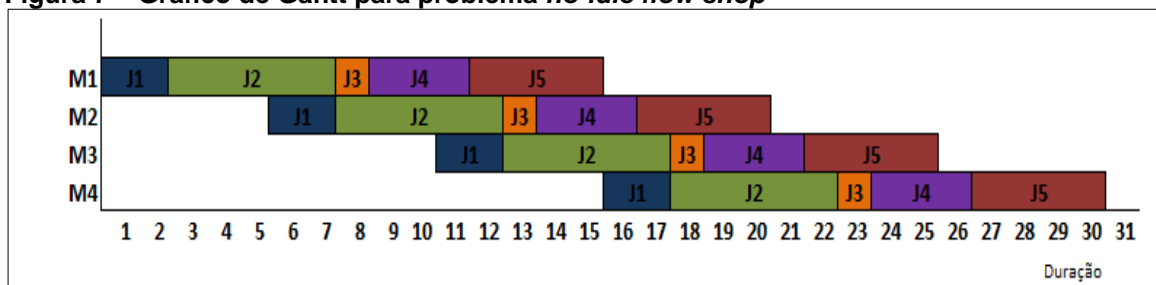
Outro caso particular do problema de programação *flow shop* é a restrição ***no-idle***, responsável por garantir que uma vez iniciada as operações em uma determinada máquina, as mesmas devem processadas sem interrupções, de modo que não haja lacunas entre o processamento de uma tarefa e outra (NAGANO; JANUÁRIO, 2013, p.271). Esta restrição é utilizada nas implementações desta pesquisa.

A restrição *no-idle*, deste modo, restringe tempos ociosos das máquinas ou tempos de *setup* desnecessários. Segundo Branco (2011, p. 23) os ambientes produtivos que decorrem de elevados tempos de *setup* e custos de utilização são os

mais compatíveis para com os problemas NIFS, uma vez em que este busca uma programação *flow shop* que optimize o aproveitamento do funcionamento das máquinas de forma eficiente.

A figura 7 apresenta o Gráfico de Gantt de um problema *no-idle flow shop* composto por cinco tarefas e quatro máquinas.

Figura 7 – Gráfico de Gantt para problema *no-idle flow shop*



Fonte: adaptado de Nagano e Januário (2013, p. 270)

É possível perceber através da Figura 7 que o instante de término do processamento de uma tarefa em uma determinada máquina coincide com o início do processamento da próxima tarefa na mesma máquina. Isto significa que as tarefas são processadas sem interrupções, o que garante que não existam *gaps* (lacunas) de tempos ociosos entre as tarefas.

Além disso, ao contrário do que acontece na restrição *no-wait flow shop*, as tarefas não necessariamente são processadas de uma máquina para outra continuamente. Pode-se perceber tal característica na Figura 7, quando o instante de término de processamento da tarefa 2 na máquina 1 (M1) não coincide com o instante de início de processamento da mesma tarefa (2) na máquina 2 (M2).

2.6 HEURÍSTICAS

Os métodos heurísticos se mostram importantes na resolução dos problemas de programação *flow shop* permutacional por encontrarem soluções práticas com resultados não necessariamente ótimos, porém, de alta qualidade. Zhou et al. (2001) descreve a solução ótima exata como não necessária para situações práticas deste tipo de problema, já que não compensam o esforço computacional e os custos incorridos na obtenção da mesma.

Para tanto, são descritas nas próximas cinco subseções as heurísticas selecionadas, usadas na experimentação computacional do presente trabalho.

2.6.1 Heurística *Shortest Processing Time* (SPT)

Uma das heurísticas mais conhecidas na literatura para a resolução do problema apresentado é conhecida como *Shortest Processing Time* (SPT), Lawrence e Sewell (1997), e Zhou et al. (2001). Este método consiste em sequenciar as tarefas através da ordem **não-decrescente** da soma dos seus tempos de processamento (tempo total de processamento para cada tarefa nas m máquinas) NAGANO (2005, p.4).

Segundo Montevechi et al. (2002, p.15), em um estudo comparativo entre heurísticas de sequenciamento, a regra SPT se apresenta mais eficiente quando o critério avaliado é a ociosidade das máquinas, com tarefas de médio tempo de processamento total. Isto significa que, no estudo de Montevechi et al. (2002), na regra STP o tempo que uma máquina permanece à espera de uma tarefa é menor.

A Tabela 1 apresenta dados de um problema *flow shop* permutacional contendo 5 tarefas e 4 máquinas, neste caso, verifica-se que o tempo total de processamento das tarefas J_i com $i = 1, \dots, 5$ são, respectivamente: 13, 20, 17, 18, e 19.

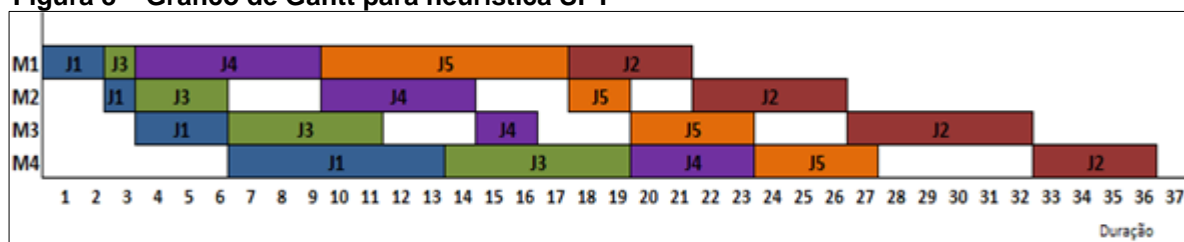
Tabela 1 – Tempos de processamento de 5 tarefas (J) em 4 máquinas

	J1	J2	J3	J4	J5
Máquina 1	2	4	1	6	8
Máquina 2	1	5	3	5	2
Máquina 3	3	6	5	2	4
Máquina 4	7	4	6	4	4

Fonte: Próprio autor (2017)

Dessa forma, quando se aplica o método SPT, o sequenciamento das tarefas deve ser definido por J1, J3, J4, J5, J2, considerando a ordem não-decrescente dos tempos totais de processamento. Desde modo, a Figura 8 apresenta o sequenciamento das tarefas com a lógica de ordenação da SPT sem restrição.

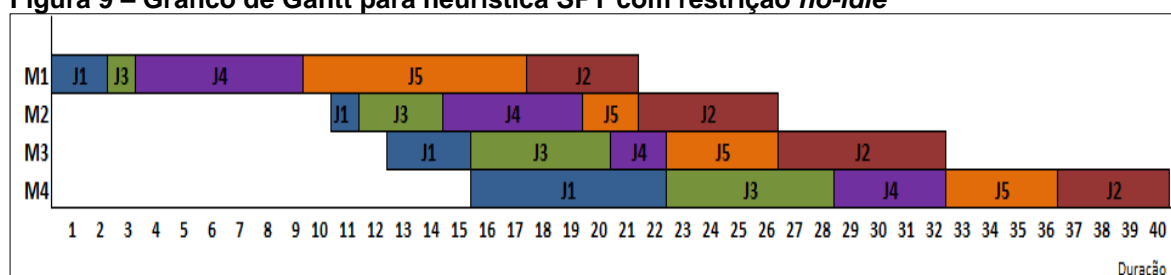
Figura 8 – Gráfico de Gantt para heurística SPT



Fonte: Próprio autor (2017)

Quando se implementa a restrição *no-idle* na heurística SPT para o mesmo caso, a ordenação das tarefas se apresenta no gráfico de Gantt da Figura 9.

Figura 9 – Gráfico de Gantt para heurística SPT com restrição *no-idle*



Fonte: Próprio autor (2017)

É possível perceber a consideração da restrição *no-idle* quando na Figura 9 não existem tempos ociosos das máquinas entre o processamento das tarefas, embora o atraso do início do processamento da tarefa J1 em 8 unidades de tempo na máquina 2, 9 unidades de tempo na máquina 3 e 9 unidades de tempo na máquina 4.

A Tabela 2 apresenta os valores de *makespan* e *mean flowtime* para a heurística SPT sem restrição (Figura 8) e com restrição *no-idle* (Figura 9).

Tabela 2 – Valores em Unidade de Tempo (u.t.) para *makespan* e *mean flowtime* - STP

	SPT	SPT com restrição <i>no-idle</i>
Makespan	36 u.t	40 u.t
Mean Flowtime	23,6 u.t	31,6 u.t

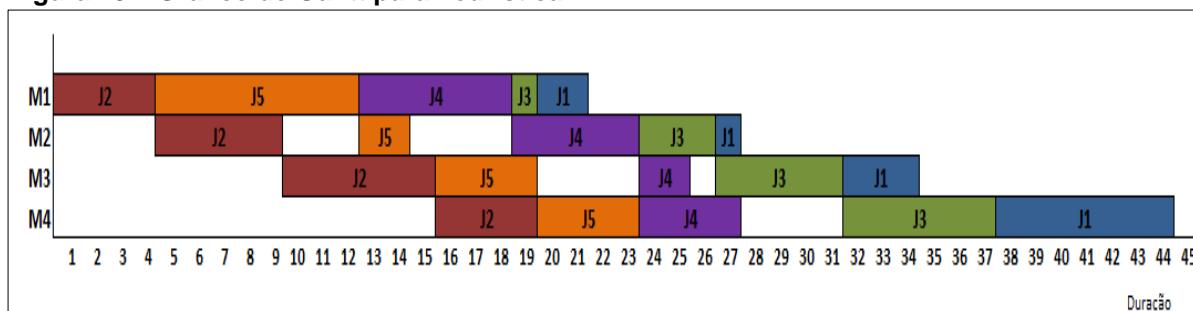
Fonte: Próprio autor (2017)

2.6.2 Heurística *Longest Processing Time* (LPT)

A heurística *Longest Processing Time* (LPT), Lawrence e Sewell (1997), ao contrário da lógica da SPT, realiza o sequenciamento das tarefas, priorizando a ordem **não-crescente** da soma dos tempos de processamento de cada tarefa, nas m máquinas.

