

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO DE ENGENHARIA DE PRODUÇÃO
CURSO DE ENGENHARIA DE PRODUÇÃO

MARCOS ROGERIO CELINI JÚNIOR

HEURÍSTICAS PARA O PROBLEMA DE MINIMIZAÇÃO DA SOMA DE
ADIANTAMENTOS E ATRASOS NO AMBIENTE DE MÁQUINAS
PARALELAS NÃO RELACIONADAS

TRABALHO DE CONCLUSÃO DE CURSO

PONTA GROSSA

2017

MARCOS ROGERIO CELINI JÚNIOR

**HEURÍSTICAS PARA O PROBLEMA DE MINIMIZAÇÃO DA SOMA DE
ADIANTAMENTOS E ATRASOS NO AMBIENTE DE MÁQUINAS
PARALELAS NÃO RELACIONADAS**

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

Orientador: Prof. Dr. Everton Luiz de Melo

PONTA GROSSA

2017



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



TERMO DE APROVAÇÃO DE TCC

**HEURÍSTICAS PARA O PROBLEMA DE MINIMIZAÇÃO DA SOMA DE
ADIANTAMENTOS E ATRASOS NO AMBIENTE DE MÁQUINAS PARALELAS NÃO
RELACIONADAS**

por

Marcos Rogério Celini Júnior

Este Trabalho de Conclusão de Curso (TCC) foi apresentado em 04 de dezembro de 2017 como requisito parcial para a obtenção do título de Bacharel em Engenharia de Produção. O candidato foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Prof. Dr. Everton Luiz de Melo
Prof. Orientador

Prof. Dr. Fabio Jose Ceron Branco
Membro titular

Prof. Dr. Fabio Neves Puglieri
Membro titular

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

Aos meus pais, por tornarem o
sonho possível.

AGRADECIMENTOS

Primeiramente agradeço a Deus pela força concedida e pelas realizações obtidas nos últimos anos.

Aos meus pais e a minha irmã, Kamila pela compreensão nos momentos de ausência, pelo apoio em todos os aspectos de minha vida e pelo amor incondicional.

Aos meus avós pela ajuda durante todos os anos desta graduação.

Agradeço ao meu orientador Prof. Dr. Everton Luiz de Melo, pelo aprendizado e pela paciência nos últimos anos.

A minha amiga Manuele, pela cooperação na realização deste trabalho, pela amizade, companheirismo e carinho.

Ao grande amigo Danilo pelo companheirismo nos últimos anos, pela amizade e pela força.

A todos os meus amigos, que de certa forma me ajudaram e apoiaram em todos os aspectos e me deram forças para não desistir.

Por fim, agradeço a todos que de alguma forma contribuíram para a realização deste trabalho, que representa um marco em minha vida.

RESUMO

CELINI JÚNIOR, Marcos Rogerio. **Heurísticas para o problema de minimização da soma de adiantamentos e atrasos no ambiente de máquinas paralelas não relacionadas**. 2017. 74 f. Trabalho de Conclusão de Curso (Bacharelado em Engenharia de Produção) – Universidade Tecnológica Federal do Paraná, Ponta Grossa, 2017.

A programação da produção é amplamente utilizada pelo setor de planejamento e controle da produção nas indústrias, pois permite definir as melhores ordenações de tarefas visando otimizar algum objetivo. Este trabalho traz o desenvolvimento de métodos de resolução para o problema de minimização da soma de adiantamentos e atrasos no ambiente de máquinas paralelas não relacionadas. Esses métodos devem ser capazes de contornar o esforço computacional encontrado na resolução do problema por métodos exatos considerando que atrasos e adiantamentos possuam a mesma proporção de importância. Para a realização do trabalho foi feita uma pesquisa bibliográfica e, a partir dela, um modelo matemático foi implementado utilizando um *software* de programação matemática, o Lingo. Adicionalmente, métodos heurísticos construtivos foram utilizados para fornecer uma solução inicial para uma heurística de melhoria proposta. Tais métodos heurísticos, implementados utilizando a linguagem de programação C, objetivam obter resultados satisfatórios na resolução do problema consumindo baixo tempo de processamento. Em alguns casos foi possível fazer a comparação das soluções obtidas pelos métodos heurísticos com a solução ótima, enquanto para outros não foi possível obter nenhuma solução com o modelo matemático pela complexidade do problema. Dentre as heurísticas construtivas, pode-se considerar que a Heurística Construtiva 2 obteve melhores resultados para a função objetivo, enquanto as outras duas obtiveram resultados satisfatórios. Quanto à heurística de melhoria, pode-se afirmar que a mesma alcançou seu objetivo, conseguindo melhorias de até 61%.

Palavras-chave: Otimização. Programação de tarefas. Máquinas paralelas não relacionadas. Heurísticas.

ABSTRACT

CELINI JÚNIOR, Marcos Rogerio. **Heuristics for the minimizing of the advances and delays's sum problem in the unrelated parallel machines environment.** 2017. 74 p. Work of Conclusion Course (Graduation in Industrial Engineering) – Federal Technology University – Paraná, Ponta Grossa, 2017.

Production scheduling is widely used by industry planning and production control because it allows defining the best ordering of tasks to optimize some objective. This work brings the development of resolution methods for the problem of minimizing the sum of advances and delays in the unrelated parallel machines environment. These methods must be able to bypass the computational effort encountered in solving the problem by exact methods considering that delays and advances have the same proportion of importance. For the accomplishment of the work a bibliographical research was made and, from it, a mathematical model was implemented using software of mathematical programming - Lingo. In addition, constructive heuristic methods were used to provide an initial solution to a proposed improvement heuristic. Such heuristic methods, implemented using the programming language C, aim to obtain satisfactory results in the resolution of the problem, consuming low processing time. In some cases it was possible to compare the solutions obtained by the heuristic methods with the optimal solution, while for others it was not possible to obtain any solution with the mathematical model because of the complexity of the problem. Among the constructive heuristics, it can be considered that Constructive Heuristic 2 obtained better results for the objective function, while the other two obtained satisfactory results. As for the heuristic of improvement, it can be said that it reached its goal, achieving improvements of up to 63%.

Keywords: Optimization. Scheduling. Unrelated parallel machines. Heuristics.

LISTA DE FIGURAS

Figura 1 - A função da produção	17
Figura 2 - Gráfico de Gantt para máquinas paralelas não relacionadas	22
Figura 3 - Gráfico de Gantt para o ambiente <i>flow shop</i>	23
Figura 4 - Gráfico de Gantt para o ambiente <i>job shop</i>	24
Figura 5 - Representação do <i>makespan</i> em máquinas paralelas	26
Figura 6 - Classificação da pesquisa.....	34
Figura 7 - Etapas de desenvolvimento	35
Figura 8 - Algoritmo da Heurística Construtiva 1	42
Figura 9 - Alocação das tarefas conforme Heurística Construtiva 1	43
Figura 10 - Gráfico de Gantt para exemplo da Heurística Construtiva 1	44
Figura 11 - Algoritmo da Heurística Construtiva 2.....	45
Figura 12 - Gráfico de Gantt para exemplo da Heurística Construtiva 2	46
Figura 13 - Algoritmo da Heurística Construtiva 3.....	47
Figura 14 - Gráfico de Gantt para exemplo da Heurística Construtiva 3	49
Figura 15 - Algoritmo da Heurística de Melhoria	50
Figura 16 - Solução da Heurística Construtiva 1	56
Figura 17 - Solução da Heurística Construtiva 1 com aplicação da melhoria	58
Figura 18 - Solução da Heurística Construtiva 2.....	60
Figura 19 - Solução da Heurística Construtiva 2 com aplicação da melhoria	61
Figura 20 - Solução da Heurística Construtiva 3.....	63
Figura 21 - Solução da Heurística Construtiva 3 com aplicação da melhoria	64

LISTA DE TABELAS

Tabela 1 - Regras de ordenação utilizadas	37
Tabela 2 - Exemplo de instância	39
Tabela 3 - Relação das instâncias	40
Tabela 4 - Tempos e datas de entrega para execução da Heurística Construtiva 1 ..	43
Tabela 5 - Resultados para M_A e M_C na Heurística Construtiva 2.....	51
Tabela 6 - Resultados para M_A após troca.....	52
Tabela 7 - Resultados para o modelo matemático	54
Tabela 8 - Resultados da Heurística Construtiva 1	55
Tabela 9 - Resultados da Heurística Construtiva 1 após melhoria.....	57
Tabela 10 - Resultados da Heurística Construtiva 2	59
Tabela 11 - Resultados da Heurística Construtiva 2 após melhoria.....	60
Tabela 12 - Resultados da Heurística Construtiva 3	62
Tabela 13 - Resultados da Heurística Construtiva 3 após melhoria.....	64
Tabela 14 - Comparação entre os resultados obtidos.....	65
Tabela 15 - Melhorias por heurística	66
Tabela 16 - Comparação entre os resultados após aplicação da melhoria.....	66

LISTA DE ABREVIATURAS

CNC	Comando Numérico Computadorizado
EDD	<i>Earliest Due Date</i>
FIFO	<i>First In First Out</i>
IPI	Índice de Prioridade
LPT	<i>Largest Processing Time</i>
MP	Matéria Prima
PCP	Planejamento e Controle da Produção
PO	Pesquisa Operacional
SPT	<i>Shortest Processing Time</i>

SUMÁRIO

1 INTRODUÇÃO	12
1.1 PROBLEMA	13
1.2 JUSTIFICATIVA	13
1.3 OBJETIVOS	14
1.3.1 Objetivo Geral	14
1.3.2 Objetivos Específicos	15
1.4 DELIMITAÇÃO DO TEMA.....	15
1.5 ESTRUTURA DO TRABALHO.....	15
2 REVISÃO BIBLIOGRÁFICA	17
2.1 O PCP E A SUA IMPORTÂNCIA	17
2.2 A PO E O PCP	18
2.3 PROGRAMAÇÃO DE TAREFAS	19
2.4 AMBIENTES DE PRODUÇÃO.....	20
2.4.1 Máquina Única	21
2.4.2 Máquinas Paralelas.....	21
2.4.3 <i>Flow Shop</i>	22
2.4.4 <i>Job Shop</i>	23
2.5 OBJETIVOS DE OTIMIZAÇÃO	24
2.5.1 <i>Makespan</i>	25
2.5.2 <i>Flowtime</i>	26
2.5.3 Atraso e variações.....	27
2.6 ADIANTAMENTOS E ATRASOS EM MÁQUINAS PARALELAS.....	27
2.7 ESFORÇO COMPUTACIONAL E HEURÍSTICAS	29
2.8 REGRAS DE PRIORIZAÇÃO DE TAREFAS	30
2.8.1 Regras para Tempo de Processamento.....	31
2.8.2 Regras de Ordenação para Outros Fatores	31
2.9 HEURÍSTICAS DE MELHORIA.....	32
2.10 BUSCA LOCAL	32
3 METODOLOGIA	34
3.1 CLASSIFICAÇÃO DA PESQUISA	34
3.2 ETAPAS DE DESENVOLVIMENTO	35
3.2.1 Revisão Bibliográfica.....	36
3.2.2 Identificação de Regras e Heurísticas para a Resolução do Problema.....	36
3.2.3 Levantamento de Dados	37
3.2.4 Implementação dos Métodos Utilizando Recursos Computacionais	37
3.2.5 Validação dos Resultados para os Problemas de Menor Porte	38
4 DESENVOLVIMENTO	39
4.1 INSTÂNCIAS.....	39

4.2 IMPLEMENTAÇÃO DO MODELO MATEMÁTICO.....	41
4.3 MÉTODOS HEURÍSTICOS.....	41
4.3.1 Heurística Construtiva 1	42
4.3.2 Heurística Construtiva 2	44
4.3.3 Heurística Construtiva 3	46
4.3.4 Heurística de Melhoria.....	49
5 APRESENTAÇÃO E ANÁLISE DOS RESULTADOS	53
5.1 RESULTADOS DO MODELO MATEMÁTICO	53
5.2 RESULTADOS PARA A HEURÍSTICA CONSTRUTIVA 1	55
5.3 RESULTADOS PARA A HEURÍSTICA CONSTRUTIVA 2.....	58
5.4 RESULTADOS PARA A HEURÍSTICA CONSTRUTIVA 3.....	62
5.5 ANÁLISES COMPARATIVAS	65
6 CONCLUSÕES	68
REFERÊNCIAS.....	70

1 INTRODUÇÃO

As decisões tomadas dentro de uma organização se dividem em três níveis: o estratégico, em que as decisões são tomadas para o longo prazo; o tático, em que as decisões são tomadas para um período menor – o médio prazo; e o operacional, em que as decisões são tomadas no curto prazo (SOUZA; BORGONHONI, 2007), o que normalmente se tem atrelada a programação da produção.

Atualmente pode-se dizer que o setor industrial está passando pela quarta revolução industrial, ou seja, existe a aplicação da automação, inovações, controle e tecnologia da informação na manufatura presenciando, de acordo com Dombrowski (2014), um alto nível de complexidade que necessita uma maior eficiência, resultando na necessidade de decisões mais precisas.

Dombrowski (2014) ainda afirma que a quarta revolução industrial traz a integração entre os produtos e os processos. Processos inteligentes, que exigem a economia de tempo, precisam de planejamento e programação de forma que obtenham eficiência retomando-se então as decisões operacionais para sequenciamento de tarefas, que dependendo do caso, se mostram preocupantes.

A alocação e o sequenciamento de atividades a serem realizadas levam em consideração muitos aspectos como, por exemplo, o tempo de entrega, a precedência de realização de tarefas e a disponibilidade de máquinas e recursos e, justamente por isso, normalmente apresentam alta complexidade (ARENALES et al., 2011).

Pinedo (2008) afirma que o sequenciamento de atividades pode ocorrer em diferentes ambientes de produção. Além do ambiente de produção, para realizar o sequenciamento e a alocação, deve-se levar em consideração também um objetivo que se precisa otimizar.

