UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

DEPARTAMENTO DE ENGENHARIA DE PRODUÇÃO

ENGENHARIA DE PRODUÇÃO

AMANDA DA SILVA CORREIA

AVALIAÇÃO DE REGRAS DE ORDENAÇÃO PARA O PROBLEMA *FLOWSHOP* COM *SETUP* SEPARADO

TRABALHO DE CONCLUSÃO DE CURSO

PONTA GROSSA

2018

AMANDA DA SILVA CORREIA

AVALIAÇÃO DE REGRAS DE ORDENAÇÃO PARA O PROBLEMA *FLOWSHOP* COM *SETUP* SEPARADO

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

Orientador: Prof. Dr. Fábio José Ceron Branco.

PONTA GROSSA

2018

	Ministério da Educação **UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ** **CÂMPUS PONTA GROSSA** Departamento Acadêmico de Engenharia de Produção	

TERMO DE APROVAÇÃO DE TCC

Avaliação de Regras de Ordenação para o Problema *Flowshop* com *Setup* Separado

por

Amanda da Silva Correia

Este Trabalho de Conclusão de Curso (TCC) foi apresentado em 27 de junho de 2018 como requisito parcial para a obtenção do título de Bacharel em Engenharia de Produção. A candidata foi arguida pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

_____
Prof. Dr. Fábio José Ceron Branco
Prof. Orientador

_____
Prof. Dra. Yslene Rocha Kachba
Membro titular

_____
Prof. Dr. Shih Yung Chin
Membro titular

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

AGRADECIMENTOS

Este documento representa a finalização de um ciclo da minha vida, que me proporcionou crescimento pessoal e profissional a cada dia durante todos esses anos vividos em Ponta Grossa. Além disso, este ciclo me mostrou o valor de pessoas tão queridas que estiveram comigo mesmo antes da faculdade e de pessoas que pude conhecer durante essa trajetória, e por isso devo meus sinceros agradecimentos à elas.

Infelizmente, estes parágrafos não atenderão todas as pessoas que, diretamente ou indiretamente, fizeram parte dessa fase tão importante, mas tenham plena certeza de que em cada momento da participação de todas essas pessoas estive grata.

Em um primeiro momento, agradeço meus pais, Edna da Silva Correia e Antônio Costa Correia, minha irmã Júlia da Silva Correia e toda minha família, pelo apoio diário, por acreditarem nos meus sonhos e fazerem parte deles. E que sem medir esforços, estiveram sempre ao meu lado mesmo que distantes.

Agradeço às minhas amigas de infância, Ana Luísa Ribeiro, Geovana Marrafão, Giovanna Raineri e Natália Cavalher pelas palavras de consolo nos momentos difíceis, pela felicidade compartilhada nos momentos de vitória, e principalmente, por nossa sincera amizade.

Aos meus colegas de sala e a todos com quem compartilhei momentos durante a faculdade, meus agradecimentos pela união durante esses anos e por todos os momentos que passamos juntos.

Ao meu namorado Rodrigo Machado de Oliveira e sua família que me acolheram como parte da família desde o dia que nos conhecemos e me ampararam durante todos esses anos que estive longe dos meus pais, e por isso se tornaram minha família ponta grossense.

Não poderia deixar de agradecer também ao meu professor orientador Fábio José Ceron Branco, por ensinar, em todos os momentos, com amor, paciência e alegria, e assim tornar possível a realização deste trabalho de conclusão.

Enfim, obrigada a todos que por algum motivo, fizeram parte dessa etapa e contribuíram de alguma forma nesta fase da minha vida.

RESUMO

CORREIA, Amanda da Silva. **Avaliação de Regras de Ordenação para o Problema *Flowshop* com *Setup* Separado.** 2018. 99 f. Trabalho de Conclusão de Curso (Bacharelado em Engenharia de Produção) - Universidade Tecnológica Federal do Paraná. Ponta Grossa, 2018.

O presente trabalho caracteriza-se pelo estudo comparativo de desempenho das regras de ordenação LPT, SPT e NEH adaptadas para a resolução de problemas de sequenciamento de tarefas *flowshop,* com tempo de *setup* separado e independente da sequência. Para realizar a análise de desempenho é considerado não só o tempo de processamento de cada tarefa para cada máquina disponível, mas também o tempo necessário para realizar a configuração das máquinas utilizadas. Com o intuito de identificar o melhor método de sequenciamento a ser utilizado, com o auxílio de um *software* de modelagem computacional, analisou-se em um banco de dados fictício o desempenho dos métodos baseado nos resultados de *makespan*, de acordo com ferramentas estatísticas como porcentagem de sucesso e desvio relativo médio. Após os cálculos realizados, observou-se na primeira análise, dos métodos de ordenação SPT e LPT, o melhor desempenho do método heurístico LPT com a consideração do tempo de *setup* das tarefas somado aos tempos de processamento, enquanto que para o método construtivo NEH, a ordenação LPT sem considerar o tempo de *setup* para definição do sequenciamento das tarefas se mostrou mais efetivo. Concluiu-se também a melhor performance dos métodos construtivos quando comparado aos métodos de ordenação SPT e LPT, por resultarem em 100% das instâncias os menores valores de *makespan*.

Palavras-Chave: *Scheduling*. *Setup* Independente. *Flowshop*. Heurísticas. *Makespan*.

ABSTRACT

CORREIA, Amanda da Silva. **Evaluation of Sequencing Rules for Flowshop Problem with Separated Setup.** 2018. 99 p. Work of Conclusion Course (Graduation in Engenharia de Produção) - Federal Technology University – Paraná. Ponta Grossa, 2018.

This work is characterized by the comparative performance study of LPT, SPT and NEH sequencing rules adapted for the resolution of flowshop sequencing tasks, with separate and sequence independent setup time. To carry out the performance analysis, is considered not only the processing time of each job for each machine available, but also the time needed to perform the configuration of the used machines. In order to identify the best sequencing method to be used, the performance of the methods based on the makespan results was analyzed using a computer modeling software, according to statistical tools such as percentage of success and mean relative deviation. After the calculations, we found in the first analysis, of SPT and LPT heuristic methods, the best performance of LPT heuristic method with consideration of the setup time of the tasks added to the processing times. For NEH method, the LPT sequencing rules without considering the setup time for defining the sequencing of the tasks was more effective. It was also concluded the best performance of the constructive methods when compared to SPT and LPT heuristics methods, to result in 100% of problems the lowest makespan values.

Keywords: Scheduling. Sequence independent setup. Flowshop. Heuristics. Makespan.

LISTA DE FIGURAS

LISTA DE GRÁFICOS

LISTA DE QUADROS

LISTA DE SIGLAS

CAPP	*Computer Aided Process Planning*
CDS	Campbell, Dudek e Smith
CI	Circuito Integrado
DRM	Desvio Relativo Médio
F	Fahrenheit
FO	Função Objetivo
LPT	*Longest Processing Time*
m	Número de máquinas disponíveis
MOFSP	Problema de programação *flowshop* multi-objetivo
n	Número de tarefas
n!	Número de sequências possíveis
N&M	Nagano e Moccellin
NEH	Nawaz, Enscore, Ham
NIFS	*No-idle flowshop*
NWFS	*No-wait flowshop*
PCB	Placa de circuito impresso
PCP	Planejamento e Controle da Produção
PMP	Planejamento-mestre da produção
PS	Porcentagem de sucesso
RA	*Rapid acess procedure*
SDBST	*Sequence dependent batch setup times*
SDFST	*Sequence dependent family setup times*
SDGST	*Sequence dependent group setup times*
SIBST	*Sequence independent batch setup times*
SIFST	*Sequence independent family setup times*
SIGST	*Sequence independent group setup times*
SDJST	*Sequence dependent job setup times*
SIJST	*Sequence independent job setup times*
SPT	*Shortest Processing Time*

LISTA DE SÍMBOLOS

C_{max}	*Makespan*
F	*Flowtime*
TT	Atraso Total
I	*Idle Time*
E	Precocidade
$\bar{C}$	Tempo médio de conclusão
C^w	Tempo total de conclusão ponderada
$\bar{C}^w$	Tempo médio de conclusão ponderada
$\bar{F}$	*Flowtime* ponderado
F^w	*Flowtime* total ponderado
T_{max}	Atraso máximo
$\bar{T}$	Atraso médio
T^w	Atraso total ponderado
$\bar{T}^w$	Atraso médio ponderado
E_{max}	Precocidade máxima
$\bar{E}$	Precocidade média
E^w	Precocidade total ponderada
$\bar{E}^w$	Precocidade média ponderada
ctv	Variação do tempo de conclusão
L	*Lateness*
S	Tempo de *setup*

SUMÁRIO

1 INTRODUÇÃO

O propósito do planejamento e controle da produção (PCP) é conseguir que a produção seja realizada de maneira eficaz, bem como produzir os produtos e/ou realizar serviços da maneira que deve. Como gerente, tem-se como uma das suas funções conseguir desempenhar esse planejamento e controle da melhor maneira para as atividades referentes a cada projeto e com os recursos que estão disponíveis.

Além disso, o planejamento e controle da produção possui um papel decisivo para as futuras ações e decisões a serem tomadas pela empresa, e que pode auxiliar no momento de competição no ambiente empresarial, que é resultante da constante e crescente oferta de materiais e de produtos em toda e qualquer parte do mundo.

Como primeiro nível operacional de curto prazo do PCP, define-se a programação da produção (em inglês, *scheduling*), que é encarregada de definir não só a quantidade como também quando comprar, fabricar ou montar cada item necessário para compor os produtos que são fabricados. Outra função que é abrangida neste nível é o sequenciamento de tarefas.

Frente a alguns modelos de programação da produção e restrições do ambiente fabril, a maior parcela do tempo total de processamento pode se referir ao tempo de espera, e não ao processo principal de transformação da produção em questão, a partir disso compreende-se a importância de uma boa programação e melhor utilização dos recursos disponíveis.

Dessa maneira, o *scheduling* tem como intuito alocar as tarefas de fabricação, bem como máquinas e ferramentas, aos processos, sendo que estes estão sujeitos a restrições tais como tempo de liberação, datas de vencimento e carga da máquina.

A alocação de tarefas envolve, no âmbito geral, a resolução de objetivos, que geralmente se tornam conflitantes, o que dificulta plena satisfação com a resolução de um problema de programação de tarefas. O objetivo é agendar as operações em cada máquina de forma a minimizar ou maximizar alguma medida de desempenho, sendo os mais utilizados, o cumprimento das datas de entrega, a minimização de atrasos, do tempo de ociosidade, do tempo total de programação e entre outros.

O sequenciamento de tarefas pode ainda se fazer necessário ou não nos sistemas de produção existentes, entre os mais conhecidos, o *jobshop, openshop* e *flowshop*. Esses sistemas regem por um definição de ordem de execução das tarefas

e quantidade de estágios que as tarefas precisam percorrer.

O sistema de produção *flowshop* é uma área de pesquisa que vem sendo bem explorada por possuir um vasto campo não só de estudo, mas principalmente de aplicação na indústria. Um problema *flowshop* é definido com um mesmo fluxo que todas as tarefas precisam passar nas **m** máquinas disponíveis, e se considerado como *flowshop* permutacional, existe **n!** sequências possíveis.

Problemas de agendamento de *flowshop* com tempos de configurações (*setup*) surgem de forma natural em muitas situações de indústrias, sendo que, em cada indústria e seu segmento de produção, o escopo do planejamento de todo o processo pode variar.

Dentro de situações de sequenciamento de tarefas, pode-se, ainda, analisar o tempo de *setup* necessário para cada tarefa em cada máquina. O *setup* envolve operações que podem ser ditas como não ligadas diretamente ao processo produtivo mas que se tornam necessárias, já que incluem tanto limpeza, como substituição de ferramentas de processamento, transporte de operações de uma máquina a outra, liberação de peças para as máquinas e entre outros.

Esse tempo, não é levado em consideração em sistemas de produção em que o *setup* é consideravelmente pequeno quando comparado ao tempo de processamento, entretanto em diversos casos dentro de um contexto industrial, não é ideal que essa informação seja simplesmente ignorada ou incluída ao tempo de processamento das tarefas como um desvio médio, frente a essa situação, trabalha-se com o *setup* separado.

Existem algumas classificações possíveis para esses tempos de configuração, como *setup* dependente e independente, quando o tempo de *setup* depende ou não da tarefa sucessora, bem como, e principalmente, da que está sendo processada no momento. E também, *setup* antecipado ou não antecipado, determinando o tempo em que pode ser realizado, se a configuração pode ser realizada durante o tempo livre da máquina mesmo que a tarefa ainda esteja em processo na máquina anterior ou não.

Na literatura, encontra-se exemplos simples cujo desempenho do sistema de produção é afetado pela eficiência da sequência de operação na máquina e pela

eficácia da programação de configuração das máquinas e, portanto, essas informações precisam ser explicitamente consideradas quando o problema é modelado.

1.1 PROBLEMA

Nos tipos de *scheduling* mais utilizados há focalização de um ambiente produtivo que não considera o tempo de preparação da máquina para realização das tarefas do processo, ou quando considerado, não é levado em consideração em sua ampla abrangência. Dessa maneira o presente estudo tem como intuito verificar os impactos gerados na linha de produção a partir do sequenciamento das tarefas em um problema *flowshop* com *setup* separado.

1.2 JUSTIFICATIVA

Ao que se encontra na literatura, não podemos dizer que se encaixa perfeitamente nas situações reais encontradas nas indústrias, já que o enfoque é dado para os problemas de *flowshop* com um único objetivo sem considerar o tempo de preparação das máquinas.

Dessa maneira, visto como necessidade e realidade de muitas indústrias, o tempo de *setup* separado, este trabalho justifica-se não só pela carência de análise profunda do assunto na literatura quando comparado a outros problemas, mas principalmente pela melhoria que uma proposta de sequenciamento de tarefas eficaz pode resultar para qualquer indústria.

Os resultados de um sequenciamento de tarefas inadequado têm influência não só na motivação, como na atitude da força de trabalho e na produtividade geral, e conclui-se a necessidade de uma apropriada alocação de tarefas dentro de uma empresa. Isso torna um fator essencial não só pela demanda de trabalho e/ou pelo respeito às restrições de escalonamento das atividades de cada processo, mas também por influenciar nos custos de produção e no melhor aproveitamento dos recursos disponíveis.

1.3 OBJETIVO GERAL

No presente trabalho, o objetivo principal é identificar entre as regras de ordenações LPT, SPT e NEH, e suas variações, a melhor configuração em termos de minimização do tempo total de programação, em problemas *flowshop* com *setup* separado e independente da sequência.

1.4 OBJETIVOS ESPECÍFICOS

Como objetivos específicos, definimos como:

- analisar os métodos de ordenação LPT, SPT e NEH para o problema com *setup*;
- verificar e comparar o desempenho desses métodos a serem utilizados;
- modelar um cenário que possibilite o melhor ambiente produtivo.

1.5 DELIMITAÇÃO DO TEMA

O trabalho em questão tem como intuito abranger uma das áreas da Engenharia de Produção, o Planejamento e Controle da Produção (PCP), que engloba dentre vários temas, o *scheduling*, que tem como objetivo principal definir e analisar o sequenciamento das tarefas dos processos industriais.

Dentre diversos possíveis assuntos que podem ser abordados neste quesito, tem-se como objetivo geral, analisar problemas e métodos heurísticos que tratam de um sequenciamento de tarefas de uma linha de produção com tempo de *setup* separado e independente da sequência.

Desta forma, pode-se afirmar que o presente trabalho deve apresentar uma análise dos métodos heurísticos adequados para a resolução de problemas de *scheduling* com *setup* separado e independente.

1.6 ESTRUTURA DO TRABALHO

Este trabalho está dividido em cinco capítulos. O Capítulo 2 apresenta uma

revisão da literatura dos principais temas relacionados ao problema descrito anteriormente, o Capítulo 3 apresenta a metodologia aplicada para cumprimento dos objetivos deste trabalho de conclusão de curso enquanto o Capítulo 4 expõe os resultados encontrados durante a execução deste estudo, bem como gera as discussões, e por fim, o Capítulo 5 tem como intuito concluir a efetividade do trabalho e realizar considerações finais.

2 REFERENCIAL TEÓRICO

Esta seção do presente trabalho possui sete tópicos e tem como propósito esclarecer cada assunto pertinente ao tema principal do trabalho de conclusão de curso por meio da revisão de literatura realizada. Os tópicos apresentados seguem como linha de raciocínio a delimitação do tema deste estudo, ou seja, inicia-se com a revisão da literatura de uma das áreas da engenharia de produção, o planejamento de controle da produção, que ao longo do referencial teórico, vai sendo aprofundado de acordo com o enfoque deste trabalho.

2.1 PLANEJAMENTO E CONTROLE DA PRODUÇÃO

Slack *et al.* (1999, p.230) afirmam que, o propósito do planejamento e controle da produção (PCP) é conseguir que a produção seja realizada de maneira eficaz, bem como produzir os produtos e/ou serviços da maneira que deve. Ademais, os autores acreditam que os gerentes de produção desempenham funções no sistema de entrada, transformação e saída para realização das atividades referentes a cada projeto, sendo estas, a forma e a natureza como isso ocorrerá e os recursos que estão disponíveis. Entretanto, não se pode afirmar que isto é uma preocupação que ocorre no andamento da produção no dia-a-dia.

A constante e crescente oferta, tanto de materiais quanto de produtos, em toda e qualquer parte do mundo define para Russomano (2000, p.47) um momento de competição entre as empresas, com desafios categóricos. E o planejamento e controle da produção possui um papel determinante para as futuras ações e decisões a serem tomadas, ao auxiliar na tomada de decisão diante das dificuldades que podem ser enfrentadas pela empresa.

De acordo com Slack, Chambers, Jhonston (2009) e Côrrea e Gianesi (2012), existem seis desafios e limitações para a administração da produção, e como o PCP pode auxiliar na prevenção e precaução das problemáticas, que estão descritas a seguir:

- Custos: Questão limitante tanto para a empresa quanto para o cliente. O estabelecimento de níveis adequados de estoque, planejamento de compras e definição de prioridades, contribuem para a redução dos custos nas empresas;

- Qualidade: A oferta de produtos livres de defeitos, dentro das conformidades e limites de tolerâncias, possui relação indireta com a administração da produção, e isso acontece por meio dos registros existentes e sistemas de rastreabilidade para identificar as causas raízes que geram defeitos;
- Capacidade: As atividades das empresas são realizadas de acordo com a capacidade projetada para determinadas situações. Isso pode ser definido diante da utilização do planejamento e controle da empresa, que conciliam duas entidades, o conjunto de demandas e a capacidade de fornecimento;
- Tempo: Existe um intervalo definido para que produtos e serviços sejam produzidos, sendo um fator relevante para o cliente em questão de velocidade na entrega. Essa questão possui relações diretas com os níveis de estoque, a formação de filas, bem como no tempo de aguardo nos processamentos. O PCP foca em uma gestão bem realizada que permite a sincronização das diversas etapas e processos inclusos;
- Confiabilidade: Ainda relacionado com o item anterior, é função da administração da produção dispor de informações e mecanismos de suporte para cumprir as promessas realizadas ao cliente, na sua maioria, relacionadas à entrega;
- Flexibilidade: Essa característica dentro de um sistema de produção, significa estar apto à reações rápidas às mudanças não planejadas. O suporte para determinadas situações não é unicamente da administração da produção, mas em seu âmbito, sistemas de informações mais ágeis e adequados direcionam bem certas situações. Os recursos estruturais são os músculos da flexibilidade produtiva, mas o sistema de administração da produção é seu sistema nervoso (Corrêa e Gianesi 2012, p.14).

Diante desses desafios, Yang, Arndt e Lanza (2016) acrescentam o fato que, além de um PCP ser flexível, este precisa ser capaz de modelar e simular com dinâmica as informações para que se possa analisar os potenciais riscos que são causados por diversas incertezas da produção.

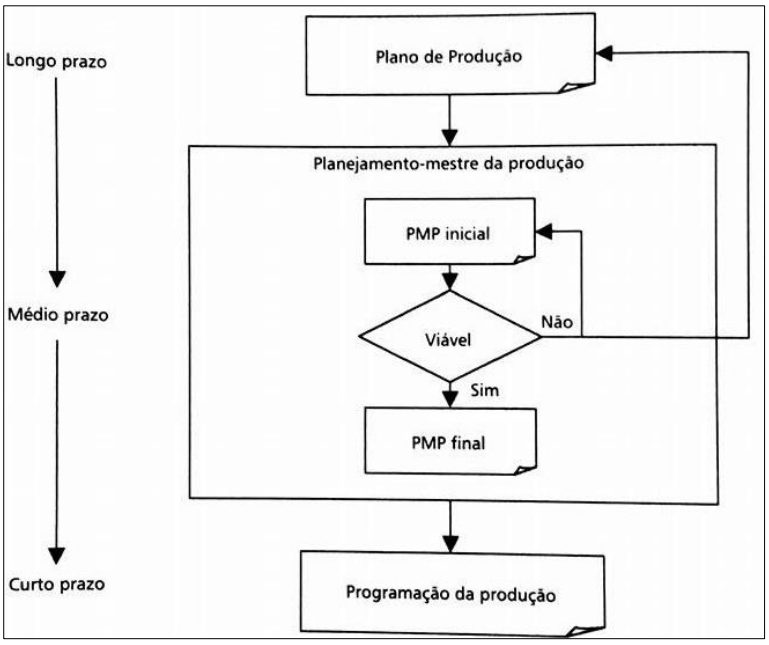
Em alguns sistemas de produção, como a produção contínua com ciclos tecnológicos limitados - que acontece com numerosa variedade de componentes a produzir, que se encaixam nas várias versões do produto final realizado de acordo com a demanda do mercado e necessidade de produção em alta escala devido à

distribuição mundial - a exigência pela flexibilidade é alta, além do baixo custo de produção e produção rápida. Dessa maneira, essas instalações participam de um plano de produção mais amplo (SAVINO *et al*., 2010).

Para Arnold (1996), o passo sucessivo após o planejamento de produção é preparar um Programa Mestre de Produção, pois se trata de uma importante ferramenta de planejamento e forma a base de comunicação entre vendas e manufatura. Como complemento, Tubino (2009) contribui com a afirmação que um planejamento-mestre da produção tem como intuito desdobrar os planos estratégicos de longo prazo em planos específicos, este deve ser feito após o plano de produção, para poder guiar as ações a serem tomadas.

Pela Figura 1, de acordo com Tubino (2009, p.51), é possível a visualização de que o Planejamento-Mestre da Produção (PMP) realiza a conexão entre o plano de produção (planejamento estratégico) e as atividades consideradas operacionais (programação da produção). O PMP possui duas funções básicas: realizar a análise e validação da capacidade de médio prazo do sistema produtivo que está sendo estudado e implementar a tática que foi escolhida para um próximo período, como identificar as quantidades dos produtos que foram acabados e que devem ser produzidos novamente.

Figura 1 - Programação da produção e horizontes de planejamento



Fonte: Tubino (2009).

Slack *et al*. (1999) adicionam às observações que, o longo prazo tem ênfase voltada para o que se deve fazer, os recursos que serão necessários e, portanto, quais objetivos serão alcançados. A médio prazo, ocorre a desagregação do plano, o que pode resultar em reformulação do plano inicial. No curto prazo são realizadas as atividades de controle, que também pode definir a programação da produção, tópico a ser apresentado a seguir.

2.2 PROGRAMAÇÃO DA PRODUÇÃO

Segundo Tubino (2009) o primeiro nível operacional de curto prazo, dentro da hierarquia do planejamento e controle de produção, é a programação. A programação da produção está encarregada de definir quanto e quando comprar, fabricar ou montar cada item necessário à composição dos produtos acabados com base no plano mestre de produção e registros de controle de estoques.

De antemão, o mesmo autor complementa que dessa maneira, o sequenciamento de tarefas é abrangido pelo âmbito geral da programação da produção, e contribui para seu melhor desenvolvimento. Na literatura é possível encontrar outros termos que definem o sequenciamento, tal como alocação de tarefas e *scheduling*.

Ao relacionar com as limitações citadas no tópico anterior, Yang, Arndt e Lanza (2016) declaram que no setor fabril, um sistema de produção gasta até 90% do tempo total de processamento em espera e apenas 10% no processo principal, o de transformação. Existe, portanto, um *link* entre os dados dispostos e a definição de objetivo da programação para Arnold (1996), que é ter como compromisso os prazos de entregar e melhor utilização dos recursos.

Maenhout e Vanhoucke (2015) complementam ao dizer que, programar projetos sob restrições de recursos é trabalhar com suposições feitas com relação a disponibilidade destes recursos.

É de responsabilidade do planejamento do processo a seleção da sequência de processos de fabricação de acordo com a especificação de projeto do produto, já o *scheduling* alocará então os recursos de fabricação, bem como máquinas, tarefas e ferramentas, aos processos, sendo que estes estão sujeitos a restrições tais como tempo de liberação, datas de vencimento e carga da máquina (ZHANG; WONG,

2016).

Os problemas de programação envolvem, em geral, a perseguição de objetivos múltiplos e frequentemente conflitantes (FRENCH, 1982), com isso, dificilmente uma programação de tarefas será totalmente satisfatória. Como principais objetivos, pode-se citar, o cumprimento das datas de entrega, minimização na quantidade de produtos/serviços atrasados, minimizar o período total de programação, o tempo de ociosidade e custo de inventário.

Os objetivos estão associados aos custos de produção e, portanto, podem também ser expressos em função deles, conclui Santos e França (1995), com os objetivos. Os custos que se tornam mais relevantes são os de preparação de máquina, de ociosidade de máquina, custos de estoques e penalidade por atrasos.

Saygin e Kilic (1999) afirmam, historicamente, que a maioria dos estudos relatados nas décadas anteriores enfatizavam o desalinhamento entre as programações geradas no nível de planejamento e o chão de fábrica, em sua maioria. Perante estes relatos, se viu necessário um sistema integrado de planejamento e programação de processos, a fim de tornar os planos de processos mais congruentes com a realidade. Portanto, a prática atual preza pelo planejamento do processo e, em seguida, a realização da programação.

O planejamento do processo fornece a entrada fundamental para a programação, ou seja, o planejamento do processo enfatiza os requisitos tecnológicos de uma tarefa, enquanto a programação envolve os aspectos de cronometragem.

O objetivo é agendar as operações de forma a minimizar alguma medida de desempenho, como o tempo de preparação, o tempo total de conclusão, o atraso máximo, o atraso total, o atraso ponderado e a soma ponderada de antecedência e atraso, entre outros (CHENG; GUPTA; WANG, 2009).

Resultados de uma programação de tarefas inadequada tem influência não só na motivação, como na atitude da força de trabalho e a produtividade geral. Então, conclui-se a necessidade de uma apropriada programação de atividades dentro de uma empresa, já que se torna um motor para o padrão de demanda de trabalho, segundo afirmações de Maenhout e Vanhoucke (2015).

Pinedo (2005) ressalta que o planejamento e alocação de tarefas auxiliam nas tomadas de decisões, que participam de maneira importante e imprescindível na

aquisição e produção em qualquer área e devem ser realizadas de maneira que otimize os objetivos e alcance as metas pré-definidas.