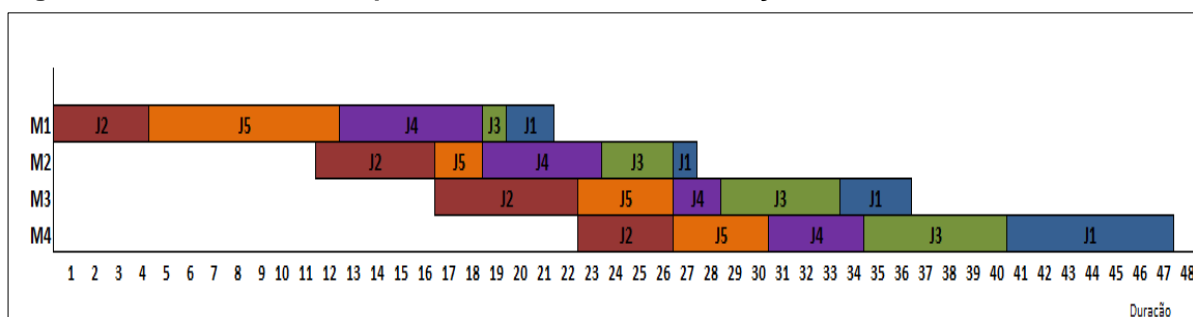
Através dos dados apresentados na Tabela 1, o sequenciamento das tarefas pelo método LPT definiu: J2, J5, J4, J3, J1. A Figura 10 apresenta o gráfico de Gantt para o método LPT sem restrição. Implementando a restrição *no-idle* para o mesmo caso, o sequenciamento se apresenta na Figura 11.

Figura 10 – Gráfico de Gantt para heurística LPT



Fonte: Próprio autor (2017)

Figura 11 – Gráfico de Gantt para heurística LPT com restrição *no-idle*



Fonte: Próprio autor (2017)

Da mesma forma, a consideração da restrição *no-idle* na Figura 11 representa a não existência de tempos ociosos entre os processamentos das tarefas nas máquinas. A Tabela 3 apresenta os valores de *makespan* e *mean flowtime* para as Figuras 10 e 11.

Tabela 3 – Valores em Unidade de Tempo (u.t.) para *makespan* e *mean flowtime* - LPT

	LPT	LPT com restrição <i>no-idle</i>
<i>Makespan</i>	44 u.t	47 u.t
<i>Mean Flowtime</i>	30 u.t	35,4 u.t

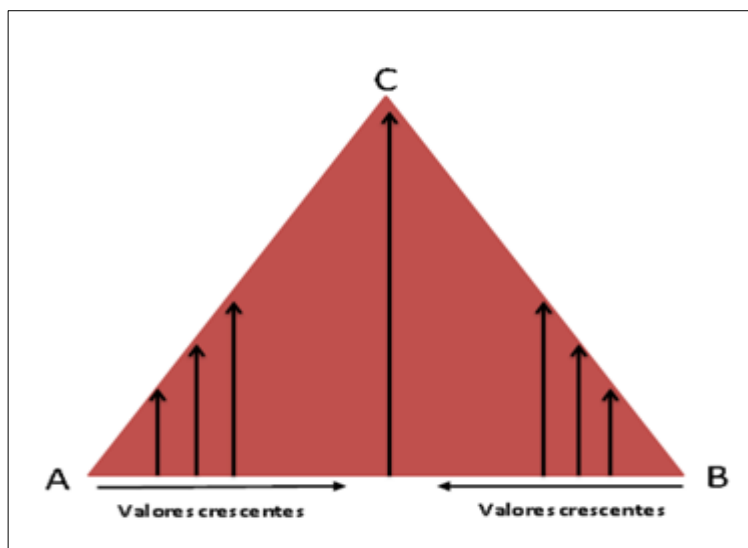
Fonte: Próprio autor (2017)

2.6.3 Heurística Triangular

A heurística triangular é chamada desta forma, pois, apresenta sua lógica de construção baseada no formato de um triângulo, em que as extremidades apresentam menores somas de tempo de processamento, e na região central as maiores.

No funcionamento da heurística, as somas dos tempos de processamento das tarefas são colocadas uma a uma nas extremidades da sequência solução, sendo alocada a tarefa com o menor valor na primeira posição da extremidade A, conforme Figura 12. Posteriormente, o segundo menor valor da soma dos tempos de processamento é alocado na primeira posição da extremidade B (Figura 12). O princípio de construção segue o mesmo critério: o terceiro menor valor na segunda posição da extremidade A, o quarto menor valor na segunda posição da extremidade B, e assim por diante. A sequência de alocação continua, ao alternar o preenchimento entre as extremidades A e B, e finaliza com a maior soma dos tempos de processamento das tarefas alocada na posição central em C (Figura 12) (BRANCO E SANTOS, 2016).

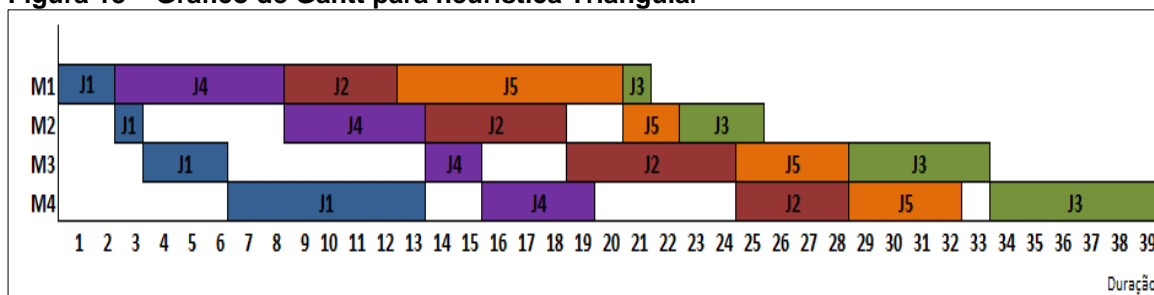
Figura 12 – Princípio de construção Heurística Triangular



Fonte: adaptado de Branco e Santos (2016)

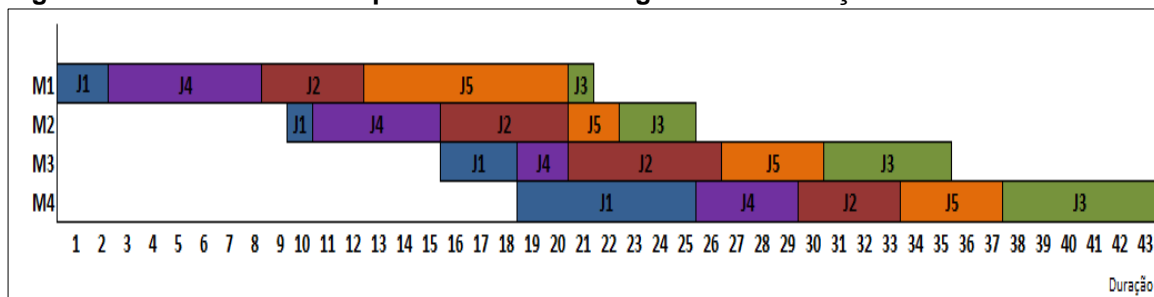
Utilizando os dados da Tabela 1, apresenta-se na Figura 13 o sequenciamento das tarefas para a Heurística Triangular sem restrição e na Figura 14 a seqüência para a mesma heurística com restrição *no-idle*.

Figura 13 – Gráfico de Gantt para heurística Triangular



Fonte: Próprio autor (2017)

Figura 14 – Gráfico de Gantt para heurística Triangular com restrição *no-idle*



Fonte: Próprio autor (2017)

A Tabela 4 apresenta os valores de *makespan* e *mean flowtime* para as Figuras 13 e 14. Novamente é possível perceber, comparando os valores da

Tabela 4, que pela restrição *no-idle* é necessário atrasar o início de processamento das tarefas a partir da máquina 2, para que não haja *gaps* entre os processamentos das tarefas em cada máquina.

Esta característica faz com que a implementação da restrição *no-idle* ocasione em um valor de *makespan* e *mean flowtime* maior, quando comparado ao sequenciamento da mesma heurística, sem restrição. Tal observação foi percebida em todos os casos apresentados até então no presente trabalho.