Este objetivo é caracterizado de acordo com a necessidade do processo e escolhê-lo de forma correta é essencial para a alocação e o sequenciamento. De acordo com Arenales et al. (2011) os objetivos são muitos, como a minimização da soma de adiantamentos e atrasos de tarefas, abordado neste trabalho, a minimização de perdas monetárias causadas pelo atraso, a minimização do atraso, a minimização do tempo de fluxo e muitos outros.

Conforme o número de máquinas e tarefas aumenta e o ambiente a ser sequenciado se torna mais complexo, os *softwares* de modelagem matemática utilizados para encontrar a solução ótima para a alocação e o sequenciamento

encontram mais dificuldades de resolução. Isto se dá pelo fato de ser necessário um alto esforço computacional (STEBEL, 2006). O resultado disto é que quanto mais dificuldade existe, mais tempo se leva para fornecer a solução e, então, esperar pela solução ótima torna-se inviável do ponto de vista prático.

Para contornar este problema, faz-se necessário o uso de métodos heurísticos para que se encontrem resultados satisfatórios de programação da produção em tempo aceitável (STEBEL, 2006).

De acordo com Cormen et al. (2015), não são conhecidos algoritmos que forneçam a solução ótima de maneira eficiente para a resolução dos problemas de alocação e sequenciamento de tarefas em geral. São criadas então heurísticas que encontram respostas satisfatórias para o problema. Estas heurísticas obedecem a regras de alocação e sequenciamento que, de maneira lógica, otimizam o critério escolhido.

Considerando o contexto apresentado, este trabalho busca desenvolver uma heurística que designe as tarefas às máquinas e as ordene de forma satisfatória visando minimizar o objetivo desejado.

1.1 PROBLEMA

O problema abordado neste trabalho consiste na dificuldade de construir um método de resolução que possa ser implementado e consiga obter soluções satisfatórias para o problema de programação de tarefas no ambiente de máquinas paralelas não relacionadas de forma eficiente. Tem-se então, como ponto de partida para o estudo, a problemática que visa definir qual o melhor método de ordenação e sequenciamento para minimizar a soma de adiantamentos e atrasos no ambiente de máquinas paralelas não relacionadas.

1.2 JUSTIFICATIVA

Os sistemas flexíveis de manufatura buscam atingir eficiência ao produzir variados produtos (ARENALES et al., 2011).

Tubino (2000) afirma que o Planejamento e Controle da Produção (PCP), além de efetuar e controlar os planos de produção, deve atender o plano mestre de produção, considerando, inclusive a gestão de Matéria Prima (MP), além da administração do estoque, do sequenciamento da produção e da emissão e liberação de ordens (OLIVEIRA, 2014).

Sendo assim, obter uma programação que aloque e sequeencie a produção de forma eficiente e considere a necessidade do ambiente pode evitar muitos problemas relacionados a essas variáveis (OLIVEIRA, 2014). Isto se dá pelo fato de que o controle de MP pode ser feito de maneira correta, o estoque pode ser programado com base em dados confiáveis e as ordens liberadas com base em certezas.

Atrasar tarefas ou adiantá-las pode ser uma fonte de despesas inesperadas e desperdícios, tanto de recursos gerais, quanto financeiros quando se considera as particularidades dos produtos e os estoques, que de acordo com Machline (1981) podem ser divididos em operacional (transação), de especulação (proteção financeira) e de precaução (reserva).

Considera-se ainda que os estoques de operação são consequências naturais das atividades produtivas e em muitos casos, são inevitáveis (MACHILINE, 1981). Para este caso, ter uma programação de tarefas adequada pode permitir uma gestão mais específica e equilibrada.

Considerando a necessidade crescente da eficiência na produção, os aspectos econômicos e as particularidades do ambiente estudado, necessita-se de pelo menos de um método eficaz para programar a produção em tempo aceitável.

1.3 OBJETIVOS

1.3.1 Objetivo Geral

O objetivo geral deste trabalho de conclusão de curso é desenvolver, adaptar e analisar métodos eficazes de alocação e ordenação de tarefas para máquinas paralelas não relacionadas visando minimizar a soma de adiantamentos e atrasos.

1.3.2 Objetivos Específicos

Os objetivos específicos utilizados para atingir o objetivo geral neste trabalho são:

- i. Identificar, na literatura modelos, e métodos já existentes para a resolução do problema a ser estudado;
- ii. Implementar um modelo matemático em um *software* de programação matemática;
- iii. Implementar métodos heurísticos para a resolução do problema através de recursos computacionais; e
- iv. Validar para os problemas de pequeno porte os resultados obtidos pelos métodos heurísticos através de comparações com os resultados obtidos por programação matemática.

1.4 DELIMITAÇÃO DO TEMA

Este trabalho delimitou-se em coletar informações sobre o problema de minimização da soma de adiantamentos e atrasos no ambiente de máquinas paralelas não relacionadas e métodos para sua resolução na literatura, criar métodos heurísticos de resolução e implementar e comparar os resultados obtidos com resultados ótimos.

1.5 ESTRUTURA DO TRABALHO

A estrutura deste trabalho de conclusão de curso se dá em seis capítulos.

O Capítulo 1 traz os itens introdutórios, a contextualização do tema, a explicação do problema, a justificativa do estudo, os objetivos geral e específicos e, ainda, a delimitação do projeto.

O Capítulo 2 traz o referencial teórico para a elaboração deste trabalho de conclusão de curso. Foram considerados os temas: PCP, a pesquisa operacional (PO), a programação da produção, os ambientes de produção e problemas relacionados a eles, os objetivos de minimização e as soluções ótimas, esforço computacional e o uso de heurísticas.

O Capítulo 3, denominado metodologia, apresenta a classificação desta pesquisa, as etapas de desenvolvimento e como os dados são analisados.

O Capítulo 4 traz o desenvolvimento do trabalho. Cada uma das etapas denotadas na metodologia é desenvolvida, para posteriormente ser discutida.

O Capítulo 5, Apresentação e Análise dos Resultados, traz os resultados obtidos após a realização das etapas do desenvolvimento, as análises individuais e comparativas e, por fim, as discussões dos resultados.

O Capítulo 6 apresenta as conclusões obtidas com as análises dos resultados, a eficácia dos métodos e se os objetivos foram atingidos.

Por fim são apresentadas as referências utilizadas em todos os capítulos para a construção deste trabalho.

2 REVISÃO BIBLIOGRÁFICA

O Capítulo 2 apresenta a pesquisa bibliográfica realizada acerca dos temas diretamente relacionados a este trabalho.

Os principais itens abordados são: O PCP e a sua importância na produção, a PO e o seu uso na resolução de problemas de PCP, a programação da produção, os ambientes de produção e problemas relacionados a eles, os objetivos de minimização e as soluções ótimas, esforço computacional e o uso de heurísticas, heurísticas construtivas e busca local, além de algumas regras de ordenação de tarefas.

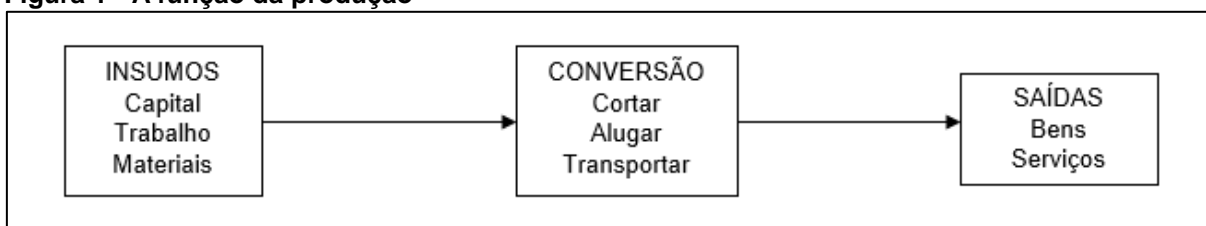
2.1 O PCP E A SUA IMPORTÂNCIA

Para Slack, Chambers e Johnston (2002), administrar a produção é cuidar da produção de bens e serviços e, acima de tudo, é um assunto com conteúdo prático para tratar de problemas reais das organizações.

O termo “administração da produção” é similar ao termo “planejamento e controle da produção” e estes são designados para as atividades, decisões e responsabilidades atribuídos aos gerentes do sistema produtivo (SLACK; CHAMBERS; JOHNSTON, 2002).

Tubino (2000) classifica um sistema de produção de acordo com uma estrutura funcional que engloba três departamentos – Marketing, Finanças e Produção. O sistema de produção, estudado neste trabalho, é o responsável por gerar os bens e serviços e sua função é demonstrada na Figura 1.

Figura 1 - A função da produção



Fonte: Adaptado de Tubino (2000)

Observando a Figura 1, a estrutura básica de um sistema de produção consiste em adicionar valor aos bens e serviços e, assim, o planejamento e controle são necessários para eliminar atividades que não agreguem valor.

De acordo com Silva e Fernandes (2008), uma empresa deve se adaptar e se modernizar para que seu sistema produtivo seja eficaz. A adaptação e modernização não vêm apenas em tecnologias de produção automatizadas, elas vêm também em técnicas de gerenciamento e no uso de tecnologias para promover o melhor aproveitamento dos recursos.

O PCP administra diversas informações que são oriundas de diversos lugares. Estas informações vão desde a demanda e engenharia do produto até indicadores produtivos e parâmetros de capacidade. Coordenar e aplicar os recursos para a produção eficaz, ou seja, que agreguem valor, é uma tarefa muito importante para que a empresa consiga sobreviver (TUBINO, 2000).

2.2 A PO E O PCP

A PO tem por objetivo resolver problemas reais através de modelos matemáticos construídos a partir de coleta de dados e algumas necessidades específicas (HILLIER; LIEBERMAN, 2010).

Após o levantamento dos dados, é feita a modelagem destes, ou seja, eles são traduzidos em funções matemáticas. Os modelos matemáticos montados para a resolução dos problemas são compostos por uma função objetivo, que visa minimizar ou maximizar uma função matemática que tem como variáveis os dados levantados na coleta prévia; e pelas restrições, que matematicamente são apresentadas em formas de equações ou inequações que também têm como variáveis os dados levantados na coleta prévia (HILLIER; LIEBERMAN, 2010).

Algoritmos matemáticos são utilizados para a resolução dos problemas fornecendo como resultado final, nas melhores situações, a solução ótima para o problema considerado.

A utilização de PO se dá em diversos setores e, com a globalização, a eficiência tornou-se fundamental, intensificando sua aplicação (JOAQUIM JUNIOR et al., 2010).

Kahn et al. (2006) afirmam que quanto mais a inovação nos produtos ocorre, mais sofisticados eles se tornam e, assim, exigem mais tarefas para serem processados. Programar a produção considerando muitas variáveis se tornou então um problema da PO.

A PO pode ser utilizada para resolver diversos problemas dentro do planejamento e controle (HILLIER; LIEBERMAN, 2010). São exemplos: determinação do *mix* ideal de produtos a serem processados para maximizar o lucro, métodos de previsão de demanda com objetivo de diminuir o erro, diminuir a distância percorrida em transporte de produtos na logística e melhor utilização de recursos para redução dos custos e aumento da eficiência, como, por exemplo, determinar um novo planejamento de corte de metais visando obter menor desperdício (ABUARA; MORABITO, 2008).

Atualmente, a PO também é utilizada para programar a produção quando se trata da alocação de tarefas a máquinas visando otimizar algum critério.

2.3 PROGRAMAÇÃO DE TAREFAS

De acordo com Arenales et al. (2011), as decisões que mais ocorrem no nível operacional em uma organização são as de designação de tarefas a máquinas e a programação destas em cada máquina disponível para processamento. A programação das tarefas nas máquinas envolve definir: a ordem em que elas serão processadas; caso exista, a relação de precedência entre as tarefas; o tempo de processamento; o ambiente de produção em questão com relação às máquinas; e o objetivo a ser otimizado.

Ronconi e Kawamura (2010) afirmam que os problemas que envolvem programação de tarefa têm sido mais estudados e ganhado maior importância nos últimos tempos pelo advento da ideologia *Lean Manufacturing*, principalmente quando se pensa no *Just in time*.

Pinedo (2008) divide a programação da produção em três parâmetros diferentes e complementares: o ambiente das máquinas, as características do processo e a função objetivo que será otimizada, ou seja, o critério escolhido para se otimizar.

A programação da produção envolve, além da designação e dimensionamento de tarefas em máquinas, a administração dos estoques por determinar em cada instante qual tarefa estará pronta e, com isso, pode-se dimensionar os lotes (PINEDO, 2008).

A complexidade dos problemas de programação da produção costuma ser alta (GAREY; JOHNSON; SETHI, 1976) e muitas vezes métodos que fornecem a melhor solução, ou seja, a solução ótima, não são aplicáveis pelo tempo que demorariam. Criam-se então heurísticas ou meta-heurísticas para que se obtenha uma solução boa para o problema em um tempo aceitável (SILVA; OCHI; MARTINS, 2008).

Normalmente os critérios de otimização estão ligados aos instantes e ao cumprimento da data de entrega das tarefas e, conseqüentemente, do produto que está sendo processado (ARENALES et al., 2011).

Algumas medidas de desempenho, como o *makespan* e o tempo de fluxo total, que serão abordadas posteriormente, são normalmente minimizadas independente do ambiente de produção (ARENALES et al., 2011) por serem medidas de desempenho críticas para o processo.

2.4 AMBIENTES DE PRODUÇÃO

Os sistemas flexíveis de manufatura são um conjunto integrado de dispositivos de manuseio de materiais e máquinas de Comando Numérico Computadorizado (CNC) controlado por computadores e têm por objetivo atingir a eficiência da produção automatizada. Espera-se que se atinja grande escala considerando a flexibilidade para que se consiga processar diversos tipos de partes dos produtos ao mesmo tempo (ARENALES et al., 2011).

Estes sistemas vêm sendo mais utilizados a cada dia e a sua presença nos ambientes produtivos trouxe novas características nos ambientes da produção que devem ser levadas em consideração no momento de programação (ARENALES et al., 2011).

Os ambientes de produção podem ser os mais variados e os mais vistos na literatura são máquina única, máquinas paralelas, *flow shop* e *job shop*, detalhados na seqüência.