Algumas ferramentas eram utilizadas para o auxílio da seleção das sequências de operação de produção de todos as **n** tarefas e alocar as operações às máquinas com o objetivo de otimizar algum critério de desempenho. Zacarrelli (1987) comentou que os grafos, de Gantt e de Montagem, ganhavam destaque pelo auxílio e facilidade de utilização, pois são conceitos simples que utilizam de métodos gráficos e tabelas.

Em meados de 1903 o gráfico de Gantt foi adaptado e tornou-se mais claro e compreensivo para que sua implementação na indústria ocorresse, como forma de planejamento, organização e controle da produção. O gráfico de Gantt lista na vertical os fatores de produção, pelos quais são distribuídas as tarefas, sendo máquinas, operários, grupos de trabalhadores ou de acordo com a necessidade da empresa e como complemento, na linha horizontal, as tarefas que devem ser feitas são dispostas de acordo com o intervalo de tempo.

Outras literaturas, como Tubino (2009) e Ullman (1975), para que a programação da produção possa exercer suas funções dentro de qualquer empresa, existe dependência de algumas técnicas matemáticas e métodos heurísticos, devido a programação de tarefas se caracterizar por um problema de NP-completo, sendo por meio desses métodos possível o alcance de uma solução ótima ou de boa qualidade.

Ademais, segundo Garey e Johnson (1979), o grau de dificuldade de um problema de programação de tarefas depende e varia tanto de acordo com a estrutura de um sistema multiprocessador, do número de processadores paralelos, da uniformidade do tempo de computação da tarefa, quanto do tempo de comunicação entre as tarefas.

Outrossim, Maccarthy e Liu (1993) salientam que as restrições tecnológicas sobre os trabalhos quanto aos objetivos de programação devem ser especificados e são determinadas, principalmente, pelo padrão de fluxo das tarefas nas máquinas.

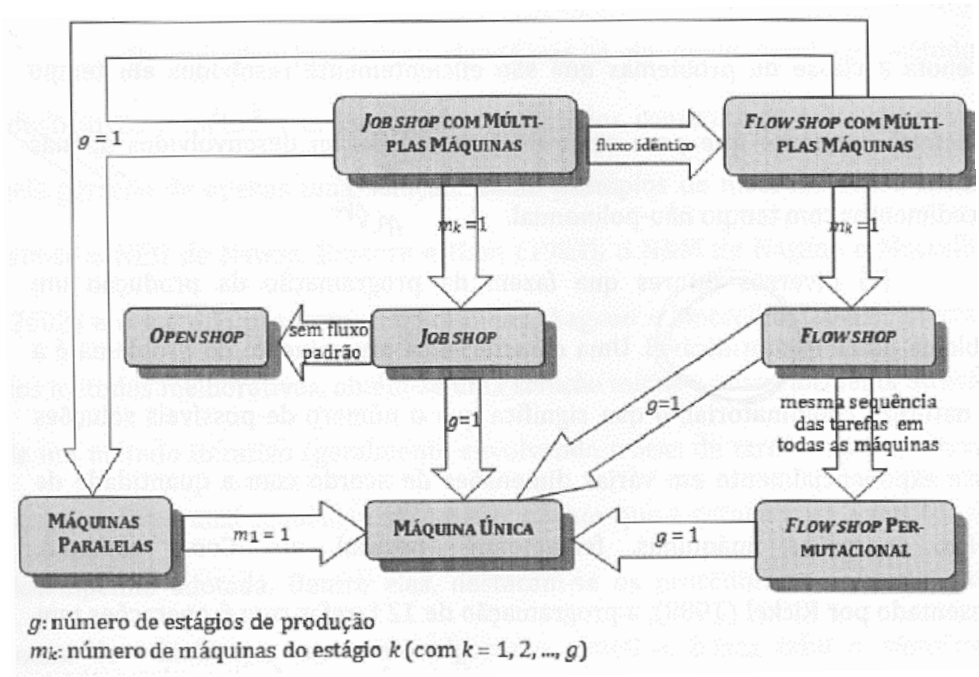
A classificação dos tipos de sequenciamento pode ocorrer conforme explicado a seguir, segundo MacCarthy e Liu (1993) e como ilustrado pela Figura 2.

- *Jobshop*: as tarefas possuem um próprio padrão de fluxo, individualmente, e/ou

o percurso pelas máquinas.

- *Flowshop*: cada tarefa possui um padrão de fluxo inalterável.
- *Openshop*: as tarefas não possuem nenhum tipo de padrão no fluxo que seja especificado.
- *Flowshop* Permutacional: esta classificação é caracterizada pela ordem de processamento em todas as máquinas estritamente igual.
- Máquina única: neste caso, somente uma máquina está disponível para uso.

Figura 2 - Relação entre as classes de problemas de programação



Fonte: Adaptação realizada por Moccelin e Nagano (2003) da figura apresentada por MacCarthy e Liu (1993).

O próximo item aborda o problema de programação foco deste trabalho de pesquisa, a programação *flowshop*.

2.3 *FLOWSHOP*

O sistema de produção *flowshop* é uma área de pesquisa importante no ramo de produção, não só teoricamente, no campo de estudo, mas também devido a sua aplicação, tanto em problemas industriais quanto na vida real (YENISEY; YAGMAHAN, 2014).

Como contextualização na parte histórica, Cheng, Gupta e Wang (2009), Vallada, Ruiz e Framinan (2015), e Riahi *et al*.(2017) enfatizam que o pioneiro na área de pesquisa da permutação *flowshop* foi Johnson (1954), que contribuiu com um problema de duas máquinas com o critério de minimização do tempo total de processamento (*makespan*) e, desde então, esse campo de pesquisa tem se tornado muito ativo.

Xu *et al*. (2017) trazem como conhecimento o fato de que muitos processos de produção, como na indústria siderúrgica, petroquímica e mecânica podem ser modelados com o problema de programação de *flowshop* permutacional.

Um problema quando considerado *flowshop* determina que exista um conjunto de máquinas que estão ordenadas, assim sendo, a primeira tarefa é executada na primeira máquina, posteriormente na segunda máquina e assim até a máquina **m** (GAREY; JOHNSON; SETHI, 1976).

Por conseguinte, Taillard (1990) e Buzzo e Moccellin (2000) descrevem um problema de programação *flowshop* como um problema que define um mesmo fluxo de execução para todas as tarefas nas **m** máquinas disponíveis, e um problema se torna permutacional quando a determinação desta sequência se encontra dentro as **n!** sequências possíveis das tarefas. Devido a sua natureza de problema combinatorial, um problema de *flowshop* é considerado como NP-completo.

Se dentro de um processo produtivo, existem dez tarefas, 3.628.800 soluções de sequenciamento existem para serem estudadas, e dentre elas, uma pode ser considerada como um ótimo resultado.

Ainda na mesma linha de raciocínio de Taillard (1990), pode-se elencar sete hipóteses consideradas usuais em questão de programação de tarefas *flowshop* com tempos de *setup* agregados aos tempos de processamento e fabricação de apenas uma unidade em cada tarefa ou lotes que não podem ser fracionados, descritas a seguir:

1. As máquinas estão disponíveis continuamente, sem interrupções;
2. Em cada máquina apenas uma tarefa pode ser processada por vez;
3. As tarefas só podem ser processadas por uma máquina de cada vez;
4. Existe predeterminação dos tempos de processamento das tarefas em

todas as máquinas e estes tempos são fixos;

5. Todas as tarefas são liberadas para processamento na primeira máquina no mesmo instante, portanto, a partir desse momento qualquer uma pode ser programada e processada;

6. Cada operação possui um tempo de preparação para cada máquina, e estes estão inclusos em seus tempos de processamento e não dependem da sequência das operações nas máquinas;

7. Não deve ocorrer interrupções das tarefas quando operadas nas diversas máquinas.

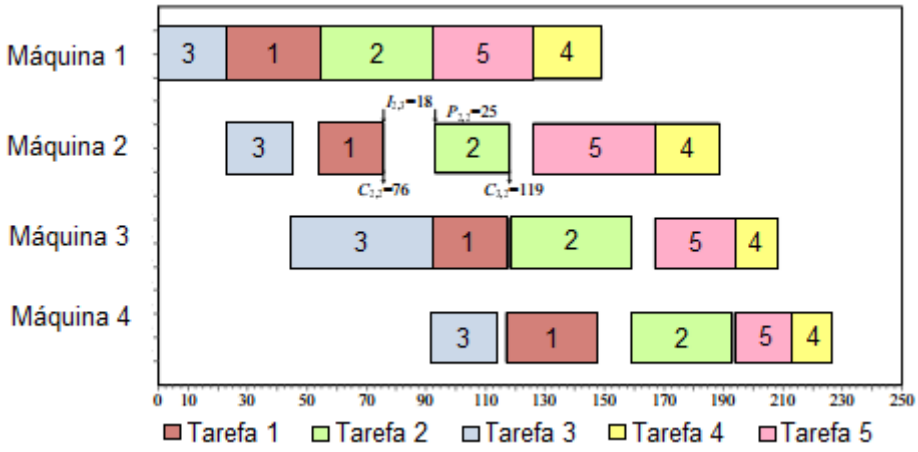
Um exemplo de Gráfico de Gantt para a programação *flowshop* pode ser visto na Figura 3. Neste caso, 5 tarefas são representadas em um fluxo de 4 máquinas, com os tempos de processamento descritos no Quadro 1.

Quadro 1 - Tempos de processamento para o exemplo de um problema de programação *Flowshop*

	Tarefas (n)				
Máquinas (m)	**1**	**2**	**3**	**4**	**5**
1	31	39	23	23	33
2	22	25	22	22	41
3	25	41	47	14	27
4	30	34	22	13	19

Fonte: Adaptado de Ruiz, Vallada e Fernández-martínez (2009).

Figura 3 - Programação de tarefas *flowshop* com 5 tarefas e 4 máquinas



Fonte: Adaptado de Ruiz, Vallada e Fernández-martínez (2009).

Ruiz, Vallada e Fernández-martínez (2009) ilustram pela Figura 3 um problema *flowshop* com base nos tempos do Quadro 1, informando também a variável $C_{i,j}$, que revela o tempo total de processamento da tarefa na posição i na máquina j,

o $I_{i,j}$, tempo de espera entre a execução da tarefa na posição i e da tarefa na posição j e $P_{i,j}$, indicando o tempo de processamento da tarefa i na máquina j.

Assim como qualquer outro tipo de sequenciamento, a permutação *flowshop* tem como intuito encontrar uma ordenação que atenda o objetivo da linha de produção, de minimização ou maximização, como declaram Riahi *et al.* (2017).

Segundo Maccarthy e Liu (1993), a teoria clássica de programação abrange o fato que a estrutura de planejamento e/ou o horizonte temporal geralmente não é considerado, e o transforma em um problema com capacidades ilimitadas. A suposição é implícita não apenas sobre a tomada de decisão ser de curto prazo em um ambiente estático e determinista, mas também o fato de que os pesquisadores possuem conhecimento sobre o aumento da complexidade do problema em caso de consideração do ambiente dinâmico, em situações de falta de mão de obra, matéria prima, falha em máquina e outros.

Arroyo e Pereira (2010) e Riahi *et al.* (2017) ressaltam que o *flowshop* que se concentra em um único objetivo despertou a atenção de muitos pesquisadores até a década de 1980, uma vez que há ajuda de métodos exatos e metaheurísticas.

Branco (2011) conclui que nas últimas décadas, um extenso esforço de pesquisa tem sido dedicado aos problemas de *flowshop* permutacional. Algumas técnicas de programação matemática, tais como programação linear inteiro, técnicas de enumeração do tipo *branch-and-bound* têm sido empregadas para se obter a solução ótima. Entretanto, tais técnicas não são eficientes em termos computacionais em problemas de médio e grande porte. Assim, muitos métodos heurísticos têm sido propostos, os quais, de maneira geral, podem ser classificados em dois grupos: métodos construtivos e metaheurísticos.

Vários tipos de metaheurísticas foram propostas para resolver problemas de sequenciamento de tarefas *flowshop*. Estes incluem algoritmos genéticos (Reeves (1995), Chen, Vempati e Aljaber (1995), Murata, Ishibuchi e Tanaka (1996), Jaszkiewicz (2002), Ishibuchi, Yoshida e Murata (2003), Azadeh *et al*. (2015), Jung, Woo e Kim (2017), Ramesh, Kannan e Baskar (2012), Ghodratnama, Jolai e Tavakkoli-moghaddam (2015), e Memari, Rahim e Ahmad (2015)), busca tabu (Al-Anzi e Allahverdi (2006) e Gao, Chen e Deng (2013)), recozimento simulado (Al-Anzi e Allahverdi (2006)), enxame de partículas (Al-Anzi e Allahverdi (2009), Kennedy e

Eberhart (1995), Chih *et al.* (2014), Chih (2015) e Hatami, Ruiz e Andrés-romano (2015)), colônia de formigas (Yagmahan e Yenisey (2010)), algoritmos iterados (Geiger (2011)) e outros.

Já para métodos heurísticos construtivos, tem-se como propostas, o método N&M introduzido por Nagano e Moccellin (2002), o algoritmo IG desenvolvido por Ruiz e Stützle (2007), a heurística construtiva NEH (Nawaz, Enscore e Ham (1983)), as baseadas na abordagem de McCormick (Leisten (1990)). A heurística PF_NEH baseada em PF a abordagem foi apresentada e melhorada pelo procedimento NEH pelos autores Liang, Pan e Chen (2010). Tasgetiren *et al.* (2017) propuseram a heurística PFT_NEH semelhante a contribuição dos autores Ruiz e Stützle (2007).

2.3.1 Algoritmo NEH

A busca por uma solução para o problema *flowshop*, com o intuito de encontrar uma sequência ótima ou quase ótima, produziu técnicas de solução exata e aproximada, concluem Nawaz, Enscore e Ham (1983). Para esses autores, as técnicas exatas resolvem o problema em princípio, mas, na maioria dos casos, o tempo de computação e a memória necessária para acompanhar os cálculos é proibitivo mesmo para problemas pequenos. Por outro lado, algoritmos heurísticos, embora não forneçam necessariamente a solução ideal para o problema, são para a maior parte, uma maneira eficiente e econômica de obter uma boa solução para um problema de sequenciamento *flowshop*.

Em uma situação de *flowshop*, em que todos os trabalhos devem passar por todas as máquinas na mesma ordem, determinados algoritmos heurísticos propõem que os trabalhos com maior tempo total do processo recebam maior prioridade do que os trabalhos com menos tempo de processamento, afirmam Nawaz, Enscore e Ham (1983).

Com base nessa premissa, o algoritmo NEH foi apresentado por Nawaz, Enscore e Ham (1983), que garantiu sequências com boas soluções em comparação com outras heurísticas existentes. Taillard (1990) também comparou a qualidade e a complexidade das soluções do método construtivo NEH com as heurísticas Palmer, proposta por Palmer (1965), que tem como sugestão um índice de ordem de declive para sequenciar **n** tarefas em **m** máquinas, com base nos tempos de processamento

de tarefas em máquinas diferentes, o método Gupta sugerida pelo autor Gupta (1971), um algoritmo semelhante ao de Palmer, exceto pelo fato de definir o índice de inclinação de uma maneira ligeiramente diferente, o algoritmo CDS, proposto por Campbell, Dudek e Smith (1970), que é uma generalização heurística do algoritmo de Johnson, sendo que este procedimento gera um conjunto de **m**-1 problemas de duas máquinas artificiais a partir do problema original da **m**-máquina, cada um dos quais é então resolvido pelo algoritmo de Johnson, e comparado também com a heurística RA, sugerida por Dannenbring (1977).

A comparação realizada por Taillard (1990) entre essas heurísticas está representada por meio do Quadro 2. A complexidade inclui o cálculo do *makespan* enquanto a qualidade das soluções é dada em porcentagem acima da média dos ótimos valores (para os problemas de 10 tarefas da máquina) ou do *makespan* obtido após 1000 iterações de heurísticas tabus.

Quadro 2 - Comparação entre heurísticas clássicas

	Complexidade	**Qualidade**					
Problemas	-	500	100	100	100	50	50
Tarefas	**n**	9	10	20	20	40	50
Máquina	**m**	10	10	10	20	10	10
Gupta	$n \log(n) + nm$	13.4	12.8	19.6	18.8	18.9	17.1
Johnson	$n \log(n) + nm$	10.9	11.8	16.7	16.8	17.3	16.3
RA	$n \log(n) + nm$	8.5	9.1	12.5	13.4	13.5	11.2
Palmer	$n \log(n) + nm$	8.3	9.0	13.3	12.5	10.9	10.7
CDS	$nm^2 + mn \log(n)$	4.5	5.2	9.7	8.6	9.9	9.3
NEH	n^2m	2.1	2.2	3.9	3.8	2.6	2.1

Fonte: Adaptado de Taillard (1990).

Taillard (1990) conclui, portanto, que o algoritmo NEH parece ser a melhor heurística polinomial na prática. As heurísticas RA ou Palmer também podem ser úteis quando tempos de computação curtos são necessários. Framinan, Leisten e Rajendran (2003) também confirmaram a eficácia da regra de prioridade da heurística NEH ao testá-lo com 177 sequências iniciais diferentes com o critério *makespan*, *idletime* e *flowtime*, respectivamente.

Nawaz, Enscore e Ham (1983) que sugeriram este método de satisfatória eficácia, descreveram os passos necessários para realiza-lo:

1. Ordenar as **n** tarefas em ordem descrescente de tempos de processamento nas máquinas.

2. Selecionar as duas primeiras tarefas (k=1 e k=2) e programá-las a fim de minimizar o *makespan* parcial.
3. Para k=3 até **n** fazer:
   -Inserir a k-ésima tarefa na sequência, entre os **n!** possíveis, que minimiza a parcial *makespan*.

A medida que os passos são seguidos, Nawaz, Enscore e Ham (1983) garantem a realização das duas fases desse método, sendo a primeira descrita pelo primeiro passo, e a segunda fase a partir do segundo passo até que todas as tarefas sejam sequenciadas, resultando em um solução tão boa ou próxima a solução ótima.

2.4 FUNÇÕES OBJETIVOS

Ravi, Tunçel e Huang (2016) comentam que vários critérios de desempenho podem ser usados para programar **n** tarefas independentes em máquinas paralelas idênticas. Em aplicações em que o inventário de material em processo é altamente valorizado (resultando em custos de estoque potencialmente em processo enorme), ou em aplicações onde o tempo total gasto no sistema deve ser minimizado, uma função objetivo fundamental seria a minimização do tempo total de fluxo (*flowtime*), essa função objetivo (FO) pode ser minimizada com o método heurístico *Shortest Processing Time* (SPT). Outra função objetivo fundamental é a minimização do *makespan*. A minimização do *makespan* mostrou ser um problema de NP-hard em suas diversas versões estudadas por vários pesquisadores. Graham (1966) examina o problema de minimização de *makespan* para um conjunto de tarefas com uma ordem parcial (restrições de precedência) em um conjunto de máquinas idênticas paralelas. Graham (1969) desenvolveu limites para soluções obtidas pela aplicação da regra *Longest Processing Time* (LPT) ao problema de minimização de *makespan* sem precedência restrições e obteve bons resultados.

Ainda que o problema de programação de *flowshop* seja alvo de muitas pesquisas, Ciavotta, Minella e Ruiz (2013) consideram, também, como objeto de críticas devido ao demasiado embasamento teórico, e a deficiência em se encaixar com as configurações reais de uma indústria.

Na realidade industrial, a maioria dos problemas de programação envolve, naturalmente, múltiplos objetivos, nomeado como Problema de Programação

Flowshop Multi-objetivo (MOFSP) e tem sido alvo de muitas pesquisas no decorrer dos últimos anos e, portanto, várias técnicas de otimização para os MOFSPs foram desenvolvidas (YENISEY; YAGMAHAN, 2014).

Embora em casos multi-objetivos os pesquisadores comumente assumem todos os trabalhos que seguem todos os objetivos, em alguns casos reais, segundo Torkashvand, Naderi e Hosseini (2017), os empregos podem pertencer a diferentes classes ou conjuntos e cada conjunto tem seu próprio objetivo.

Algumas das funções objetivos que podem fazer parte de cada problema de *scheduling*, estão nomeadas, explicadas e abreviadas na sequência:

- *Makespan* (C_{max}): Tempo de conclusão das tarefas;
- *Flowtime* (F): Tempo total do fluxo das tarefas;
- Atraso Total (TT): Tempo total de atraso das tarefas;
- *Idle Time* (I): Tempo de ociosidade da máquina;
- Precocidade (E): Adiantamento de execução das tarefas.
- *Lateness* (L): Tempo de *delay* para conclusão das tarefas;

Outros objetivos podem ser utilizados, segundo os autores Yenisey e Yagmahan (2014), e que normalmente são derivados dos citados acima, como, o tempo médio de conclusão ($\bar{C}$), o tempo total de conclusão ponderada (C^w), o tempo médio de conclusão ponderada ($\bar{C}^w$), *flowtime* médio ($\bar{F}$), *flowtime* ponderado total (F^w), atraso máximo (T_{max}), atraso médio ($\bar{T}$), atraso ponderado total (T^w), atraso ponderado médio ($\bar{T}^w$), precocidade máxima (E_{max}), precocidade média ($\bar{E}$), precocidade ponderada total (E^w), precocidade média ponderada ($\bar{E}^w$) e variação do tempo de conclusão (ctv).

Para maior entendimento, apesar da utilização neste presente trabalho da função objetivo de *makespan*, os autores ainda contribuem com as definições detalhadas das funções objetivas, representadas pelas equações a seguir:

- Funções baseadas no tempo de conclusão das tarefas:

  $C(\pi) = \sum_{i=1}^{n} C_i(\pi)$ (1)

  $\bar{C}(\pi) = \left(1/n\right) \cdot \sum_{i=1}^{n} C_i(\pi)$ (2)

  $C^w(\pi) = \sum_{i=1}^{n} w_i C_i(\pi)$ (3)

$$\bar{C}^{w}(\pi) = \left(1/n\right) \cdot \sum_{i=1}^{n} w_i C_i(\pi) \quad (4)$$

- Funções baseadas no tempo de fluxo total das tarefas:

$$F(\pi) = \sum_{i=1}^{n} (C_i(\pi) - r_i) = \sum_{i=1}^{n} F_i \quad (5)$$

$$T(\pi) = \sum_{i=1}^{n} T_i(\pi) \quad (6)$$

$$E(\pi) = \sum_{i=1}^{n} E_i(\pi) \quad (7)$$

$$T_{max}(\pi) = \max_{i} T_i(\pi) \quad (8)$$

$$n_t(\pi) = \sum_{i=1}^{n} U_i(\pi) \quad (9)$$

$$I_j = \max_{i=1,..,n} C_{ij}(\pi) - \sum_{i=1,...,n} p_{ij} \quad (10)$$

$$I = \sum I_j \quad (11)$$

$$ctv = \left(1/n\right) \cdot \sum_{i=1}^{n} \left(C_i(\pi) - \bar{F}(\pi)\right)^2 \quad (12)$$

Como encontrado na literatura, Yenisey e Yagmahan (2014) diferem as funções objetivos dos problemas de *scheduling* citadas anteriormente, e podem ser classificadas em três grupos:

i) Baseados em tempo de conclusão (C_{max}, F, etc.);
ii) Baseados na data de entrega (L, TT, etc.);
iii) Baseados em custos de inventário e utilização.

O primeiro e segundo grupos de objetivos são mais comumente usados nos problemas de programação de *flowshop*, como afirmam Xu *et al*. (2017) e Yenisey e Yagmahan (2014).

A otimização multi-objetivo foi originalmente concebida com a busca de soluções *Pareto-optimal solution* ou soluções eficientes. Tais soluções são não dominadas, isto é, nenhuma outra solução é superior a elas quando todos os objetivos são tomados em consideração (ARROYO; PEREIRA, 2010).

2.5 *NO-IDLE*

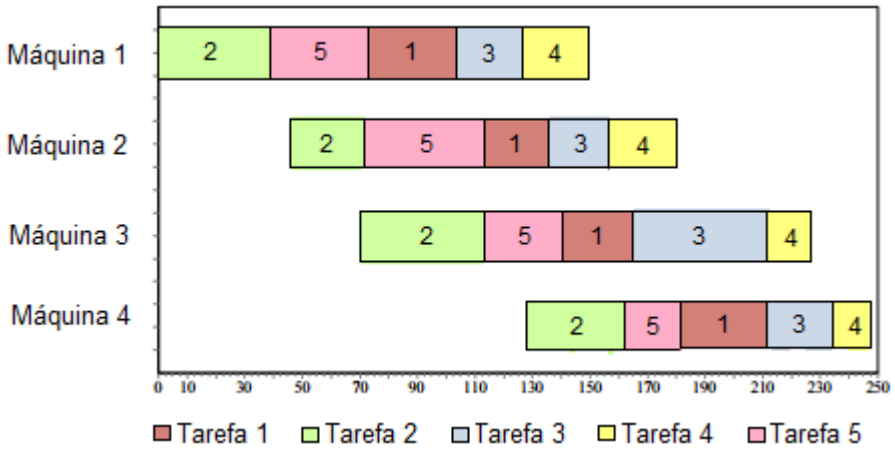
A característica dos problemas de programação de tarefas *no-idle flowshop*

(NIFS), como encontrado nos trabalhos de Saadani, Guinet, Moalla (2003), Branco (2011), Tagestiren *et al*. (2013), e Pan e Ruiz (2014), determinam que na programação estabelecida não deve ser permitido, em nenhuma máquina, um intervalo de tempo inativo entre operações consecutivas. Em outras palavras, quando o processamento for iniciado, as máquinas devem realizar todas as operações sem permitir qualquer tempo de ociosidade até que elas estejam completadas.

Para garantir essa restrição, normalmente, as horas de início das tarefas devem ser adiadas, o que acarreta em um maior tempo total de programação (*makespan*). Na prática, não é possível assumir que todas as máquinas no *flowshop* possuem essa restrição, essa informação não seria realista (PAN; RUIZ, 2014).

Os autores Ruiz, Vallada e Fernández-martínez (2009) ao relatar sobre esse tipo de problema de *flowshop* também o ilustraram, baseado nos tempos de processamento descritos no Quadro 1, como é possível visualizar na Figura 4, com resultado de *makespan* igual a 247 unidades de tempo.

Figura 4 - Representação de um problema de programação *Flowshop No-Idle*



Fonte: Adaptado de Ruiz, Vallada e Fernández-martínez (2009).

Esse tipo de comportamento, como descrito por Branco (2011), ocorre por possibilidade de ser tecnicamente inviável ou não-econômico realizar a paralização das máquinas, ambiente comum em processos que detém de tempo de preparação (*setup times*), custo de utilização da máquina elevado ou devido a restrições tecnológicas, e implica, assim, que desligá-la ou prepará-la mais vezes que o necessário provoca um processo bastante oneroso.

Na literatura, encontram-se a trajetória histórica dos problemas *no-idle*. Sendo os primeiros autores a tratar deste tipo de problema, Adiri e Pohoryles (1982), com a principal contribuição de um algoritmo que obteve a solução ótima para o problema

de minimização do tempo total de fluxo das tarefas. Além disso, apresentaram resultados para problema com mais de duas máquinas, porém em casos especiais de dominância entre elas.