Tabela 4 – Valores em Unidade de Tempo (u.t.) para *makespan* e *mean flowtime* da heurística Triangular

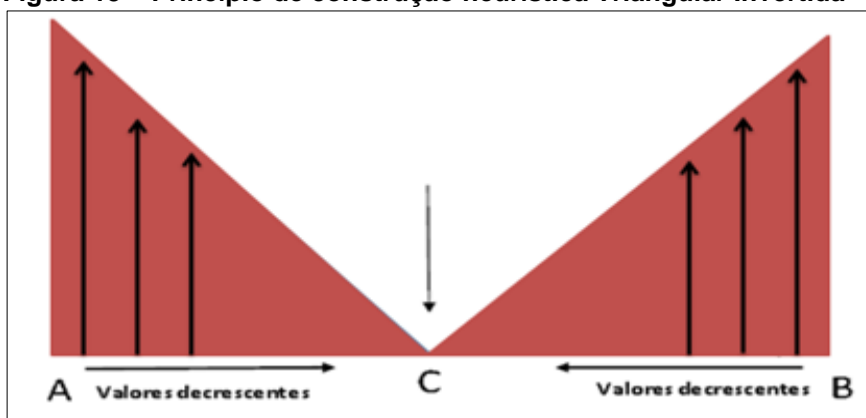
	Triangular	Triangular com restrição <i>no-idle</i>
<i>Makespan</i>	39 u.t	43 u.t
<i>Mean Flowtime</i>	26,2 u.t	33,4 u.t

Fonte: Próprio autor (2017)

2.6.4 Heurística Triangular Invertida

Da mesma forma que a heurística Triangular, a heurística Triangular Invertida é assim denominada pois sua lógica de construção baseia-se no formato de um triângulo, neste caso **invertido**, como apresenta a Figura 15. Nesta heurística, a tarefa com menor valor de soma dos tempos de processamento é alocada na posição central.

Figura 15 – Princípio de construção heurística Triangular Invertida



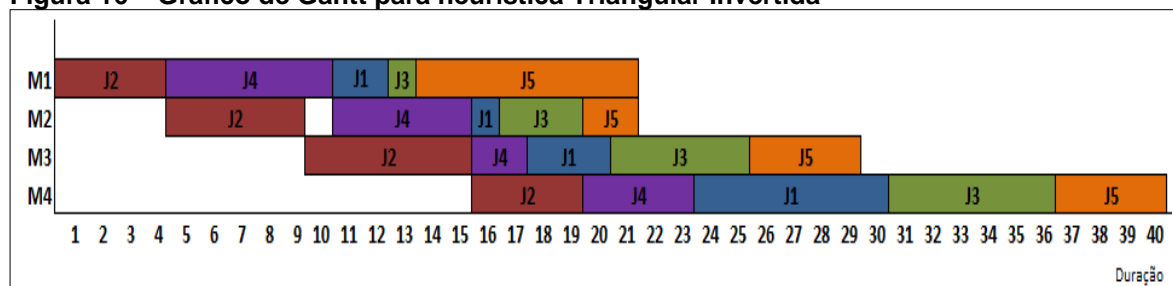
Fonte: adaptado de Branco e Santos (2016).

Conforme a Figura 15, a ordenação da heurística Triangular Invertida começa com a alocação da tarefa com maior valor de soma dos tempos de processamento

na extremidade A, o segundo maior valor da soma dos tempos de processamento é alocado na primeira posição da extremidade B. Por esta lógica, o terceiro maior valor da soma dos tempos de processamento alocado na segunda posição da extremidade A, e o quarto maior valor na segunda posição da extremidade B. A construção da heurística segue desta forma para as somas dos tempos de processamento das n tarefas, até que a tarefa de menor valor de soma seja alocada na posição central C (Figura 15) (BRANCO E SANTOS, 2016).

Como exemplo de ordenação da heurística Triangular Invertida, apresenta-se a Figura 16, com a heurística implementada sem a restrição *no-idle*, e novamente com os dados contidos na Tabela 1.

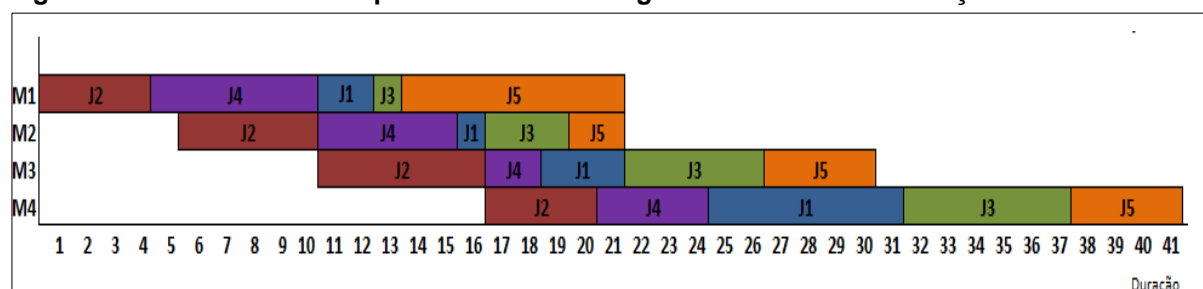
Figura 16 – Gráfico de Gantt para heurística Triangular Invertida



Fonte: Próprio autor (2017)

Na Figura 17 é possível perceber a implementação da restrição *no-idle* pela eliminação do único *gap*, entre a tarefa J2 e J4 na máquina 2 (M2). Para a eliminação de tal *gap* houve o atraso do início de processamento da tarefa J2 na máquina 2, 3 e 4 em uma unidade de tempo. Neste caso, há uma diferença de uma unidade de tempo tanto para o valor de *makespan* quanto do *mean flowtime*, quando se compara a implementação da heurística Triangular Invertida com e sem restrição *no-idle*.

Figura 17 – Gráfico de Gantt para heurística Triangular Invertida com restrição *no-idle*



Fonte: Próprio autor (2017)

2.6.5 Heurística Randômica

Na heurística Randômica não existe nenhum parâmetro a ser seguido na ordenação. A sequencia se dá por n tarefas ordenadas de maneira aleatória em m máquinas. Para o caso da restrição *no-idle* devem-se restringir tempos ociosos das máquinas, de modo que não hajam lacunas entre o processamento de uma tarefa e outra (NAGANO; JANUÁRIO, 2013, p.271).

Para o presente trabalho, a heurística Randômica será implementada em cinco tentativas, para todos os problemas, de modo que seja considerada a melhor solução em termos de desempenho dos resultados entre tais.

3 METODOLOGIA

A pesquisa científica pode ser definida de diversas formas e representa de modo geral, um meio de construção do conhecimento através de um método científico, que possui a capacidade de ser disseminada. De acordo com Gil (2008, p. 8) um método científico pode ser definido como “um conjunto de procedimentos intelectuais e técnicos, adotados para se atingir o conhecimento”.

Neste sentido, a pesquisa científica pode ser considerada ainda segundo Gil (2008, p. 27) como uma busca para o progresso da ciência de modo formalizado e generalizado. Minayo (1993) considera a pesquisa científica como “atividade básica na sua indagação e descoberta da realidade” (SILVA e MENEZES apud Minayo, 2005, p. 19).

Para tanto, é apresentada a forma clássica da classificação para o presente trabalho de acordo com Silva e Menezes (2005, p.20):

- Do ponto de vista da sua natureza:

A presente pesquisa pode ser classificada de natureza aplicada no sentido em que busca desenvolver conhecimentos com aplicação prática para a solução do problema específico conhecido na literatura como problema de programação da produção *flow shop*.

- Do ponto de vista da forma de abordagem:

Neste sentido, a pesquisa é classificada na forma de abordagem quantitativa uma vez em que a sua análise é realizada a partir da experimentação computacional da solução do problema apresentado na forma de recursos quantificáveis. Serão utilizadas comparações entre os valores de tempos computacionais obtidos na experimentação por exemplo.

- Do ponto de vista de seus objetivos:

É considerada nesta forma, como uma pesquisa descritiva e explicativa já que pretende realizar a um método experimental computacional para o aprofundamento do conhecimento, juntamente com o levantamento das técnicas padronizadas e a geração de dados.

Sendo assim, o presente trabalho, como procedimento técnico de pesquisa bibliográfica, possui a necessidade de desenvolver seus estudos a partir da revisão

bibliográfica do material já publicado sobre o tema abordado, seguido da realização das seguintes etapas:

- i) Experimento computacional: geração do banco de dados através de um *software* gerador de dados, simulando o cenário teste de um problema *flow shop* permutacional, composto por n tarefas e m máquinas. Além da implementação computacional dos métodos heurísticos SPT, LPT, Triângular, Triângular Invertida e Randômica, descritos anteriormente, para a solução do problema proposto;
- ii) Adaptação e Verificação: adaptar os métodos utilizados para outros problemas a fim de verificar a eficiência computacional e a qualidade das soluções obtidas;
- iii) Análise de resultados e posicionamento: realizar a análise crítica das soluções obtidas na experimentação juntamente com a comparação da eficiência computacional entre os métodos aplicados, emitir o parecer e a conclusão do trabalho. Para tanto serão utilizadas ferramentas estatísticas encontradas na literatura para a avaliação do desempenho comparativo entre os procedimentos implementados.

3.1 DESENVOLVIMENTO

A experimentação computacional proposta pelo presente trabalho foi realizada através de um conjunto teste de dados, gerados pelo *software* Gerador de Dados. O problema trata da programação *flow shop* permutacional com a restrição *no-idle* de um conjunto de n tarefas a serem processadas por um conjunto de m máquinas. A combinação $n \times m$ determinou a formação de 120 classes de 100 problemas.