2.4.1 Máquina Única

Um ambiente produtivo é considerado como de máquina única quando apenas em uma máquina é realizado o sequenciamento das tarefas. Esta configuração pode ser útil quando há realmente apenas uma máquina no ambiente ou, quando se deseja controlar uma máquina gargalo dentro de um processo produtivo (PINEDO, 2008).

O ambiente de máquina única é um caso especial dos outros ambientes existentes, visto que eles podem ser decompostos em subproblemas que lidam com apenas uma máquina. Normalmente, de acordo com Ronconi e Kawamura (2010), o ambiente é decomposto em máquina única quando envolve produtos com meia vida curta, envio de exportações e misturas químicas e físicas.

A complexidade do sequenciamento de tarefas no ambiente de máquina única, quando comparada à complexidade dos outros modelos, é menor e os resultados e *insights* obtidos podem frequentemente ter aplicação em outros ambientes, como por exemplo em máquinas paralelas (DING et al., 2017).

2.4.2 Máquinas Paralelas

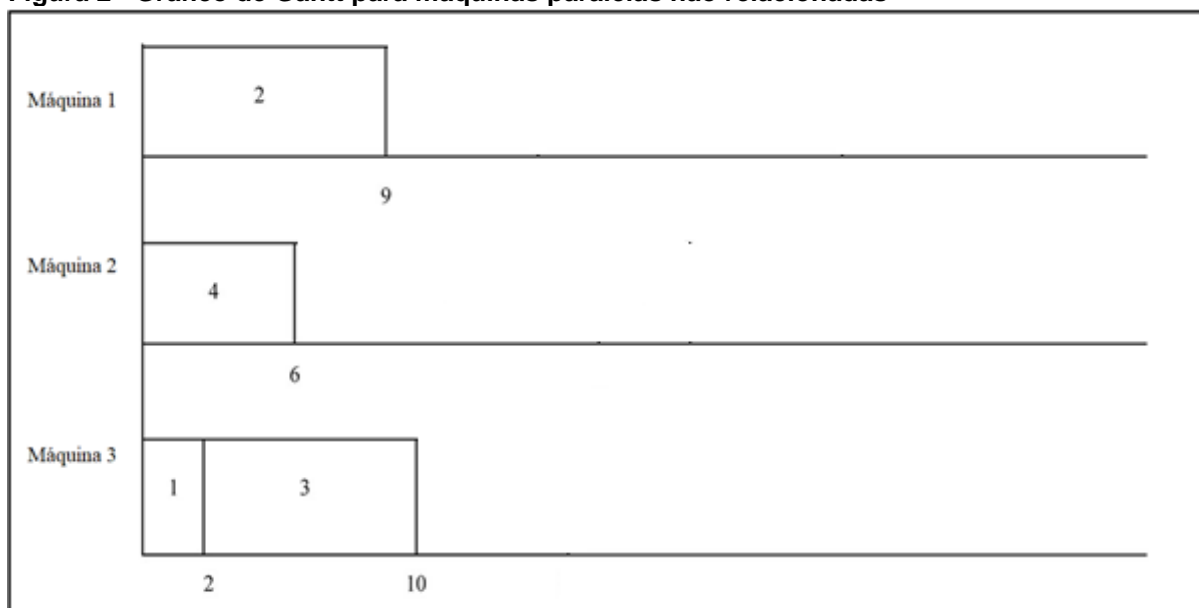
Um ambiente produtivo é considerado como de máquinas paralelas quando existem duas ou mais máquinas capazes de realizar o mesmo processamento de um produto, que para Müller, Dias e Araújo (2002) significa que há diferentes máquinas capazes de desempenhar a mesma função a realizam isso com tempos de processamento diferentes. As máquinas paralelas fazem parte de uma das classes mais difíceis de se resolver em *scheduling* (HAMZADAY; YILDIZ, 2017).

De acordo com Arenales et al. (2011), as máquinas paralelas podem ser classificadas em três categorias: idênticas, relacionadas e não relacionadas. As máquinas paralelas são idênticas quando o preparo e o processamento apresentam o mesmo tempo para todas as máquinas que executam a tarefa. São relacionadas quando as máquinas apresentam tempos de preparo e processamento proporcionais. São não relacionadas quando não existe relação alguma entre os tempos de processamento e preparo para a realização da tarefa. Nos casos das máquinas

relacionadas e não relacionadas, normalmente as máquinas paralelas são do mesmo tipo, entretanto com diferença, por exemplo, de potência ou tempo de vida.

Um gráfico de Gantt é usado na Figura 2 para explicar um exemplo de alocação de tarefas no ambiente de máquinas paralelas não relacionadas.

Figura 2 - Gráfico de Gantt para máquinas paralelas não relacionadas



Fonte: Adaptado de Senthilkumar e Narayanan (2011)

Na Figura 2 Máquina 1, Máquina 2 e Máquina 3 representam as máquinas disponíveis para o processamento das 4 tarefas existentes, que são representadas no gráfico de Gantt, pelos seus números.

Ressalta-se que cada tarefa pode ser processada em cada uma das máquinas disponíveis, entretanto cada uma possui um tempo diferente para processamento.

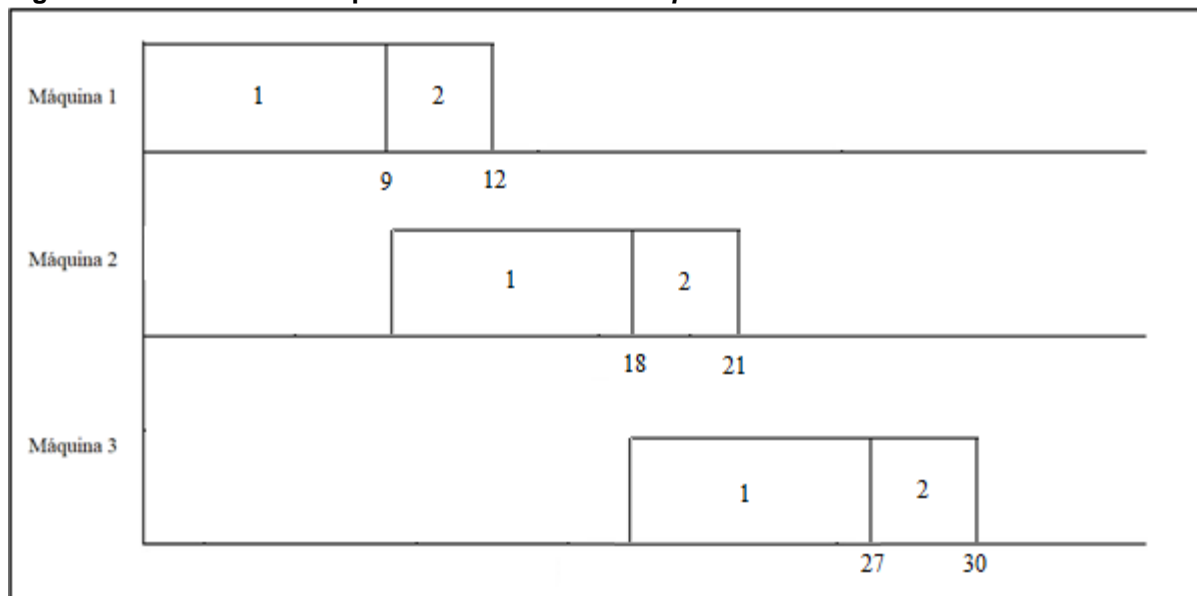
Os modelos matemáticos para máquinas paralelas relacionadas e máquinas paralelas não relacionadas costumam ser mais complexos pela necessidade de averiguar as opções de alocação e também de sequenciamento considerando diferentes tempos de processamento a fim de se obter melhores resultados (ARENALES et al., 2011).

2.4.3 Flow Shop

Neste ambiente de produção todas as tarefas apresentam o mesmo roteiro de processamento em todas as máquinas (ARENALES et al., 2011). A Figura 3 mostra

um gráfico de Gantt com quatro tarefas a serem processadas em três máquinas no ambiente *flow shop*.

Figura 3 - Gráfico de Gantt para o ambiente *flow shop*



Fonte: Adaptado de Reyna et al. (2014)

Observa-se na Figura 3, que as tarefas apresentam uma mesma ordem de processamento em todas as máquinas mesmo que seja possível o processamento antes do programado, ou seja, elas apresentam um mesmo roteiro. Isto pode ser observado, por exemplo, na tarefa 2, que é a segunda a ser processada nas três máquinas disponíveis.

Normalmente encontrar a solução de um problema neste ambiente é determinar uma sequência de processamento das tarefas que possa ser mantida para todas as máquinas otimizando uma medida de desempenho (BUZZO; MOCCELLIN, 2000).

De acordo com Lin e Zhang (2016), o *flow shop* desempenha um papel muito importante em sistemas de produção e é amplamente estudado para que se consiga melhores resultados.

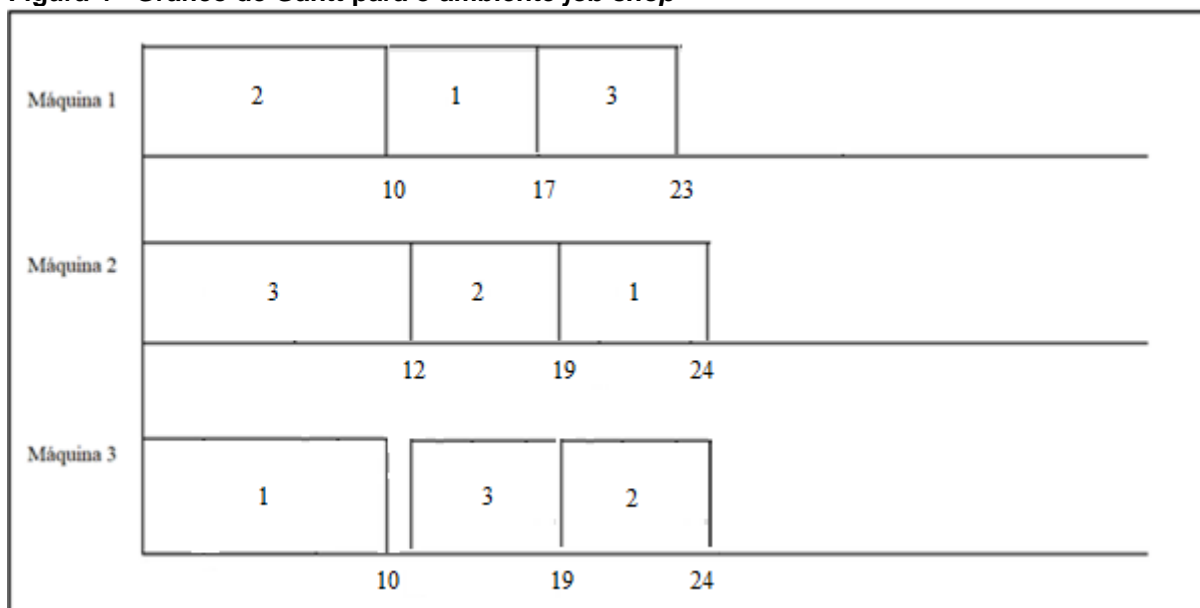
2.4.4 Job Shop

Os problemas mais complexos e que aparecem com mais frequência dentro da indústria vêm atraindo mais atenção para os estudiosos da área de programação de tarefas (WANG; YU, 2010).

O ambiente *job shop* é o ambiente produtivo em que há a presença de n tarefas e m máquinas e cada tarefa deve ser processada em todas as máquinas considerando um roteiro (ARENALES et al., 2011).

A Figura 4 mostra um gráfico de Gantt para o ambiente de produção *job shop*.

Figura 4 - Gráfico de Gantt para o ambiente *job shop*



Fonte: Adaptado de Arenales et al. (2011)

Na Figura 4 consegue-se observar que, para cada tarefa, existe um roteiro de processamento diferente, e que todas as tarefas passam por todas as máquinas em algum momento como, por exemplo, a tarefa 2, que é processada primeiro na máquina 1, depois na máquina 2 e por fim, na máquina 3.

Para facilitar a resolução de problemas no ambiente *job shop* assume-se que as máquinas estão constantemente prontas para receber novas tarefas para processar. Modelos flexíveis estão sendo considerados para lidar e promover sequenciamentos mais reais (WANG; YU, 2010).

O ambiente *flow shop*, estudado na subseção anterior é um caso particular do ambiente *job shop*

2.5 OBJETIVOS DE OTIMIZAÇÃO

Os problemas de PO e, conseqüentemente os problemas de programação da produção exigem que um critério de otimização seja estabelecido (OLIVEIRA, 2014).

Em programação da produção existem muitos critérios de otimização que podem ser abordados e estes são escolhidos de acordo com a necessidade do processo e com o ambiente em que a produção ocorre.

Na literatura são encontrados diferentes critérios de otimização, como por exemplo, a minimização dos desgastes causados pelo processamento de acordo com a sequência (SANTOS, 2016), a otimização de cargas em fornos de fundição (SILVA; MORABITO, 2004) e até mesmo, a otimização dos custos de energia elétrica na programação da captação, armazenamento e distribuição de água (SOLER et al., 2016).

A existência de muitos critérios mostra a subjetividade deste tipo de programação e enfoca a necessidade da utilização dela para resolver problemas de escalonamento de tarefas.

Os critérios de otimização mais utilizados na literatura são apresentados na sequência.