Já os métodos heurísticos para o problema *no-idle flowshop* com **m** máquinas foram inicialmente tratados por Woollam (1986) com o critério de minimização do *makespan*. Na publicação em questão, vários métodos foram selecionados na literatura, incluindo alguns como o conhecido método NEH. Cinco métodos foram adaptados e os resultados computacionais foram obtidos para problemas com até 25 tarefas e 25 máquinas (25 × 25), considerados de pequeno porte. No entanto, para tais casos, o método NEH adaptado para o problema NIFS produziu os melhores resultados.

Ainda no final da década de 90, Cepek, Okada e Vlach (2000) observaram resultados incoerentes no trabalho de Adiri e Pohoryles (1982). Ademais, demonstraram que, em casos de minimização do tempo total de fluxo com duas máquinas, os métodos permutacionais geram melhores soluções (PAN; RUIZ, 2014).

Essa configuração, de trabalho contínuo sem intervalos ociosos, é um ambiente interessante em muitas indústrias. Ambiente que decorre tipicamente das características da tecnologia de processamento utilizada, como temperatura da matéria-prima, metal ou vidro em fusão, ou da necessidade de maximizar o rendimento, a título de informação, uma linha de fundição ou uma loja de montagem da indústria automobilística (SAADANI; GUINET; MOALLA, 2003).

Alguns exemplos industriais desta condição *no-idle* foram levantados pelos autores Branco (2011) e Pan e Ruiz (2014). Os equipamentos utilizados na produção de circuitos integrados por meio de fotolitografia – técnica que cria o molde desejado na lâmica semicondutora ou circuito integrado (CI) - são equipamentos caros e sua marcha lenta é evitada a todo e qualquer custo.

A restrição também ocorre na fundição, comentados pelos autores Tasgetiren *et al*. (2013), que produz blocos de motor de caminhão. A fabricação de fundição requer a produção de moldes de areia e núcleos de areia. Os moldes são preenchidos com metal em fusão e os núcleos evitam que o metal preencha alguns espaços no molde (como o espaço do pistão se a fundição for um bloco do motor). A linha de fundição e algumas máquinas de produção de núcleo devem trabalhar sem tempos ociosos devido a razões técnicas e econômicas.

Alguns outros exemplos surgem em indústrias onde se emprega máquinas

menos dispendiosas, citados pelos autores Pan e Ruiz (2014). No entanto, as máquinas não podem ser interrompidas e reiniciadas. Por exemplo, fornos de rolos de cerâmica consomem grandes quantidades de gás natural quando em operação. O regime mais baixo no funcionamento do motor não é uma opção neste caso, porque leva vários dias para parar e reiniciar o forno devido à inércia térmica notoriamente elevada. Em tais casos, a inatividade deve ser evitada.

Outro exemplo prático é o forno utilizado no processamento de fibra de vidro em que os lotes de vidro são reduzidos a vidro fundido, porque leva três dias para aquecer o forno de volta para a temperatura requerida de 2800 F, o forno deve permanecer, portanto, em operação durante toda a temporada de produção.

2.6 *NO-WAIT*

Os autores Aldowaisan, Allahverdi (2004) mencionam que uma ampla pesquisa tem sido realizada para as variantes do problema de *flowshop* regular com a suposição que existe um espaço infinito de armazenagem entre as máquinas. Entretanto, existem inúmeras situações em que o processamento descontínuo não é permitido por razões tecnológicas, essa variação é chamada de *no-wait flowshop* (NWFP).

Como definição para problemas de ausência de espera, Lin e Ying (2016) sugere que ocorre quando as operações de um processo necessitam ser realizadas continuamente do início ao fim, sem que haja interrupção das tarefas entre máquinas. Dessa maneira, o início de uma operação pode ser atrasada para que seu final coincida com o início da tarefa sucedente.

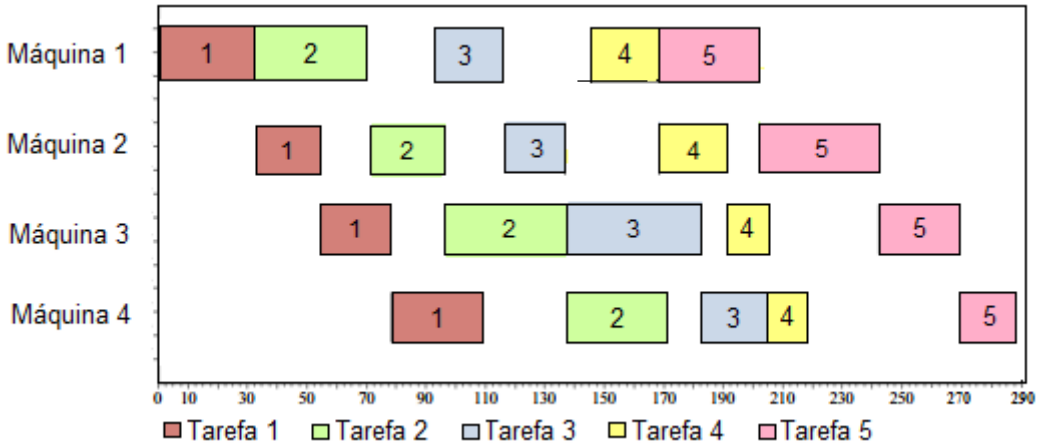
O NWFP se aplica a diversos ramos de indústrias, incluindo indústrias de metal, plástico, química e de alimentos. Por exemplo, no caso de laminagem de aço, o metal aquecido deve passar continuamente por uma sequência de operações antes de ser arrefecido de modo a evitar defeitos na composição do material. Também na indústria de processamento de alimentos, a operação de conservação deve seguir imediatamente a operação de cozimento para garantir a frescura (ALDOWAISAN; ALLAHVERDI, 2004).

Ademais, Framinan e Nagano (2008) e Nagano, Silva e Lorena (2014), agregam ao dizer que aplicações adicionais podem ser encontradas em ambientes

avançados de manufatura, como sistemas de produção *just-in-time*, flexíveis e células robóticas onde as tarefas são processadas continuamente sem espera *in-process*, como também utilizadas na indústria eletrônica, particularmente em placa de circuito impresso (PCB) e fabricação de semicondutores.

A Figura 5 ilustra o exemplo do processo sem interrupções nas consecutivas máquinas, em um problema de 5 tarefas e 4 máquinas referente aos tempos de processamento do Quadro 1, com resultado de *makespan* equivalente a 287 unidades de tempo.

Figura 5 - Exemplo de NWFS



Fonte: Adaptado de Ruiz, Vallada e Fernández-martínez (2009).

Os pioneiros para estudar o tipo de problema NWFS foram os autores van Deman e Baker (1974), que apresentaram em sua pesquisa eles um algoritmo *branch-and-bound* a fim de estabelecer as sequências parciais das operações de acordo com a utilização de limitantes inferiores.

Encontra-se ainda em algumas das principais literaturas, como van Deman e Baker (1974), Adiri e Pohoryles (1982), Rajendran e Chaudhuri (1990), Aldowaisan e Allahverdi (1998), Bertolissi (2000), Allahverdi e Aldowaisan (2000, 2001) e Aldowaisan e Allahverdi (2004), o enfoque para a minimização do *flowtime*.

Adiri e Pohoryles (1982) além de realizar uma pesquisa focada em *no-idle flowshop*, também realizaram para *no-wait flowshop* e definiram propriedades que são relevantes para obter sequências de tarefas ótimas, para problemas de duas máquinas e teoremas para métodos polinomiais para **m** máquinas com séries crescentes ou decrescentes de máquinas dominantes.

Allahverdi e Aldowaisan (2000, 2001) e Aldowaisan e Allahverdi (1998) analisaram o problema para duas e três maquinas com tempos de *setup*. Foram obtidas soluções ótimas para casos especiais estabelecendo uma relação de dominância local, e fornecendo solução heurística para o caso genérico.

Quando o critério de *makespan* é posto em questão, as primeiras pesquisas são referências dos autores Reddi e Ramamoorthy (1972) e Wismer (1972). Posteriormente, Bonney e Gundry (1976), King e Spachis (1980), Gangadharan e Rajendran (1993) e Rajendran (1994).

Bonney e Gundry (1976) utilizaram relações geométricas e extensões de algoritmos para problemas de máquina única. King e Spachis (1980) apresentaram métodos heurísticos com formas diferenciadas de ordenações de tarefas. Gangadharan e Rajendran (1993) desenvolveram dois métodos heurísticos que se mostraram melhores que as heurísticas desenvolvidas por King e Spachis (1980). Ambos foram compostos de duas fases: uma primeira de ordenação inicial das tarefas; e uma segunda de construção da sequência final. Rajendran (1994) apresentou uma extensão do método de Bonney e Gundry (1976) e de King e Spachis (1980).

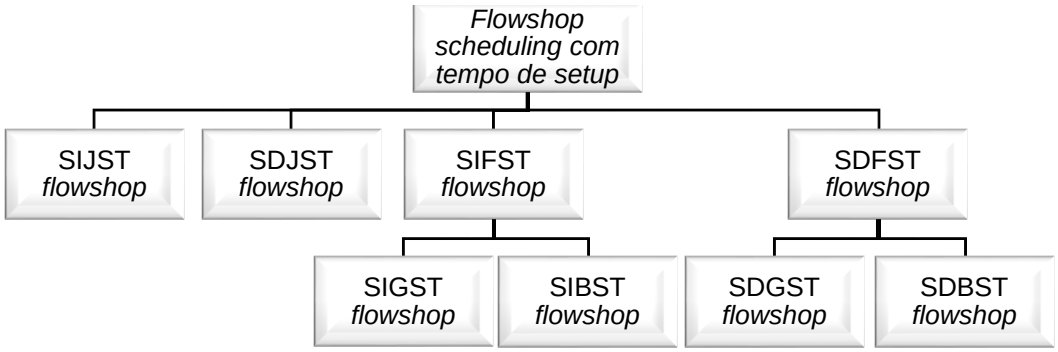
Conforme mencionado acima, várias pesquisas têm considerado somente um dos critérios de avaliação (*makespan* ou *total flow time*). Entretanto, quando é levado em consideração ambos os critérios simultaneamente, o problema torna-se mais realístico e complexo, sendo este conhecido na literatura como análise multi-critério. As primeiras pesquisas com análise multi-critério foram direcionadas inicialmente para os casos de máquina única (NAGAR; HADDOCK; HERAGU, 1995).

2.7 *SETUP*

A natureza dos tempos de *setup* e remoção dos problemas de *flowshop* foram discutidas e sua representação em três campos dos variados problemas de sequenciamento do *flowshop* foi descrita, bem como fornecida uma hierarquia de complexidade de *scheduling* pelos autores Cheng, Gupta e Wang (2009). A Figura 6

ilustra estas possibilidades, com explicação na sequência.

Figura 6 - Classificação de problema de *scheduling* com tempo de *setup*



Fonte: Adaptado de Cheng, Gupta e Wang (2009).

● *Flowshop* com tempos de preparação de tarefas independentes de sequência (SIJST *flowshop*);

● *Flowshop* com tempos de configuração de tarefas dependentes da sequência (SDJST *flowshop*);

● *Flowshop* com tempos de configuração da família de sequência independente (SIFST *flowshop*);

-*Flowshop* com tempos de configuração de grupos independentes de sequências (SIGST *flowshop*);

-*Flowshop* com tempos de configuração de lote independentes de sequência (SIBST *flowshop*);

● *Flowshop* com tempos de configuração de famílias dependentes da sequência (SDFST *flowshop*);

-*Flowshop* com tempo de configuração de grupo dependente de sequência (SDGST *flowshop*);

-*Flowshop* com tempo de configuração de lote dependente da sequência (SDBST *flowshop*).

Cheng, Gupta e Wang (2009) dão continuidade ao assunto ao explicitar que os problemas de agendamento de *flowshop* com os tempos de configuração surgem naturalmente em muitas situações práticas.

Assumir que os tempos de processamento, bem como outros parâmetros de um conjunto de operações, são valores constantes conhecidos antecipadamente é

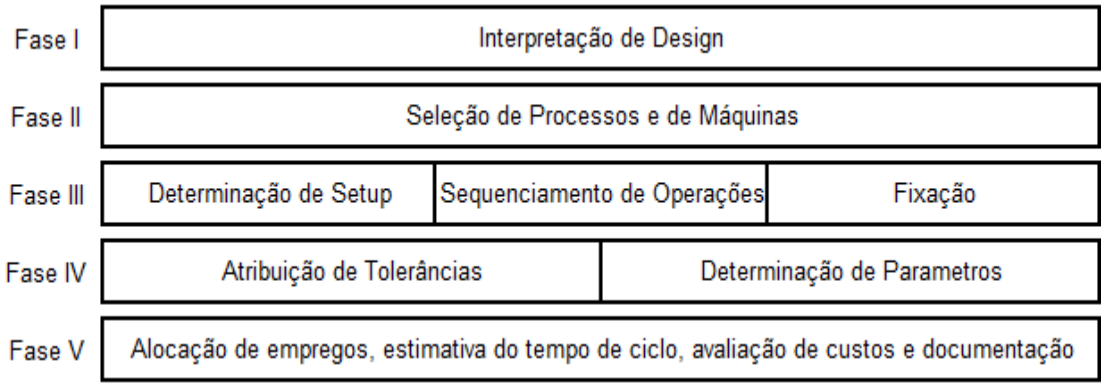
inadequado, visto que para modelar muitos processos industriais modernos um trabalho executado sob a mesma ou quase a mesma condição possui um tempo de processamento variável. Devido a situações como perda de eficiência da máquina durante o processo, eleva-se o tempo de processamento total, e outros exemplo na área de manutenção e limpeza, que caso venha a atrasar, exige um esforço adicional para realizá-lo. Na literatura, o fenômeno é chamado de deterioração do tempo de processamento do trabalho (CHENG; SUN; HE, 2007).

O planejamento de *setup* é uma fase intermediária do planejamento do processo e do planejamento de configuração automatizada e o planejamento de *setup* automático constitui o núcleo de um sistema pelo planejamento de processo assistido por computador (CAPP). No entanto, ainda não há um sistema de planejamento de configuração automática que poderia chegar ao nível de aplicações comerciais. Entretanto, tem sido uma área de pesquisa ativa nas últimas duas décadas, e um elevado número de métodos são relatados na literatura (XU; HUANG; RONG, 2007).

O planejamento do *setup* é proveniente da decomposição do planejamento de processo, caracterizado pela complexidade das tarefas, portanto para que seja desenvolvido requer um acompanhamento completo (XU; HUANG; RONG, 2007).

Xu, Huang e Rong (2007) destacam que, de acordo com as indústrias e seu segmento de produção, o escopo do planejamento de todo o processo pode variar. Todavia, a Figura 7 ilustra um esquema básico da divisão que ocorre dentro do planejamento de processo. Cada fase que está disposta a seguir pode vir a tomar decisões de maneira independente, emite informação para as fases inferiores e recebe resultado das fases superiores.

Figura 7 - Decomposição do Planejamento de Processo



Fonte: Adaptação realizada da figura apresentada por XU; HUANG; RONG (2007).

O termo - planejamento de instalação - amplamente utilizado na literatura de pesquisa de planejamento de processo, refere-se à Fase III no esquema ilustrado. No entanto, não há um padrão bem reconhecido quanto ao seu escopo.

Na concepção dos autores Ciavotta, Minella e Ruiz (2013), o *setup* envolve operações que não estão ligadas diretamente ao processo produtivo, ou seja, são operações não produtivas, mas que necessitam ser realizadas nas máquinas. São atividades que incluem, mas não se limitam a, limpeza, substituição de ferramentas de processamento, transporte de tarefas de uma máquina para a próxima máquina, fixação e liberação de peças para as máquinas. Normalmente, o tempo de *setup* não é levado em consideração, mas na maioria do contexto industrial, não é possível nem ideal que esta informação seja ignorada.

Os autores Ciavotta, Minella e Ruiz (2013) ainda contribuem ao declarar que o *setup* pode ser classificado em duas categorias principais, sendo *setup* independente e o *setup* dependente. Dentro da primeira classificação se encaixa o tempo de *setup* que depende apenas da operação que está a ser processada e pode ser apenas considerado junto com o tempo de processamento da tarefa, enquanto que na segunda classificação, o tempo de *setup* depende não só da tarefa sucessora, mas principalmente da que está sendo processada no momento, tornando um processo mais complexo e, portanto, a possibilidade de descrever diversos cenários operacionais.

Além disso, as configurações podem ser antecipadas ou não antecipadas. No primeiro caso, as configurações podem ser executadas assim que a máquina estiver livre e antes da próxima tarefa na sequência ser carregado, conforme afirmam Ciavotta, Minella e Ruiz (2013).

Huang, Cai e Zhang (2010) comentam que são poucas as pesquisas que consideram o tempo de configuração da máquina (*setup*), e quanto esse tempo é considerado, ignoram-se a disponibilidade dos trabalhadores qualificados para a realização desta atividade. Entretanto, esta informação se torna importante já que os operadores precisam ser atribuídos de uma máquina para outra, e diante dessas situações, o planejamento da produção se tornou uma atividade complexa, especialmente quando os *setups* das operações são dependentes da sequência de tarefas, além dos variados tipos de recursos que devem ser atribuídos

simultaneamente.

Para demonstrar a importância do tempo de *setup* e o porquê de levá-lo em consideração na decisão do sequenciamento de tarefas, os autores Huang, Cai e Zhang (2010) desenvolveram um exemplo com quatro tarefas, duas máquinas e um servidor disponível, representado no Quadro 3.

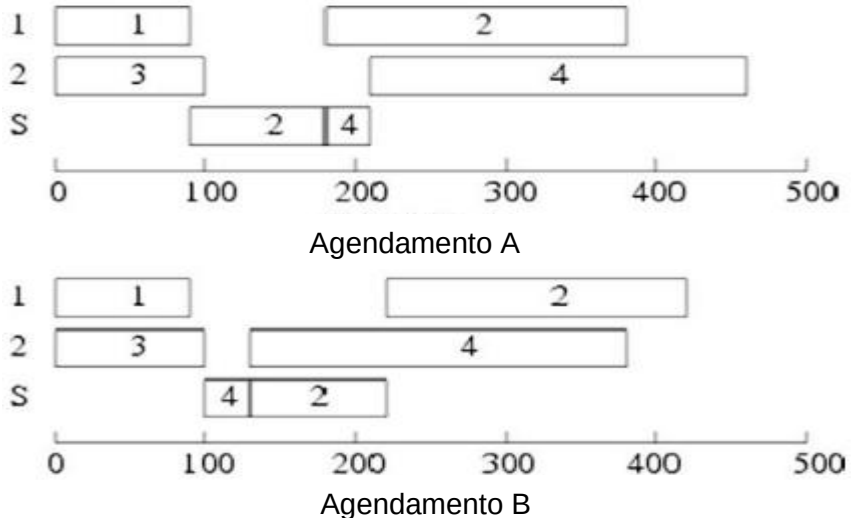
Quadro 3 - Exemplo de sequenciamento de tarefas considerando o tempo de *setup*

Tarefa	Máquina	Tempo de Processamento (min)
1	**1**	90
2	**1**	200
3	**2**	100
4	**2**	250

Fonte: Adaptação do exemplo proposto por Huang, Cai e Zhang (2010).

Sendo que os valores do *setup* dependendo da sequência (S) são: S_{12}=90, S_{34}=30, S_{21}=500, S_{43}=500. Obviamente, seria melhor processar a tarefa 1 antes da tarefa 2 na máquina 1. Se o servidor executar a operação de configuração para tarefa 2 imediatamente após a conclusão da tarefa 1 e, em seguida, executar a operação de configuração para a tarefa 4 (agendamento A na Figura 8), o tempo total da programação do sistema é 460 unidades. No entanto, se o servidor tiver a máquina 1 aguardar e executar a operação de configuração para a tarefa 4 primeiro (agendamento B na Figura 8), a configuração do sistema é de 420 unidades.

Figura 8 - Dois diferentes sequenciamentos levando em consideração o tempo de *setup*



Fonte: Adaptado de Huang, Cai e Zhang (2010).

Pode-se concluir, com este simples exemplo que o desempenho do sistema

é afetado pela eficiência da sequência de operação na máquina e pela eficácia da programação de configuração das máquinas no servidor e, portanto, essas informações precisam ser explicitamente consideradas quando o problema é modelado.

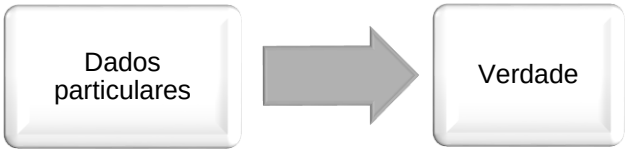
3 METODOLOGIA

Esse capítulo do trabalho está disposto em quatro seções que possuem como objetivo principal citar as etapas necessárias para conclusão do estudo em questão, bem como descrever a metodologia utilizada.

3.1 CLASSIFICAÇÃO DA PESQUISA

A metodologia utilizada para que o estudo referente ao presente trabalho pudesse ser realizado se enquadra no método científico indutivo, definido como o levantamento de dados para concluir uma verdade ou o que se julga verdade. Sendo assim, a observação dos fenômenos acontece para que de maneira subsequente a descoberta da relação entre esses fenômenos ocorra e por fim, generalizar esta correspondência, esta metodologia pode ser melhor compreendida por meio da ilustração da Figura 9.

Figura 9 - Processo do método científico indutivo



Fonte: Autoria Própria.

Além disso, a classificação quanto a natureza da pesquisa realizada se enquadra na definição de pesquisa aplicada. Pois contribui para o conhecimento já existente na literatura e apenas realiza o acúmulo de informações.

A pesquisa aplicada é fundamentalmente motivada pela necessidade de resolver problemas concretos, mais imediatos, ou não. Tem, portanto, finalidade prática, ao contrário da pesquisa pura, motivada basicamente pela curiosidade intelectual do pesquisador e situada sobretudo no nível da especulação (Vergara, 1998).

Visto que os dados utilizados no estudo são numéricos e que por meio deles busca-se confirmação das hipóteses encontradas na literatura, pode-se concluir que a pesquisa realizada possui abordagem quantitativa.

O objetivo do presente trabalho pode ser dito como exploratório, já que tem

como finalidade proporcionar à literatura uma visão geral do assunto tratado, por meio de levantamento bibliográfico e documental.

Em relação aos procedimentos, o presente trabalho é definido como uma pesquisa bibliográfica, que a partir de dados obtidos em fontes bibliográficas, o estudo pode ser elaborado. As informações necessárias foram coletadas por meio do acesso literário via livros e base de dados de artigos científicos, como *Science Direct*, *Scielo* e *Periódicos Capes*, e de teses, como o Banco de Dados da Universidade de São Paulo. Tanto os livros escolhidos quanto os artigos foram filtrados de acordo com o alinhamento com o tema, além de buscar pelos autores mais representativos referente a cada tópico do trabalho. Além disso, outro nível de filtragem utilizado para os artigos utilizados foi referente ao fator de impacto.

3.2 LEVANTAMENTO DE DADOS

O presente trabalho é baseado em um cenário fictício, descrevendo um problema de programação *flowshop* utilizando oito mil dados aleatórios respeitando as especificações citadas posteriormente que se referem tanto a tempos de processamento quanto a tempos de *setup* necessários para realização de cada atividade em determinada máquina.

O banco de dados original de Taillard engloba 120 instâncias, separados em 12 classes distintas, sendo que cada classe possui 10 problemas. O autor trabalha com 9 classes combinando o número de máquinas (**m**) = {5, 10, 20} e o número de tarefas (**n**) = {20, 50, 100}, 2 classes combinam **m** = {10, 20} e **n** = {200} e por fim, uma classe combina **m** = {20} e **n** = {500}.

Neste trabalho utilizou-se uma adaptação do banco de dados deste autor, ou seja, a amostra utilizada é resultado de combinações de máquinas e tarefas, sendo que para cada variável, estipulou-se determinada variação de quantidade. A quantidade inicial de máquinas foi de cinco, com variação de 5 em 5 máquinas, por 4 vezes, enquanto que para as tarefas a quantidade inicial foi de 20, variando de 20 em 20, por 5 vezes. De forma geral, trabalhou-se com as seguintes quantidades das variáveis em questão:

m = 5, 10, 15 e 20;

n = 20, 40, 60, 80 e 100.

Possibilitando, assim, vinte classes geradas da combinação de **m**x**n** para análise do estudo, equivalentes a 5x20, 10x20, 15x20, 20x20, 5x40, 10x40, 15x40, 20x40, 5x60, 10x60, 15x60, 20x60, 5x80, 10x80, 15x80, 20x80, 5x100, 10x100, 15x100 e 20x100.

Para toda e qualquer combinação de máquinas e tarefas, os tempos de processamento foram gerados com intervalos de variação entre 1 e 100 unidades de tempo, sendo que para cada possível combinação, um total de 100 problemas diferentes foram armazenados, com o total de duas mil instâncias.

Enquanto que para os tempos de *setup*, trabalhou-se com três grupos de diferentes intervalos de variação dos tempos, o primeiro possui intervalo de variação entre 1 e 25 unidades de tempo, o segundo entre 1 e 50 unidades de tempo e por fim, o terceiro trabalha com tempos intervalados entre 1 e 100 unidades de tempo. Para cada grupo gerou-se, assim como para os tempos de processamento, 100 problemas diferentes para cada combinação de máquinas e tarefas, totalizando, para os três grupos utilizados, seis mil instâncias.

3.3 OPERACIONALIZAÇÃO DAS VARIÁVEIS

Em sequência do levantamento de dados, estes puderam ser operacionalizados por meio do algoritmo implementado no *software DevPascal* de autoria própria, disponibilizado no Apêndice A, utilizando um computador com as seguintes especificações:

- Processador Intel ® Core™ i5-3317U CPU @ 1.70 GHz;

- Memória RAM de 6 GB;

- Sistema Operacional de 64 bits, processador com base em x64.

A fim de verificar o comportamento do sistema de produção clássico em virtude da função objetivo de *makespan* para os métodos de ordenação SPT, LPT e NEH, serão utilizadas estatísticas de desempenho apresentados na próxima seção.

A medida de desempenho escolhida foi a de *makespan* por garantir maior

confiabilidade nas entregas e uma utilização eficiente da linha produtiva. Em questão dos métodos heurísticos, a seleção do SPT e LPT se deve pela complexidade de um problema *flowshop* com *setup* separado, considerado um problema NP-Completo, além disso, apesar da simplicidade dos métodos de ordenação, colaboram com um satisfatório desempenho nos resultados para atender a FO de minimização de *makespan*.

Enquanto que o método NEH está sendo analisado por ser um método construtivo de bastante referência na criação de novas heurísticas que utilizam como parâmetro as ordenações SPT e/ou LPT. Como método construtivo, possui duas fases, a primeira se baseia na ordenação dos métodos de sequenciamento SPT ou LPT e na segunda fase o NEH avança construindo a sequência à medida que a ordem das tarefas vai sendo alterada com base nos resultados obtidos da FO. Por explorar novas sequências além da ordenação inicial SPT ou LPT, este método garante, portanto, o encontro de soluções de alta qualidade.