As combinações $n \times m$ do conjunto de dados gerados são determinadas por: 10x5, 10x10, 10x15, 10x20, 10x25, 10x30, 20x5, 20x10, 20x15, 20x20, 20x25, 20x30, 30x5, 30x10, 30x15, 30x20, 30x25, 30x30, 40x5, 40x10, 40x15, 40x20, 40x25, 40x30, 50x5, 50x10, 50x15, 50x20, 50x25, 50x30, 60x5, 60x10, 60x15, 60x20, 60x25, 60x30, 70x5, 70x10, 70x15, 70x20, 70x25, 70x30, 80x5, 80x10, 80x15, 80x20, 80x25, 80x30, 90x5, 90x10, 90x15, 90x20, 90x25, 90x30, 100x5, 100x10, 100x15, 100x20, 100x25, 100x30, 110x5, 110x10, 110x15, 110x20, 110x25, 110x30, 120x5, 120x10, 120x15, 120x20, 120x25, 120x30, 130x5, 130x10, 130x15,

130x20, 130x25, 130x30, 140x5, 140x10, 140x15, 140x20, 140x25, 140x30, 150x5, 150x10, 150x15, 150x20, 150x25, 150x30, 160x5, 160x10, 160x15, 160x20, 160x25, 160x30, 170x5, 170x10, 170x15, 170x20, 170x25, 170x30, 180x5, 180x10, 180x15, 180x20, 180x25, 180x30, 190x5, 190x10, 190x15, 190x20, 190x25, 190x30, 200x5, 200x10, 200x15, 200x20, 200x25, 200x30.

Desta forma, a experimentação computacional partiu de um conjunto de instâncias composto por 12.000 arquivos com as combinações entre o número de tarefas e máquinas descritas anteriormente. A programação foi desenvolvida em linguagem Pascal, no *software* DevPascal por meio de diferentes algoritmos.

Para a solução do problema proposto, foram executadas as heurísticas, descritas no referencial teórico do presente trabalho: SPT, LPT, Triângular, Triângular Invertida e Randômica. A função biobjetivo implementada é apresentada pela Equação 1, e pretende otimizar a solução através da interação entre o tempo total da programação – *makespan* – e o tempo médio de fluxo – *mean flowtime*.

$$FO = (\alpha \cdot Makespan + (1 - \alpha) \cdot (Mean Flowtime)) \quad (1)$$

Para a Equação 1, o parâmetro α que pondera o valor entre *makespan* e *flowtime*, tem o seu valor implementado no intervalo de 0 a 1, com variação de 0,2. Ao invés do critério tempo de fluxo (*flowtime*), foi utilizado na função objetivo (FO) o **tempo médio de fluxo** (*mean flowtime*) com a finalidade de amenizar a discrepância de valor no caso de se utilizar os critérios *makespan* e *flowtime*.

Após a implementação das heurísticas propostas para todos os valores de α , foram avaliados o desempenho de cada algoritmo através dos resultados da FO e dos tempos computacionais médios obtidos. Foi utilizado um microcomputador com processador Intel Core i-5 6200U 2.3GHz, com memória RAM de 6Gb, sistema operacional Windows 10.

3.2 ANÁLISE DOS RESULTADOS

A avaliação do desempenho dos algoritmos implementados considerou os critérios porcentagem de sucesso (PS), desvio relativo médio (DRM), e tempo médio de computação (TMC), em que se define:

- **PS:** soma de vezes em que uma heurística apresenta melhor solução dentro do intervalo de 100 problemas avaliados.
- **DRM:** relação entre uma determinada heurística com a que apresenta melhor solução para o problema, sendo que o menor valor de DRM representa a heurística com melhor desempenho do caso. O cálculo de DRM é apresentado pela Equação 2.

$$DRM = \frac{Dh - Dm}{Dm} \times 100 \quad (2)$$

Onde:

Dh: valor da FO da heurística analisada;

Dm: valor da FO da heurística que apresenta melhor solução.

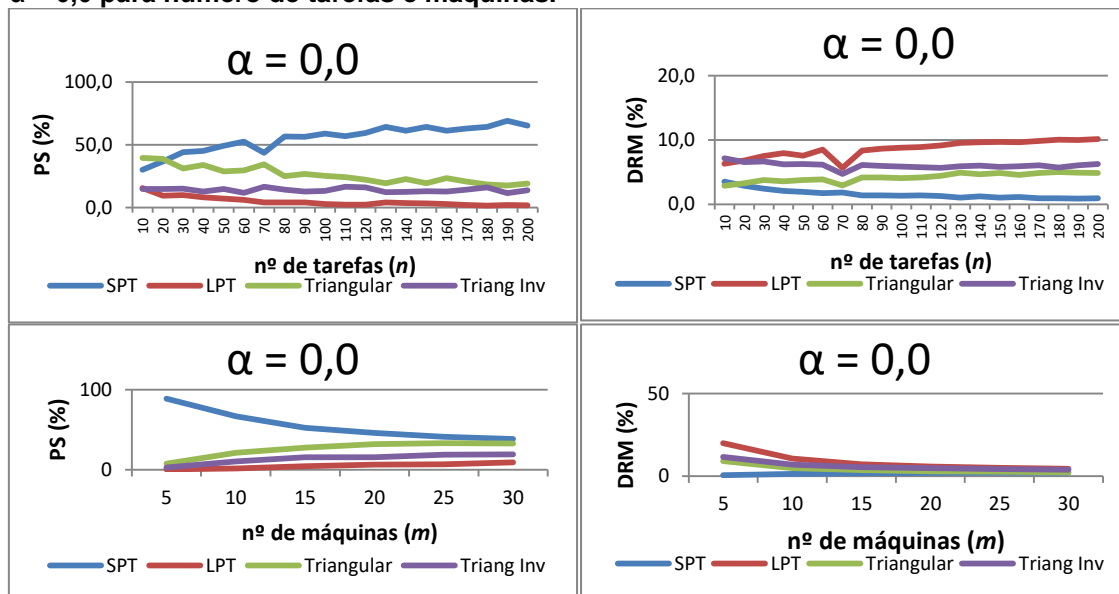
- **TMC:** tempo decorrido pelo software para a geração das saídas de soluções dos problemas.

Tais avaliações foram aplicadas aos seis valores do parâmetro α (0; 0,2; 0,4; 0,6; 0,8; e 1,0), para cada número de tarefas ($n= 10, 20, 30, \dots, 200$) e máquinas ($m= 5, 10, 15, 20, 25, 30$) em que cada conjunto $n \times m$ apresenta 600 instâncias de geração da FO. Para tanto são apresentados, nas tabelas e gráficos seguintes, os dados obtidos nas avaliações por meio dos critérios descritos nesta seção, de modo que seja identificado o melhor método heurístico para a solução do problema proposto. A base de dados utilizada para construção dos gráficos analisados é apresentada nos Apêndices A e B.

As figuras 18 e 19 apresentam a comparação dos métodos para $\alpha = 0,0$ e $0,2$. Em tal análise percebe-se as melhores eficiências das heurísticas SPT e Triangular nas soluções obtidas para o problema proposto. Para o parâmetro $\alpha = 0,0$ (Figura 18), analisando-se o número de tarefas, a heurística SPT demonstra melhores resultados, com maiores valores de PS e menores valores para DRM, exceto para 10 e 20 tarefas onde tal resultado está para a heurística Triangular. Quando se têm 70 tarefas, observa-se um certo crescimento das eficiências das heurísticas Triangular, Triangular Invertida e LPT, com aumento dos valores de PS e redução dos valores de DRM, neste mesmo ponto o resultado se inverte para a SPT. Em outra análise, se ressalta a queda do distanciamento entre todas as heurísticas com

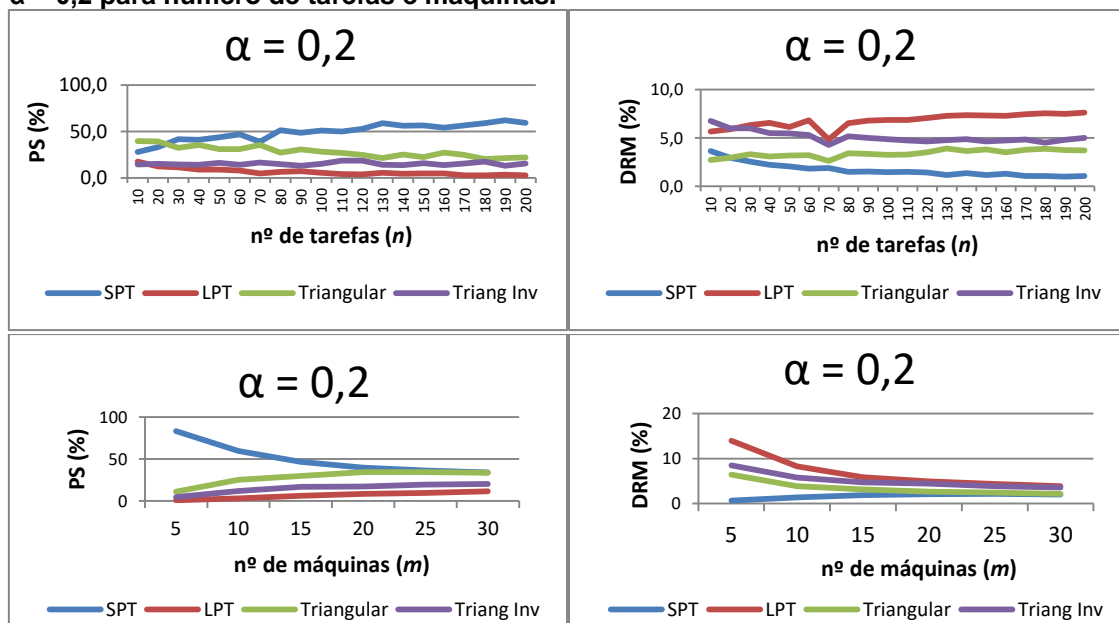
o aumento do número de máquinas a partir de 15. Tais resultados se apresentam semelhantes para o parâmetro $\alpha = 0,2$ (Figura 19).