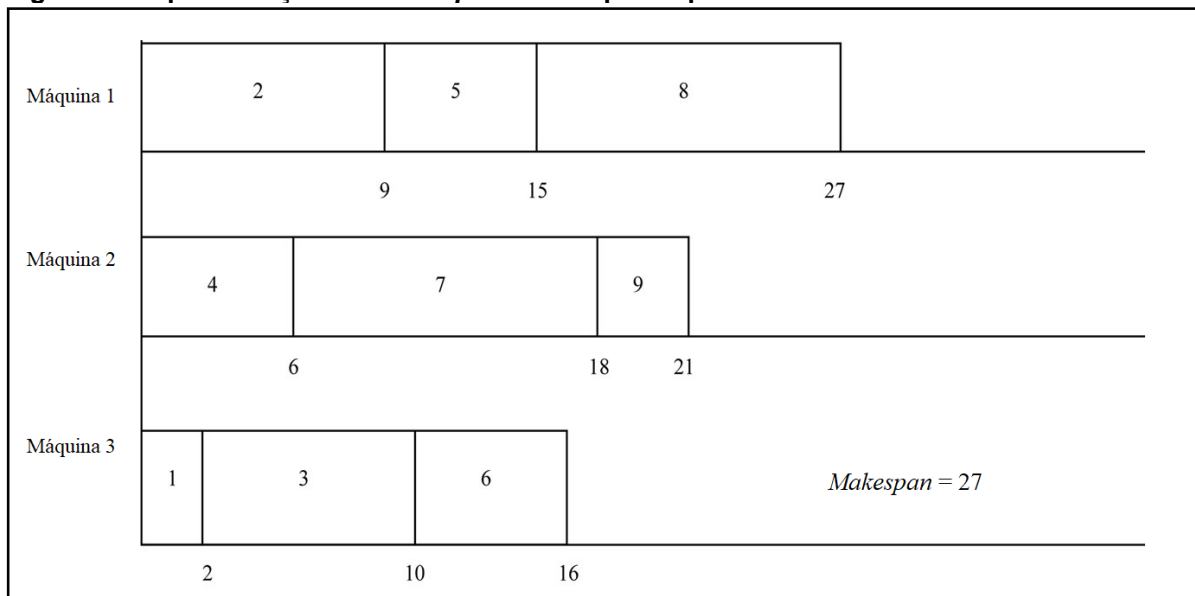
2.5.1 *Makespan*

De acordo com Lee e Wang (2017), a minimização do *makespan* em problemas de programação da produção tem sido nos últimos tempos largamente utilizada.

Arenales et al. (2011) definem o *makespan* como o instante em que o processamento de todas as tarefas a serem processadas termina. Esta é uma medida do sistema de produção.

A Figura 5 ilustra o *makespan* em um gráfico de Gantt.

Figura 5 - Representação do *makespan* em máquinas paralelas



Fonte: Adaptado de Senthilkumar e Narayanan (2011)

Na Figura 5 o eixo vertical representa as máquinas para processamento e o eixo horizontal, o tempo. O *makespan* vale 27 unidades de tempo e está no instante em que a tarefa 8 termina de ser processada na máquina 1, visto que após seu instante de término, nenhuma tarefa está sendo processada.

2.5.2 Flowtime

O *flowtime* pode ser considerado como a soma de todos os tempos de conclusão de todas as tarefas (GUPTA; HO; WEBSTER, 2000). Ele é o tempo de fluxo total e é utilizado na indústria para medir o estoque em processamento (ARENALES et al., 2011).

Minimizá-lo é importante, principalmente quando não se tem muito espaço para o estoque em processamento, ou ainda, quando a eficiência estocástica é necessária, como por exemplo, em casos em que é preciso manter apenas o estoque necessário por determinado período, por falta de espaço.

Ainda na Figura 5, o *flowtime* seria dado pela soma dos tempos de conclusão de todas as 9 tarefas.

2.5.3 Atraso e variações

O atraso de tarefas pode ser definido como a entrega de uma tarefa ocorrer em um período posterior ao que ela deveria ser entregue por qualquer motivo.

Verifica-se que, muitas vezes, quando há atraso na entrega de algum lote encomendado, é prevista uma multa que pode ser calculada como porcentagem do valor da encomenda ou, ainda, contabilizado a cada unidade de atraso (SLACK; CHAMBERS; JOHNSTON, 2002). Tendo isto em vista, faz sentido sequenciar as tarefas com o objetivo de minimizar o atraso.

Uma variação deste modelo ocorre quando se pondera em nível de criticidade, normalmente por unidades monetárias, as tarefas que serão mais críticas caso atrasem. Estas serão priorizadas na programação da produção visando minimizar seus atrasos.

Outras variações deste modelo são possíveis, como o número de tarefas atrasadas e o atraso máximo. Nestes casos, respectivamente, busca-se reduzir a quantidade de tarefas atrasadas ou minimizar a tarefa que mais atrasa.

2.6 ADIANTAMENTOS E ATRASOS EM MÁQUINAS PARALELAS

Algumas indústrias, como a de alguns produtos químicos, necessitam que o processamento ocorra no momento exato para que a próxima etapa seja processada imediatamente após o término da sua predecessora, visto que o produto pode ser perecível, não ter um espaço destinado para ficar estocado, ou ainda, como mencionam Ronconi e Kawamura (2010), perder a qualidade. Nestes casos, o objetivo a ser otimizado é a minimização das somas dos adiantamentos e atrasos. Este sistema é análogo às filosofias do sistema *Just in Time* em que os estoques intermediários devem ser minimizados ou, na melhor das hipóteses, eliminados (RONCONI; KAWAMURA, 2010).

Pode-se definir como adiantamentos e atrasos em uma sequência predefinida a diferença entre o instante de término de uma tarefa e a sua previsão de entrega, ou seja, o instante em que a tarefa precisa estar pronta.

Quando esta diferença é negativa, considera-se um adiantamento, pois o instante de término é menor do que a sua previsão de entrega. O oposto é válido para os atrasos (COLIN; SHIMIZU, 2000).

O modelo matemático utilizado para minimizar a soma dos adiantamentos e atrasos no ambiente de máquinas paralelas não relacionadas é apresentado a seguir. Para o entendimento deste modelo matemático, primeiramente é necessário que se estabeleçam alguns parâmetros e variáveis, que segundo Arenales et al. (2011), são listados a seguir.

Parâmetros:

- p_{ik} Tempo de processamento da tarefa i ($i=1, \dots, n$) na máquina k ($k=1, \dots, m$);
 s_{ijk} Tempo de setup da máquina k para processar a tarefa j imediatamente depois da tarefa i ;
 d_i Data de entrega da tarefa i ;
 G Número suficientemente grande;

Variáveis:

- C_{ik} Instante de término do processamento da tarefa i na máquina k ;
 X_{ijk} Assume 1 se a tarefa i precede imediatamente a tarefa j na máquina k e 0, caso contrário;
 T_i Atraso da tarefa i ;
 E_i Adiantamento da tarefa i .

Sendo assim, o modelo matemático é apresentado conforme segue:

$$\text{Minimizar } \sum_{i=1}^n (E_i + T_i) \quad (1)$$

$$\text{Sujeito a: } \sum_{k=1}^m \sum_{i=0}^n X_{ijk} = 1 \quad j=1, \dots, n \quad (2)$$

$$\sum_{j=1}^n X_{0jk} \leq 1 \quad k=1, \dots, m \quad (3)$$

$$\sum_{i=0; i \neq h}^n X_{ihk} - \sum_{j=0; j \neq h}^n X_{hjk} = 0 \quad h=1, \dots, n; k=1, \dots, m \quad (4)$$

$$C_{jk} \geq C_{ik} - G + (s_{ijk} + p_{jk} + G)X_{ijk} \quad i=0, \dots, n; j=1, \dots, n; k=1, \dots, m \quad (5)$$

$$T_i \geq C_{ik} - d_i \quad i=1, \dots, n; k=1, \dots, m \quad (6)$$

$$E_i \geq d_i - C_{ik} \quad i=1, \dots, n; k=1, \dots, m \quad (7)$$

$$X_{ijk} \in \{0,1\} \quad i=0, \dots, n; j=0, \dots, n; k=1, \dots, m \quad (8)$$

$$C_{ik} \geq 0 \quad i=0, \dots, n; k=1, \dots, m \quad (9)$$

$$E_i \geq 0 \quad i=1, \dots, n \quad (10)$$

$$T_i \geq 0 \quad i=1, \dots, n \quad (11)$$

Para este modelo matemático, considera-se uma tarefa fictícia – a tarefa inicial de cada máquina. Isto se dá para garantir que em todas as máquinas, haja apenas uma predecessora para cada tarefa.

Considera-se ainda para efeitos de cálculo, unidades de atraso e de adiantamentos, ou seja, quantas unidades de tempo uma tarefa atrasou ou adiantou, resultando em atrasos e adiantamentos sempre positivos ou iguais a zero.

A função objetivo (1) minimiza a soma dos atrasos e adiantamentos das tarefas. As restrições do tipo (2) garantem que cada tarefa possua apenas uma predecessora em uma máquina. As restrições do tipo (3) garantem que cada tarefa tenha no máximo uma única sequência de tarefas. As restrições do tipo (4) garantem que cada tarefa tenha apenas uma única sucessora, exceto pela tarefa zero (0). Nas restrições do tipo (5) têm-se a garantia de que cada tarefa vai receber instantes de término adequados. As restrições dos tipos (6) e (7) determinam os atrasos e os adiantamentos, respectivamente. As restrições do tipo (8) estabelecem os tipos das variáveis. As restrições dos tipos (9) a (11) estabelecem os domínios das variáveis.

2.7 ESFORÇO COMPUTACIONAL E HEURÍSTICAS

De acordo com a teoria da complexidade computacional os problemas de programação da produção são normalmente considerados como NP-Difícil (GAREY; JOHNSON; SETHI, 1976), ou seja, na maioria dos casos, a sua resolução de maneira eficiente é possível apenas em casos restritos. Estes casos envolvem números pequenos de tarefas e máquinas (BUZZO; MUCCELLIN, 2000).

Assim, quanto maior o número de tarefas e de máquinas, maior a quantidade de soluções possíveis e, conseqüentemente, maior a dificuldade em se obter a melhor solução.

O esforço computacional para problemas complexos é muito alto e isso dificulta que sejam encontradas soluções ótimas num tempo eficiente.

Mesmo que nos últimos quarenta anos tenha existido um esforço relevante em pesquisas da área e tenham sido desenvolvidas técnicas de programação matemática (SELEN; HOTT, 1986), ainda se faz necessário o uso de heurísticas para se obter a alocação e o sequenciamento de tarefas.

As heurísticas são regras que simplificam a tomada de decisão (SBICCA, 2014), ou seja, algoritmos que por meio de cálculos possibilitam o alcance de resultados aceitáveis, mesmo que não sejam os melhores possíveis para um problema

O seu uso vai além de problemas de programação da produção. Pode ser observado também em problemas de alocação em arranjos físicos e *layout* e até mesmo para facilitar a tomada de decisão em aspectos econômicos (SBICCA, 2014).

As heurísticas podem ser simples ou envolver procedimentos de melhoria e têm se mostrado cada vez mais bem-sucedidas na literatura fazendo com que utilizá-las na programação da produção seja essencial para que se consiga resultados satisfatórios viáveis.

2.8 REGRAS DE PRIORIZAÇÃO DE TAREFAS

Considerando a necessidade da utilização de heurísticas, podem ser aplicadas para a construção destas regras de alocação de tarefas que podem elencar, segundo Tubino (2000), informações sobre os lotes de produção, como datas de entrega, tempo de processamento das tarefas de acordo com o estado do sistema produtivo e ordem de chegada.

As regras mais utilizadas de acordo com a literatura são abordadas em seguida.

2.8.1 Regras para Tempo de Processamento

As regras mais utilizadas encontradas na literatura que levam em consideração o tempo de processamento para realizar a ordenação de tarefas são a *Shortest Processing Time* (SPT) e *Largest Processing Time* (LPT) (OLIVEIRA, 2014).

A regra SPT aloca as tarefas às máquinas disponíveis para execução em ordem não decrescente de tempos de processamento (SANTOS; FRANÇA, 1995), ou seja, tarefas com menores tempos de processamento serão executadas primeiramente.

A regra como LPT aloca as tarefas às máquinas disponíveis em ordem não crescente de tempos de processamento, de forma oposta à regra SPT. Desta maneira, tarefas com maiores tempos de processamento serão processadas primeiramente (MÜLLER; DIAS; ARAÚJO, 2002).

As regras SPT e LPT normalmente ordenam as tarefas considerando que estas já estejam alocadas em alguma máquina. A alocação das tarefas nas máquinas deve ser feita separadamente e podem, por exemplo, determinar que cada tarefa seja alocada na máquina capaz de processá-la em menor tempo.

2.8.2 Regras de Ordenação para Outros Fatores

Outras regras de ordenação são encontradas na literatura considerando diferentes aspectos do processo, desde a data de entrega até a prioridade de diferentes lotes (OLIVEIRA, 2014).

A *Earliest Due Date* (EDD) é uma das regras mais utilizadas e leva em consideração a data de entrega da tarefa. Para esta regra, as tarefas são alocadas a uma determinada máquina em ordem não decrescente das suas respectivas datas de entrega (MÜLLER; DIAS; ARAÚJO, 2002).

A regra Índice de Prioridade (IPI) atribui a um cliente ou produto um índice, que o prioriza quando comparado aos outros. As tarefas relacionadas aos produtos ou clientes com maior índice de prioridade são processadas primeiro (OLIVEIRA, 2014).

A regra *First In First Out* (FIFO) considera a ordem de chegada dos lotes para processamento. Os lotes que chegam primeiro para serem processados serão

processados primeiro (OLIVEIRA, 2014). O mesmo é válido para as tarefas em uma determinada máquina.

Mais de uma regra pode ser combinada para compor uma heurística. O que muda é o momento em que cada regra será válida e conseqüentemente, o resultado final esperado.

2.9 HEURÍSTICAS DE MELHORIA

As heurísticas de melhoria podem ser consideradas tentativas de modificar soluções atuais obtidas a partir de outras heurísticas cujo objetivo seja encontrar uma solução melhor (OLIVEIRA; MORABITO, 2006).

Normalmente as melhorias são propostas ao observar o comportamento das soluções iniciais. São introduzidas ou realizadas, então, pequenas modificações que ajam diretamente nos comportamentos observados com o intuito de se melhorar os resultados (CUNHA, 2015). Leva-se em consideração para propor as melhorias, espaço de solução, soluções viáveis e inviáveis e, por fim, vizinhança de soluções (CUNHA, 2015).