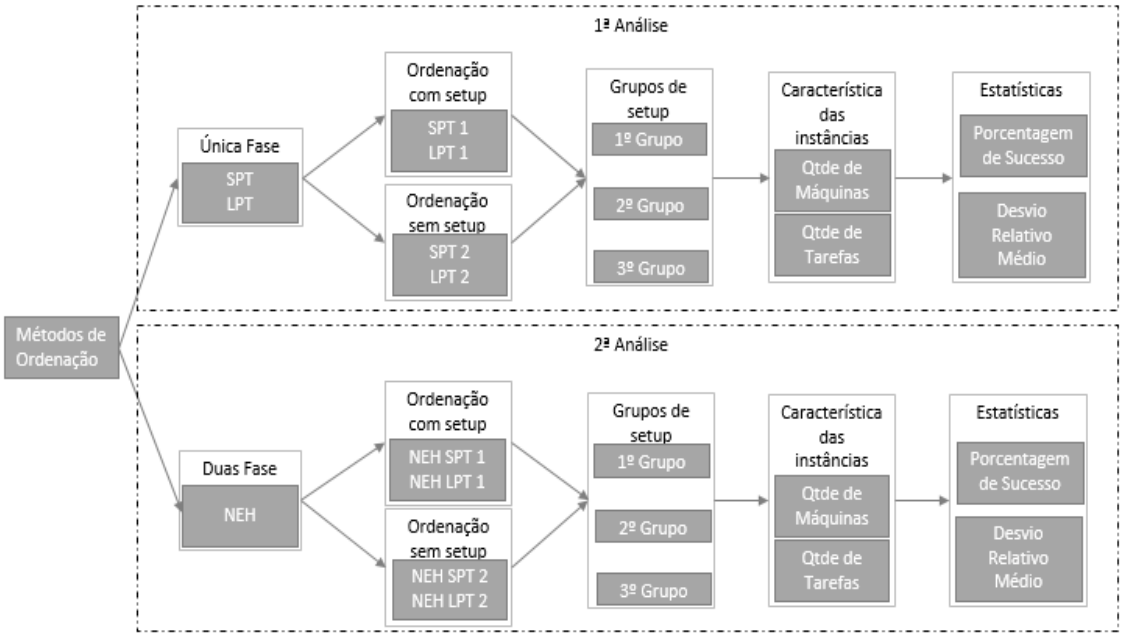
Ademais, vale salientar também que os tempos de *setup* são separados dos tempos de processamento e podem ser antecipados, ou seja, a medida que determinada máquina esteja livre, a configuração para uma tarefa **n**, mesmo que esta ainda esteja sendo executada em outra máquina, pode ser realizada. Como a própria definição deste tipo de *setup* sugere, ocorre a antecipação da configuração, o que possibilita redução do tempo utilizado no processo produtivo.

3.4 ANÁLISE DOS RESULTADOS

Diante da obtenção dos resultados por meio do *software* com implementação em linguagem *Pascal*, os dados foram analisados a fim de verificar e comparar o comportamento de cada método.

A análise foi realizada utilizando alguns critérios que envolvem a quantidade de fases dos métodos heurísticos, utilização, ou não, do tempo de *setup* na ordenação, a combinação com os grupos de tempos de *setup* definidos anteriormente e a caracterização das instâncias, para serem analisado com o auxílio das estatísticas de desempenho, como ilustrado na Figura 10.

Figura 3 - Fluxograma da análise de resultados.



Fonte: Autoria Própria.

Como visualizado na Figura 10, inicialmente foram realizadas duas análises separadamente, seguindo o critério de configuração dos métodos heurísticos, ou seja, os métodos SPT e LPT que possuem apenas uma fase, foram analisados primeiramente, e em um segundo momento apenas o método construtivo NEH foi observado por utilizar duas fases na ordenação das tarefas.

Em cada análise utilizou-se a ordenação inicial crescente (SPT) e descrescente (LPT) baseada apenas nos tempos de processamento e baseada nos tempos de processamento e tempos de *setup*. Cada método e sua variação foi analisado, em diferentes momentos, com cada grupo de *setup* que se refere ao possível intervalo de variação dos tempos. Por fim, os resultados obtidos foram agrupados de acordo com a caracterização das instâncias, primeiramente agrupou-se instâncias com a mesma quantidade de máquinas e em seguida com a mesma quantidade de tarefas para que então os resultados fossem observados por meio das estatísticas de desempenho.de porcentagem de sucesso (PS) e desvio relativo médio (DRM).O conjunto de resultados dessas estatísticas foram visualizados de forma gráfica, para melhor entendimento da correlação e comparação entre os métodos heurísticos utilizados, com o auxílio da ferramenta *Excel* 2013.

A Porcentagem de Sucesso é definida pela quantidade de vezes que um método se mostrou mais efetivo, dividido pela quantidade total dos problemas

analisados. Para determinar o sucesso de um determinado método, escolhe-se o menor valor obtido da FO, visto o caso de uma função objetivo focada em minimização de critério, e todos os resultados são comparados com este valor. Se o valor analisado equivale ao menor valor, então é considerado como um sucesso e recebe o valor 1 (um), caso contrário recebe o valor 0 (zero).

Sendo assim, podemos considerar o raciocínio lógico do sucesso como mostra a equação 13:

Se $x_i = x_{min;}$

Então, $x_i=1;$ (13)

Se não, $x_i=0$

A partir de então podemos definir a equação de Porcentagem de Sucesso como:

$$PS(\%) = \frac{\sum_{i=1}^{n} sucesso}{n} \quad (14)$$

sendo n equivalente a quantidade total de problemas analisados.

Já o Desvio Relativo analisa a variação entre os valores encontrados em um determinado método e o valor de sucesso em âmbito geral. Ou seja, o ideal para que um método seja considerado bom é que o valor desse desvio seja o menor possível, ou até mesmo zero. O cálculo do desvio é encontrado na Equação 15, sendo que x_i equivale aos valores obtidos no método em questão e x^* representa o menor valor dentre todos os métodos analisados.

$$Desvio_i = x_i - x^* \quad (15)$$

Seguindo a lógica proposta, a equação referente ao DRM é a seguinte:

$$DR_i(\%) = \frac{x_i - x^*}{x^*} \times 100 \quad (16)$$

Após obtermos o cálculo e o conhecimento das estatísticas citadas para cada método heurístico, se torna possível a comparação dentre todas as instâncias e por fim, a conclusão de qual método heurístico se mostra mais eficiente em cada análise realizada e, também, entre a 1ª e a 2ª análise.

4 RESULTADOS E DISCUSSÕES

Os resultados da análise de qual método de ordenação de tarefas proporciona um menor tempo total da programação da produção, considera, além do tempo de processamento, o tempo de *setup* para o cálculo de *makespan*, e foram representados de forma gráfica, para a melhor visualização dos desempenhos alcançados.

Vale ressaltar que as comparações realizadas foram em torno do sistema de produção clássico (sem restrição *no-wait* ou *no-idle*) e dos métodos de ordenação SPT. LPT e NEH. Os métodos de ordenação de apenas uma fase, ou seja, o SPT e o LPT foram utilizados de duas maneiras distintas, em um primeiro momento, as ordenações foram baseadas na somatória tanto do tempo de processamento, quanto do tempo de *setup* das tarefas. Assim como as ordenações também foram definidas com base apenas nos tempos de processamento.

Ao passo que o NEH foi analisado de quatro maneiras diferentes em sua primeira fase, ou seja, observou-se o método NEH baseado na ordenação SPT e LPT e suas variações, ao considerar o tempo de *setup* para definição da ordem das tarefas, ou levar em consideração apenas os tempos de processamento de cada operação nas maquinas disponíveis. Enquanto que a segunda fase deste método aconteceu sem alterações nas maneiras analisadas, construindo a sequência em cada etapa com base no menor valor obtido na FO.

Entretanto, a maneira como a somatória é calculada para definição da ordenação, não interfere no fato de que o cálculo do *makespan* leva em consideração o tempo de *setup* que cada tarefa demanda em cada máquina necessária no processo. Ou seja, por mais que a ordenação não seja definida pela soma dos tempos de processamento e tempos de *setup* demandado por cada tarefa, ainda assim o *setup* é levado em consideração no cálculo das medidas de desempenho.

Como os métodos analisados podem ser diferenciados pela quantidade de fases utilizadas para definição da sequência de tarefas, foram também analisados separadamente. Inicialmente analisou-se os métodos de ordenação SPT e LPT, seguido da análise do método construtivo NEH. Complementarmente, realizou-se a comparação entre todas as soluções de cada análise de métodos, a fim de se avaliar a melhora ou não da qualidade da função objetivo, sendo possível identificar qual tipo

de método heurístico atende melhor a minimização do *makespan*.

4.1 MÉTODOS DE ORDENAÇÃO SPT e LPT

Os métodos de sequenciamento estão representados pelas cores ilustradas na Figura 11, recebendo como denominação SPT 1 e LPT 1 os métodos que levam em consideração tanto o tempo de processamento quanto o tempo de *setup* para definir o sequenciamento das tarefas, enquanto que a legenda de SPT 2 e LPT 2, faz referência aos métodos de sequenciamento que se baseiam apenas na somatória dos tempos de processamento para definição da ordem de tarefas.

Figura 4 - Cores representativas dos métodos de sequenciamento

Fonte: Autoria Própria.

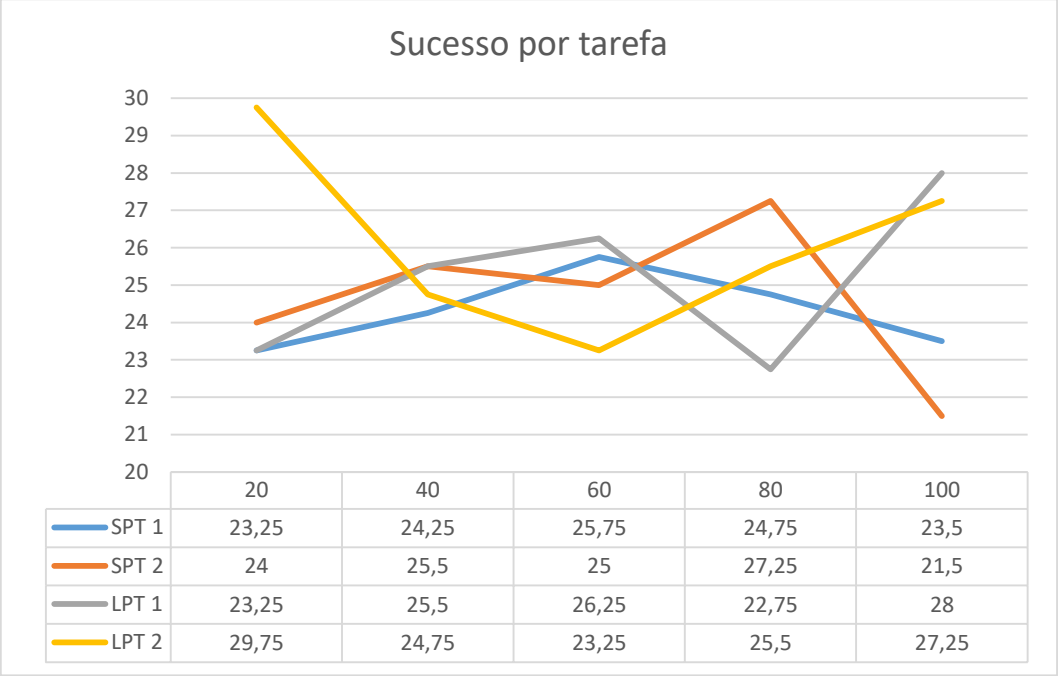
Realizou-se a média dos valores para a PS e DRM para cada uma dessas categorias formadas, e assim, um total de 12 gráficos de análise foram gerados já que para cada grupo de valores de *setup* analisamos gráficos de porcentagem de sucesso e desvio relativo médio para instâncias com a mesma quantidade de tarefas, bem como com a mesma quantidade de máquinas, e estes são apresentados em seguida, separados em cada seção por grupos de acordo com os valores de *setup*.

4.1.1 Primeiro Grupo

Este grupo de instâncias de tempos de *setup* se refere a tempos com intervalos entre 1 e 25 unidades de tempo, que ao se unir com as instâncias de tempos de processamento foram executados pelo algoritmo *Pascal* desenvolvido e foram analisados para cada estatística em estudo.

O Gráfico 1 ilustra a PS para a categoria dos problemas com a mesma quantidade de tarefas, para os métodos heurísticos SPT 1, SPT 2, LPT 1 e LPT 2.

Gráfico 1 - Sucesso por tarefa do 1º grupo



	20	40	60	80	100
SPT 1	23,25	24,25	25,75	24,75	23,5
SPT 2	24	25,5	25	27,25	21,5
LPT 1	23,25	25,5	26,25	22,75	28
LPT 2	29,75	24,75	23,25	25,5	27,25

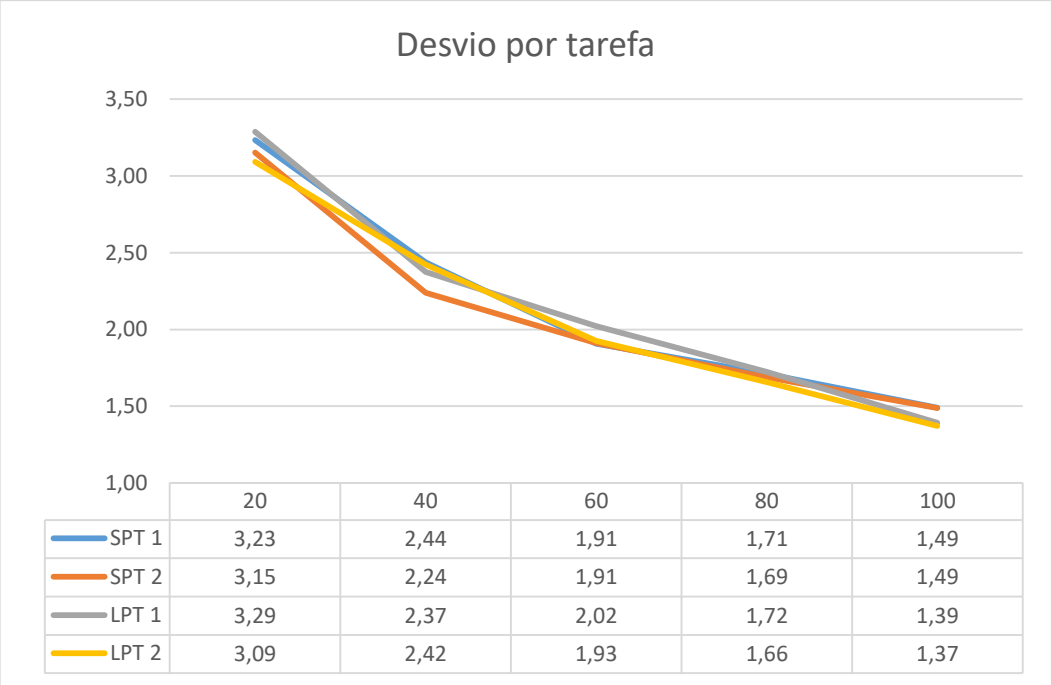
Fonte: Autoria Própria.

É possível perceber que os valores, de uma forma geral, são muito próximos, com exceção dos resultados para a quantidade de tarefas equivalente a 20, bem como no último valor de variação das tarefas, sendo **n**=100, que possuem maiores diferenças entre os pontos plotados no gráfico.

Apesar disso é visto que o método de sequenciamento de tarefas que demonstra mais sucesso é o LPT, especialmente o LPT 2, que não leva em consideração os tempos de *setup* para definição de ordenação das tarefas.

O Gráfico 2 ilustra o desvio relativo médio dos valores encontrados como solução para cada método de sequenciamento com o valor de sucesso obtido entre a comparação entre os métodos. Após constatação realizada com auxílio do Gráfico 1, percebe-se veracidade nas afirmações já que de forma geral, os métodos de sequenciamento demonstraram comportamento parecido, em que as linhas do gráfico possuem, de certa forma, desenhos e tendências parecidas, com a proximidade de valores encontrados entre eles.

Gráfico 2 - DRM por tarefa do 1º grupo



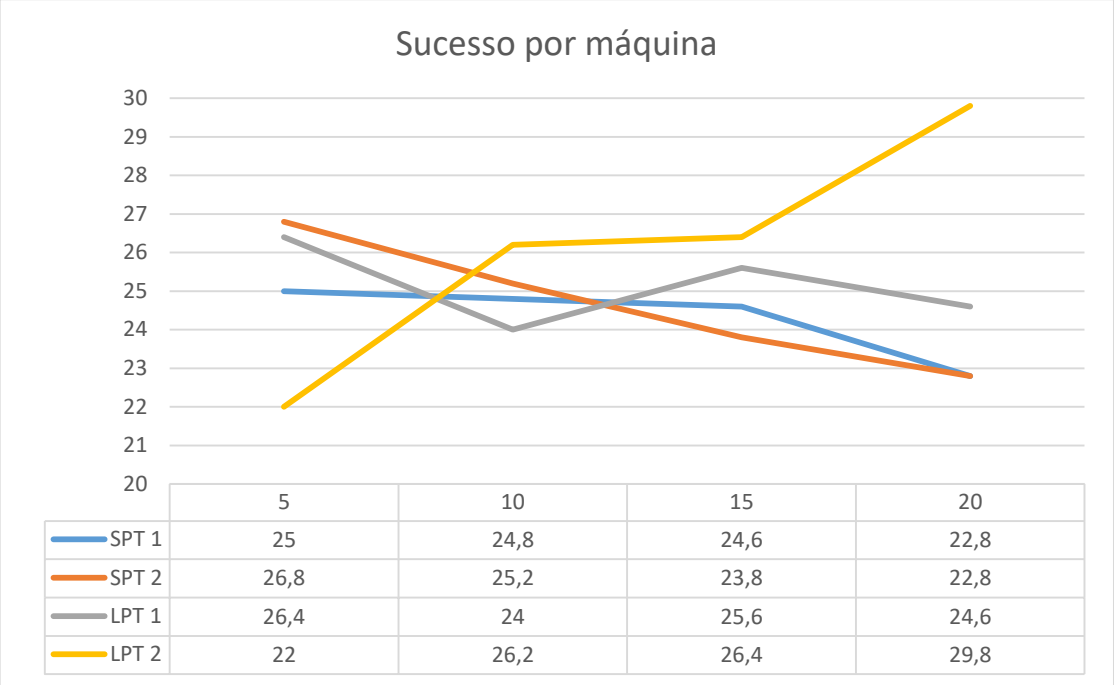
	20	40	60	80	100
SPT 1	3,23	2,44	1,91	1,71	1,49
SPT 2	3,15	2,24	1,91	1,69	1,49
LPT 1	3,29	2,37	2,02	1,72	1,39
LPT 2	3,09	2,42	1,93	1,66	1,37

Fonte: Autoria Própria.

Quando tratamos de análise realizada com compilação dos dados baseados na quantidade de máquinas de cada instância, também é possível analisar o desempenho dos métodos de sequenciamento por meio das mesmas estatísticas definidas anteriormente. Os resultados encontrados para o primeiro grupo de *setup* analisado, juntamente com os tempos de processamento, podem ser visualizados e analisados por meio dos gráficos gerados.

No Gráfico 3 ilustra-se a porcentagem de sucesso obtida entre os métodos de ordenação analisados. O comportamento entre os métodos é um pouco distinto do gráfico de sucesso por tarefas analisado anteriormente, em outras palavras, quando as estatísticas são analisadas pelos dados compilados pela quantidade de máquinas, os métodos se mostram menos competitivos entre si, o que resulta em valores bem diversificados.

Gráfico 3 - Sucesso por máquina do 1º grupo



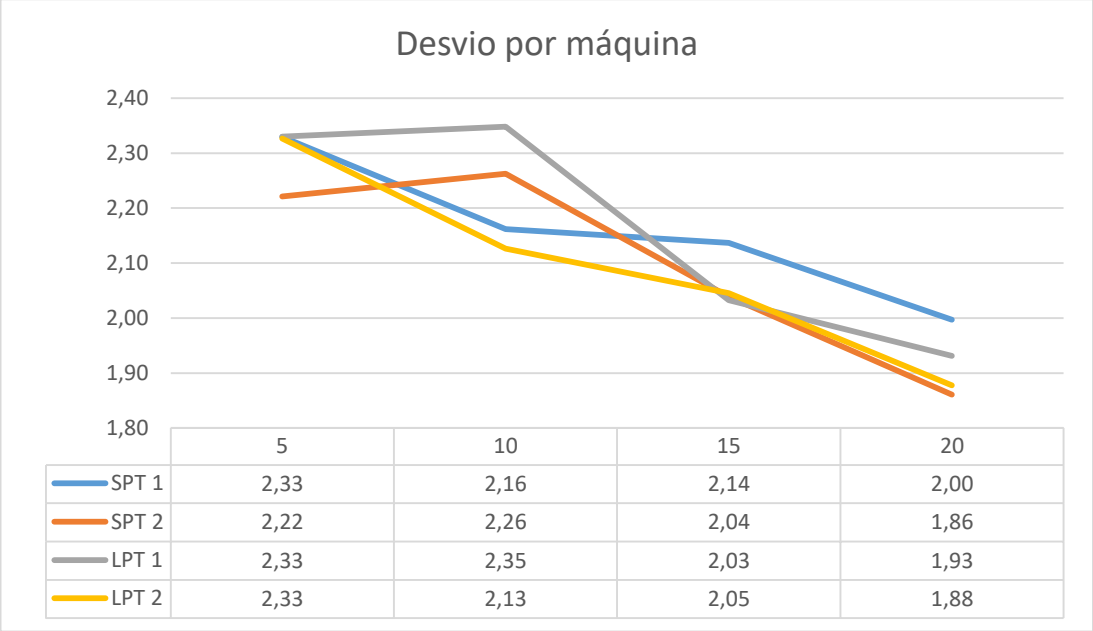
	5	10	15	20
SPT 1	25	24,8	24,6	22,8
SPT 2	26,8	25,2	23,8	22,8
LPT 1	26,4	24	25,6	24,6
LPT 2	22	26,2	26,4	29,8

Fonte: Autoria Própria.

Ainda assim, ao realizar a análise dos dados baseado na quantidade de máquinas, especificamente, e não de tarefas, fica ainda mais evidente o desempenho positivo do método de sequenciamento LPT, tanto na versão que o tempo de *setup* e o tempo de processamento são levados em consideração para ordenação das tarefas quanto na versão em que apenas o tempo de processamento determina o sequenciamento.

Assim como verificado pelas estatísticas de quantidade de sucesso, no Gráfico 4, que representa o desvio relativo médio por máquina para este primeiro grupo de *setup* estudado, verifica-se melhor desempenho do LPT diante outro método analisado.

Gráfico 4 - DRM por máquina do 1º grupo



	5	10	15	20
SPT 1	2,33	2,16	2,14	2,00
SPT 2	2,22	2,26	2,04	1,86
LPT 1	2,33	2,35	2,03	1,93
LPT 2	2,33	2,13	2,05	1,88

Fonte: Autoria Própria.

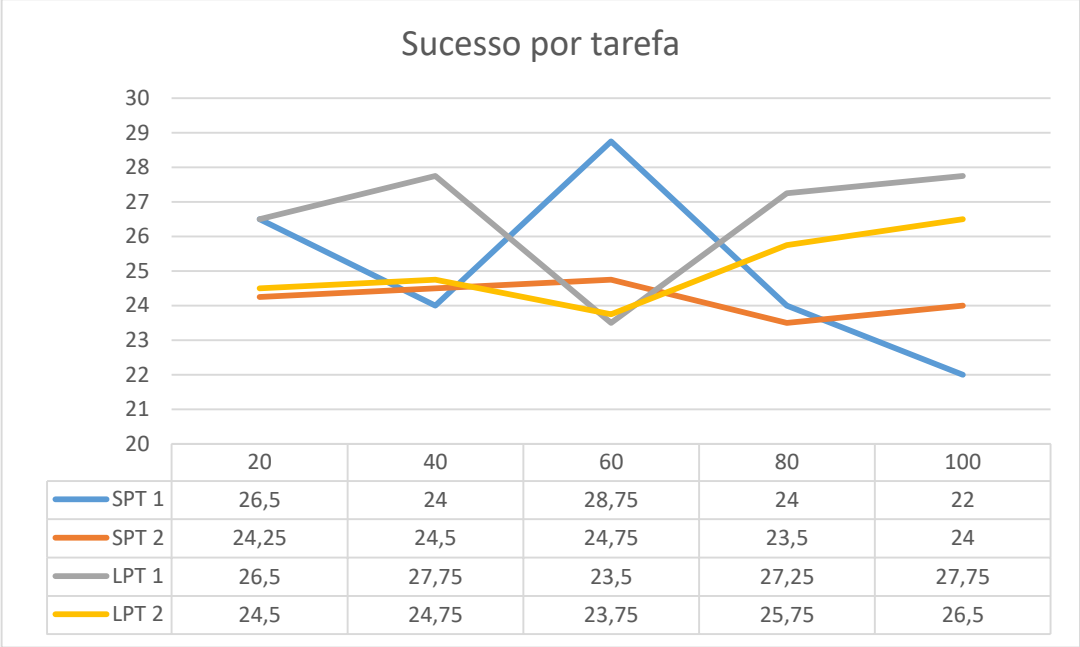
De forma geral, neste primeiro momento, o método de sequenciamento que se demonstrou mais efetivo, de forma a minimizar a função objetivo *makespan*, foi o LPT 2, que determina o sequenciamento com base apenas nos tempos de processamento, que considera o tempo de *setup* apenas para cálculo da FO.

4.1.2 Segundo Grupo

O segundo grupo de *setup* tem como característica o intervalo de variação entre 1 e 50 unidades de tempo dos possíveis tempos de *setup* a serem gerados, e dessa forma foram analisados, com os tempos de processamentos, com o auxílio do algoritmo construído para cálculo do *makespan*.

A primeira estatística analisada foi a quantidade de sucesso para cada método de sequenciamento de tarefas dos resultados compilados das instâncias que possuem a mesma quantidade de tarefas, e essa estatística por ser visualizada no Gráfico 5.