Figura 18 – Gráficos de comparação entre as heurísticas para PS e DRM com o parâmetro $\alpha = 0,0$ para número de tarefas e máquinas.



Fonte: Próprio autor, 2017

Figura 19 – Gráficos de comparação entre as heurísticas para PS e DRM com o parâmetro $\alpha = 0,2$ para número de tarefas e máquinas.

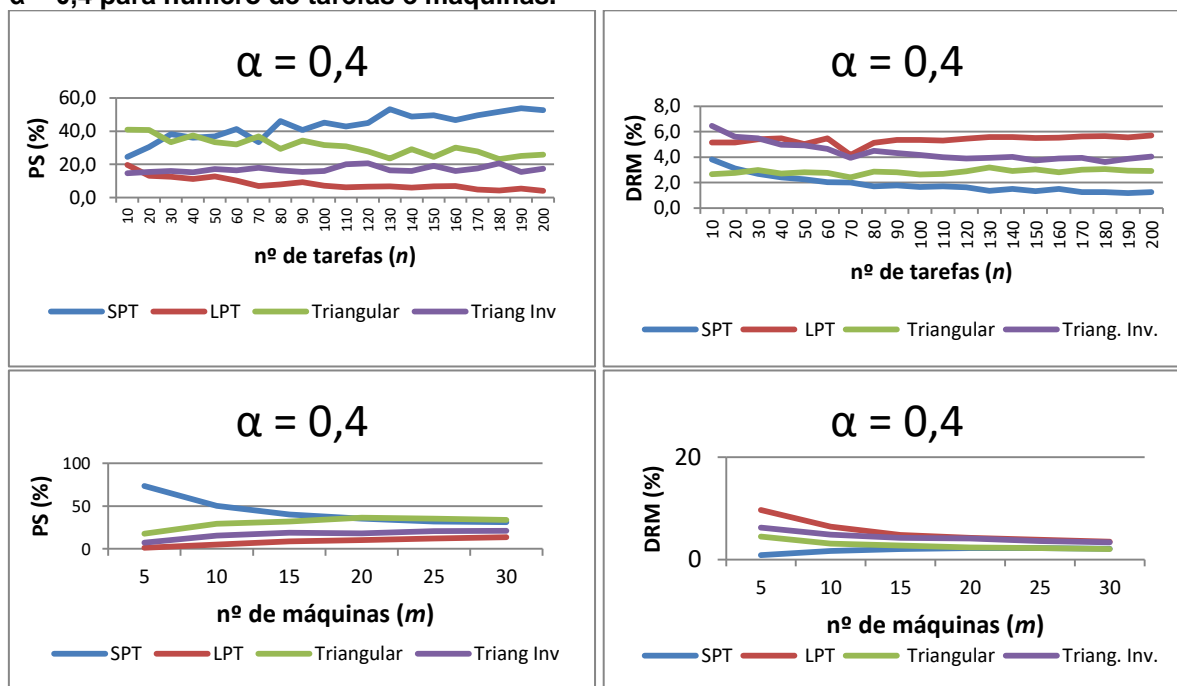


Fonte: Próprio autor, 2017.

Para $\alpha = 0,4$ (Figura 20) os gráficos apresentam novamente os mesmos resultados descritos anteriormente em $\alpha = 0,0$ e $\alpha = 0,2$ das Figuras 18 e 19, respectivamente. Neste caso, a heurística Triangular se mostra mais eficiente do

que a SPT em quatro pontos (10, 20, 40 e 70 tarefas). As heurísticas LPT e Triangular Invertida igualmente demonstraram os piores resultados para o problema.

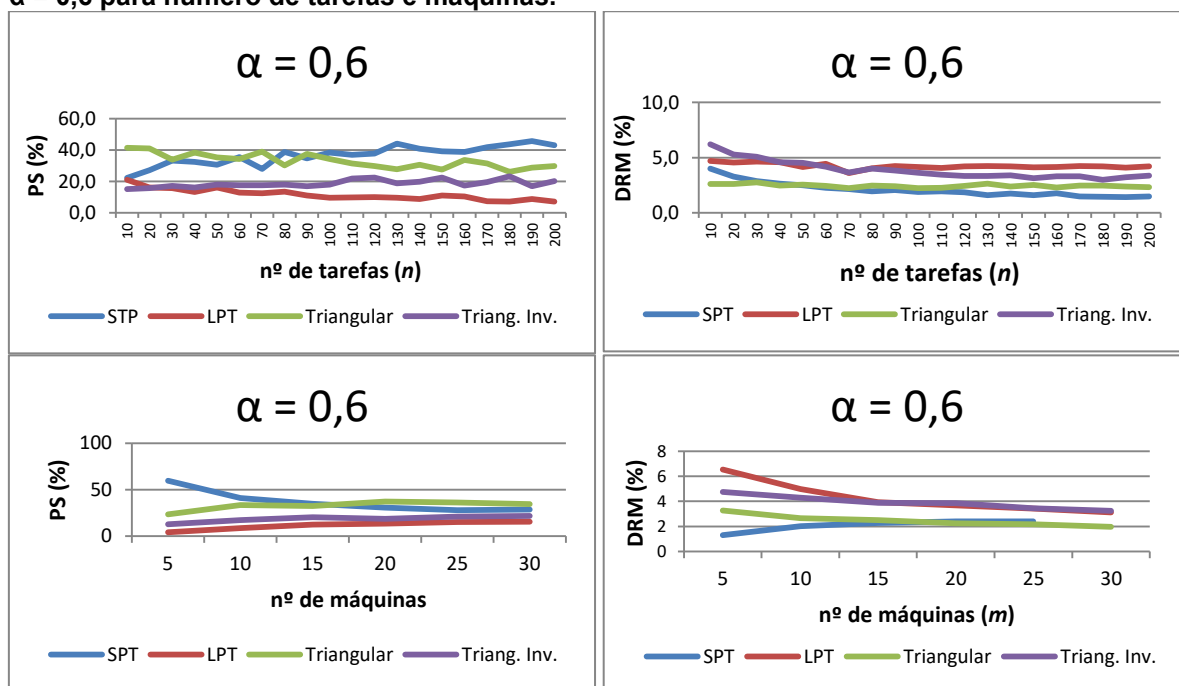
Figura 20 – Gráficos de comparação entre as heurísticas para PS e DRM com o parâmetro $\alpha = 0,4$ para número de tarefas e máquinas.



Fonte: Próprio autor, 2017.

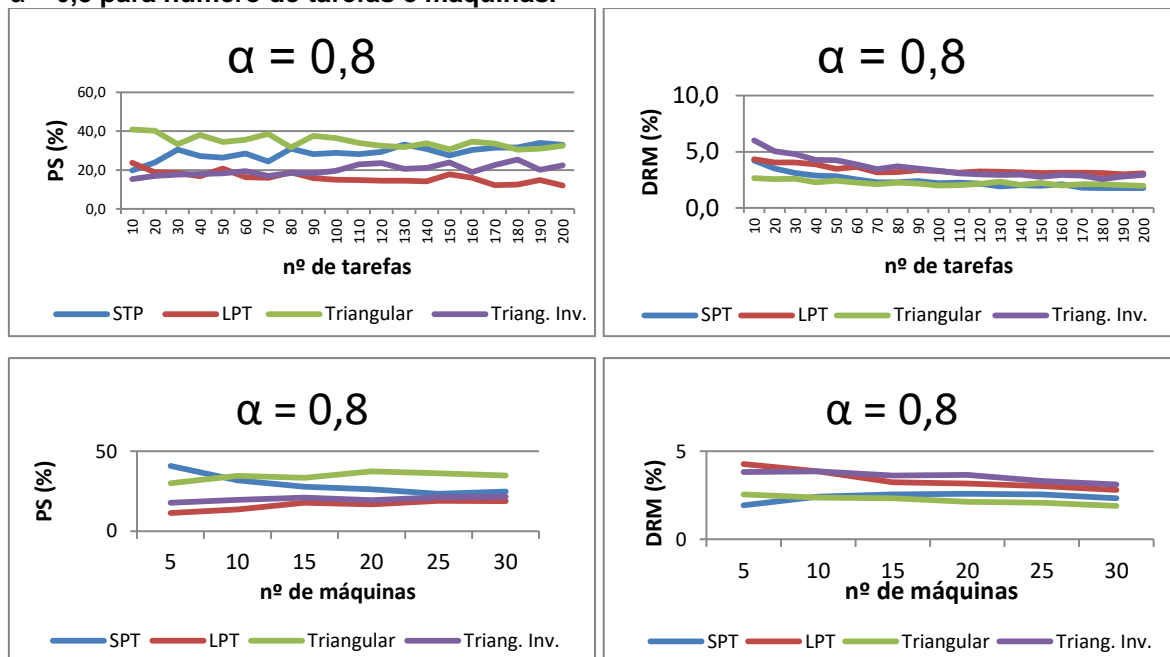
Na Figura 21, para $\alpha = 0,6$ percebe-se novamente a boa eficiência das heurísticas Triangular e SPT, sendo que a primeira apresenta PS superior principalmente até o número de tarefas $n = 120$, e esse fato demonstra a superioridade da SPT para problemas mais complexos, com maior número de tarefas, neste caso. Na análise pelo número de máquinas a heurística SPT somente é favorecida para $m = 5$ e 10. Já para o parâmetro $\alpha = 0,8$ (Figura 22) ressalta-se a superioridade da Triangular principalmente para menores números de tarefas e a partir de $m = 10$. Para os valores do parâmetro α testados até então ($\alpha = 0; 0,2; 0,4; 0,6; e 0,8$), a heurística Randômica não apresentou nenhuma porcentagem de sucesso, por este motivo tal heurística não está presente nos gráficos das Figuras 18, 19, 20, 21 e 22, já que os valores de DRM seriam muito altos para serem plotados.