As heurísticas de melhoria observadas na literatura possuem aplicações em diversas áreas, assim como as heurísticas simples e construtivas. Sua aplicação pode ser observada, por exemplo, em problemas de roteamento de veículos (SOSA; GALVÃO; GANDELMAN, 2007), problemas no manejo de florestas industriais (MOREIRA; RODRIGUEZ, 2010) e problemas de produção em geral, como o de carregamento de paletes do produtor (OLIVEIRA; MORABITO, 2006).

A construção de heurísticas de melhorias pode ser realizada utilizando, por exemplo, uma busca local.

2.10 BUSCA LOCAL

Uma busca local pode ser considerada como um algoritmo que objetiva encontrar uma solução melhor do que uma já encontrada anteriormente percorrendo uma vizinhança de soluções que possuem características muito próximas (VIEIRA, 2006).

Vieira (2006) afirma que a busca local necessita que uma solução inicial para um problema tenha sido gerada antes da aplicação do método de melhoria, seja aleatoriamente ou através de uma heurística já desenvolvida para resolver o problema em estudo.

Tendo uma solução inicial, um algoritmo de busca local pode ser aplicado de forma que percorra a vizinhança desta solução buscando uma outra solução que, para um problema de minimização, tenha um valor menor do que a solução alcançada inicialmente (VIEIRA, 2006). O algoritmo, então promove iterações para realizar trocas avaliando as vizinhanças. Ao encontrar uma solução melhor do que a solução inicial, esta solução se torna uma nova solução corrente e o algoritmo continua até percorrendo a vizinhança até que não se obtenham mais melhoramentos, o que categoriza a solução encontrada como um ótimo local (VIEIRA, 2006).

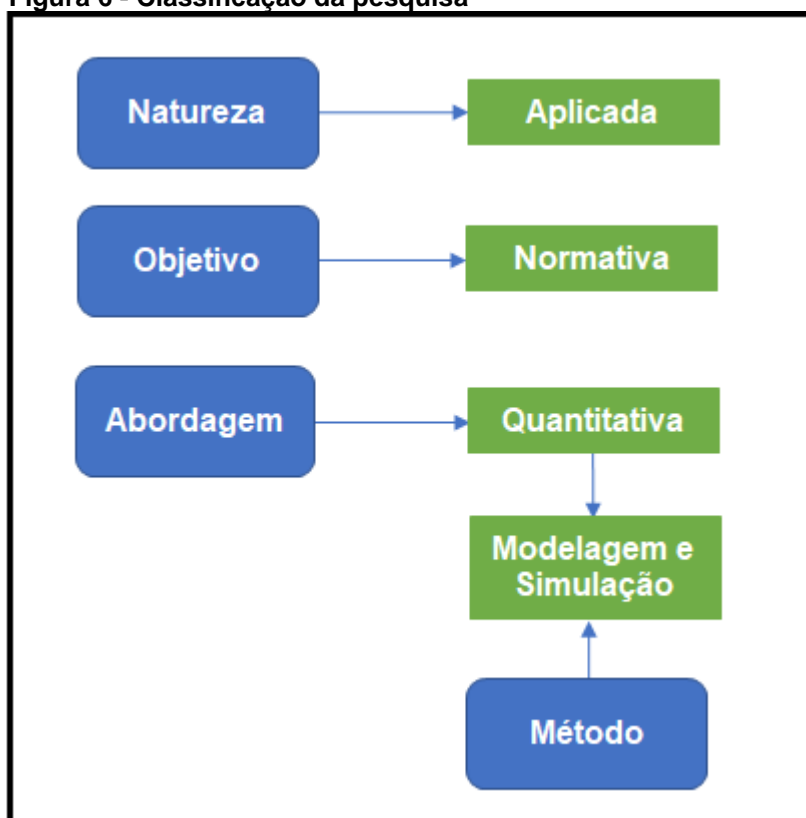
3 METODOLOGIA

Este capítulo apresenta a metodologia empregada neste trabalho. Nas seções e subseções seguintes estão descritas todas as etapas realizadas, bem como a classificação da pesquisa. As seções abordadas são a classificação da pesquisa e as etapas de desenvolvimento.

3.1 CLASSIFICAÇÃO DA PESQUISA

Para a classificação desta pesquisa, foram considerados quatro abordagens, que estão resumidas na Figura 6.

Figura 6 - Classificação da pesquisa



Fonte: Autoria própria

A Figura 6 traz como base a classificação para esta pesquisa proposta por Turrioni e Mello (2012).

Quanto à natureza, esta pesquisa pode ser classificada como aplicada (TURRIONI; MELLO, 2012) por apresentar interesse prático, em que se busque

também a solução de problemas reais, mesmo que o presente trabalho ainda esteja atrelado ao incremento do conhecimento científico.

Quanto ao objetivo, a pesquisa pode ser classificada como normativa, visto que de acordo com Bertrand e Franzoo (2002), busca aperfeiçoar resultados já disponíveis na literatura existente, ou ainda, encontrar uma solução ótima para problemas.

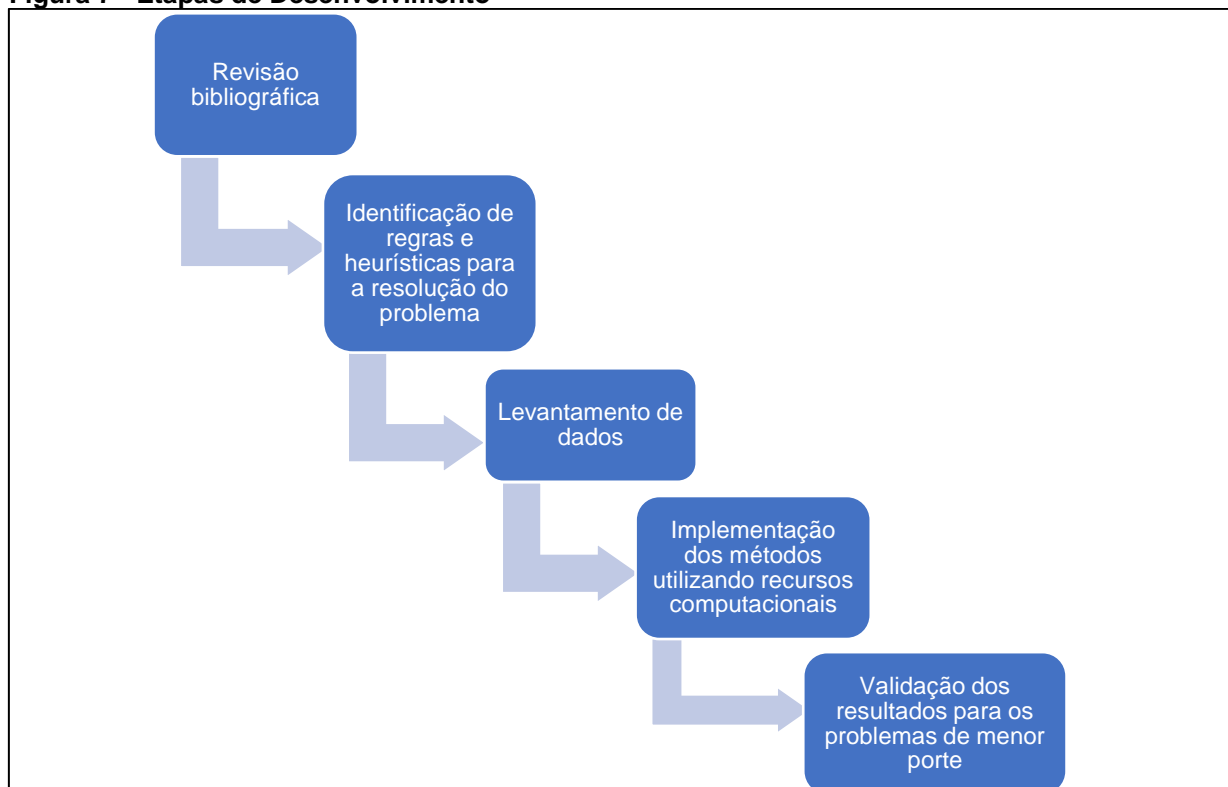
Quanto à forma de abordar o problema, pode-se considerar, de acordo com Turrioni e Mello (2012), a pesquisa como quantitativa, pois os resultados são quantificáveis.

Do ponto de vista dos métodos, utilizando a visão de Turrioni e Mello (2012), pode-se considerar que a pesquisa é feita a partir da modelagem e simulação, por testar através de um modelo, como um sistema responde a variações propostas.

3.2 ETAPAS DE DESENVOLVIMENTO

As etapas de desenvolvimento do trabalho estão delimitadas no fluxograma exibido na Figura 7.

Figura 7 - Etapas de Desenvolvimento



Fonte: Autoria própria

O esquema representado será desenvolvido e explicado na sequência.

3.2.1 Revisão Bibliográfica

A revisão bibliográfica foi realizada para embasar teoricamente o trabalho acerca dos principais temas relacionados a ele.

Foram pesquisados autores do assunto considerados clássicos pela academia, como Tubino, Pinedo, Hillier e Lieberman, e artigos de produção científica, em sua maioria publicados nos últimos dez anos em periódicos como o *Computers & Industrial Engineering*, para que se garanta uma revisão bibliográfica atual e com qualis entre A1 e B2.

Utilizando este filtro pode-se assegurar que a pesquisa tenha maior confiabilidade, visto que se baseia em produções com pertinência adequada na área de estudo e esteja atualizada.

Utilizou-se a plataforma Sucupira para verificar o qualis de periódicos encontrados na área de avaliação considerada pela Capes Engenharias III, grupo do qual a Engenharia de Produção faz parte.

Após isto, as bases de dados Science Direct e Scielo foram utilizadas para encontrar os artigos dos periódicos consultados. Nelas foram inseridas combinações das palavras-chave programação de tarefas, otimização e heurísticas e de outros tópicos relevantes abordados, os periódicos de interesse e também os filtros com relação ao ano de publicação – publicados nos últimos 10 anos – para a pesquisa dos artigos.

Foram selecionados artigos que continham informações compatíveis com o trabalho, os quais foram utilizados na revisão bibliográfica.

3.2.2 Identificação de Regras e Heurísticas para a Resolução do Problema

Algumas regras de ordenação e alocação de tarefas que podem ser aplicadas para a construção de heurísticas que resolvam o problema apresentado foram identificadas na revisão bibliográfica. Para este trabalho, foram consideradas algumas regras de ordenação que são exibidas na Tabela 1.

Tabela 1 - Regras de ordenação utilizadas

Regras consideradas	Número de utilizações
EDD	1
FIFO	2

Fonte: Autoria Própria

A Tabela 1 traz também o número de vezes que cada regra foi utilizada na composição de heurísticas, considerando que uma heurística pode ser composta de uma ou mais regras de ordenação.

3.2.3 Levantamento de Dados

Para implementar as heurísticas são necessários dados. Estes dados incluem informações relevantes para a ordenação, como a quantidade de máquinas e tarefas e outros parâmetros, que podem elencar datas de entrega e tempos de processamento das tarefas em cada máquina.

Para isso, foram utilizadas dez instâncias geradas aleatoriamente por Melo (2016) que relacionam tarefas a serem processadas a máquinas disponíveis para processamento.

Esta relação ocorre ao atribuir os tempos individuais de processamento de cada tarefa em cada máquina e expor as datas de entrega de cada tarefa, representada pelo instante que cada tarefa deve estar processada.

A utilização dos dados se deu no processo de implementação do modelo matemático e das heurísticas, abordadas no Capítulo 4.

3.2.4 Implementação dos Métodos Utilizando Recursos Computacionais

Primeiramente foi implementado no *software* Lingo 17.0 o modelo matemático que tem por objetivo fornecer a solução ótima para o problema, ou seja, fornecer o valor mínimo que a soma de adiantamentos e atrasos no ambiente de máquinas paralelas não relacionadas pode assumir considerando as tarefas, máquinas, tempo de processamento e datas de entrega. Este modelo foi implementado, utilizando as instâncias como parâmetros.

Em seguida as heurísticas idealizadas foram traduzidas para a linguagem de programação C e implementadas utilizando o *software* DEV-C++ 5.1 e, para cada solução, foram gerados gráficos de Gantt ilustrando a solução de atribuição das tarefas às máquinas.

3.2.5 Validação dos Resultados para os Problemas de Menor Porte

Após a implementação do modelo matemático e das heurísticas, os resultados obtidos foram comparados, bem como os tempos de obtenção dos resultados para todos os casos.

Os resultados e discussões estão expostos no Capítulo 5.

4 DESENVOLVIMENTO

O presente capítulo traz o desenvolvimento deste trabalho. Aqui estão detalhadas todas as etapas que levaram o trabalho a ser concluído. As seções presentes neste capítulo são o detalhamento sobre as instâncias utilizadas, a implementação do modelo matemático, além do desenvolvimento e da implementação das heurísticas para a resolução do problema.

4.1 INSTÂNCIAS

Conforme mencionado na Subseção 3.2.3, foram utilizadas dez instâncias que incluem os parâmetros necessários para realizar a programação de tarefas para minimizar a soma de adiantamentos e atrasos no ambiente estudado.

As instâncias podem ser consideradas dados de demanda e informações de capacidade de um sistema produtivo emitidos pelo setor responsável, como por exemplo, o PCP.

Os parâmetros incluídos nas instâncias são os números de tarefas e de máquinas, os tempos de processamento de cada tarefa em cada uma das máquinas disponíveis e, por fim, as datas de entrega de cada tarefa, ou seja, o momento em que as tarefas devem ficar prontas.

Um exemplo de instância está na Tabela 2. Nesta instância, são consideradas seis tarefas que devem ser processadas em duas máquinas.