Gráfico 5 - Sucesso por tarefa do 2º grupo



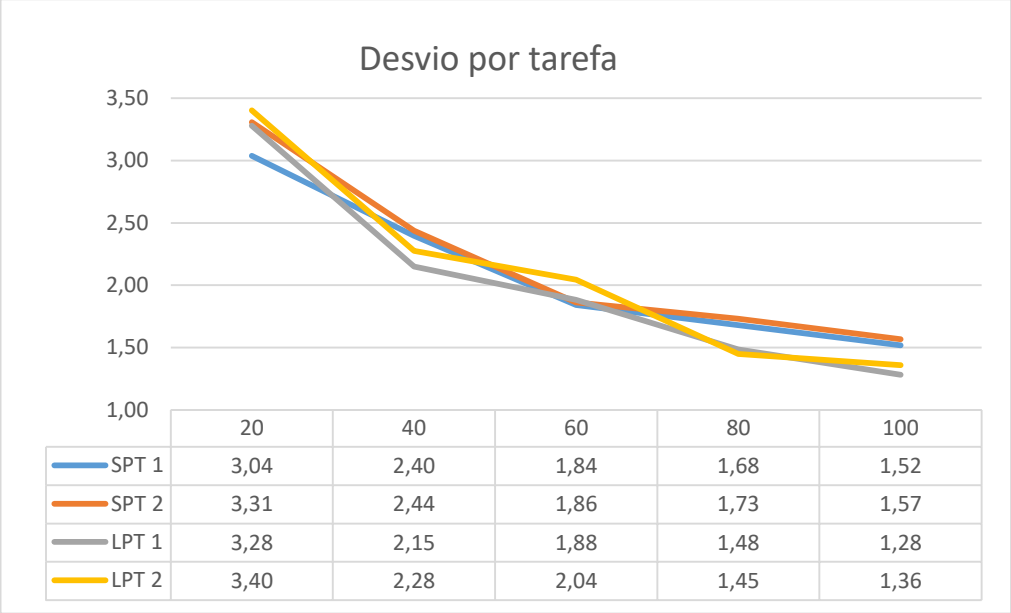
	20	40	60	80	100
SPT 1	26,5	24	28,75	24	22
SPT 2	24,25	24,5	24,75	23,5	24
LPT 1	26,5	27,75	23,5	27,25	27,75
LPT 2	24,5	24,75	23,75	25,75	26,5

Fonte: Autoria Própria.

Ao analisar este gráfico representado, verifica-se uma importante divergência de valores, ou em outras palavras, identifica-se oscilação de resultados entre as quantidades de tarefas estipuladas. Entretanto, de certa forma ainda é possível perceber que o método de sequenciamento LPT 1 demonstrou melhor desempenho, acompanhado em sequência de um empate entre o método LPT 2 e SPT 1.

Analisou-se também, pelo Gráfico 6, o desvio relativo médio do valor de sucesso encontrado na comparação dos quatros métodos heurísticos em estudo.

Gráfico 6 - DRM por tarefa do 2º grupo



	20	40	60	80	100
SPT 1	3,04	2,40	1,84	1,68	1,52
SPT 2	3,31	2,44	1,86	1,73	1,57
LPT 1	3,28	2,15	1,88	1,48	1,28
LPT 2	3,40	2,28	2,04	1,45	1,36

Fonte: Autoria Própria.

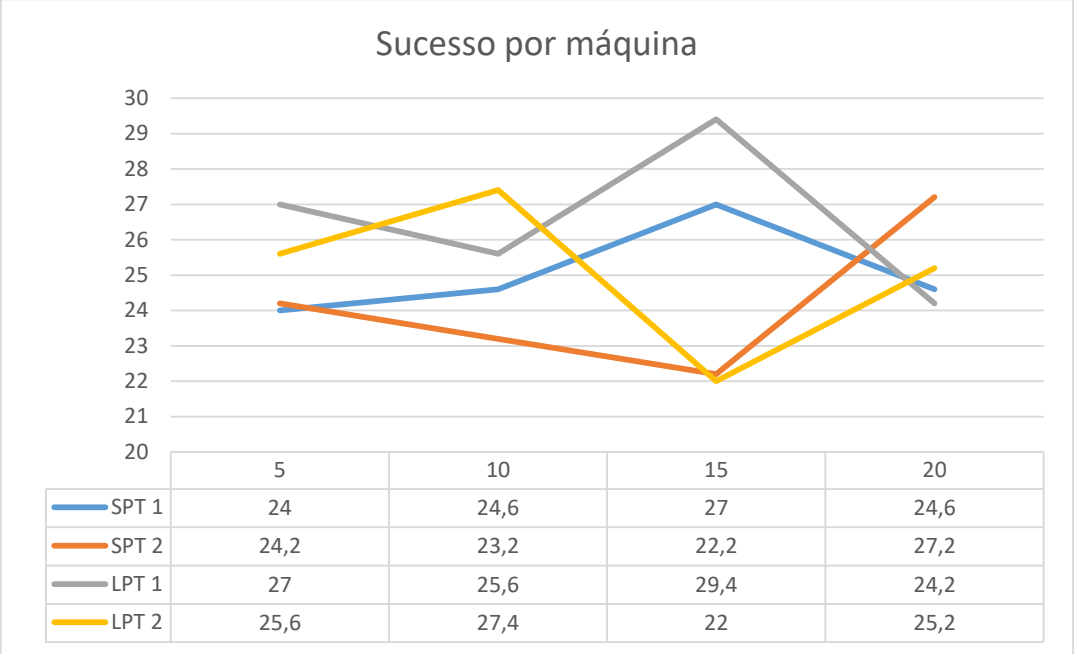
De forma muito semelhante os métodos trabalham com praticamente o mesmo intervalo de desvio do melhor resultado encontrado, o que justifica a proximidade das linhas representantes de cada método de sequenciamento.

Interligado com o resultado da quantidade de sucesso apresentada anteriormente, o método que se mostra mais efetivo, ou seja, o que possui menor desvio em relação ao valor de sucesso encontrado neste grupo de análise, é o LPT 1. Já em relação aos dois métodos de sequenciamento de tarefas que demonstraram, após o LPT 1, um bom e mesmo resultado de quantidade de sucesso, puderam nesta análise de DRM serem desempatados e então classificados justamente, sendo o SPT 1 o segundo método com menor desvio relativo e o LPT 2 o terceiro.

Em sequência, a análise realizada aconteceu em torno das instâncias que possuem a mesma quantidade de máquinas, independentemente da quantidade de tarefas que possuem, sendo analisadas as mesmas estatísticas de desempenho.

Assim como nos gráficos representativos das estatísticas para instâncias unidas por quantidade de tarefas, por meio do Gráfico 7, verifica-se certa semelhança nos resultados dos métodos de sequenciamento, e ainda assim, mesmo que com pouca diferença, o método que se destaca com um melhor desempenho é o LPT 1, seguido do método SPT 1 e LPT 2, com resultados semelhantes.

Gráfico 7 - Sucesso por máquina do 2º grupo

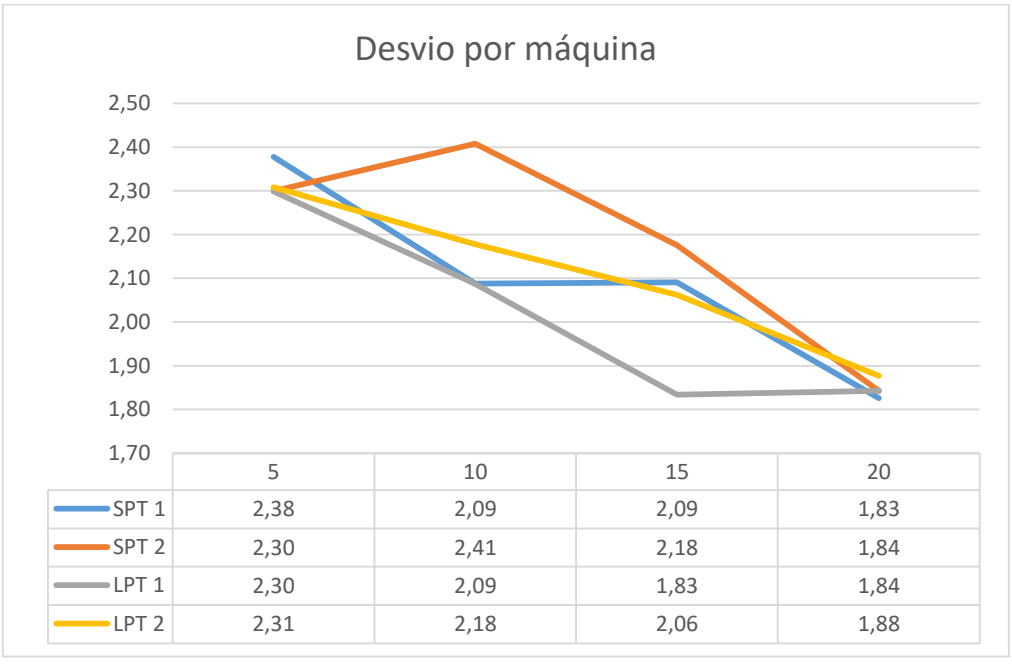


	5	10	15	20
SPT 1	24	24,6	27	24,6
SPT 2	24,2	23,2	22,2	27,2
LPT 1	27	25,6	29,4	24,2
LPT 2	25,6	27,4	22	25,2

Fonte: Autoria Própria.

Sendo assim, os resultados encontrados para os desvios relativos também se assemelham aos já vistos nas análises anteriores desse grupo de *setup*, ilustrado no Gráfico 8.

Gráfico 8 - DRM por máquina do 2º grupo



	5	10	15	20
SPT 1	2,38	2,09	2,09	1,83
SPT 2	2,30	2,41	2,18	1,84
LPT 1	2,30	2,09	1,83	1,84
LPT 2	2,31	2,18	2,06	1,88

Fonte: Autoria Própria.

Ou seja, de certa forma os resultados são muito próximos, com algumas

exceções pontuais, tanto para a quantidade de máquinas igual a 10, como para **m**=15, mas ainda assim seguem o mesmo comportamento.

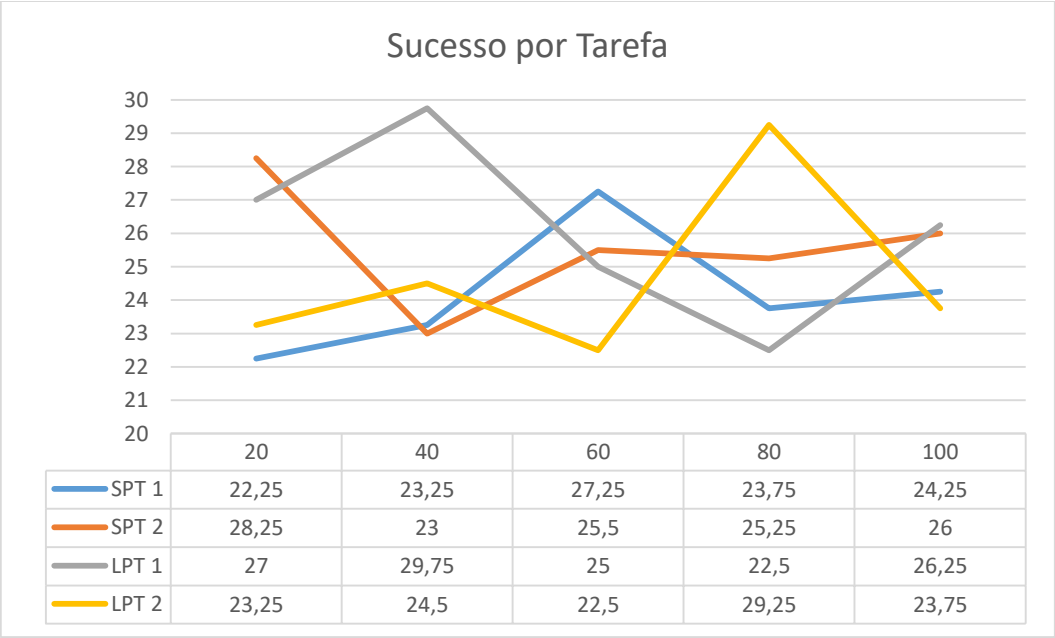
No Gráfico 8 fica ainda mais evidente a melhor performance dos resultados do método de sequenciamento LPT 1, visto que sua linha representativa é a que demonstra menor valor, ou seja, a que está mais perto dos valores de sucesso encontrados na comparação realizadas entre os métodos de ordenação.

4.1.3 Terceiro Grupo

O último grupo de *setup* criado para a análise dos métodos de sequenciamento deste presente estudo se baseia na geração de dados de tempos de *setup* com intervalo de variação entre 1 e 100 unidades de tempo, ou seja, é o grupo com maior nível de possível variação dos tempos. A análise para esse grupo consiste no mesmo sistema adotado para os outros dois grupos anteriores, sendo então a análise de porcentagem quantidade de sucesso bem como do desvio relativo médio.

Quando comparado aos outros grupos de *setup* utilizados no presente estudo, verifica-se que uma maior discrepância de resultados é encontrada no terceiro grupo, ilustrada no Gráfico 9.

Gráfico 9 - Sucesso por tarefa do 3º grupo



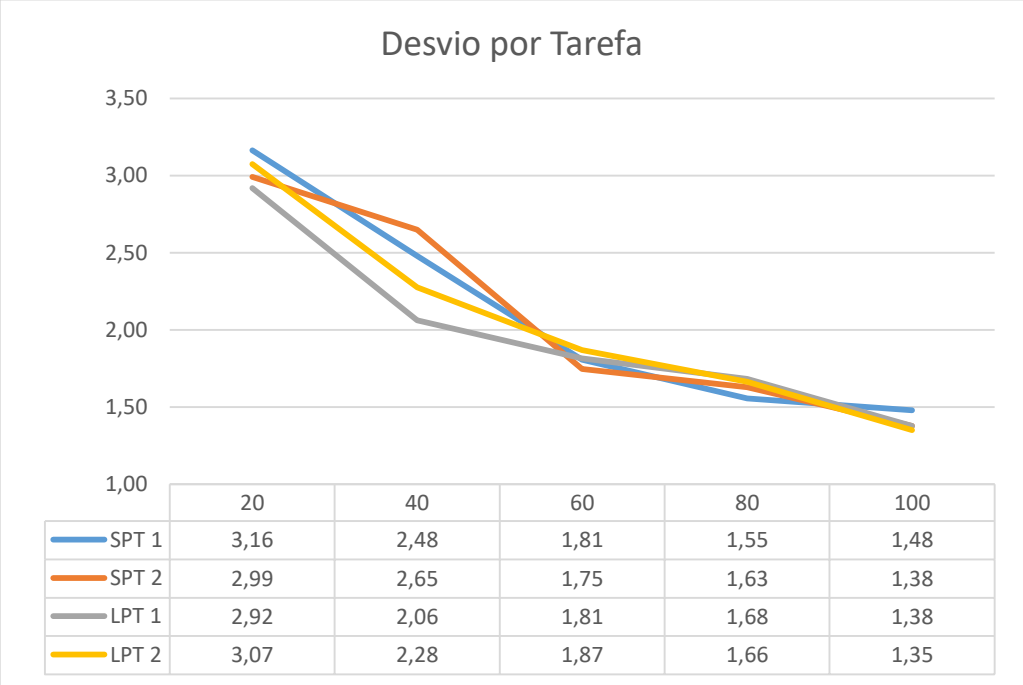
	20	40	60	80	100
SPT 1	22,25	23,25	27,25	23,75	24,25
SPT 2	28,25	23	25,5	25,25	26
LPT 1	27	29,75	25	22,5	26,25
LPT 2	23,25	24,5	22,5	29,25	23,75

Fonte: Autoria Própria.

Os métodos de sequenciamento se demonstram muito competitivos uns com os outros e dificulta a verificação de qual método desempenhou melhor quanto a esta estatística, entretanto com os dados numéricos conclui-se que o método LPT 1 resultou em uma porcentagem maior de sucesso quando em comparação aos demais.

Verifica-se, com auxílio do Gráfico 10, que ainda assim os métodos de ordenação possuem basicamente mesma proporção de desvio relativo com o melhor resultado.

Gráfico 10 - DRM por tarefa do 3º grupo



	20	40	60	80	100
SPT 1	3,16	2,48	1,81	1,55	1,48
SPT 2	2,99	2,65	1,75	1,63	1,38
LPT 1	2,92	2,06	1,81	1,68	1,38
LPT 2	3,07	2,28	1,87	1,66	1,35

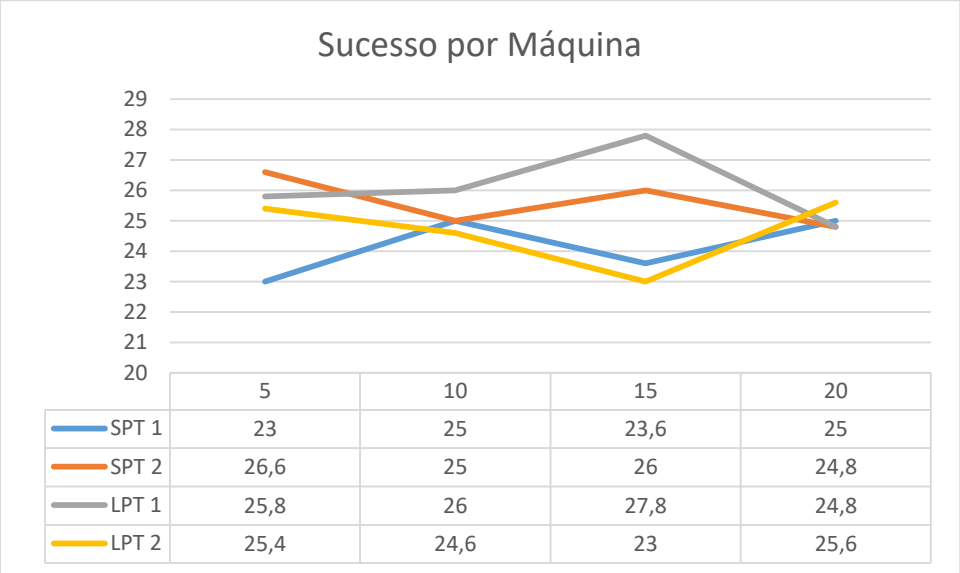
Fonte: Autoria Própria.

Isso muito se deve pelo fato da quantidade de sucesso dos métodos possuírem elevada oscilação, ou seja, serem bastante competitivos entre si. Contudo, é possível identificar que o desvio relativo médio está condizente com o resultado da estatística de porcentagem de sucesso, já que o método LPT 1 apresenta menor variância em relação ao valor de sucesso encontrado na comparação dos métodos analisados.

Por outro lado, quando analisamos as estatísticas pelas instâncias compiladas pelo número de máquina, ao invés de tarefas, os resultados de sucesso se ilustram de forma espelhada para os métodos SPT 1 e SPT 2, e para LPT 1 e LPT 2. Ao contrário do que pôde ser concluído no gráfico de sucesso por tarefa, em que os

métodos se mostravam muito competitivos, o Gráfico 11, de sucesso por máquina, na realidade, mostra que os métodos SPT 2 e LPT 1 resultaram em um melhor desempenho do que os demais.

Gráfico 11 - Sucesso por máquina do 3º grupo



	5	10	15	20
SPT 1	23	25	23,6	25
SPT 2	26,6	25	26	24,8
LPT 1	25,8	26	27,8	24,8
LPT 2	25,4	24,6	23	25,6

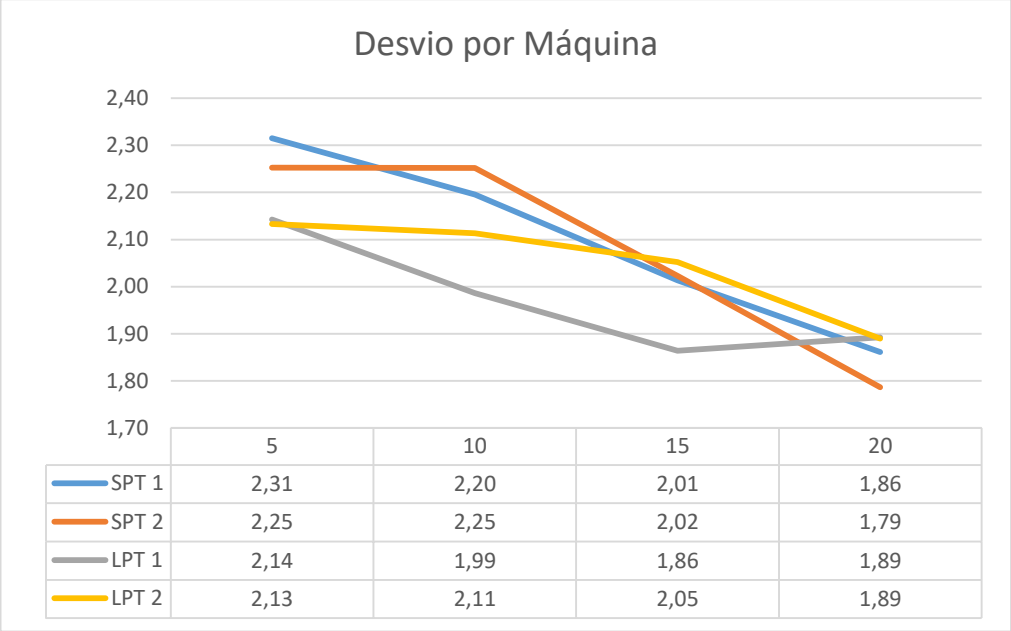
Fonte: Autoria Própria.

Ainda que dois métodos de ordenação tenham se mostrados mais competitivos, é evidente que a linha que representa o método LPT 1 no gráfico possui maiores valores, portanto, maior quantidade de sucesso e consequentemente, melhor desempenho de minimização de *makespan*.

Em conjunto com a estatística de sucesso, o desvio relativo médio auxilia na determinação do método mais adequado de forma a atender a FO desejada. No caso do terceiro grupo, para as instâncias agrupadas por quantidade de máquina, o desvio relativo, ilustrado no Gráfico 12, auxilia na conclusão do desempenho de métodos de ordenação.

No gráfico ilustrativo do desvio relativo por máquina é visível os baixos valores representados pelo método LPT 1, ou seja, esse método resultou em valores mais próximos ou iguais ao menor valor, também chamado de sucesso, encontrado entre os métodos de ordenação analisados.

Gráfico 12 - DRM por máquina do 3º grupo



	5	10	15	20
SPT 1	2,31	2,20	2,01	1,86
SPT 2	2,25	2,25	2,02	1,79
LPT 1	2,14	1,99	1,86	1,89
LPT 2	2,13	2,11	2,05	1,89

Fonte: Autoria Própria.

De forma geral, ao analisar cada estatística em cada grupo de tempos de *setup*, pode-se perceber que as estatísticas são complementares, de forma que a análise se torna mais completa, assim como quando os dados compilados por quantidade de máquinas são analisados juntamente com os dados unidos pela quantidade de tarefas.

Desse modo é possível verificar com mais veracidade, o comportamento para cada restrição, e em cada estatística, e assim consequentemente, averiguar o comportamento de cada método heurístico analisado neste presente estudo.

4.2 MÉTODO CONSTRUTIVO

O método construtivo NEH, também utilizado neste presente estudo, está representado pelas cores ilustradas na Figura 12, sendo que o método NEH SPT 1 tem sua ordenação inicial baseada no método SPT que considera o *setup* no sequenciamento das tarefas, o método NEH SPT 2 baseia-se na ordenação SPT que considera apenas os tempos de processamento no sequenciamento, enquanto que o NEH LPT 1 tem como referência o uso, durante a primeira fase, da ordenação LPT considerando, além dos tempos de processo, o *setup* de cada operação nas máquinas

disponíveis, e por fim, o NEH LPT 2 utiliza como base a ordenação LPT sem considerar o tempo de *setup*.

Figura 5 - Cores representativas dos métodos construtivos

Fonte: Autoria Própria.

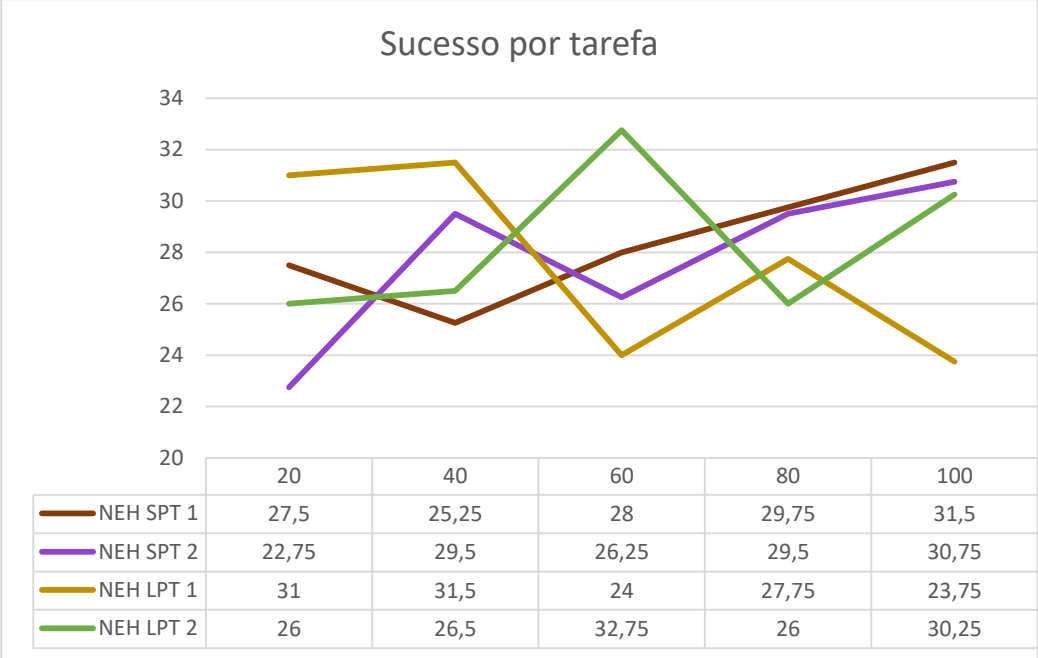
Assim como apresentado na seção secundária anterior, analisou-se por meio da porcentagem de sucesso e do desvio relativo médio, o desempenho dos métodos construtivos para duas categorias, a de instâncias com a mesma quantidade de tarefas, assim como para a de instâncias que possuem a mesma quantidade de máquinas disponíveis.

Desse modo, gerou-se gráficos para analisar cada estatística de acordo com a categoria de tarefas ou máquinas, estes serão apresentados a seguir em seções separadas por grupos de *setup*, definidos pelo intervalo de variação de unidades de tempo.

4.2.1 Primeiro Grupo

Para o grupo de tempos de *setup* em que o intervalo de variação é de 1 a 25 unidades de tempo – utilizados apenas para cálculo da medida de desempenho *makespan* para os métodos NEH SPT 2 e NEH LPT 2, enquanto que os métodos NEH SPT 1 e NEH LPT 1 não só os utilizaram para calcular o desempenho, mas também para definir o sequenciamento de tarefas, baseado nos métodos heurísticos LPT ou SPT – observou-se primeiramente a estatística de PS para o conjunto de instâncias que tem em comum a quantidade de tarefas, ilustrada no Gráfico 13.