Figura 21 – Gráficos de comparação entre as heurísticas para PS e DRM com o parâmetro $\alpha = 0,6$ para número de tarefas e máquinas.



Fonte: Próprio autor, 2017.

Figura 22 – Gráficos de comparação entre as heurísticas para PS e DRM com o parâmetro $\alpha = 0,8$ para número de tarefas e máquinas.

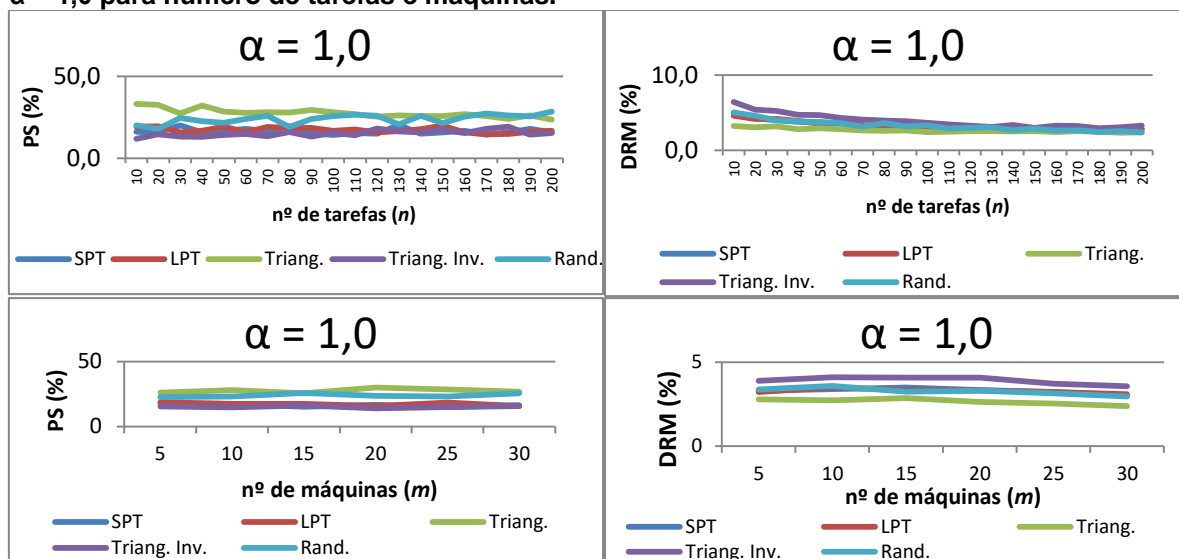


Fonte: Próprio autor, 2017.

A Figura 23 apresenta os gráficos para o parâmetro $\alpha = 1,0$, com prioridade máxima do *makespan* da FO. Para este caso, embora a heurística Triangular demonstre novamente superioridade, destaca-se o aparecimento da heurística

Randômica com bons resultados tanto para valores altos de PS e consequentemente para baixos valores de DRM. Tais ocorrências podem ser explicadas pelo fato de que os métodos testados no presente trabalho não apresentem tão boa eficiência para *makespan*. Além disso, a heurística Randômica está sendo testada cinco vezes, garantindo desta forma, alguma qualidade.

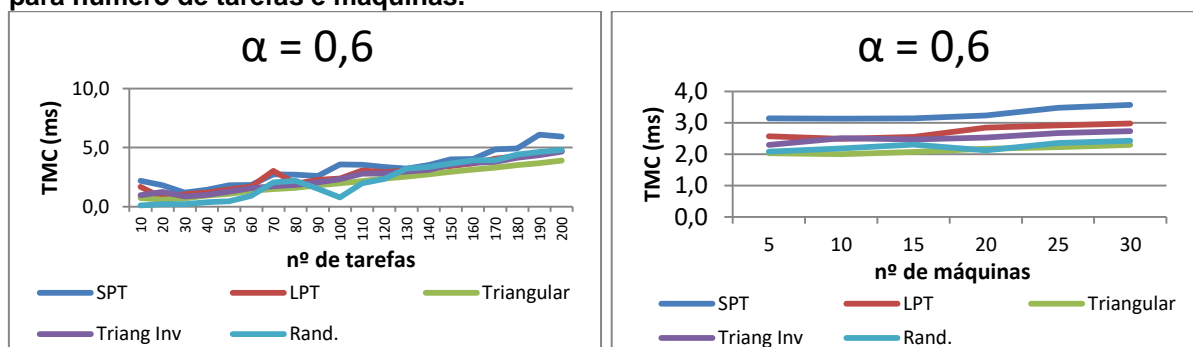
Figura 23 – Gráficos de comparação entre as heurísticas para PS e DRM com o parâmetro $\alpha = 1,0$ para número de tarefas e máquinas.



Fonte: Próprio autor, 2017.

Para a avaliação do TMC apresenta-se a Figura 24, com os gráficos dos valores de TMC para as heurísticas SPT, LPT, Triangular, Triangular Invertida, e Randômica com $\alpha = 0,6$. Para todos os valores do parâmetro α , os comportamentos dos tempos médios de computação foram semelhantes, por este motivo a Figura 24 é a única a ser exposta para tal critério de avaliação (TMC). É possível perceber através dos gráficos da Figura 24 que não existem diferenças significativas entre os valores de TMC para as cinco heurísticas implementadas, em que todos são considerados viáveis. Além disso, apresenta-se um crescimento no TMC para todas as heurísticas conforme aumenta-se o número de tarefas e máquinas, em problemas mais complexos.

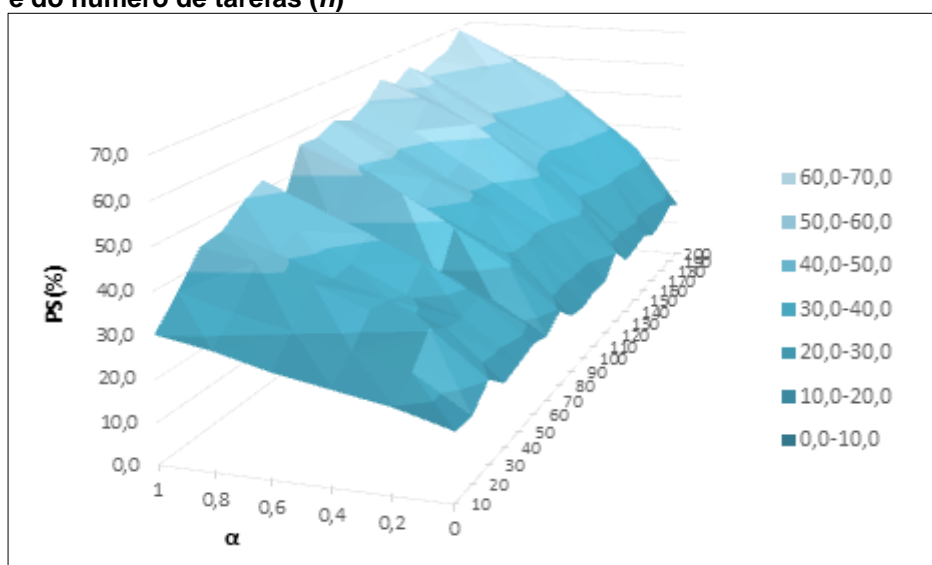
Figura 24 – Gráficos de comparação entre as heurísticas para TMC com o parâmetro $\alpha = 0,6$ para número de tarefas e máquinas.



Fonte: Próprio autor, 2017.

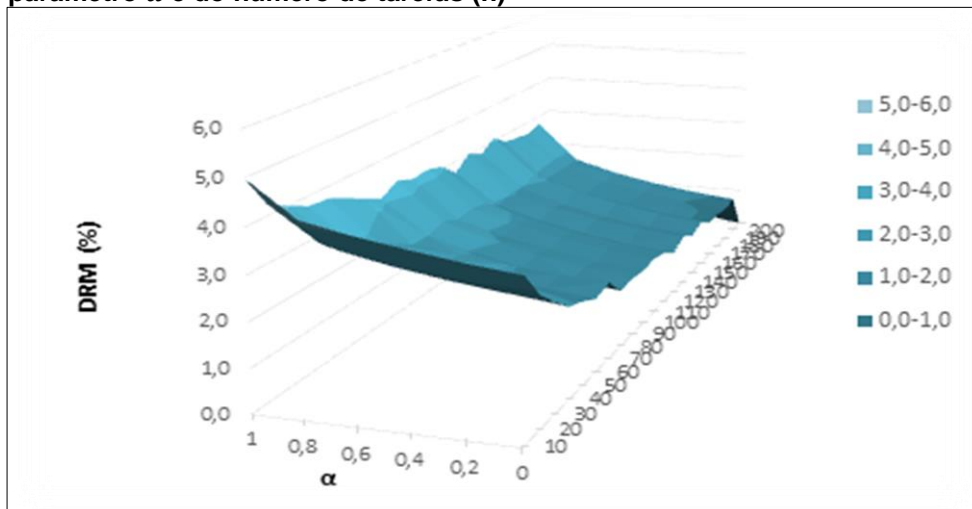
Como as heurísticas SPT e Triangular demonstraram as melhores soluções na minimização da FO apresentada pela Equação 2, somente estas foram foco da próxima análise, em que se observa a eficiência de cada heurística com a variação do valor de α e do número de tarefas (n). A Figura 25 expõe o comportamento da heurística SPT com as variações e, permite perceber que os maiores valores de PS se dão principalmente para os problemas mais complexos, com maior número de tarefas. Tal observação também pode ser confirmada pela Figura 26, em que conforme se aumenta o número de tarefas (n) decresce os valores de DRM para SPT. Em relação à variação do parâmetro α , para os problemas mais complexos se confirma o favorecimento desta heurística para a função-objetivo de *mean flowtime*.