Tabela 2 - Exemplo de instância

Tarefa	Tempo Máquina 1 (unidades de tempo)	Tempo Máquina 2 (unidades de tempo)	Data de entrega
1	82	28	53
2	62	92	83
3	96	43	92
4	28	37	57
5	92	5	39
6	3	54	36

Fonte: Adaptado de Melo (2016)

Nessa tabela, a primeira coluna, Tarefa, denota o nome da tarefa a ser processada; a segunda e a terceira coluna, respectivamente, Tempo Máquina 1 e Tempo Máquina 2, representam os tempos de processamento de cada uma das tarefas nas máquinas 1 e 2, nesta ordem; e a quarta coluna, Data de Entrega, representa o instante em que cada tarefa deve ser processada.

As informações referentes às quantidades de tarefas e máquinas podem ser observadas na Tabela 3.

Tabela 3 - Relação das instâncias

Instância	Número de tarefas	Número de máquinas
1	4	2
2	6	2
3	8	2
4	12	4
5	16	4
6	20	4
7	30	4
8	50	10
9	100	10
10	200	10

Fonte: Autoria própria

Conforme pode ser observado na Tabela 3, o número de tarefas variou para todas as instâncias seguindo uma ordem crescente, enquanto o número de máquinas, ora se alterou, ora não se alterou.

Este comportamento é abordado, principalmente pelo fato de a quantidade de soluções possíveis aumentar conforme aumenta o número de tarefas consideradas ou o número de máquinas envolvidas.

Outra razão para este comportamento é o fato de se desejar assemelhar o problema a cenários reais. Em muitas indústrias, as máquinas processam muitas tarefas e, simular um cenário que conte apenas com um baixo número de tarefas pode ser considerado desnecessário quando se deseja observar a aplicação da teoria em ambientes produtivos reais.

4.2 IMPLEMENTAÇÃO DO MODELO MATEMÁTICO

O modelo matemático apresentado na Seção 2.6 foi implementado no *software* Lingo versão 17.0, licenciado para estudante e instalado em um computador com processador AMD A4-3305M APU HD *Graphics* 1.9GHz, dois núcleos e sistema operacional Windows 10 Pro de 32 bits.

O modelo foi implementado considerando dez instâncias com números de tarefas e máquinas diferentes. Foram registrados o valor obtido com a programação e o tempo levado para a obtenção do valor.

Estipulou-se um limite de 3600 segundos para a obtenção da resposta. Isso significa que, após este tempo, caso o programa não tivesse obtido a solução ótima, o valor considerado como solução da instância que estava sendo analisada seria o limitante superior encontrado nas iterações realizadas até o momento, podendo ser ou não o valor ótimo.

Este limitante é caracterizado como a melhor solução possível para o problema encontrada nas iterações até o momento em que a implementação foi interrompida.

4.3 MÉTODOS HEURÍSTICOS

Celini Júnior e Silva (2016) desenvolveram três heurísticas para a resolução do problema no ambiente estudado que foram utilizadas, com a permissão dos autores, neste trabalho. Para programá-las foi utilizado um código em linguagem C implementado no *software* DEV-C++ versão 5.1 no mesmo computador e nas mesmas condições em que foi implementado o modelo matemático.

Inicialmente as heurísticas foram executadas e seus resultados para a função objetivo foram armazenados e utilizados para comparação no Capítulo 5.

Após isto, foi desenvolvido um método de melhoria dos resultados que funciona como busca local e implementado para comparação e análise dos resultados.

As definições das heurísticas e do método de melhoria estão descritas nas subseções seguintes.

4.3.1 Heurística Construtiva 1

A heurística construtiva 1 foi a primeira a ser desenvolvida com o intuito de minimizar a soma de adiantamentos e atrasos processando as tarefas com maior rapidez (CELINI JÚNIOR; SILVA, 2016).

O seu funcionamento se dá ao verificar, primeiramente, para cada tarefa, qual o tempo de processamento em cada uma das máquinas disponíveis para sua execução. Após esta verificação, é selecionada, para cada tarefa, a máquina que tem capacidade de processá-la em menor tempo. (CELINI JÚNIOR; SILVA, 2016).

Sendo assim, as tarefas são processadas nas máquinas que as executam com maior velocidade, seguindo para cada máquina a regra FIFO.

A Figura 8 traz o algoritmo que exemplifica a lógica de funcionamento da heurística.

Figura 8 - Algoritmo da Heurística Construtiva 1

```

Entrada: instância
Saída: solução
1 início
2   para cada tarefa i
3     Identificar máquina k que a execute com o menor tempo de processamento
4     Alocar a tarefa i na máquina k identificada
5   fim para
6   retorne solução
7 fim

```

Fonte: Autoria própria

A seção Entrada representa todos os parâmetros utilizados para que a heurística seja implementada. Neste caso, a única entrada é a instância, pois esta considera e relaciona o número de tarefas e máquinas com os tempos de processamento e as datas de entrega para cada caso.

A seção Saída representa, como o próprio nome diz, a saída do programa, ou seja, a solução após todas as iterações terem sido realizadas. Para todos os métodos desenvolvidos, a seção Saída representa a alocação e a ordenação das tarefas nas máquinas e para estas alocações e ordenações, a soma dos adiantamentos e atrasos obtidos.

Um exemplo da utilização do método pode ser observado na Tabela 4, que traz os tempos de processamento para três tarefas em três máquinas paralelas não relacionadas e, na Figura 9, com a alocação das tarefas.

Tabela 4 - Tempos e datas de entrega para execução da Heurística Construtiva 1

Máquina	T_1	T_2	T_3	T_4	T_5
M_A	30	8	43	55	35
M_B	42	60	32	52	27
M_C	56	75	40	39	25
Data de entrega	42	35	33	50	52

Fonte: Autoria própria

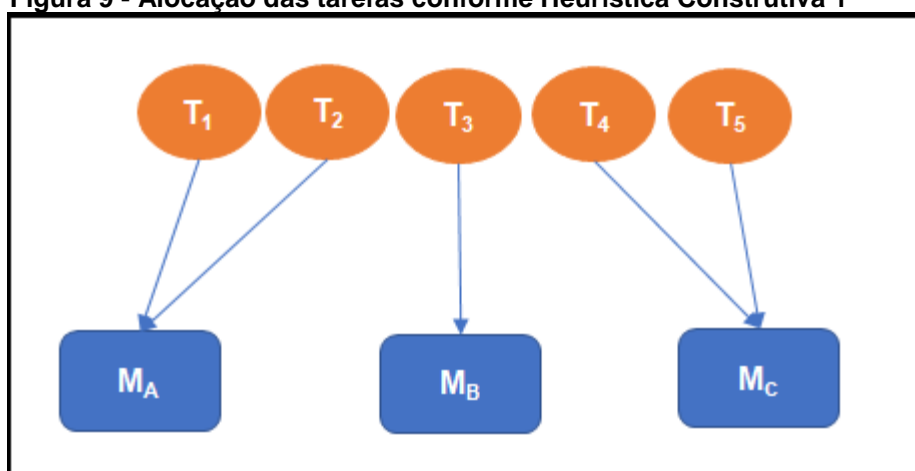
Na Tabela 4, M_A , M_B e M_C representam, respectivamente, as máquinas A, B e C disponíveis para o processamento das tarefas 1, 2, 3, 4 e 5 representadas por T_1 , T_2 e T_3 , T_4 e T_5 na mesma ordem.

Ao aplicar a Heurística Construtiva 1, identifica-se primeiramente qual máquina tem capacidade de processar cada tarefa em menor tempo. Cada tarefa é alocada à máquina mais veloz para ela.

Obtém-se que T_1 e T_2 são processadas em menor tempo em M_A , T_3 é processada em menor tempo em M_B e T_4 e T_5 são processadas em menor tempo em M_C .

Aloca-se então as tarefas conforme a Figura 9.

Figura 9 - Alocação das tarefas conforme Heurística Construtiva 1

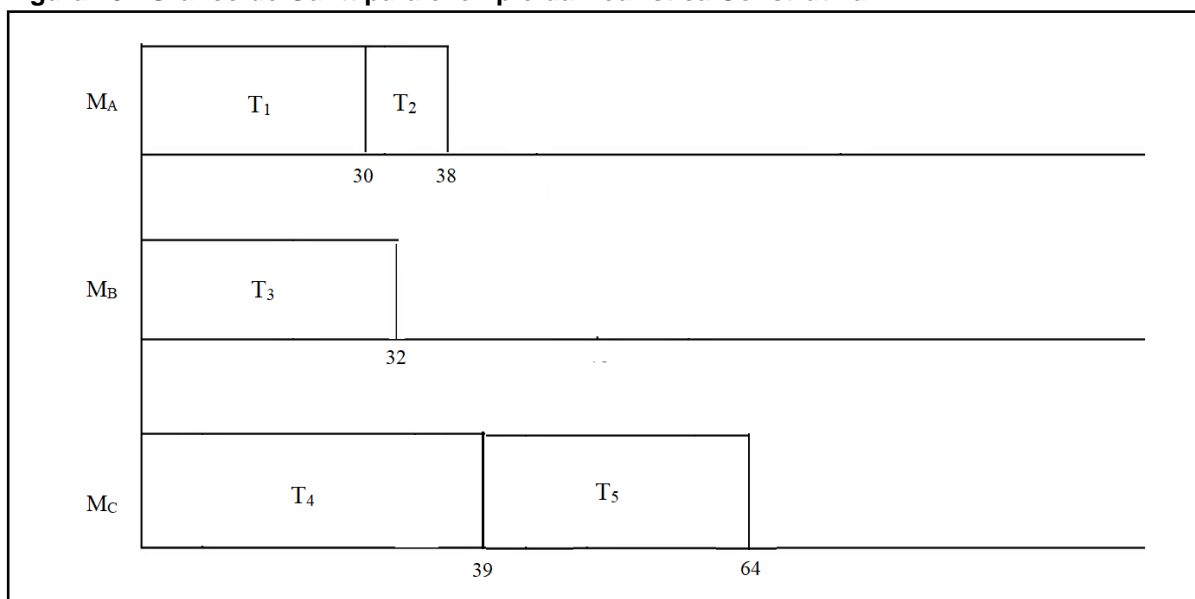


Fonte: Autoria própria

Observa-se que neste caso, todas as máquinas têm uma tarefa alocada. Isto não precisa acontecer, visto que as máquinas não precisam ser todas ocupadas. Apenas deve-se considerar os tempos de processamento e alocar as tarefas nas máquinas mais rápidas, mesmo que isto signifique que algum outro recurso fique ocioso.

O gráfico de Gantt que representa a alocação e a ordenação deste exemplo está contido na Figura 10.

Figura 10 - Gráfico de Gantt para exemplo da Heurística Construtiva 1



Fonte: Autoria própria

De acordo com a figura, as tarefas estão alocadas nas máquinas seguindo a regra FIFO, ou seja, as primeiras tarefas percorridas pela heurística são as primeiras a serem alocadas.

4.3.2 Heurística Construtiva 2

A Heurística Construtiva 2 foi desenvolvida com o intuito de propor uma melhoria à heurística apresentada anteriormente. Assim como na primeira heurística, deve-se verificar primeiramente, para cada tarefa, qual o tempo de processamento em cada uma das máquinas em que elas podem ser executadas (CELINI JÚNIOR; SILVA, 2016).

Após essa verificação, as tarefas são alocadas às máquinas com capacidade de processá-las com mais rapidez. Aplica-se então, a regra EDD para cada máquina.

Ou seja, serão processadas primeiro as tarefas com data de entrega mais próxima. Este método tem como objetivo principal processar as tarefas com datas de entrega mais próximas nas máquinas que conseguem executá-las com maior eficiência (CELINI JÚNIOR; SILVA, 2016).

Utilizar a regra EDD na solução objetivo deste trabalho faz sentido, principalmente pelo fato de se processar as tarefas que precisam ser processadas primeiro. Este comportamento pode evitar que atrasos e adiantamentos desnecessários aconteçam.

A Figura 11 traz o algoritmo com a lógica utilizada para implementação da Heurística 2.

Figura 11 - Algoritmo da Heurística Construtiva 2

```

Entrada: instância
Saída: solução
1 início
2   para cada tarefa i
3     | Identificar máquina k que a execute com o menor tempo de processamento
4     | Alocar a tarefa i na máquina k identificada
5   fim para
6   para cada máquina k
7     | Aplicar a Regra EDD para as tarefas alocadas à máquina
8   fim para
6   retorne solução
7 fim

```

Fonte: Autoria própria

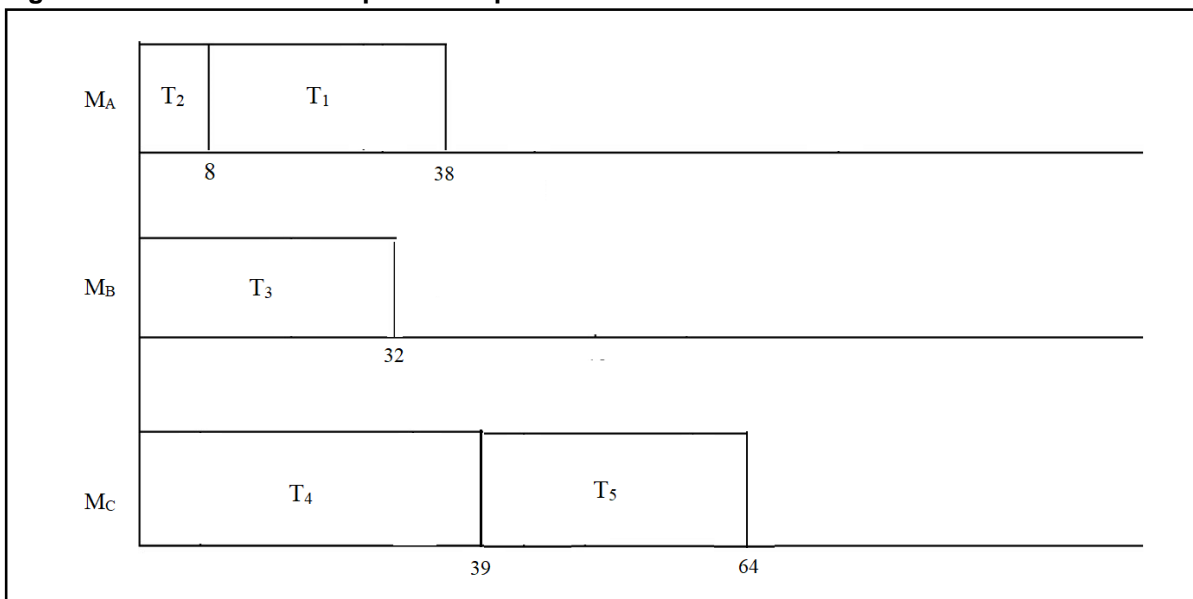
As seções Entrada e Saída no algoritmo são iguais às do algoritmo apresentado na primeira heurística, ou seja, os parâmetros e a exibição da solução, respectivamente.