Gráfico 13 - Sucesso por tarefa do 1º grupo para o método NEH



	20	40	60	80	100
NEH SPT 1	27,5	25,25	28	29,75	31,5
NEH SPT 2	22,75	29,5	26,25	29,5	30,75
NEH LPT 1	31	31,5	24	27,75	23,75
NEH LPT 2	26	26,5	32,75	26	30,25

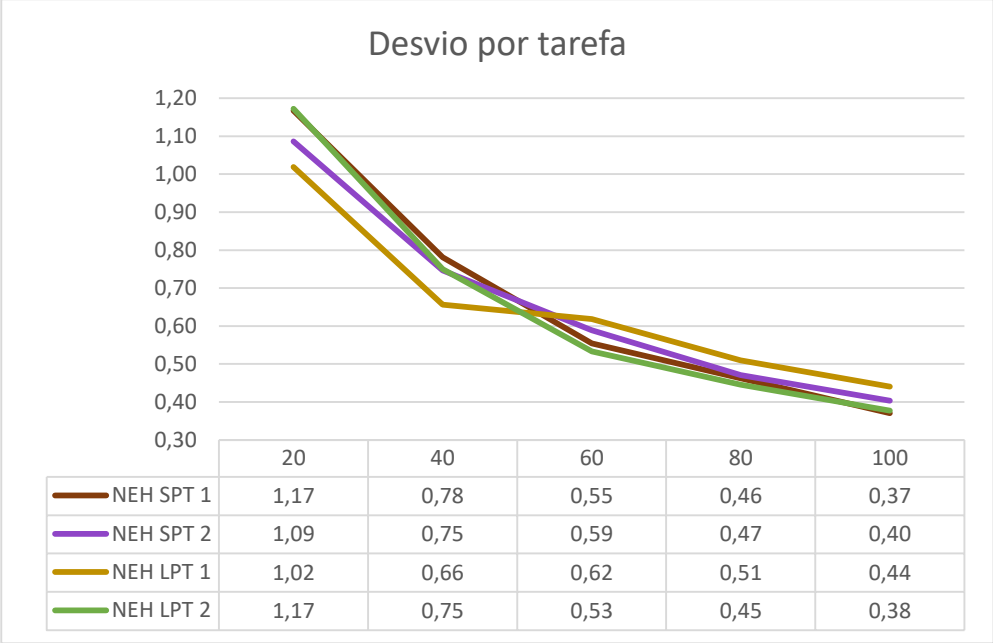
Fonte: Autoria Própria.

Nesta primeira análise, apesar da pequena diferença entre os métodos construtivos NEH, observa-se uma maior estabilidade do método que baseia a ordenação inicial no método heurístico SPT 1, em outras palavras, o método que define o sequenciamento de tarefas em ordem crescente pela soma dos tempos de processamento e tempos de *setup*.

Este método que se destaca, inicia em decréscimo entre **n**=20 e **n**=40 e após esse momento seu desempenho é definido por progressão até a última classe de tarefas. Por essa razão é que se mostrou mais eficiente, já que os demais métodos possuem momentos com grandes oscilações, ou seja, da mesma forma que possuem maior desempenho em uma determinada quantidade de tarefas, se mostram menos eficientes em outras, resultando em um grande desequilíbrio.

De forma a enriquecer a análise dos desempenhos alcançados, o Gráfico 14 auxilia a entender a performance desses métodos, apresentando o DRM, isto é, o desvio dos valores encontrados por cada método em relação ao sucesso da análise, em outras palavras, o menor valor de *makespan*.

Gráfico 14 - DRM por tarefa do 1º grupo para o método NEH



	20	40	60	80	100
NEH SPT 1	1,17	0,78	0,55	0,46	0,37
NEH SPT 2	1,09	0,75	0,59	0,47	0,40
NEH LPT 1	1,02	0,66	0,62	0,51	0,44
NEH LPT 2	1,17	0,75	0,53	0,45	0,38

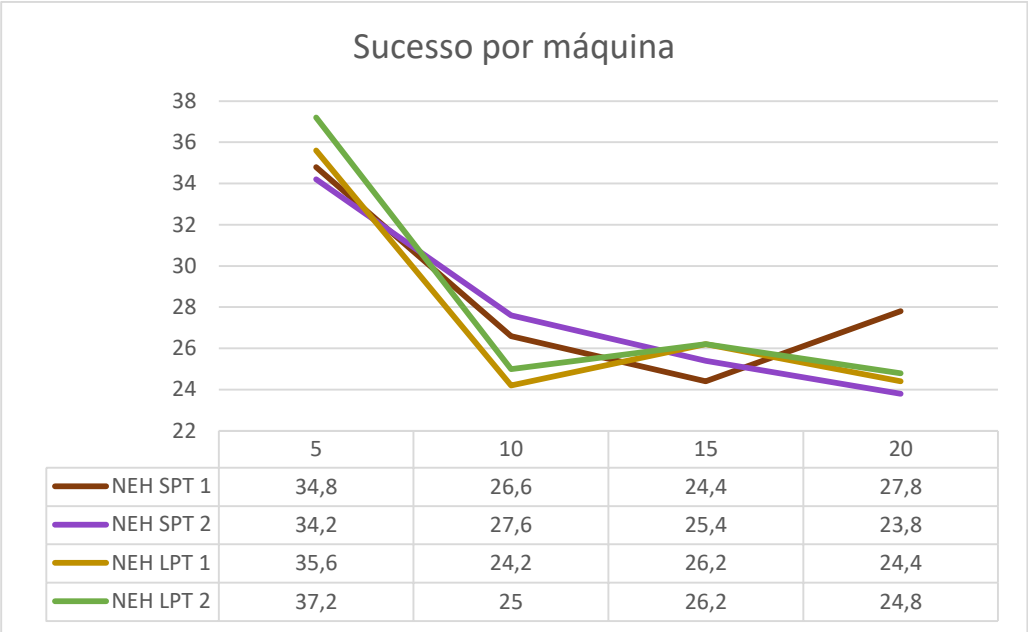
Fonte: Autoria Própria.

O gráfico de desvio relativo médio por tarefa deste primeiro grupo de tempos de *setup* afirma a análise realizada por meio da estatística de porcentagem de sucesso, já que o método NEH SPT 1 entre as duas primeiras categorias de tarefas possui um valor mais distante do sucesso encontrado e, portanto, maior desvio, enquanto que, à medida que o número de tarefas aumenta demonstra maior proximidade do melhor valor de *makespan* encontrado na comparação dos métodos construtivos analisados.

Apesar disso, o NEH SPT 1 não foi o método que durante toda a análise por quantidade de tarefas se manteve mais próximo dos melhores resultados, já que em média o método baseado na ordenação inicial LPT levando em consideração o *setup* para definição do sequenciamento de tarefas, o NEH LPT 1, resultou em um menor valor de DRM, indicando o método com menor discrepância em relação aos melhores resultados, apesar de não obter o sucesso.

Da mesma forma que foi realizado para as categorias de tarefas, a análise de PS e DRM também foi efetuada para as instâncias com a mesma quantidade de máquinas. Em um primeiro momento, analisou-se a porcentagem de sucesso, ilustrada pelo Gráfico 15.

Gráfico 15 - Sucesso por máquina do 1º grupo para o método NEH



	5	10	15	20
NEH SPT 1	34,8	26,6	24,4	27,8
NEH SPT 2	34,2	27,6	25,4	23,8
NEH LPT 1	35,6	24,2	26,2	24,4
NEH LPT 2	37,2	25	26,2	24,8

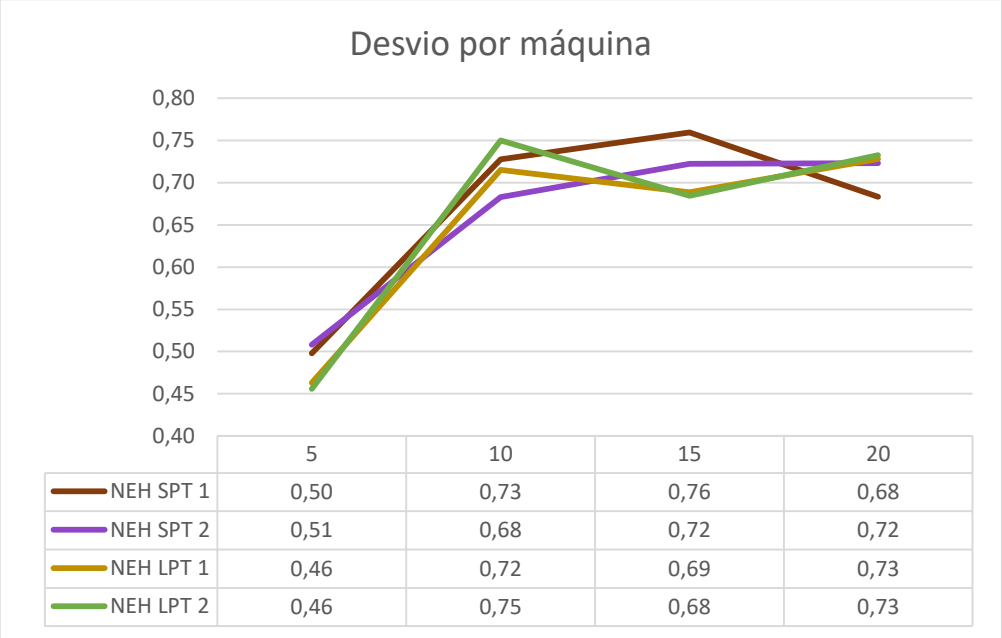
Fonte: Autoria Própria.

Ainda que o conjunto de análise tenha sido alterado de quantidade de tarefas por quantidade de máquinas, o desempenho dos métodos não se mostrou distinto do apresentado anteriormente, sendo que o método NEH SPT 1 resultou em uma maior quantidade de sucessos obtidos. No gráfico observa-se um comportamento semelhante entre os métodos até **n**=10, o que não implica em resultados iguais, mas sim em uma mesma tendência de desempenho já que todos os métodos apresentaram decréscimo nos resultados obtidos.

Apesar disso, é visto que enquanto os demais métodos declinam em suas respectivas soluções, o método NEH SPT 1 apresenta diferença positiva em média de 3,46 pontos, fator determinante para declaração do método com melhor performance.

O desvio relativo médio contribui para o entendimento do desempenho dos métodos analisados nesta seção, por isso, conjuntamente com a análise da PS, realizou-se o estudo do DRM para o conjunto de instâncias com a mesma quantidade de máquinas, esta análise pode, portanto, ser verificada por meio do Gráfico 16, que ilustra graficamente os resultados obtidos.

Gráfico 16 - DRM por máquina do 1º grupo para o método NEH



	5	10	15	20
NEH SPT 1	0,50	0,73	0,76	0,68
NEH SPT 2	0,51	0,68	0,72	0,72
NEH LPT 1	0,46	0,72	0,69	0,73
NEH LPT 2	0,46	0,75	0,68	0,73

Fonte: Autoria Própria.

Nota-se que a partir de **n**=10 todos os métodos aumentam a relação de desvio ao valor de sucesso e além disso, não é possível definir reflexão clara da situação de cada método quando comparado ao gráfico de porcentagem de sucesso, já que as ordenações que na análise anterior se mostraram mais eficientes, ou seja, obtiveram maiores valores em PS, não são as que, em média, desviaram menos quando comparado aos menores valores de *makespan* encontrados.

Como já discutido na análise realizada para as categorias de tarefas, apesar de não apresentarem os melhores valores nas comparações realizadas, os métodos construtivos NEH que baseiam a ordenação inicial no sequenciamento definido pelo LPT são os que em média se mantém mais próximos dos melhores valores da medida de desempenho de sucesso aplicada neste presente estudo.

Para este primeiro grupo de tempos de *setup* define-se então, tanto para as instâncias com mesma quantidade de tarefas, quanto para instâncias com mesma quantidade de máquinas, que o método NEH SPT 1 obteve maiores resultados na estatística de porcentagem de sucesso, apesar de possuir maior DRM, sendo que o método NEH LPT 1 foi o que se manteve mais próximo dos melhores resultados.

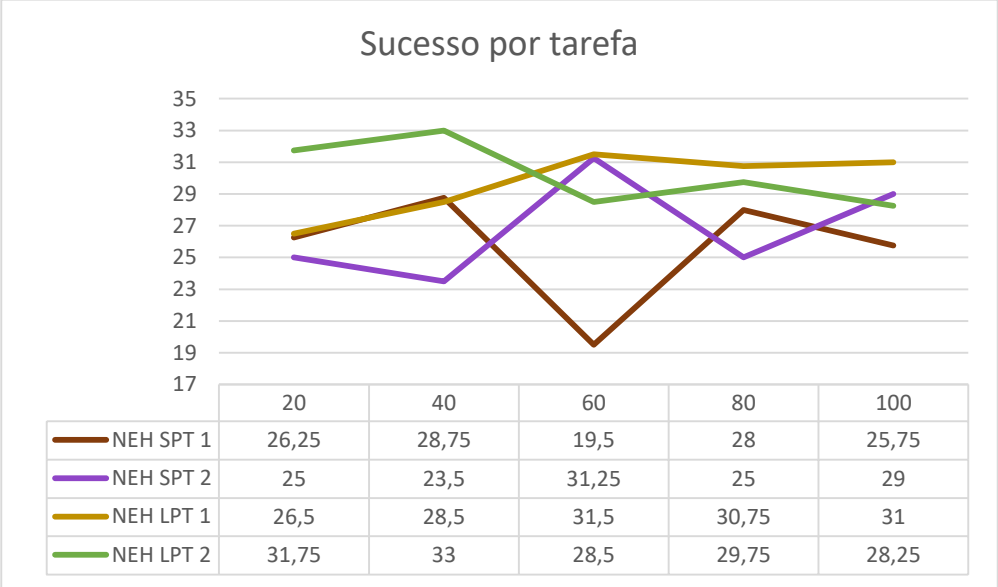
4.2.2 Segundo Grupo

Os tempos de *setup* que possuem como intervalo de variação de 1 a 50

unidades de tempo foram definidos como o segundo grupo de análise deste trabalho e para fins de comparação entre os métodos construtivos NEH, assim como realizado anteriormente, nesta seção terciária será apresentada a análise das estatísticas de PS e DRM.

De forma a averiguar neste grupo a porcentagem de sucesso dos métodos construtivos, analisou-se primeiramente os resultados obtidos nas instâncias de mesma quantidade de tarefas, e estes estão ilustrados no Gráfico 17.

Gráfico 17 - Sucesso por tarefa do 2º grupo para o método NEH



	20	40	60	80	100
NEH SPT 1	26,25	28,75	19,5	28	25,75
NEH SPT 2	25	23,5	31,25	25	29
NEH LPT 1	26,5	28,5	31,5	30,75	31
NEH LPT 2	31,75	33	28,5	29,75	28,25

Fonte: Autoria Própria.

Mais claramente que na análise do 1º grupo para os métodos construtivos NEH, as ordenações iniciais baseadas no LPT são as que apresentaram melhores resultados de sucesso, visto que as linhas que representam os métodos NEH LPT 1 e NEH LPT 2, em praticamente todas as classes de tarefas, se encontram em evidência. Já que quanto maior o valor do eixo vertical, que representa a quantidade de sucesso obtida pelos métodos analisados, menor são os valores de *makespan* encontrados, e por isso são também os métodos que melhor atendem a função objetiva que é de minimização.

Mesmo que haja uma pequena diferença entre a estatística de porcentagem de sucesso do NEH LPT 1 e NEH LPT 2, o método que se desempenhou mais positivamente liderando a comparação realizada é o NEH LPT 2, baseado na ordenação decrescente inicial apenas dos tempos de processamento das tarefas.

Esta conclusão também foi retratada no Gráfico 18, que ilustra o desvio relativo médio deste estudo.

Gráfico 18 - DRM por tarefa do 2º grupo para o método NEH

	20	40	60	80	100
NEH SPT 1	1,08	0,81	0,60	0,47	0,38
NEH SPT 2	1,10	0,75	0,53	0,50	0,38
NEH LPT 1	1,09	0,76	0,55	0,48	0,40
NEH LPT 2	1,05	0,70	0,56	0,46	0,40

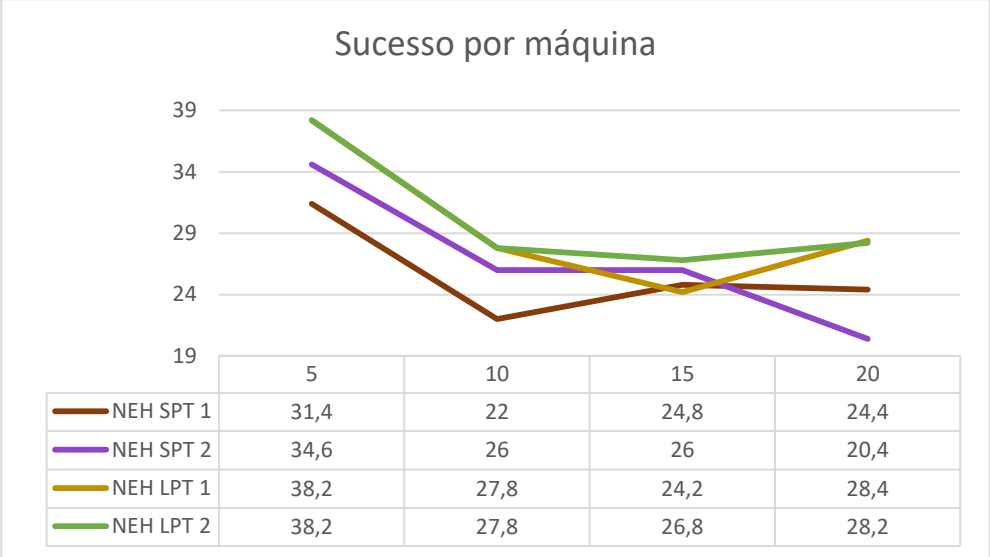
Fonte: Autoria Própria.

Quando se trata de desvio relativo comparado aos valores de sucesso encontrados pelos métodos analisados também é possível identificar a performance satisfatória dos métodos baseados na ordenação LPT, ainda mais da ordenação que não considera os tempos de *setup*.

Diferente do que foi apresentado para o primeiro grupo, em que os métodos com melhores resultados obtidos em valores de *makespan* não eram os mesmos que variavam menos em relação a todos os valores de sucesso, quando se trata do grupo de intervalos de variação entre 1 e 50 unidades de tempo, a performance dos métodos é retratada de maneira similar tanto para a estatística de PS quanto para a de DRM.

Além das análises realizadas para as classes de tarefas, ou seja, para instâncias que possuem a mesma quantidade de tarefas, também foram efetuadas a análise das estatísticas de estudo para instâncias com a mesma quantidade de **m**, sendo que a primeira análise está ilustrada no Gráfico 19.

Gráfico 19 - Sucesso por máquina do 2º grupo para o método NEH



	5	10	15	20
NEH SPT 1	31,4	22	24,8	24,4
NEH SPT 2	34,6	26	26	20,4
NEH LPT 1	38,2	27,8	24,2	28,4
NEH LPT 2	38,2	27,8	26,8	28,2

Fonte: Autoria Própria.

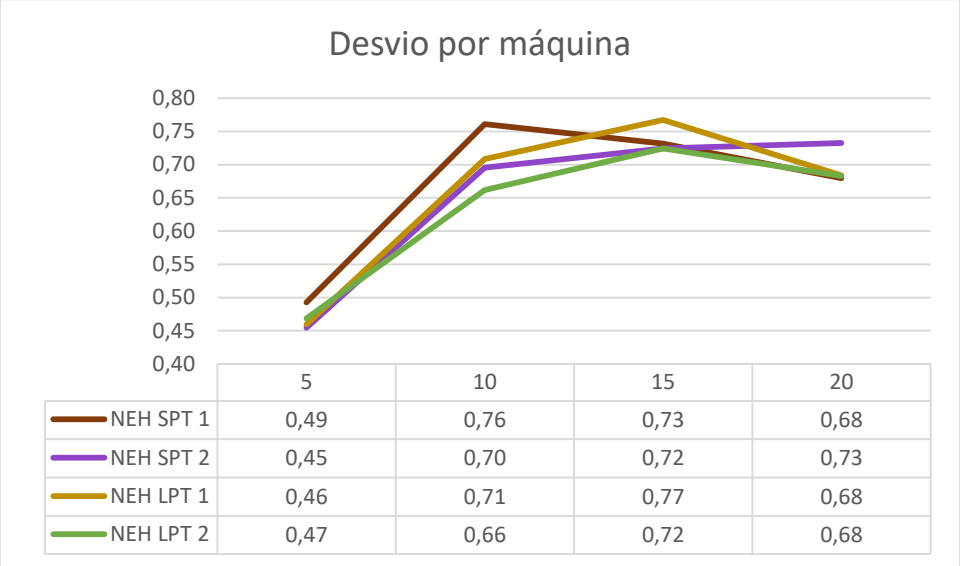
Assim como quando analisado por categorias de tarefas, à medida que analisamos as estatísticas de desempenho tal como a porcentagem de sucesso para as categorias de máquinas, verifica-se, com auxílio dos gráficos, que os métodos construtivos NEH baseados na ordenação inicial do método SPT não se apresentam tão eficientes quanto aos métodos que tem como princípio a ordenação inicial LPT.

No Gráfico 19 é evidenciado a satisfação obtida pelos resultados do método NEH LPT 2, já que a linha que o representa se mantém a frente dos demais métodos, alcançando os maiores valores nas três primeiras quantidades de **m**.

Observa-se também que em **m**=5 e **m**=10, os métodos NEH LPT 1 e NEH LPT 2 alcançaram o mesmo valor de sucesso, e apesar de na última categoria analisada (**m**=20), mesmo que em pequena proporção, o NEH LPT 1 obter um resultado mais satisfatório, quando a quantidade de máquinas é de 15, o NEH LPT 2 ganha vantagem em 2,6 pontos, garantindo, portanto, a posição do método com maior participação na porcentagem de sucesso.

O apontamento de desempenho dos métodos analisados neste trabalho para a classe de quantidade de máquinas também estão baseados na estatística de desvio relativo médio, ilustrado no Gráfico 20, auxiliando a identificar quais métodos dentro desta categoria no 2º grupo estão mais próximo ou mais distantes dos melhores resultados obtidos para todas as instâncias em estudo.

Gráfico 20 - DRM por máquina do 2º grupo para o método NEH



	5	10	15	20
NEH SPT 1	0,49	0,76	0,73	0,68
NEH SPT 2	0,45	0,70	0,72	0,73
NEH LPT 1	0,46	0,71	0,77	0,68
NEH LPT 2	0,47	0,66	0,72	0,68

Fonte: Autoria Própria.

Se comparado com a análise realizada para a categoria de tarefas, os resultados se diferenciam, de forma que não acompanham a tendência dos resultados de porcentagem de sucesso, já que os métodos construtivos que se mostraram mais eficientes ao obter maiores resultados de PS não são, em sua totalidade, os mesmos que se desviam menos dos melhores resultados.

Pode-se tomar como afirmativa, nesse caso, que os métodos que se baseiam em ordenações que consideram apenas os tempos de processamento para definição do sequenciamento de tarefas, sendo estes o NEH SPT 2 e NEH LPT 2, são os que menos se desviam dos melhores resultados de *makespan*, tomando como frente a ordenação inicial decrescente.

Ainda assim, de forma geral, o método que se mostrou mais eficiente nas duas estatísticas citadas, com melhores resultados de medida de desempenho bem como com valores mais próximos dos sucessos até mesmo quando não se mostrou mais eficiente, foi o NEH LPT 2, o método construtivo NEH com ordenação inicial baseada no LPT sem considerar os tempos de *setup* para o sequenciamento de tarefas.

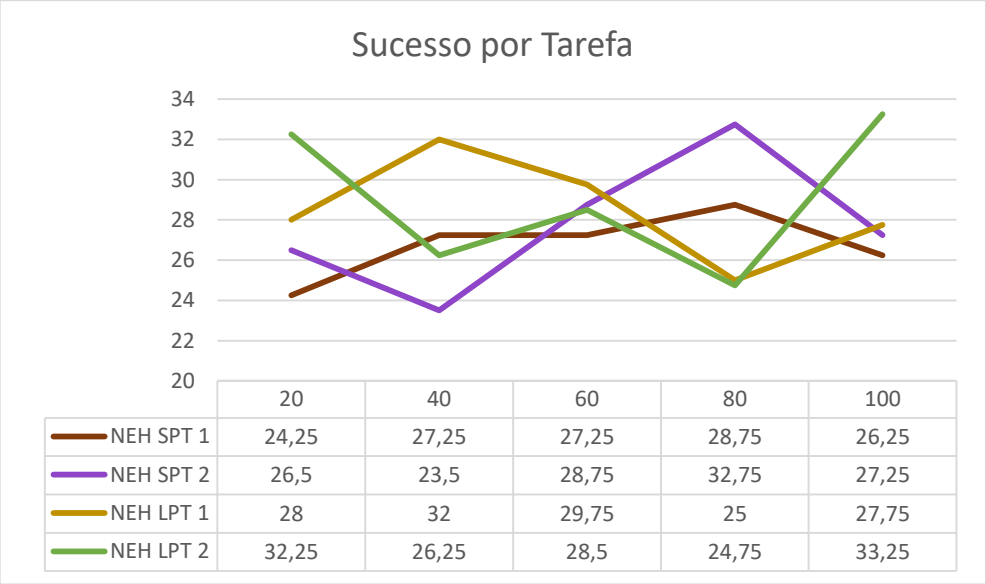
4.2.3 Terceiro Grupo

De modo a finalizar a análise do estudo dos quatro métodos construtivos NEH, observou-se as estatísticas de desempenho para o grupo de tempos de *setup*

com intervalo de variação entre 1 e 100 unidades de tempo, denominado como o terceiro grupo, e assim então serão apresentadas nesta seção.

O primeiro gráfico gerado, com intuito de auxiliar a verificação das performances dos métodos analisados neste estudo, é baseado nos valores resultantes da porcentagem de sucesso das instâncias classificadas pela quantidade de tarefas, ilustrado no Gráfico 21.

Gráfico 21 - Sucesso por tarefa do 3º grupo para o método NEH



	20	40	60	80	100
NEH SPT 1	24,25	27,25	27,25	28,75	26,25
NEH SPT 2	26,5	23,5	28,75	32,75	27,25
NEH LPT 1	28	32	29,75	25	27,75
NEH LPT 2	32,25	26,25	28,5	24,75	33,25

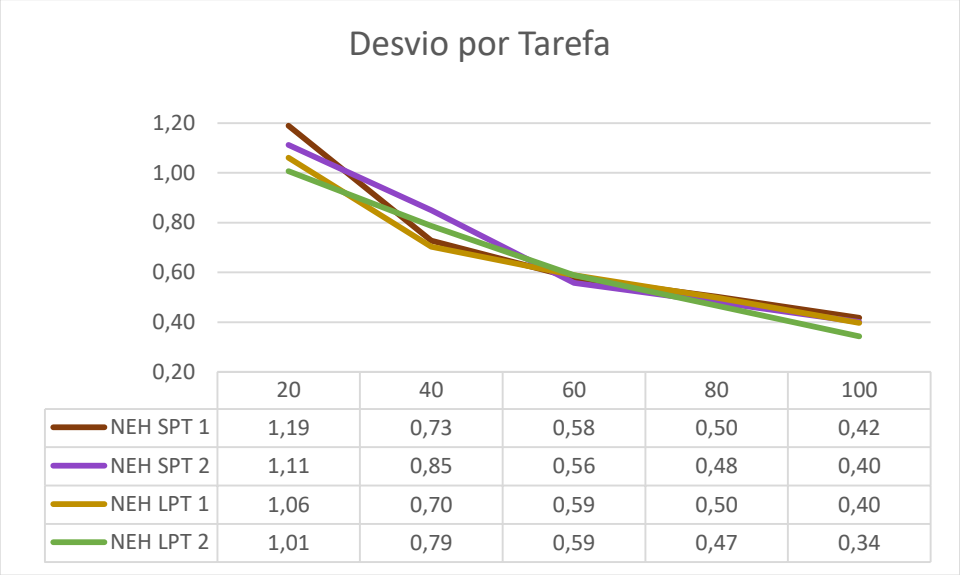
Fonte: Autoria Própria.