Figura 25 – Gráfico de superfície da PS (%) da heurística SPT para as variações do parâmetro α e do número de tarefas (n)



Fonte: Próprio autor (2017)

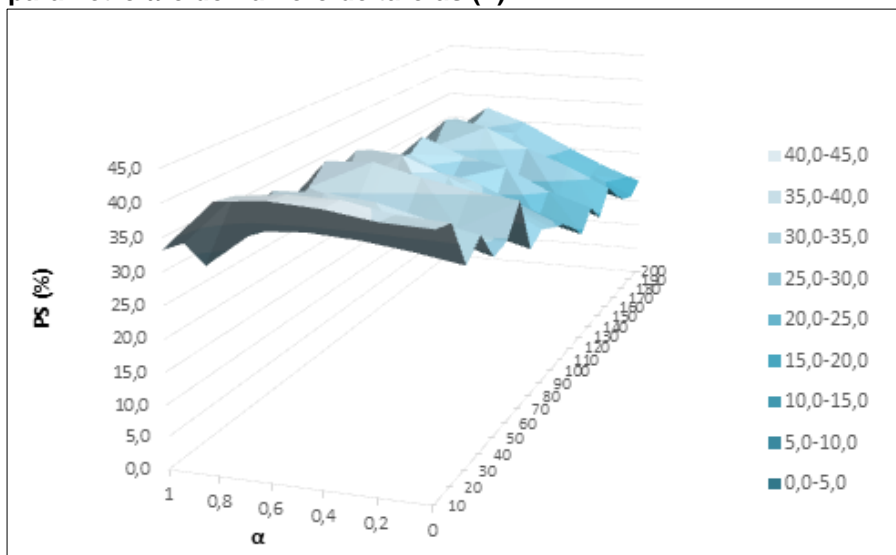
Figura 26 – Gráfico de superfície do DRM (%) da heurística SPT para as variações do parâmetro α e do número de tarefas (n)



Fonte: Próprio autor (2017)

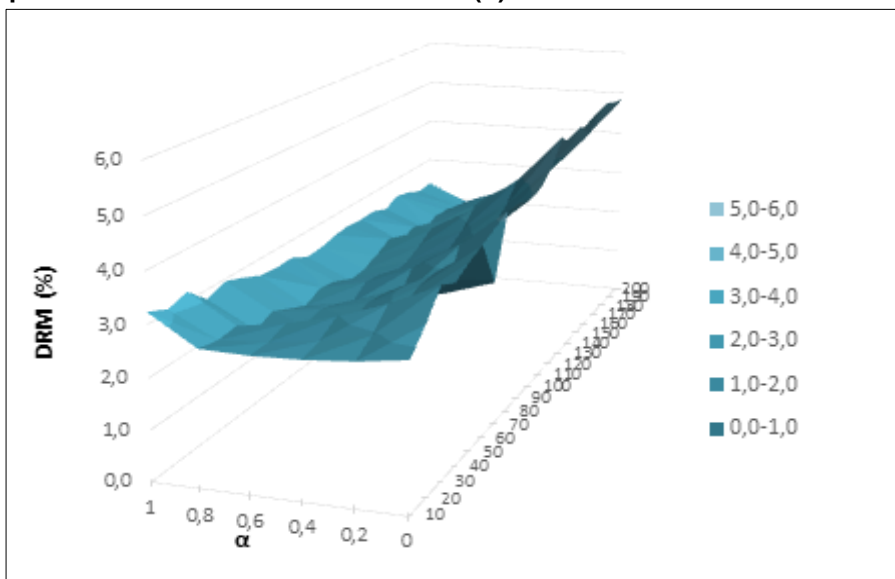
Da mesma forma, as Figuras 27 e 28 demonstram o comportamento da heurística Triangular para as variações do parâmetro α e do número de tarefas (n). Esta análise confirma as anteriores quando se dá os maiores valores de PS para os problemas de menor número de tarefas e os menores valores de DRM quando se tem o parâmetro α no favorecimento da função-objetivo *makespan*.

Figura 27 – Gráfico de superfície da PS (%) da heurística Triangular para as variações do parâmetro α e do número de tarefas (n)



Fonte: Próprio autor (2017)

Figura 28 – Gráfico de superfície do DRM (%) da heurística Triangular para as variações do parâmetro α e do número de tarefas (n)



Fonte: Próprio autor (2017)

4 CONCLUSÃO

O presente trabalho apresentou o problema de programação de operações *flow shop* e, através de implementação computacional, permitiu que se verificasse a eficiência dos métodos de solução aplicados e, dessa forma, atinge os objetivos propostos.

Foi possível adaptar as heurísticas SPT, LPT, Triangular, Triangular Invertida e Randômica, por meio de diferentes algoritmos, para o problema de programação *flow shop* permutacional com a restrição *no-idle*. Foram geradas soluções para as 12.000 instâncias de dados, derivadas da combinação $n \times m$ de tarefas e máquinas, com base na função biobjetivo descrita na Equação 1.

Permitiu-se analisar o comportamento das heurísticas, aplicadas aos seis valores do parâmetro α (0; 0,2; 0,4; 0,6; 0,8; e 1,0), e verificar a qualidade das soluções geradas em cada caso. Deste modo, a experimentação computacional foi desenvolvida de maneira robusta, já que a análise estatística considerou o conjunto suficientemente grande de dados testados, 12.000 instâncias solucionadas por cinco heurísticas diferentes, sendo cada uma destas testadas por seis valores do parâmetro α (360.000 soluções geradas). Tal contexto experimental aliado ao fato de que as heurísticas Triangular e Triangular Invertida, referenciadas no presente trabalho, são praticamente inéditas na literatura do assunto, apontam a possibilidade de publicação.

Na avaliação do desempenho dos algoritmos implementados, através dos critérios PS e DRM é possível concluir, de forma geral, que as heurísticas SPT e Triangular demonstram melhor eficiência para o problema *flow shop* permutacional com a restrição *no-idle*. De $\alpha = 0,0$ á $\alpha = 0,4$, em que se favorece o critério de *mean flowtime*, é evidente o melhor desempenho da heurística SPT, especialmente para maiores números de tarefas (n) e menores números de máquinas (m).

De $\alpha = 0,6$ á $\alpha = 1,0$, em que o critério *makespan* da FO é favorecido, é possível perceber o melhor desempenho da heurística Triangular, sobretudo nas instâncias que apresentam menores números de tarefas (n) e para as de maiores números de máquinas (m). Ainda assim, com $\alpha = 1,0$, único valor do parâmetro α que somente favorece o critério *makespan* da FO, é o primeiro momento em que a

heurística Randômica apresenta soluções razoáveis. Tal fato pode significar o baixo desempenho das demais heurísticas implementadas para o parâmetro *makespan*. Dessa forma, como a heurística SPT demonstrou superioridade no desempenho dos problemas para o critério *mean flowtime*, em que as heurísticas testadas no presente trabalho se evidenciaram mais eficientes, é possível afirmar que a heurística SPT possui melhor desempenho na maior parte dos problemas testados.

Nas avaliações do TMC, de forma geral é possível concluir que para as heurísticas SPT, LPT, Triangular, Triangular Invertida e Randômica os valores encontrados para tal critério são razoáveis e viáveis, sendo aplicável a solução prática dos problemas de programação *flow shop* permutacional com a restrição *no-idle*.

REFERÊNCIAS

- BRANCO, F.J.C. **Avaliação de métodos heurísticos para o problema *no-wait flowshop* com o critério de minimização da duração total da programação.** 2006. Dissertação (Mestrado) – Escola de Engenharia de São Carlos, Departamento de Engenharia de Produção, Universidade de São Paulo. São Carlos, 2006.
- BRANCO, F.J.C. **Um novo método heurístico construtivo de alto desempenho para o problema *no-idle flow shop*.** 2011. Dissertação (Doutorado) – Escola de Engenharia de São Carlos, Departamento de Engenharia de Produção, Universidade de São Paulo. São Carlos, 2011.
- BRANCO, F. J. C.; DOS SANTOS, A. L. **Avaliação de Ordenções Iniciais Para o Problema Flowshop com Restrição No-Wait e Minimização de Função-Objetivo Ponderada Entre Makespan e Flowtime.** Simpósio Brasileiro de Pesquisa Operacional. 2016.
- CORRÊA, H. L.; CORRÊA, C. A. **Administração de Produção e de Operações: manufatura e serviços: uma abordagem estratégica.** 2. ed. São Paulo: Atlas, 2013.
- CORRÊA, H. L.; GIANESI, I. G. N. **Just in Time, MRP II e OPT: um enfoque estratégico.** 2. ed. São Paulo: Atlas, 2011.
- CRUZ, F. E. ; BRANCO, F. J. C. Análise de Métodos Para Solução Problema de Programação de Operações *Flow shop*. In: **SIMPEP**, 2009, Bauru. Simpósio de Engenharia de Produção, 2009.
- FATTAHI, P.; HOSSEINI, S. M. H.; JOLAI, F. A mathematical model and extension algorithm for assembly flexible flow shop scheduling problem. ***The International Journal of Advanced Manufacturing Technology***, 65(5-8), p. 787–802, 2012.
- FUCHIGAMI, H. Y. **Flexible flow line com tempos de setup: métodos heurísticos.** 2010. Dissertação (Doutorado) – Escola de Engenharia de São Carlos, Departamento de Engenharia de Produção, Universidade de São Paulo. São Carlos, 2010.
- FUCHIGAMI, H. Y. **Métodos heurísticos construtivos para o problema de programação da produção em sistemas *flow shop* híbridos com tempos de preparação das máquinas assimétricos e dependentes da sequência.** 2005. Dissertação (Mestrado) – Escola de Engenharia de São Carlos, Departamento de Engenharia de Produção, Universidade de São Paulo. São Carlos, 2005.
- FUCHIGAMI, H. Y.; ALVES, A.; HORDONES, P. A. Métodos heurísticos para a minimização do número de tarefas atrasadas em *flow shop* permutacional. In: SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL. 12., 2015, Porto de Galinhas, PE. **Anais da SBPO**. Porto de Galinhas: 2015. p. 86-97.
- GAITHER, N; FRAZIER, G. **Administração da produção e operações.** 8ª ed. São Paulo: Pioneira Thomson Learning, 2004.