Retomando o exemplo visto na subseção anterior, que trazia T_1 e T_2 alocadas em M_A , T_3 em M_B e T_4 e T_5 em M_C pode-se aplicar a Heurística Construtiva 2 na ordenação das tarefas que utilizam a mesma máquina para serem processadas.

De acordo com a heurística, após alocar as tarefas, deve-se aplicar a regra EDD em cada máquina que contém tarefas alocadas.

A Figura 12 traz para as máquinas num gráfico de Gantt a nova ordenação das tarefas.

Figura 12 - Gráfico de Gantt para exemplo da Heurística Construtiva 2



Fonte: Autoria própria

A lógica da primeira heurística ordena as tarefas nas máquinas conforme seus parâmetros são lidos, diferentemente da segunda. Se neste caso, a primeira tarefa lida tivesse sido a T_1 , ela seria processada primeiro em M_A .

Aplicando então a Heurística 2, em que a ordenação nas máquinas ocorre por data de entrega mais próxima, a ordem em M_A se alteraria, sendo T_2 processada primeiro que T_1 .

O caso se repete para todas as outras máquinas. Como T_4 tem menor data de entrega, a ordenação em M_C continua a mesma.

No caso de M_B que só tem uma tarefa alocada, não faz sentido aplicar a regra EDD a elas.

4.3.3 Heurística Construtiva 3

Assim como a Heurística Construtiva 2, a terceira heurística foi desenvolvida como um aprimoramento da primeira. Como nas heurísticas anteriores, a Heurística Construtiva 3 primeiramente identifica qual máquina tem capacidade de processar cada uma das tarefas com mais rapidez (CELINI JÚNIOR; SILVA, 2016).

Após encontrar a máquina correspondente ao processamento mais eficiente, verifica-se a ocupação desta máquina. As tarefas são então alocadas às máquinas

mais rápidas para realizar o processamento, desde que estas estejam desocupadas no instante em que o processo pode iniciar (CELINI JÚNIOR; SILVA, 2016).

Sendo assim, as tarefas são alocadas às máquinas que as processam mais rápido e, caso a máquina esteja ocupada no momento em que a tarefa pode começar a ser processada, será alocada à próxima máquina mais eficiente para processamento disponível desde que esta troca seja viável.

Em outras palavras, cada tarefa é alocada à máquina mais eficiente para ela e por fim, o processamento ocorre de forma similar à regra FIFO.

Como se assume a premissa de que as tarefas não sejam dependentes umas das outras e estejam prontas para serem processadas desde o instante inicial, a lógica da heurística se torna eficaz.

No geral, este método busca diminuir a fila de espera para processamento das tarefas nas máquinas, possibilitando o processamento das tarefas em todos os recursos disponíveis, ainda considerando a possibilidade de que as tarefas sejam processadas nas máquinas mais rápidas.

A Figura 13 traz o algoritmo com a lógica utilizada para implementação da Heurística Construtiva 3.

Figura 13 - Algoritmo da Heurística Construtiva 3

```

Entrada: instância
Saída: solução
1 início
2 para cada tarefa i
3   Identificar máquina k que a finalize com o menor tempo de processamento
4   se a máquina mais eficiente estiver ocupada
5     Identificar próxima máquina k que finalize a tarefa i com menor tempo
6     Verificar a viabilidade da alocação
7     se é viável trocar a tarefa
8       |Alocar a tarefa nesta máquina
9     fim se
10    senão
11      |Alocar a tarefa na máquina mais eficiente
12    fim senão
13  fim se
14  senão
15    |Alocar a tarefa na máquina mais eficiente
16  fim senão
17 fim para
18 retorne solução
19 fim

```

Fonte: Autoria própria

As seções Entrada e Saída no algoritmo, da mesma forma que nas lógicas das primeiras heurísticas apresentadas, representam os parâmetros e a exibição da solução, respectivamente.

O exemplo apresentado na Subseção 4.2.1 também pode ser retomado e sofrer a aplicação da Heurística Construtiva 3.

Ao fazer a análise dos tempos de processamento das tarefas, a heurística as percorre de uma a uma obedecendo a ordem em que as tarefas são mostradas nas instâncias, a entrada do algoritmo.

T_1 e T_2 têm os menores tempos de processamento na mesma máquina. Para este caso, T_1 é, então alocada à esta máquina (M_A) por ser lida primeiro na heurística.

T_2 é lida em seguida e logo alocada à M_A também. Entretanto M_A , neste momento, está ocupada processando T_1 . Então verifica-se se é mais viável aloca-la à próxima máquina que a processa mais rápido (M_B), ou à M_A , mesmo que ocupada.

O término de T_2 em M_A se dará no instante 38 e o de T_2 em M_B , no instante 60, confirmando que a atribuição de T_2 em M_B pode ser muito prejudicial ao problema. Aloca-se então T_2 à M_A .

A tarefa T_3 que anteriormente estava alocada à M_B continua na mesma posição, pelo fato de no momento em que for lida pela heurística, nenhuma tarefa estar alocada à sua máquina mais eficiente.

O mesmo ocorre para T_4 , que é processada com maior eficiência em M_C , máquina que no momento em que a tarefa for lida na instância estará disponível.

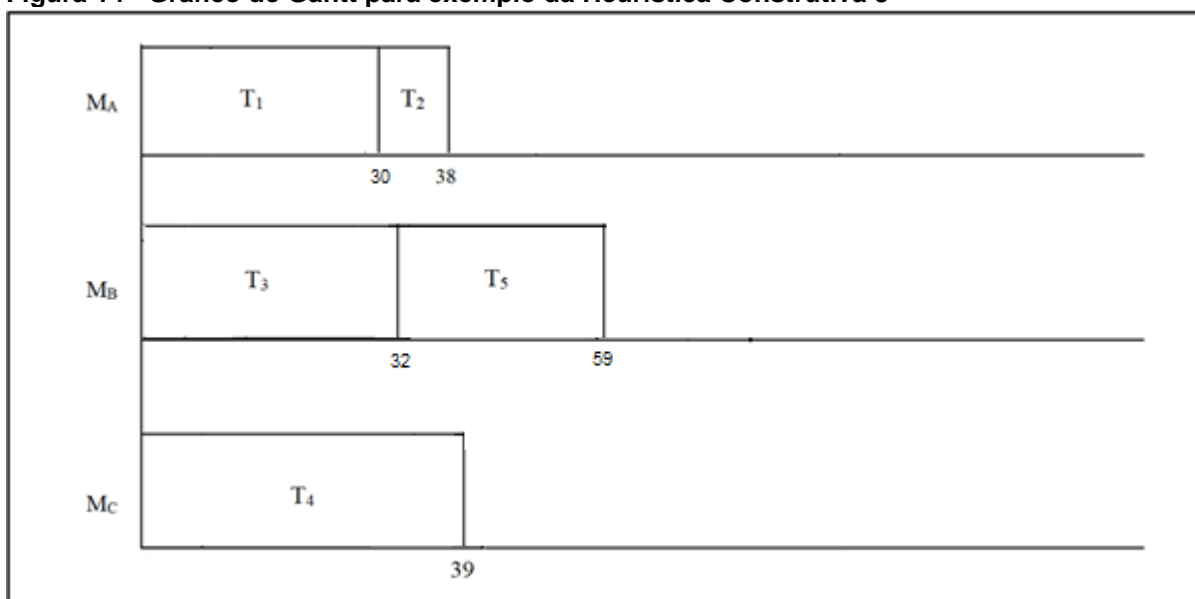
Já T_5 , é processada com maior eficiência em M_C , que está ocupada processando T_4 . As próximas máquinas mais rápidas para processá-la são M_B e M_A , respectivamente e estão ambas ocupadas.

Olha-se então para a viabilidade de se trocar M_C por alguma das outras máquinas, que no caso da troca ser feita por M_B , o término do processamento será no instante 59, e caso seja por M_A , no instante 73, contra 64 para o término do processamento em M_C .

Aloca-se então T_5 à M_B por sofrer menos com a ocupação das máquinas para processamento.

A Figura 14 traz o esquema de alocação para esta heurística.

Figura 14 - Gráfico de Gantt para exemplo da Heurística Construtiva 3



Fonte: Autoria própria

Observa-se para este caso que todas as máquinas foram utilizadas, comprovando um dos objetivos do método, que é o de lidar com o tempo de espera das tarefas para processamento com a utilização de todas as máquinas.

4.3.4 Heurística de Melhoria

A quarta heurística proposta é considerada uma heurística de melhoria, visto que resultados obtidos a partir de uma alocação inicial realizada por outras heurísticas podem ser melhorados com a execução da mesma.

Após a alocação e o sequenciamento terem sido realizados por uma heurística construtiva, verifica-se então, qual tarefa está adiantada e qual tarefa está atrasada para realizar alterações nelas.

O objetivo desta heurística é alcançar melhoria nos resultados obtidos por outras heurísticas utilizadas para a resolução do problema de minimização da soma de adiantamentos e atrasos no ambiente de máquinas paralelas não relacionadas, realizando a troca de posições de tarefas atrasadas e adiantadas com suas antecessoras e sucessoras, respectivamente até que não se obtenha mais melhorias.

As tarefas atrasadas trocam de posição com as suas predecessoras, desde que não sejam as únicas ocupantes da máquina ou as primeiras a serem alocadas.

As tarefas adiantadas trocam de posição com suas sucessoras, desde que não sejam as únicas ocupantes da máquina ou as últimas tarefas alocadas nela.

Avalia-se então, após cada troca a soma de adiantamentos e atrasos.

Se esta soma for maior do que a obtida antes da troca ser realizada, mantém-se o valor anterior e a troca é desfeita até finalizar-se o método. Caso o oposto ocorra, e a soma for menor do que a obtida antes da troca, repete-se o procedimento até que não se obtenha uma solução melhor.

Finaliza-se, então o método atribuindo o valor trocado à solução objetivo, que se torna a melhor solução encontrada na Heurística de Melhoria.

A Figura 15 traz o algoritmo com a lógica utilizada para implementação da Heurística de Melhoria.

Figura 15 - Algoritmo da Heurística de Melhoria

```

Entrada: instância
Saída: solução melhorada
1 início
2 Construir solução inicial usando uma heurística construtiva
3 enquanto houver melhoria na solução
4   para cada tarefa i
5     se tarefa i estiver atrasada e não for a primeira e/ou única na máquina
6       troca a tarefa i de posição com sua predecessora
7       se solução melhorou
8         mantém nova solução
9       fim se
10      senão
11        restaura solução anterior
12      fim senão
13    fim se
14    se tarefa i estiver adiantada e não for a última e/ou única na máquina
15      troca a tarefa i de posição com sua sucessora
16      se solução melhorou
17        mantém nova solução
18      fim se
19      senão
20        restaura solução anterior
21      fim senão
22    fim se
23  fim para
24 fim enquanto
25 retorne solução melhorada
26 fim

```

Fonte: Autoria própria

O exemplo abordado nas subseções anteriores que trazia a alocação de acordo com as heurísticas para às máquinas e ordenação das tarefas pode ser considerado para a heurística de melhoria também. Entretanto, para este caso, é

necessário considerar as unidades de tempo de atraso e de adiantamento para poder aplicá-la.

Para a aplicação da melhoria considerando a ordenação da Heurística Construtiva 2, deve-se calcular as unidades de adiantamentos e atrasos obtidas nas máquinas em que existe mais de uma tarefa alocada, requisito da heurística de melhoria.

A Tabela 5 apresenta os resultados por tarefa em M_A e M_C , únicas máquinas no exemplo em que houve a alocação de mais de uma tarefa.

Tabela 5 - Resultados para M_A e M_C na Heurística Construtiva 2

Resultado/Tarefa	T_1	T_2	T_3	T_4	T_5
Atraso	0	0	0	0	12
Adiantamento	4	27	1	11	0
Total por máquina	$M_A=31$		$M_B=1$	$M_C=23$	
Total	55				

Fonte: Autoria própria

Os cálculos de atraso e adiantamento são feitos considerando que as tarefas começam a ser processadas de acordo com a ordem definida, no instante 0. Ao término do processamento, compara-se o instante de término com a data de entrega. Se o instante de término é igual à data de entrega, a tarefa não atrasou e nem adiantou, condição ideal para este problema. A tarefa é considerada atrasada se o instante de término é maior que a data de entrega e adiantada, caso o instante de término seja menor que a data de entrega.

Pode-se observar na Tabela 5 que apenas uma das tarefas atrasou, e o restante das tarefas adiantou, o que torna possível a aplicação da heurística de melhoria.

Ao aplicar a heurística, observa-se que T_1 é a última tarefa alocada na máquina e estando adiantada, impossibilita a troca por não ter uma sucessora. Entretanto, T_2 também está adiantada e é a primeira de duas tarefas alocadas na máquina, ou seja, possui uma sucessora. T_2 é então trocada por T_1 .

O mesmo caso de T_1 e T_2 é válido para T_4 e T_5 . Realiza-se a troca entre T_5 e T_4 .

Os resultados obtidos com as trocas estão contidos na Tabela 6.

Tabela 6 - Resultados para M_A após troca

Resultado/Tarefa	T_1	T_2	T_3	T_4	T_5
Atraso	0	3	0	15	0
Adiantamento	12	0	1	0	27
Total por máquina	$M_A=15$		$M_B=1$	$M_C=42$	
Total	58				

Fonte: Autoria própria

Pela Tabela 6, observa-se que o total para M_A que processa as tarefas T_1 e T_2 o resultado melhorou consideravelmente. O adiantamento de T_1 diminuiu 15 unidades de tempo e o de T_2 se tornou um atraso de 3 unidades de tempo.