Ao realizar a análise deste gráfico, percebe-se que a maioria dos resultados dos métodos é marcada por uma considerável oscilação, demonstrando instabilidade dos sucessos dos métodos quando separados por instâncias com a mesma quantidade de tarefas. Sendo que o único método que possui uma linha de tendência mais linear é o NEH SPT 1, apesar de não resultar em grandes valores de sucesso.

Diante de dados tão variáveis ainda é possível verificar que os métodos NEH baseados na ordenação inicial LPT se destacam ao resultarem em altos valores de porcentagem de sucesso quando comparado aos demais. Entretanto, apesar de muita semelhança nos resultados obtidos, o método NEH LPT 2 se mostrou mais efetivo neste primeiro momento.

Para compreender ainda mais o comportamento dos métodos, deu-se continuidade nas análises da categoria de tarefas ao gerar o Gráfico 22 que representa o desvio relativo médio.

Gráfico 22 - DRM por tarefa do 3º grupo para o método NEH



	20	40	60	80	100
NEH SPT 1	1,19	0,73	0,58	0,50	0,42
NEH SPT 2	1,11	0,85	0,56	0,48	0,40
NEH LPT 1	1,06	0,70	0,59	0,50	0,40
NEH LPT 2	1,01	0,79	0,59	0,47	0,34

Fonte: Autoria Própria.

Ainda para a categoria de instâncias reunidas de acordo com a quantidade de tarefas, ao analisar a estatística de desempenho DRM, verifica-se a confirmação da análise realizada ao observar os resultados obtidos na PS dos métodos. Em outras palavras, os métodos que obtiveram os maiores resultados na porcentagem de sucesso, ou seja, resultaram em um menor valor de *makespan* por mais vezes que os demais, consequentemente também foram os métodos que apresentaram menor variação em média quando comparado aos resultados de sucesso.

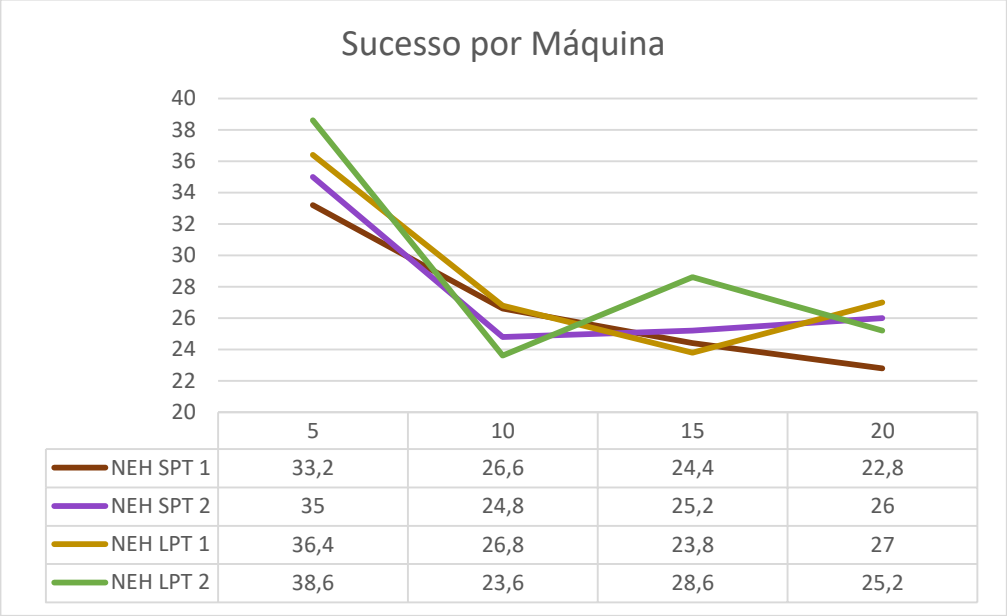
Observa-se também que todos os métodos representados no gráfico apresentaram decréscimo nos valores de desvio à medida que a quantidade de tarefas aumenta, além de apresentarem valores bem próximos entre si, como pode ser verificado pela tabela de dados contida no Gráfico 22, ou seja, o desvio dos valores encontrados pelos métodos em relação ao valor de sucesso oscila em uma mesma proporção.

De forma geral, para esta estatística, pode-se também notar que o método mais eficiente, é o NEH LPT 2, que baseado na ordenação sem considerar o tempo de *setup* apresenta uma menor variância quando comparado ao valor de sucesso de cada instância. Portanto, mesmo que não seja o método mais eficiente em todas as instâncias analisadas, sempre resulta em valores próximos do sucesso, demonstrando comportamento satisfatório.

Para fins de análise sob outra perspectiva, analisou-se para este terceiro

grupo de tempos de *setup* as mesmas estatísticas que anteriormente para a categoria de máquinas, ou seja, em instâncias que têm em comum a quantidade de **m**. Primeiramente, assim como ilustrado no Gráfico 23, verificou-se o desempenho da porcentagem de sucesso de cada método.

Gráfico 23 - Sucesso por máquina do 3º grupo para o método NEH



	5	10	15	20
NEH SPT 1	33,2	26,6	24,4	22,8
NEH SPT 2	35	24,8	25,2	26
NEH LPT 1	36,4	26,8	23,8	27
NEH LPT 2	38,6	23,6	28,6	25,2

Fonte: Autoria Própria.

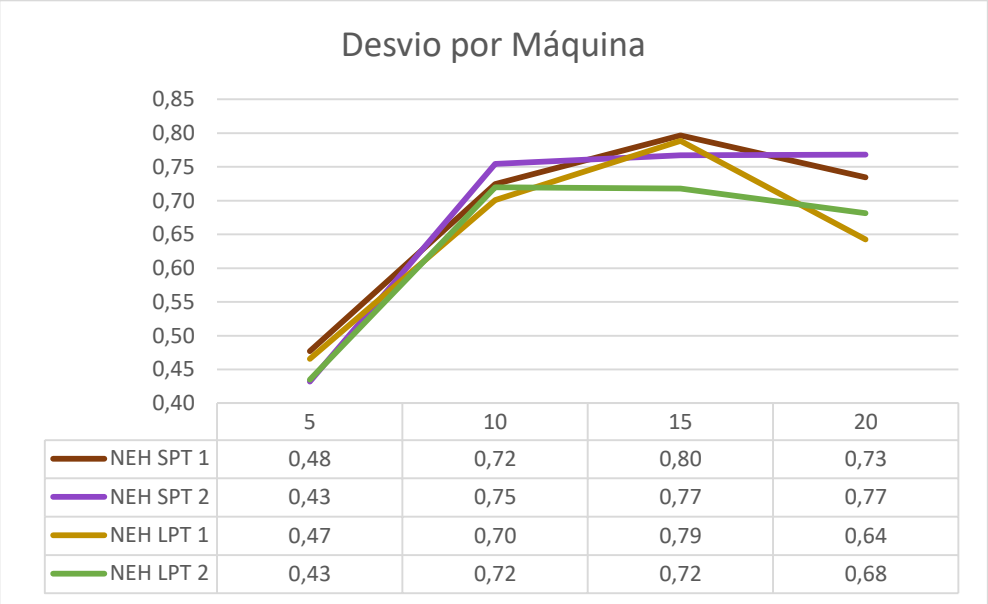
Ao realizar a troca da categoria como foco de análise para a categoria de máquinas, percebe-se uma melhora na distribuição de dados, isto é, uma maior linearidade entre os resultados obtidos, com valores não tão discrepantes como visto anteriormente na análise por tarefa.

Ainda que de forma diferente, nesta análise o método NEH LPT 2 também lidera a estatística de porcentagem de sucesso, em outras palavras, contribuiu, quando comparado com os demais métodos, de forma mais positiva para o encontro de valores satisfatórios no intuito de atender a função objetivo de minimização do *makespan*.

Em sequência do método NEH LPT 2, o método NEH LPT 1 também atinge bons resultados de porcentagem de sucesso. Portanto, ao analisar essa estatística, os métodos construtivos baseados em um sequenciamento de tarefas fundamentado na ordem decrescente de tempos de processamento e – para o método NEH LPT 1 – tempos de *setup* se mostram mais eficientes.

Por fim, ainda para a categoria de máquinas, analisou-se por meio do Gráfico 24, a estatística de DRM para os quatro métodos construtivos NEH.

Gráfico 24 - DRM por máquina do 3º grupo para o método NEH



	5	10	15	20
NEH SPT 1	0,48	0,72	0,80	0,73
NEH SPT 2	0,43	0,75	0,77	0,77
NEH LPT 1	0,47	0,70	0,79	0,64
NEH LPT 2	0,43	0,72	0,72	0,68

Fonte: Autoria Própria.

Por mais que não seja a linha representativa com menores valores de desvio em todas as quantidades de **m**, o NEH LPT 2 se mostra, em média, o método com índice de variação mais baixo se comparado com os valores de sucesso.

Fica evidente, também, que os métodos construtivos NEH baseados na ordenação inicial crescente, ou seja, SPT, são os métodos com maiores valores ilustrados no Gráfico 24, portanto, assim como observado no Gráfico 23, são métodos que demonstraram pouca eficiência para resolução de problemas cuja a função objetivo é minimização do *makespan*.

De forma geral, no terceiro grupo de tempos de *setup* os gráficos gerados para as estatísticas de PS e DRM, tanto para as categorias de máquinas quanto para as de tarefas, refletiam um mesmo cenário, em que métodos baseados no *Longest Processing Time* sem considerar o tempo de *setup* para definição do sequenciamento de tarefas são mais eficientes que os demais analisados.

4.3 ANÁLISE ENTRE OS MÉTODOS ESTUDADOS

Com o propósito de constatar a eficiência dos métodos analisados neste trabalho, e ainda, verificar a eficácia destes, para entender e definir entre os métodos

de sequenciamento SPT e LPT e os métodos construtivos qual pode ser considerado mais efetivo, estes foram comparados.

A análise entre os métodos ocorreu por meio da comparação dos resultados obtidos ao executar o programa em linguagem *Pascal* utilizado para realização do cálculo do *makespan* de acordo com as restrições que se referem ao método de sequenciamento de tarefas inicial, SPT ou LPT, ao intervalo de variação de unidade de tempo de *setup* e também no uso ou não deste tempo para a definição da ordenação das tarefas.

Assim como foi feito para analisar as estatísticas, a análise entre os tipos de métodos utilizados também foi separada por grupo de tempo de *setup*, isto é, os menores valores encontrados em cada instância entre o métodos SPT 1, SPT 2, LPT 1 e LPT 2 foram comparados com os menores valores para todas as instâncias entre o métodos NEH SPT 1, NEH SPT 2, NEH LPT 1 e NEH LPT 2 em cada um dos três grupos de tempos de *setup* definidos anteriormente.

Para cada grupo um gráfico foi gerado para ilustrar a comparação entre estes métodos, que estão representados pelas cores apresentadas na Figura 13. Os gráficos retratam a média dos valores de *makespan* encontrados nos 100 problemas de cada combinação de **m**x**n**, ou seja, a comparação tem como base as 20 combinações determinadas previamente para os métodos de ordenação SPT e LPT e para o método construtivo NEH.

Figura 6 - Cores representativas dos métodos analisados

Fonte: Autoria Própria.

A comparação realizada para o primeiro grupo, para o segundo grupo e para o terceiro grupo são apresentadas na Figura 14.

Figura 7 - Análise entre os métodos estudados.

Fonte: Autoria Própria.

Em todos os grupos de intervalos de variação de tempos de *setup*, como pode ser observados nos gráficos dispostos anteriormente, o método construtivo NEH, em todas as instâncias analisadas, obteve menores valores de *makespan* quando comparado aos métodos de sequenciamento SPT e LPT.

Os resultados do método construtivos NEH supreenderam por não definir como relevante a utilização dos tempos de *setup* na definição inicial do sequenciamento de tarefas, já que o NEH LPT 2 foi o que desempenhou melhor frente aos demais. Entretanto, apesar disso, resultou em valores mais baixos de *makespan* em todas as instâncias observadas em relação aos métodos SPT e LPT, portanto, discute-se a efetividade do método baseado na sua configuração de determinar a ordenação em duas fases, e isso independe das restrições dispostas.

Observa-se também que a medida que a quantidade de tarefas vai aumentando, a distância entre as linhas que representam os métodos são maiores, isto muito se deve pela maior possibilidade de variação entre os tempos tanto de processamento quanto de *setup*. Assim como a proporção da diferença dos

resultados obtidos aumenta quando há um maior intervalo de variação dos tempos de *setup*, em outras palavras, a diferença média entre os valores obtidos pelos métodos SPT e LPT e pelo NEH no primeiro grupo é de 738,14 unidades de tempo, para o segundo grupo esse valor equivale a 739,82 e no terceiro grupo é de 747,32 unidades de tempo.

5 CONSIDERAÇÕES FINAIS

Com o intuito de averiguar e entender as reais necessidades e situações encontradas nas indústrias, o presente trabalho buscou analisar a programação da produção ao considerar não somente os tempos de processamento, bem como os tempos de *setup* demandados pelas tarefas nas máquinas utilizadas nos processos produtivos.

Para entender o impacto causado por essa informação, geralmente omitida, no sequenciamento de tarefas e consequentemente na linha de produção, utilizou-se como base o cálculo do *makespan*, ou também definido como tempo total de um processo.

Ao atender os objetivos específicos traçados em um primeiro momento, a análise de métodos heurísticos e construtivos atuais, SPT, LPT e NEH, foi realizada para que o desempenho destes pudesse ser comparado.

Em um primeiro momento, a análise foi realizada comparando os métodos de sequenciamento de uma fase, ou seja, a comparação foi efetuada entre SPT 1, SPT 2, LPT 1 e LPT 2, e de forma geral, o LPT apresentou, em todos os grupos de tempos de *setup* do estudo, um melhor desempenho, com a maior quantidade de sucesso diante dos demais. Já quando a análise foi realizada entre os métodos construtivos NEH SPT 1, NEH SPT 2, NEH LPT 1 e NEH LPT 2, a ordenação inicial baseada na ordem decrescente obteve boa performance na maioria dos grupos, com exceção apenas do grupo de *setup* com intervalo de variação entre 1 e 25 unidades de tempo.

O sucesso representa a quantidade de vezes que resultou em um menor valor dentre os quatro métodos de sequenciamento avaliados em cada análise. Ou seja, conclui-se que o melhor desempenho foi do método *Longest Processing Time* frente à função objetivo de *makespan,* em um problema de programação *flowshop* de um sistema de produção clássico tanto para métodos heurísticos SPT e LPT quanto para métodos construtivos.

O método heurístico LPT, independentemente de quais tempos se baseou para realização da ordenação, de fato se mostrou o mais adequado para solucionar problemas de programação da produção com ênfase em minimização do tempo total do processo, como já evidenciado na revisão de literatura deste estudo. Isso se faz

verdade visto o melhor desempenho obtido na avaliação dos oito métodos de sequenciamento observados.

De formar a possibilitar a adequada análise de um problema de sequenciamento no ambiente produtivo, o exame de métodos de sequenciamento baseados em tempos de processamento e métodos baseados em tempos de processamento e tempo de *setup*, para os métodos de ordenação de apenas uma fase, evidencia a importância de considerar os tempos de configuração no processo produtivo, visto que no segundo e terceiro grupo de análise deste estudo os métodos de ordenação que englobavam os tempos de *setup* demonstraram melhor performance, ou seja, atenderam a especificação da função objetivo, a minimização do *makespan*.

Ademais, é observável que a medida que os tempos de *setup* possuem um intervalo de variação maior, estes impactam cada vez mais em um sequenciamento e em um valor de *makespan* distinto ao método de ordenação baseado na soma apenas dos tempos de processamento, pois apesar de não serem avaliados durante a ordenação de tarefas, ainda são considerados na realização das tarefas propriamente dita.

Diante dessa conclusão, é possível entender e explicar o porquê do método LPT 1 não ter sido o de melhor desempenho no primeiro grupo estudado, visto que o intervalo de variação dos tempos de *setup* desse grupo, entre 1 a 25 unidades de tempo, é relativamente baixo quando comparado ao intervalo de variação dos tempos de processamento, de 1 a 100 unidades de tempo.

Entretanto, quando tratamos de métodos que também são baseados inicialmente em métodos de ordenação como o SPT ou LPT, mas determinam a sequência de tarefas de maneira construtiva, é possível verificar que apesar do LPT possuir uma boa performance, os tempos de *setup* para determinação da ordem de tarefas não se faz tão necessário.

De certa forma, nesta segunda análise, os tempos de *setup* podem se tornar dispensáveis no auxílio do sequenciamento de tarefas devido ao fato de que o método NEH garante em suas duas fases de construção de sequência, a observação e análise, por meio dos resultados de *makespan* de cada etapa, de uma maior variedade de possibilidade de ordens de tarefas. Sendo assim, o método não fica dependente

apenas dos tempos de processamento e tempos de *setup*, considerando apenas como base inicial um sequenciamento em ordem crescente ou descrente da soma desses tempos, e entende que sequências derivadas também podem se mostrar mais eficientes.

Percebe-se ainda, por meio dos gráficos gerados durante o estudo, que os dados gerados, os cálculos realizados e o programa codificado remetem confiabilidade, à medida que as duas estatísticas analisadas se complementam. Em outras palavras, os métodos de ordenação que não resultam em elevados valores de sucesso, geralmente, são os mesmos que são ilustrados nos gráficos com altos valores de desvio relativo, assim como o contrário também se faz verdade.

Contudo, é visto que não há uma regra de ordenação clara para todos os possíveis casos de solução para problemas *flowshop*, especialmente quando possuem o tempo de *setup* separado dos tempos de processamento. De certa forma, cada método heurístico desempenha-se de maneira diferente devido a limitações traçadas por suas próprias restrições, assim como por determinadas circunstâncias de cada situação em que é utilizado.

Também foram comparados os resultados encontrados na análise 1 (SPT 1, SPT 2, LPT 1 e LPT2) com os resultados encontrados na análise 2 (NEH SPT 1, NEH SPT 2, NEH LPT 1 e NEH LPT 2), para entender qual dos métodos, construtivos ou não, se mostrou mais eficaz.

E assim foi possível verificar que, independente de qual tipo de ordenação toma-se como base, a crescente ou decrescente, ou se os tempos de *setup* são levados em consideração apenas no cálculo da medida de desempenho ou também são relevantes na definição de ordem das tarefas, o método construtivo NEH em todas as instâncias para todos os grupos de *setup* analisados, garantiu um melhor desempenho, ou seja, a performance do NEH comparado com os métodos LPT e SPT em todos os momentos alcançou de maneira mais satisfatória a função objetiva de minimização do *makespan*.

Mais uma vez neste presente estudo, portanto, a revisão da literatura se faz verdadeira, já que o método construtivo NEH é referência para criação de outros métodos heurísticos e garante em suas duas fases identificar uma melhor sequência de tarefas.

Apesar do resultado satisfatório do estudo, alcançado o objetivo geral e os objetivos específicos, para que o trabalho tenha um caráter de melhoria contínua, sugestões de trabalhos futuros e aprimoramento são dispostas para um estudo ainda mais complexo:

- Implementação do estudo em ambiente fabril;

- Aperfeiçoamento do algoritmo utilizado para compreender modelos mais complexos;

- Análise e comparação de mais métodos de ordenação atuais;

- Análise de *flowtime* com a consideração dos tempos de *setup*;

- Análise e comparação de outros métodos heurísticos.

De forma geral o trabalho atendeu as expectativas e realizou de forma positiva uma contribuição para a literatura de *scheduling*. As melhorias se farão verdade à medida que, o presente estudo seja aplicado em ambiente fabril.

REFERÊNCIAS BIBLIOGRÁFICAS

ADIRI, I E POHORYLES, D. Flowshop/no-idle or no-wait scheduling to minimize the sum of completion times, Naval Research Logistic Quartely, 29, 495-504, 1982.

AKBARI, Mehdi; RASHIDI, Hassan; ALIZADEH, Sasan H.. An enhanced genetic algorithm with new operators for task scheduling in heterogeneous computing systems. Engineering Applications Of Artificial Intelligence, [s.l.], v. 61, p.35-46, maio 2017. Elsevier BV.

AL-ANZI, Fawaz S.; ALLAHVERD, Ali. A Hybrid Tabu Search Heuristic for the Two-Stage Assembly Scheduling Problem. International Journal Of Operations Research, [s.l.], v. 3, p.109-119, maio 2006.

AL-ANZI, Fawaz S.; ALLAHVERDI, Ali. Heuristics for a two-stage assembly flowshop with bicriteria of maximum lateness and makespan. Computers & Operations Research, [s.l.], v. 36, n. 9, p.2682-2689, set. 2009. Elsevier BV.

ALLAHVERDI, Ali; ALDOWAISAN, Tariq. Minimizing total completion time in a no-wait flowshop with sequence-dependent additive changeover times. Journal Of The Operational Research Society, [s.l.], v. 52, n. 4, p.449-462, abr. 2001. Springer Nature.

ALLAHVERDI, Ali; ALDOWAISAN, Tariq. No-wait and separate setup three-machine flowshop with total completion time criterion. International Transactions In Operational Research, [s.l.], v. 7, n. 3, p.245-264, 1 maio 2000. Wiley-Blackwell.

ALDOWAISAN, Tariq; ALLAHVERDI, Ali. New heuristics for m-machine no-wait flowshop to minimize total completion time. Omega, [s.l.], v. 32, n. 5, p.345-352, out. 2004. Elsevier BV.

ALDOWAISAN, Tariq; ALLAHVERDI, Ali. Total flowtime in no-wait flowshops with separated setup times. Computers & Operations Research, [s.l.], v. 25, n. 9, p.757-765, set. 1998. Elsevier BV.

ARNOLD, J.R.T.. Introduction to Materials Management. 2. ed. New Jersey: Prentice

Hall International Editions, 1996.
ARROYO, José Elias Claudio; PEREIRA, Ana Amélia de Souza. A GRASP heuristic for the multi-objective permutation flowshop scheduling problem. The International Journal Of Advanced Manufacturing Technology, [s.l.], v. 55, n. 5-8, p.741-753, 23 dez. 2010. Springer Nature.

AZADEH, A. et al. A multi-objective optimization problem for multi-state series-parallel systems: A two-stage flow-shop manufacturing system. Reliability Engineering & System Safety, [s.l.], v. 136, p.62-74, abr. 2015. Elsevier BV.

BERTOLISSI, Edy. Heuristic algorithm for scheduling in the no-wait flow-shop. Journal Of Materials Processing Technology, [s.l.], v. 107, n. 1-3, p.459-465, nov. 2000. Elsevier BV.

BONNEY, M. C.; GUNDRY, S. W.. Solutions to the Constrained Flowshop Sequencing Problem. Journal Of The Operational Research Society, [s.l.], v. 27, n. 4, p.869-883, 1 jan. 1976. Springer Nature.

BRANCO, Fábio José Ceron; NAGANO, Marcelo Seido; MOCCELIN, João Vitor. Análise Multi-critério na programação da produção em sistemas no-wait flow shop. Gestão Industrial, Ponta Grossa, v. 4, n. 2, p.65-77, 2008.

BRANCO, Fábio José Ceron. Um novo método heurístico construtivo de alto desempenho para o problema no-idle flow shop. 2011. 112 f. Tese (Doutorado) - Curso de Engenharia de Produção, –- Escola de Engenharia de São Carlos da Universidade de São Paulo, São Carlos, 2011. Disponível em: <http://www.teses.usp.br/teses/disponiveis/18/18140/tde-05092011-094838/pt-br.php>. Acesso em: 18 abr. 2017.

BUZZO, Walther Rogério; MOCCELLIN, João Vitor. Programação da produção em sistemas flow shop utilizando um método heurístico híbrido algoritmo genético-simulated annealing. Gestão & Produção, [s.l.], v. 7, n. 3, p.364-377, dez. 2000. FapUNIFESP (SciELO).

CABRERA, Guillem et al. Promoting green internet computing throughout simulation-optimization scheduling algorithms. 2013 Winter Simulations Conference (wsc), [s.l.], dez. 2013. IEEE.

CAMPBELL, Herbert G.; DUDEK, Richard A.; SMITH, Milton L.. A Heuristic Algorithm for thenJob,mMachine Sequencing Problem. Management Science, [s.l.], v. 16, n. 10, p.630-637, jun. 1970. Institute for Operations Research and the Management Sciences (INFORMS).

CEPEK, Ondrej; OKADA, Masanori; VLACH, Milan. Note: On the two-machine no-idle flowshop problem. Naval Research Logistics, [s.l.], v. 47, n. 4, p.353-358, jun. 2000. Wiley-Blackwell.

CHEN, Chuen-lung; VEMPATI, Venkateswara S.; ALJABER, Nasser. An application of genetic algorithms for flow shop problems. European Journal Of Operational Research, [s.l.], v. 80, n. 2, p.389-396, jan. 1995.

CHENG, Mingbao; SUN, Shijie; HE, Longmin. Flow shop scheduling problems with deteriorating jobs on no-idle dominant machines. European Journal Of Operational Research, [s.l.], v. 183, n. 1, p.115-124, nov. 2007. Elsevier BV.

CHENG, T. C. Edwin; GUPTA, Jatinder N. D.; WANG, Guoqing. A REVIEW OF FLOWSHOP SCHEDULING RESEARCH WITH SETUP TIMES. Production And Operations Management, [s.l.], v. 9, n. 3, p.262-282, 5 jan. 2009. Wiley-Blackwell.

CHIH, Mingchang et al. Particle swarm optimization with time-varying acceleration coefficients for the multidimensional knapsack problem. Applied Mathematical Modelling, [s.l.], v. 38, n. 4, p.1338-1350, fev. 2014. Elsevier BV.

CHIH, Mingchang. Self-adaptive check and repair operator-based particle swarm optimization for the multidimensional knapsack problem. Applied Soft Computing, [s.l.], v. 26, p.378-389, jan. 2015. Elsevier BV.

CIAVOTTA, Michele; MINELLA, Gerardo; RUIZ, Rubén. Multi-objective sequence dependent setup times permutation flowshop: A new algorithm and a comprehensive study. European Journal Of Operational Research, [s.l.], v. 227, n. 2, p.301-313, jun. 2013. Elsevier BV.

CORRÊA, Henrique Luiz; GIANESI, Irineu Gustavo Nogueira; CAON, Mauro. Planejamento, Programação e Controle da Produção. 5. ed. São Paulo: Altas S.A, 2012.

DANNENBRING, David G.. An Evaluation of Flow Shop Sequencing

Heuristics. Management Science, [s.l.], v. 23, n. 11, p.1174-1182, jul. 1977. Institute for Operations Research and the Management Sciences (INFORMS).

FRAMINAN, J. M.; LEISTEN, R.; RAJENDRAN, C.. Different initial sequences for the heuristic of Nawaz, Enscore and Ham to minimize makespan, idletime or flowtime in the static permutation flowshop sequencing problem. International Journal Of Production Research, [s.l.], v. 41, n. 1, p.121-148, jan. 2003. Informa UK Limited.