GAO, S. W.; et al. Break the ties for NEH heuristic in solving the permutation flow shop problems. *Journal of Donghua University (English Edition)*, 25(3), p. 258–262, 2008.

GIGANTE, R. L. **Heurística construtiva para a programação de operações *flow shop* permutacional**. 2010. Dissertação (Mestrado) – Escola de Engenharia de São Carlos, Departamento de Engenharia de Produção, Universidade de São Paulo. São Carlos, 2010.

GIL, A. C. **Métodos e Técnicas de Pesquisa Social**. 6. ed. São Paulo: Atlas, 2008.

HILLIER, F.S.; LIEBERMAN, G. J. **Introdução à pesquisa operacional**. 8. ed. São Paulo: McGraw-Hill, 2006.

GUIMARÃES, I. F. G.; et al. O problema de sequenciamento da produção em um ambiente *flowshop* com linhas semi-paralelas e operação de sincronização final. In: XXXV ENCONTRO NACIONAL DE ENGENHARIA DE PRODUÇÃO. 14. 2015, Fortaleza, CE. **Anais do ENEGEP**. Fortaleza: 2015. P. 1-14.

KALCZYNSKI, P. J.; KAMBUROWSKI, J. An improved NEH heuristic to minimize makespan in permutation flow shops. *Computers & Operations Research*, 35(9), p. 3001–3008, 2008.

KALCZYNSKI, P. J.; KAMBUROWSKI, J. On no-wait and no-idle flow shops with makespan criterion. *European Journal of Operational Research*, 178(3), p. 677–685, 2007.

LAWRENCE, S. T.; SEWELL, E. C. Heuristic, Optimal, Static, and Dynamic schedules when processing times are uncertain. *Journal of Operations Management*, v. 15, p. 71 – 82, 1997.

LUGO, P. L. M.; TEIXEIRA, R. F. J. *Scheduling em hybrid flow shop* com máquinas paralelas não relacionadas, setups dependentes da sequencia e elegibilidade de máquinas via algoritmo genético híbrido. In: ENCONTRO NACIONAL DE ENGENHARIA DE PRODUÇÃO. 12. 2012, Bento Gonçalves, RS. **Anais do ENEGEP**. Bento Gonçalves: 2012. P. 1-12.

LUGO, P.L.M.; et al. Um modelo de programação inteira mista para a programação da produção em *flowshop* híbrido com *buffers* limitados. In: SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL. 45., 2013, Natal, RN. **Anais da SBPO**. Natal: 2013. p. 154-165.

MOCCELLIN, J. V. NAGANO, M. S. **Flow shop** com máquinas paralelas genéricas. In: SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL. 9. 2003, Natal, RN. **Anais da SBPO**. Natal: 2003. p. 1561-1569 .

MOCCELLIN, J. V.; NAGANO, M. S. Uma propriedade estrutural do problema de programação da produção *flow shop* permutacional com tempos de *setup*. *Pesquisa Operacional*, 27(3), p. 487–515, 2007.

MOCCELLIN, J. V.; FUCHIGAMI, H. Y. Heurísticas construtivas para programação de flexible flow line com tempos de setup independentes da sequência. In: SIMPÓSIO

BRASILEIRO DE PESQUISA OPERACIONAL. 43.,2011, Ubatuba, São Paulo. **Anais da SBPO**. Ubatuba: 2011. p. 58-69.

MONTEVECHI, J. A.; et al. Análise comparativa entre regras heurísticas de sequenciamento da produção aplicada em job shop. **Produto & Produção**, 6(2), p.12-18, 2002.

NAGANO, M. S.; BRANCO, F. J. C.; MOCCELLIN, J. V. *A high quality solution heuristic for no-wait flow shop scheduling*. In: SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL. 12.,2010, Bento Gonçalves, RS. **Anais da SBPO**. Bento Gonçalves: 2010. p. 24-35.

NAGANO, M. S.; JANUÁRIO, J. C. S. S. *Evolutionary heuristic for makespan minimization in no-idle flow shop production systems*. **Acta Scientiarum. Technology**, 35(2), p. 271–278, 2013.

NAGANO, M. S.; MOCCELLIN, J. V.; LORENA, L. A. N. Programação da Produção *Flow Shop* Permutacional com Minimização do Tempo Médio de Fluxo. In: XXXVI Simpósio Brasileiro de Pesquisa Operacional, 2004, São João Del-Rei - MG. **Anais do XXXVI SBPO**, 2004.

NAGANO, M. S.; MOCCELLIN, J. V.; LORENA, L. A. N. *Reducing in-process inventory in permutation flow shop production environment*. **Revista Produção**, p. 1-12, 2005.

NAWAZ, M.; ENSCORE, E.E.; HAM, I. A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. **Omega**, 11(1), p. 91-95, 1983.

OVANDO, A. K. 2015. **Avaliação de métodos de ordenação inicial para o problema de programação da produção em ambiente *flowshop* não permutacional com critério de minimização do *makespan***. Escola de Engenharia de São Carlos, Departamento de Engenharia de Produção Mecânica, Universidade de São Paulo. São Carlos, 2015.

PAIVA, E. L.; et al. **Estratégia de produção e de operações**: conceitos, melhores práticas, visão de futuro. 2. ed. Porto Alegre: Bookman, 2009.

PINEDO, M. L. **Planning and Scheduling in Manufacturing and Services**. 1. ed. New York: Springer, 2005.

QIN, H; et al. (2016). Permutation *flowshop* group scheduling with position-based learning effect. **Computers & Industrial Engineering**, 92, p. 1–15, 2016.

RANGEL, S.; FUCHIGAMI, H. Y. Métodos heurísticos para maximização do número de tarefas *just-in-time* em *flow shop* permutacional. In: SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL. 12.,2015, Porto de Galinhas, PE. **Anais da SBPO**. Porto de Galinhas: 2015. p. 98-109.

RANGEL, S.; FUCHIGAMI, H. Y. Uma análise de estudos de casos em sequenciamento da produção. In: SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL. 12. 2014, Salvador, BA. **Anais da SBPO**. Salvador: 2014. p. 159-170.

ROCHA, R. P. da; et al. Use of excel worksheets with user-friendly interface to minimize the makespan in a permutation flow shop production system. **Acta Scientiarum. Technology**, 36(3), p. 453, 2014.

SCARDOELLI, L. Y. 2006. **Novos métodos heurísticos para a programação de operações no-wait flow shop com critério de minimização do tempo total de fluxo**. Dissertação (Mestrado) – Escola de Engenharia de São Carlos, Departamento de Engenharia de Produção, Universidade de São Paulo. São Carlos, 2006.

SELEN, W. J.; HOTT, D. D. A new formulation and solution of the flowshop scheduling problem with no in-process waiting. **Applied mathematical modelling**, v. 10, n. 4, p. 246-248, 1986.

SELEN, W. J.; HOTT, D. D. A mixed-integer goal-programming formulation of the standard flow-shop scheduling problem. **Journal of the Operational Research Society**, v. 37, n. 12, p. 1121-1128, 1986.

SILVA, E. L.; MENEZES, E. M. **Metodologia da pesquisa e elaboração de dissertação**. 4.ed. Florianópolis: UFSC, 2005.

SLACK, N.; et al. **Administração da produção**. 3. ed. São Paulo: Atlas, 2009.

WANG, B. L.; LI, T. K.; GE, H. B. Neighborhood Search Heuristic for 2-Machine Flowshop Scheduling Problem with Limited Waiting Times. **Applied Mechanics and Materials**, 411-414, 1894–1897, 2013.

WILSON, J.M. Alternative formulations of a flow shop scheduling problem. **Journal of the Operational Research Society**, v.40, n4, p.395-399, 1989.

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

APÊNDICE A – Comparação entre os métodos SPT, LPT, Triangular, Triangular Invertida e Randômica, com variação de $\alpha = 0,0$ à $\alpha = 0,4$

APÊNDICE B – Comparação entre os métodos SPT, LPT, Triangular, Triangular Invertida e Randômica, com variação de $\alpha = 0,6$ à $\alpha = 1,0$