FRAMINAN, Jose M.; NAGANO, Marcelo S.. Evaluating the performance for makespan minimisation in no-wait flowshop sequencing. Journal Of Materials Processing Technology, [s.l.], v. 197, n. 1-3, p.1-9, fev. 2008. Elsevier BV.

FRENCH, Simon. Sequencing and Scheduling: An Introduction to the Mathematics of the Jobshop. West Sussex, England: Ellis Horwood Ltd., 1982.

GANGADHARAN, Rajesh; RAJENDRAN, Chandrasekharan. Heuristic algorithms for scheduling in the no-wait flowshop. International Journal Of Production Economics, [s.l.], v. 32, n. 3, p.285-290, nov. 1993. Elsevier BV.

GAO, Jian; CHEN, Rong; DENG, Wu. An efficient tabu search algorithm for the distributed permutation flowshop scheduling problem. International Journal Of Production Research, [s.l.], v. 51, n. 3, p.641-651, fev. 2013. Informa UK Limited.

GAREY, M. R.; JOHNSON, D. S.; SETHI, Ravi. The Complexity of Flowshop and Jobshop Scheduling. Mathematics Of Operations Research, [s.l.], v. 1, n. 2, p.117-129, maio 1976. Institute for Operations Research and the Management Sciences (INFORMS).

GEIGER, Martin Josef. Decision support for multi-objective flow shop scheduling by the Pareto Iterated Local Search methodology. Computers & Industrial Engineering, [s.l.], v. 61, n. 3, p.805-812, out. 2011. Elsevier BV.

GHODRATNAMA, A.; JOLAI, F.; TAVAKKOLI-MOGHADDAM, R.. Solving a new multi-objective multi-route flexible flow line problem by multi-objective particle swarm optimization and NSGA-II. Journal Of Manufacturing Systems, [s.l.], v. 36, p.189-202, jul. 2015. Elsevier BV.

GUPTA, Jatinder N. D.. A Functional Heuristic Algorithm for the Flowshop Scheduling Problem. Journal Of The Operational Research Society, [s.l.], v. 22, n. 1, p.39-47, mar. 1971. Informa UK Limited.

GRAHAM, R. L.. Bounds on Multiprocessing Timing Anomalies. Siam Journal On Applied Mathematics, [s.l.], v. 17, n. 2, p.416-429, mar. 1969. Society for Industrial & Applied Mathematics (SIAM).

HATAMI, Sara; RUIZ, Rubén; ANDRÉS-ROMANO, Carlos. Heuristics and metaheuristics for the distributed assembly permutation flowshop scheduling problem with sequence dependent setup times. International Journal Of Production Economics, [s.l.], v. 169, p.76-88, nov. 2015. Elsevier BV.

HUANG, Simin; CAI, Linning; ZHANG, Xiaoyue. Parallel dedicated machine scheduling problem with sequence-dependent setups and a single server. Computers & Industrial Engineering, [s.l.], v. 58, n. 1, p.165-174, fev. 2010. Elsevier BV.

ISHIBUCHI, H.; YOSHIDA, T.; MURATA, T.. Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling. Ieee Transactions On Evolutionary Computation, [s.l.], v. 7, n. 2, p.204-223, abr. 2003. Institute of Electrical and Electronics Engineers (IEEE).

JASZKIEWICZ, Andrzej. Genetic local search for multi-objective combinatorial optimization. European Journal Of Operational Research, [s.l.], v. 137, n. 1, p.50-71, fev. 2002. Elsevier BV.

JOHNSON, S. M.. Optimal two- and three-stage production schedules with setup times included. Naval Research Logistics Quarterly, [s.l.], v. 1, n. 1, p.61-68, mar. 1954. Wiley-Blackwell.

JUNG, Sunwoong; WOO, Young-bin; KIM, Byung Soo. Two-stage assembly scheduling problem for processing products with dynamic component-sizes and a setup time. Computers & Industrial Engineering, [s.l.], v. 104, p.98-113, fev. 2017. Elsevier BV.

KENNEDY, J.; EBERHART, R.. Particle swarm optimization. Proceedings Of Icnn'95 - International Conference On Neural Networks, [s.l.], v. 1944, p.1942-1948, jan. 1995. IEEE.

KING, J. R.; SPACHIS, A. S.. Heuristics for flow-shop scheduling. International Journal Of Production Research, [s.l.], v. 18, n. 3, p.345-357, maio 1980. Informa UK Limited.

LEISTEN, Rainer. Flowshop sequencing problems with limited buffer storage. International Journal Of Production Research, [s.l.], v. 28, n. 11, p.2085-2100, nov. 1990. Informa UK Limited.

LIANG, J. J.; PAN, Quan-ke; CHEN, Tie-jun. A Dynamic Multi-swarm Particle Swarm Optimizer for blocking flow shop scheduling. 2010 Ieee Fifth International Conference On Bio-inspired Computing: Theories and Applications (BIC-TA), [s.l.], set. 2010. IEEE.

LIN, Shih-wei; YING, Kuo-ching; HUANG, Chien-yi. Minimising makespan in distributed permutation flowshops using a modified iterated greedy algorithm. International Journal Of Production Research, [s.l.], v. 51, n. 16, p.5029-5038, ago. 2013. Informa UK Limited.

LIN, Shih-wei; YING, Kuo-ching. Optimization of makespan for no-wait flowshop scheduling problems using efficient matheuristics. Omega, [s.l.], v. 64, p.115-125, out. 2016. Elsevier BV.

MACCARTHY, B. L.; LIU, Jiyin. Addressing the gap in scheduling research: a review of optimization and heuristic methods in production scheduling. International Journal Of Production Research, [s.l.], v. 31, n. 1, p.59-79, jan. 1993. Informa UK Limited.

MAENHOUT, Broos; VANHOUCKE, Mario. An exact algorithm for an integrated project staffing problem with a homogeneous workforce. Journal Of Scheduling, [s.l.], v. 19, n. 2, p.107-133, 28 ago. 2015. Springer Nature.

MEMARI, Ashkan; RAHIM, Abdul Rahman Abdul; AHMAD, Robiah Binti. An Integrated Production-distribution Planning in Green Supply Chain: A Multi-objective Evolutionary Approach. Procedia Cirp, [s.l.], v. 26, p.700-705, 2015. Elsevier BV.

MURATA, Tadahiko; ISHIBUCHI, Hisao; TANAKA, Hideo. Genetic algorithms for flowshop scheduling problems. Computers & Industrial Engineering, [s.l.], v. 30, n. 4, p.1061-1071, set. 1996. Elsevier BV.

M. R. Garey, D. S. Johnson, Computers and intractability: a guide to the theory of NP-Completeness, W. H. Freeman, New York, 1979.

NAGANO, M S; MOCCELLIN, J V. A high quality solution constructive heuristic for flow shop sequencing. Journal Of The Operational Research Society, [s.l.], v. 53, n. 12, p.1374-1379, 7 nov. 2002. Springer Nature.

NAGANO, Marcelo Seido; SILVA, Augusto Almeida da; LORENA, Luiz Antonio

Nogueira. An evolutionary clustering search for the no-wait flow shop problem with sequence dependent setup times. Expert Systems With Applications, [s.l.], v. 41, n. 8, p.3628-3633, jun. 2014. Elsevier BV.

NAGAR, Amit; HADDOCK, Jorge; HERAGU, Sunderesh. Multiple and bicriteria scheduling: A literature survey. European Journal Of Operational Research, [s.l.], v. 81, n. 1, p.88-104, fev. 1995. Elsevier BV.

NAWAZ, Muhammad; ENSCORE, e Emory; HAM, Inyong. A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. Omega, [s.l.], v. 11, n. 1, p.91-95, jan. 1983. Elsevier BV.

PALMER, D. S.. Sequencing Jobs Through a Multi-Stage Process in the Minimum Total Time—A Quick Method of Obtaining a Near Optimum. Journal Of The Operational Research Society, [s.l.], v. 16, n. 1, p.101-107, mar. 1965. Informa UK Limited.

PAN, Quan-ke; RUIZ, Rubén. An effective iterated greedy algorithm for the mixed no-idle permutation flowshop scheduling problem. Omega, [s.l.], v. 44, p.41-50, abr. 2014. Elsevier BV.

PINEDO, Michael L.. Planing and Scheduling in Manufacturing and Services. New York: Springer, 2005.

RAJENDRAN, Chandrasekharan. A No-Wait Flowshop Scheduling Heuristic to Minimize Makespan. Journal Of The Operational Research Society, [s.l.], v. 45, n. 4, p.472-478, abr. 1994. Springer Nature.

RAJENDRAN, Chandrasekharan; CHAUDHURI, Dipak. Heuristic algorithms for continuous flow-shop problem. Naval Research Logistics, [s.l.], v. 37, n. 5, p.695-705, out. 1990. Wiley-Blackwell.

RAMESH, S.; KANNAN, S.; BASKAR, S.. Application of modified NSGA-II algorithm to multi-objective reactive power planning. Applied Soft Computing, [s.l.], v. 12, n. 2, p.741-753, fev. 2012. Elsevier BV.

RAVI, Peruvemba Sundaram; TUNÇEL, Levent; HUANG, Michael. Worst-case performance analysis of some approximation algorithms for minimizing makespan and flowtime. Journal Of Scheduling, [s.l.], v. 19, n. 5, p.547-561, 9 jan. 2016. Springer Nature.

REDDI, S. S.; RAMAMOORTHY, C. V.. On the Flow-Shop Sequencing Problem with No Wait in Process. Journal Of The Operational Research Society, [s.l.], v. 23, n. 3, p.323-331, set. 1972. Springer Nature.

REEVES, Colin R.. A genetic algorithm for flowshop sequencing. Computers & Operations Research, [s.l.], v. 22, n. 1, p.5-13, jan. 1995.

RIAHI, Vahid et al. Scatter search for mixed blocking flowshop scheduling. Expert Systems With Applications, [s.l.], v. 79, p.20-32, ago. 2017. Elsevier BV.

RIBAS, Imma; COMPANYS, Ramon; TORT-MARTORELL, Xavier. An efficient Discrete Artificial Bee Colony algorithm for the blocking flow shop problem with total flowtime minimization. Expert Systems With Applications, [s.l.], v. 42, n. 15-16, p.6155-6167, set. 2015. Elsevier BV.

RONCONI, D P; A ARMENTANO, V. Lower bounding schemes for flowshops with blocking in-process. Journal Of The Operational Research Society, [s.l.], v. 52, n. 11, p.1289-1297, nov. 2001. Springer Nature.

RUIZ, Rubén; STÜTZLE, Thomas. A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem. European Journal Of Operational Research, [s.l.], v. 177, n. 3, p.2033-2049, mar. 2007. Elsevier BV.

RUIZ, Rubén; VALLADA, Eva; FERNÁNDEZ-MARTÍNEZ, Carlos. Scheduling in Flowshops with No-Idle Machines. Computational Intelligence In Flow Shop And Job Shop Scheduling, [s.l.], p.21-51, 2009. Springer Berlin Heidelberg.

RUSSOMANO, Victor Henrique. Planejamento e Controle da Produção. 6. ed. São Paulo: Pioneira, 2000.

SAADANI, Nour El Houda; GUINET, Alain; MOALLA, Mohamed. Three stage no-idle flow-shops. Computers & Industrial Engineering, [s.l.], v. 44, n. 3, p.425-434, mar. 2003. Elsevier BV.

SANTOS, Hamilton Carlos Massaro; FRANÇA, Paulo Morelato. Meta-heurística para programação da produção com tempos de preparação dependentes da seqüência. Genetics And Molecular Biology, [s.l.], v. 2, n. 3, p.228-243, dez. 1995. FapUNIFESP (SciELO).

SAVINO, Matteo Mario et al. Dynamic batch scheduling in a continuous cycle-

constrained production system. International Journal Of Services Operations And Informatics, [s.l.], v. 5, n. 4, p.313-329, 2010. Inderscience Publishers.

SAYGIN, C.; KILIC, S. E.. Integrating Flexible Process Plans with Scheduling in Flexible Manufacturing Systems. The International Journal Of Advanced Manufacturing Technology, [s.l.], v. 15, n. 4, p.268-280, 27 abr. 1999. Springer Nature.

SLACK, Nigel et al. Administração da Produção. São Paulo: Atlas S.A, 1999

SLACK, Nigel; CHAMBERS, Stuart; JHONSTON, Robert. Administração da Produção. 3. ed. São Paulo: Atlas S.A, 2009.

TAILLARD, E.. Benchmarks for basic scheduling problems. European Journal Of Operational Research, [s.l.], v. 64, n. 2, p.278-285, jan. 1993. Elsevier BV.

TAILLARD, E.. Some efficient heuristic methods for the flow shop sequencing problem. European Journal Of Operational Research, [s.l.], v. 47, n. 1, p.65-74, jul. 1990. Elsevier BV.

TASGETIREN, M. Fatih et al. A variable iterated greedy algorithm with differential evolution for the no-idle permutation flowshop scheduling problem. Computers & Operations Research, [s.l.], v. 40, n. 7, p.1729-1743, jul. 2013. Elsevier BV.

TASGETIREN, M. Fatih et al. Iterated greedy algorithms for the blocking flowshop scheduling problem with makespan criterion. Computers & Operations Research, [s.l.], v. 77, p.111-126, jan. 2017. Elsevier BV.

TORKASHVAND, M.; NADERI, B.; HOSSEINI, S.a.. Modelling and scheduling multi-objective flow shop problems with interfering jobs. Applied Soft Computing, [s.l.], v. 54, p.221-228, maio 2017. Elsevier BV.

TUBINO, Dalvio Ferrari. Manual de planejamento e controle da produção. 2. ed. São Paulo: Altas S.A., 2009.

ULLMAN, J.d.. NP-complete scheduling problems. Journal Of Computer And System Sciences, [s.l.], v. 10, n. 3, p.384-393, jun. 1975. Elsevier BV.

VAN DEMAN, John M.; BAKER, Kenneth R.. Minimizing Mean Flowtime in the Flow

Shop with No Intermediate Queues. A I I e Transactions, [s.l.], v. 6, n. 1, p.28-34, mar. 1974. Informa UK Limited. http://dx.doi.org/10.1080/05695557408974929.

VALLADA, Eva; RUIZ, Rubén; FRAMINAN, Jose M.. New hard benchmark for flowshop scheduling problems minimising makespan. European Journal Of Operational Research, [s.l.], v. 240, n. 3, p.666-677, fev. 2015. Elsevier BV.

VERGARA, Sylvia Constant. Projetos e Relatórios de Pesquisa em Administração. 2ª ed. São Paulo: Atlas, 1998.

WANG, Sheng-yao et al. An effective estimation of distribution algorithm for solving the distributed permutation flow-shop scheduling problem. International Journal Of Production Economics, [s.l.], v. 145, n. 1, p.387-396, set. 2013. Elsevier BV.

WISMER, D. A.. Solution of the Flowshop-Scheduling Problem with No Intermediate Queues. Operations Research, [s.l.], v. 20, n. 3, p.689-697, jun. 1972. Institute for Operations Research and the Management Sciences (INFORMS).
WOOLLAM, C.r.. Flowshop with no idle machine time allowed. Computers & Industrial Engineering, [s.l.], v. 10, n. 1, p.69-76, 1986. Elsevier BV.

XU, Jianyou et al. An iterated local search for the multi-objective permutation flowshop scheduling problem with sequence-dependent setup times. Applied Soft Computing, [s.l.], v. 52, p.39-47, mar. 2017. Elsevier BV.

XU, Nuo; HUANG, Samuel H.; RONG, Y. Kevin. Automatic setup planning: current state-of-the-art and future perspective. International Journal Of Manufacturing Technology And Management, [s.l.], v. 11, n. 2, p.193-208, 2007. Inderscience Publishers.

YAGMAHAN, Betul; YENISEY, Mehmet Mutlu. A multi-objective ant colony system algorithm for flow shop scheduling problem. Expert Systems With Applications, [s.l.], v. 37, n. 2, p.1361-1368, mar. 2010. Elsevier BV.

YANG, Shun; ARNDT, Tobias; LANZA, Gisela. A Flexible Simulation Support for Production Planning and Control in Small and Medium Enterprises. Procedia Cirp, [s.l.], v. 56, p.389-394, 2016. Elsevier BV.

YENISEY, Mehmet Mutlu; YAGMAHAN, Betul. Multi-objective permutation flow shop scheduling problem: Literature review, classification and current trends. Omega, [s.l.], v. 45, p.119-135, jun. 2014. Elsevier BV.

ZACCARELLI, Sergio Baptista. Programação e Controle da Produção. 8. ed. São Paulo: Pioneira, 1987.

ZHANG, Luping; WONG, T.n. Solving integrated process planning and scheduling problem with constructive meta-heuristics. Information Sciences, [s.l.], v. 340-341, p.1-16, maio 2016. Elsevier BV.

APÊNDICE A – CÓDIGO DESENVOLVIDO PARA CÁLCULO DE *MAKESPAN*.

```
program tcc;

uses SysUtils, windows;

type

arrn = array[0..200] of integer;

//Variaveis

var i,j,m,n,k,X,ii,jj,xx,ax,b,mk,fof,ee,aa,aux: integer; //i e ii=contador linha, j e jj=contador coluna, m=máquina, n=tarefa, k=variável de auxílio para definição de sequência, X=variável de auxílio para cálculo de soma dos tempos, xx=contador utilizado para leitura de todos os arquivos gerados,ax e b=utilizados para realização do procedura de calculo do makespan, mk=variável que guarda valor de makespan, fof=variável auxiliar de comparação que guarda valor de makespan, ee, aa e auxi=variável de auxílio para ordenação de sequência;

ma,mb,d: array [0..200,0..200] of integer; //ma=matriz inicial de tempo de processamento de cada instância, mb=matriz inicial de tempo de setup de cada instância, d= matriz que guarda o tempo final de cada instância;

VetSoma, VetTar,s,seq,seq1,sp: array [0..200] of integer; //VetSoma=Vetor que guarda o cálculo da soma dos tempos,VetTar=Vetor que mostra a sequência inicial das tarefas, s,seq,sp=Vetor auxiliar de ordenação das tarefas, seq1=Vetor que mostra a sequência final das tarefas;

arquivoa, arquivob, arquivo1 : text; //arquivoa=arquivos que armazenam os tempos de processamento de cada instância, arquivob=arquivos que armazenam os tempos de setup de cada instância, arquivo1=arquivo que armazena o makespan de todas as instâncias.


//Procedure Clássico

procedure fo(ax:arrn;b:integer); //definição de como a procedura pode ser chamada em qualquer parte do programa.

begin

  for ii := 1 to m do for jj := 1 to n do d[ii,jj] := 0; //toda a matriz que é alimentada por valores de makespan, inicialmente é zerada.

  begin

        d[1,1] := ma[1,ax[1]]; //primeiro elemento equivale apenas ao tempo de processamento;

        for ii := 2 to b do d[1,ii] := d[1,ii-1] + mb[1,ax[ii]] + ma[1,ax[ii]]; //preenchimento da primeira linhaa da matriz, o makespan desta linha é a soma dos tempos de processamento de todas as tarefas na primeira máquina, mais os tempos de setup, com exceção do setup da primeira tarefa executada;

;
```

```
for ii := 2 to m do //preenchimento da primeira coluna da matriz
begin
if d[ii-1,1] > mb[ii,ax[1]] then //se o tempo final da tarefa atual na máquina anterior é maior que o tempo de setup da tarefa atual analisada..então
begin
d[ii,1] := d[ii-1,1] + ma[ii,ax[1]]; ];//o tempo final da tarefa analisada corresponde ao tempo final na máquina anterior mais o tempo de processamento na máquina atual.
end
else //caso contrário..
begin
d[ii,1] := mb[ii,ax[1]] + ma[ii,ax[1]]; ];//o tempo final da tarefa analisa corresponde ao tempo de setup mais o tempo de processamento na máquinal atual
end
end;
for ii := 2 to m do //contador para preenchimento dos demais campos da matriz
  for jj := 2 to b do
  begin
  if (d[ii-1,jj] > d[ii,jj-1] + mb[ii,ax[jj]]) //caso o tempo final da tarefa atual na máquina anterior seja maior que o tempo de processamento da tarefa antecedente na máquina atual mais o tempo de setup da tarefa analisada.. então
  then
    begin
      d[ii,jj] := d[ii-1,jj] + ma[ii,ax[jj]]; //o tempo final da tarefa analisada corresponde ao tempo final da tarefa atual na máquina anterior mais o tempo de processamento na máquina atual;
    end
  else //caso contrário..
    begin
      d[ii,jj] := d[ii,jj-1] + mb[ii,ax[jj]] + ma[ii,ax[jj]]; //o tempo final da tarefa analisada, correponde ao tempo final da tarefa anterior na máquina atual mais o tempo de setup e tempo de processamento da tarefa analisada na máquina atual;
    end
  end;
end;
```

```
  mk := d[m,b]; //a variavél de makespan equivale ao último elemento da tarefa que se deseja na última máquina disponível.

  fof := mk; //a variável fof equivale à variável mk.

end;


//Leitura dos arquivos txt

begin

assign(arquivo1,'C:\Users\Acer\Desktop\Arquivos Salvos\Amanda\UTFPR\9º Período\Programa\Tempos de Processamento\saida.txt'); //indica aonde armazenar os tempos de makespan;

rewrite(arquivo1);

for xx := 1 to 2000 do //contador para ler todas as instâncias geradas;

begin

assign(arquivoa,'C:\Users\Acer\Desktop\Arquivos Salvos\Amanda\UTFPR\9º Período\Programa\Tempos de Processamento\'+IntToStr(xx)+'.txt'); //indica aonde encontrar os arquivos de tempo de processamento;

reset(arquivoa);

read(arquivoa,m,n); //lê cada instância, sendo m=máquinas e n=tarefas;

{assign(arquivob,'C:\Users\Acer\Desktop\Arquivos Salvos\Amanda\UTFPR\9º Período\Programa\Tempos de Setup\1 a 25\'+IntToStr(xx)+'.txt');  //arquivo com intervalo de setup de 1 a 25

assign(arquivob,'C:\Users\Acer\Desktop\Arquivos Salvos\Amanda\UTFPR\9º Período\Programa\Tempos de Setup\1 a 50\'+IntToStr(xx)+'.txt');  //arquivos com intervalo de setup de 1 a 50}

assign(arquivob,'C:\Users\Acer\Desktop\Arquivos Salvos\Amanda\UTFPR\9º Período\Programa\Tempos de Setup\1 a 100\'+IntToStr(xx)+'.txt'); //arquivos com intervalo de setup de 1 a 100

reset(arquivob);

read(arquivob,m,n);


for i := 1 to m do //contador para percorrer todo o arquivos das instâncias

begin

    for j := 1 to n-1 do read (arquivoa,ma[i,j]);//lê todos os tempos de processamento das instâncias;

    readln (arquivoa,ma[i,n]); //lê o último tempo de processamento;
```

```
    for j := 1 to n-1 do read (arquivob,mb[i,j]);//lê todos os tempos de setup das instâncias

    readln (arquivob,mb[i,n]); //lê o ultimo tempo de setup.

end;

VetSoma[jj] := 0; //soma os tempos de processamento e os tempos de setup em todas as máquinas, para poder organizar por ordem crescente ou decrescente

VetTar [jj] := 0;//indica a sequência inicial das tarefas;

X := 0;

 for ii := 1 to m do //contador para percorrer todas as máquinas das instâncias;

     begin

     for jj := 1 to n do //contador para percorrer todas as tarefas das instâncias;

        begin

        X := ma[ii,jj] {+ mb[ii,jj]}; //realiza a soma dos tempos de processamento e tempos de setup de cada tarefa;

        VetSoma[jj] := VetSoma[jj] + X; //acumula as somas para cada linha (máquina) percorrida

        VetTar[jj] := jj; //indica sequência da tarefa atual analisada;

        end;

      end;

 for jj := 1 to n do

  begin

  writeln('A soma dos tempos de processamento e de setup da tarefa ',jj,' e ',VetSoma[jj]); //imprime na tela a soma de cada tarefa das instâncias;

  writeln('A posicao referente a tarefa ',jj,' e ',VetTar[jj]); //imprime na tela a sequência de cada tarefa das instâncias;

  end;


//Ordenação

for i := 1 to n do s[i] := 1;

for i := 1 to n do for j := 1 to n do if VetSoma[i] < VetSoma[j] then s[i] := s[i] + 1;
// realiza a ordenação das tarefas baseado no VetSoma (<LPT >SPT) ;

for i := 1 to n do for j := 1 to n do if i <> j then if s[i] = s[j] then s[j] := s[j] + 1;
//contabiliza toda vez que é necessário fazer a troca de ordem das tarefas devido a criticidade de soma descrescente ou crescente;
```

```
  for i := 1 to n do
    begin
      ee := s[i];
      seq[ee] := i; //armazena a ordem final das tarefas;
    end;
for i := 1 to n do
begin
seq1[i] := seq[i]; //o vetor final da sequência de tarefas é alimentado com base no vetor que possui a sequência inicial;
end;


//2ª fase NEH
fo(seq,2); //chama a procedure do cálculo do makespan para cálculo do primeiro par de tarefas;
aux := fof; //a variavel aux recebe o valor de fof;
seq1[1] := seq[2]; //os dois primeiros pares de tarefas são trocados;
seq1[2] := seq[1];
fo(seq1,2); //o makespan é calculado para o par de tarefas trocado.
if fof >= aux then for i := 1 to 2 do seq1[i] := seq[i]; //se o makespan do par de tarefas trocado for maior do que o makespan da sequência inicial, mantêm-se a sequência original;
for i := 3 to n do seq1[i] := seq[i]; //o vetor de sequência final recebe os mesmo valores do vetor de sequência inicial;


for i:= 3 to n do //para as demais tarefas da instância, fazer;
    begin
    aux := 0; //vetor auxiliar é zerado no início de cada I;
       fo(seq1,i); //calcula-se o makespan para cada inclusão de tarefa da sequência incial;
       aux := fof; //a variável auxiliar recebe o resultado do cálculo;
       for k := 1 to i do sp[k] := seq1[k]; //o vetor auxiliary é alimentado pela vetor de sequência final
       for k := i downto 2 do //a troca da ordem de tarefas é realizada;
```

```
        begin
        aa := sp[k];
        sp[k] := sp[k-1];
        sp [k-1] := aa;
        fo(sp,i); //calcula-se o makespan de cada ordem possível
        if fof < aux then //caso o cálculo atual seja menor do que o ultimo cálculo armazenado, então;
          begin
          aux := fof; //armazena o cálculo de makespan atual;
          for aa := 1 to i do seq1[aa] := sp[aa]; //armazena até o I atual, a sequência com menor makespan;
          end;
        end;
    end;


for i := 1 to n do write(seq1[i],' '); //a sequência final é impressa na tela;
fo(seq1,n); //calcula-se o makespan da sequência final;
writeln('Makespan da sequencia: ',fof); //valor de makespan final é impresso na tela;
writeln;
writeln(arquivo1,fof);//armazena no arquivo .txt o makespan de cada instância;
end;
close(arquivo1); //fecha o arquivo1;
close(arquivoa); //fecha o arquivoa;
close(arquivob); //fecha o arquivob;
readln(k);
end. //fim do programa.
```