Neste caso, mantém-se a troca para a formulação da solução final do problema.

Já para M_C , o resultado piorou, chegando a quase o dobro do que era antes da troca ser realizada. Para este caso, mantém-se a primeira solução obtida, descartando a troca realizada.

A análise do problema considera iguais os impactos obtidos com adiantamentos e atrasos, podendo-se então analisar o fato de alguns adiantamentos terem se tornado atrasos apenas pelas unidades de tempo.

Como o objetivo do problema estudado é obter a menor soma possível de adiantamentos e atrasos, considera-se que a troca realizada em M_A concedeu uma melhoria na solução.

Neste exemplo, como as máquinas analisadas possuem apenas duas tarefas alocadas cada uma, uma nova iteração retornará a solução inicial, que conforme já visto, é pior do que a solução melhorada da primeira iteração da heurística e, como o método mantém a melhor solução obtida como a solução final do problema, a primeira troca será dada como a solução final do exemplo.

5 APRESENTAÇÃO E ANÁLISE DOS RESULTADOS

O Capítulo 5 traz a apresentação dos resultados do que foi apresentado e discutido no Capítulo 4, ou seja, os resultados obtidos com a implementação do modelo matemático, das heurísticas construtivas estudadas e da heurística de melhoria proposta, bem como a análise destes.

5.1 RESULTADOS DO MODELO MATEMÁTICO

O modelo matemático, desde que fornecido tempo de execução suficiente, traz o melhor resultado, ou seja, o resultado ótimo para o problema de minimização da soma de adiantamentos e atrasos no ambiente de máquinas paralelas não relacionadas para as instâncias apresentadas.

Neste problema, considera-se que atrasos e adiantamentos possuem as mesmas influências para o sistema produtivo. Tarefas adiantadas são tão ruins quanto tarefas atrasadas.

Os adiantamentos e atrasos foram obtidos ao comparar na melhor ordenação, quando obtida, a diferença entre o instante em que a tarefa está programada para terminar de ser processada com a data de entrega prevista para a mesma tarefa.

O ideal para este problema é que os instantes de término das tarefas sejam iguais aos instantes em que estas devem estar prontas, ou seja, suas datas de entrega.

Caso o instante de término de uma tarefa seja maior que a sua data de entrega, pode-se considerar um atraso quantificado pela diferença do instante de término com o instante de entrega. O mesmo é válido para o adiantamento, entretanto o instante de término neste caso é menor do que a data de entrega.

Utilizando o programa Lingo, obtiveram-se os resultados contidos na Tabela 7, que também inclui os tempos de processamento para cada instância.

Conforme mencionado na Seção 4.1 deste trabalho, foi estipulado um tempo limite de 3600 segundos para que o programa obtivesse uma solução para o problema, seja esta a ótima global ou um limitante superior.

Tabela 7 - Resultados para o modelo matemático

Instância	Soma de adiantamentos e atrasos	Tempo de resolução (s)
1	72	0,31
2	57	11,78
3	66	17,49
4	[84]	3600,00
5	[124]	3600,00
6	[805]	3600,00
7	[2670]	3600,00
8	[4083]	3600,00
9	-	3600,00
10	-	3600,00

Fonte: Autoria própria

Ao analisar os resultados individuais, observa-se que as primeiras três instâncias obtiveram como resposta a solução ótima para o problema em um tempo relativamente curto. Entretanto o número de tarefas contido nas instâncias era muito reduzido – 4, 6 e 8 tarefas para serem processadas em 2 máquinas.

Os resultados entre colchetes, ou seja, os das instâncias 4, 5, 6, 7 e 8 não podem ser considerados resultados ótimos, visto que o programa não terminou a resolução em 3600 segundos. Estes resultados são os limitantes superiores obtidos ao término do tempo designado para a execução.

Isto acontece pela complexidade do problema ter aumentado, resultado este do aumento do número de tarefas.

Os programas de simulação matemática ainda não são capazes de obter a solução ótima para problemas de alta complexidade em tempos eficientes, o que pode ser observado nos resultados.

Apesar de se ter interrompido a simulação para as instâncias de 4 a 8, na maioria dos casos, as soluções consideradas foram as melhores encontradas desde aproximadamente 1800 segundos. Isso pode significar que os resultados estejam próximos do ótimo, mas não há qualquer garantia disso.

No caso das instâncias 9 e 10, que envolvem, respectivamente 100 e 200 tarefas para serem processadas em 10 máquinas, mesmo após uma hora e resolução,

não foram encontradas soluções para o problema, o que evidencia as limitações da resolução exata.

Uma hipótese para isto são as características do computador utilizado para simular o modelo matemático. Possivelmente, utilizando um computador mais veloz essa situação melhora. Entretanto acredita-se, pelas características do problema e por sua complexidade, que a situação das instâncias anteriores que não obtiveram a solução ótima se repita.

Como a maioria dos problemas reais envolvem altos números de tarefas, semelhantes aos das instâncias 9 e 10, comprova-se que o método do modelo matemático que traz soluções ótimas não é aplicável, fazendo-se necessário o uso de métodos alternativos como os desenvolvidos neste trabalho.

5.2 RESULTADOS PARA A HEURÍSTICA CONSTRUTIVA 1

Os resultados obtidos com a implementação da Heurística Construtiva 1 estão contidos na Tabela 8, bem como os tempos de processamento individuais levados para a alocação das tarefas às máquinas e seu sequenciamento.

Tabela 8 - Resultados da Heurística Construtiva 1

Instância	Resultado	Resultado do modelo matemático	Tempo de resolução (s)
1	124	72	2,13
2	194	57	2,02
3	206	66	2,23
4	417	[84]	2,52
5	495	[124]	2,69
6	1124	[805]	1,99
7	1227	[2670]	2,58
8	997	[4083]	2,54
9	4276	-	3,04
10	18505	-	3,20

Fonte: Autoria própria

Observando a Tabela 8 e comparando os resultados da Heurística Construtiva 1 aos da programação matemática, pode-se perceber que os resultados não podem

ser considerados ótimos. Ao analisar as instâncias de 1 a 3 observa-se que os resultados foram piores do que para os resultados ótimos obtidos.

Para as instâncias 4, 5 e 6 o resultado obtido é, também, pior do que os menores resultados obtidos nos 3600 segundos em que o modelo matemático foi implementado no Lingo.

Já para as instâncias 7 e 8, os resultados obtidos com a Heurística Construtiva 1 são melhores do que os obtidos pelo Lingo. Mostra-se, então, que para estas instâncias, nas condições de implementação, o resultado do modelo matemático em 3600 segundos não é o ponto ótimo global.

Pode-se observar também para este caso a diferença entre os tempos de obtenção dos resultados do modelo matemático e dos métodos heurísticos.

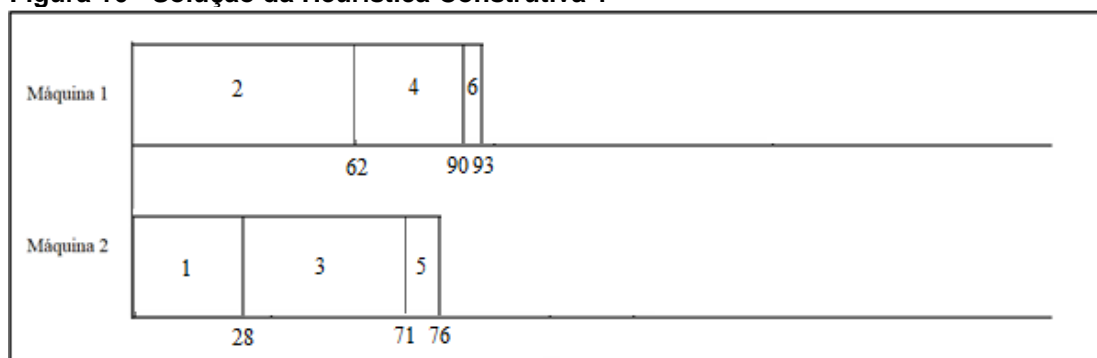
A eficácia da heurística é muito bem recebida pelo fato de se ter obtido valores menores do que os limitantes da programação matemática em um tempo muito curto, o que se assemelha ao aceitável em problemas reais.

Com base nestes resultados, pode-se confiar na heurística para a resolução de problemas com um alto número de tarefas e máquinas para alocação em um curto período de tempo.

O tempo para obtenção da resposta para a primeira instância foi maior do que o do modelo matemático. Entretanto a situação muda para todas as outras, visto que para a heurística o tempo de obtenção de resposta tem uma pequena variação, mantendo-se sempre em poucos segundos diferentemente do modelo matemático, que já para um pequeno número de tarefas tem um grande aumento, perdendo sua eficiência.

A Figura 16 traz um gráfico de Gantt com a solução para a instância 2 utilizando a Heurística Construtiva 1.

Figura 16 - Solução da Heurística Construtiva 1



Fonte: Autoria própria

Na Figura 16, os números representam as tarefas e M1 e M2 representam as duas máquinas disponíveis para processamento das tarefas.

Pode-se observar que, neste caso, as primeiras tarefas que serão processadas em cada máquina, são as primeiras tarefas lidas nas instâncias, as tarefas 1 em M2, 2 em M1, 3 em M2 e assim sucessivamente.

Isto se dá pelo fato de a heurística utilizar para ordenação a regra FIFO.

Aplicou-se então a Heurística de Melhoria sobre os resultados da Heurística Construtiva 1 e obtiveram-se os resultados contidos na Tabela 9.

Tabela 9 - Resultados da Heurística Construtiva 1 após melhoria

Instância	Resultado	Resultado após melhoria	Melhoria (%)	Tempo de resolução (s)
1	124	124	0,00	2,28
2	194	91	53,09	2,62
3	206	126	38,83	2,80
4	417	293	29,74	3,56
5	495	487	1,62	2,88
6	1124	678	39,68	2,63
7	1227	732	40,34	3,75
8	997	564	43,43	3,07
9	4276	2855	33,23	5,14
10	18505	11116	39,93	7,70

Fonte: Autoria própria

Os resultados para a Heurística Construtiva 1 foram retomados na Tabela 9 para que se observe as mudanças nos valores da solução com relação a Heurística de Melhoria em algumas das instâncias, como a 2, abordada na Figura 17, antes da aplicação do método de melhoria.

A maioria das instâncias obteve melhorias nos resultados, ainda que os tempos de obtenção da solução também sejam maiores.

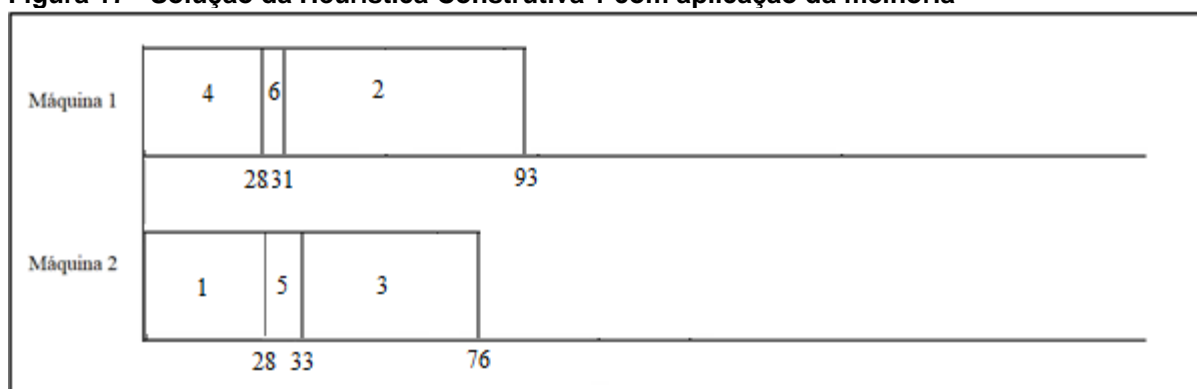
Os tempos de resolução aumentaram para as instâncias com maiores números de tarefas. Isto se dá porque ao aumentar o número de tarefas nas máquinas, aumenta-se também a possibilidade de trocas entre tarefas atrasadas e adiantadas com suas vizinhas obterem maior sucesso, pois aumentam as possibilidades.

Entretanto, o aumento no tempo continua tendo uma significância bem reduzida quando comparado aos valores de tempo da solução do modelo matemático ou quando se utiliza o método para aplicar a problemas reais, principalmente ao se observar a melhoria percentual dos resultados, que chega a 53,09%.

Observa-se ainda, que na Instância 1 que o resultado da Heurística 1 foi mantido ao não se obter uma solução melhor com as trocas.

A Figura 17 traz o gráfico de Gantt da Instância 2 após as ordenações nas máquinas terem sido alteradas com as trocas do método de melhoria.

Figura 17 - Solução da Heurística Construtiva 1 com aplicação da melhoria



Fonte: Autoria própria

Pode-se observar após a melhoria que a ordem de alocação das tarefas se alterou nas duas máquinas.

Em M1, a tarefa 2 que era processada primeiro passou a ser processada por último com a nova ordenação. Em M2 a única alteração observada foi entre as tarefas 5 e 3, que trocaram de posição de processamento.

Estas trocas de posição na alocação inicial propiciaram uma mudança na solução de quase 100 unidades de soma de adiantamentos e atrasos.

5.3 RESULTADOS PARA A HEURÍSTICA CONSTRUTIVA 2

Os resultados obtidos com a implementação da Heurística 2 são mostrados na Tabela 10, onde também são exibidos os tempos de obtenção das soluções para cada instância.

Tabela 10 - Resultados da Heurística Construtiva 2

Instância	Resultado	Resultado do modelo matemático	Tempo de resolução (s)
1	80	72	1,65
2	139	57	2,02
3	159	66	2,48
4	363	[84]	2,73
5	555	[124]	2,62
6	1063	[805]	2,65
7	757	[2670]	2,84
8	556	[4083]	2,09
9	2494	-	2,98
10	8619	-	3,87

Fonte: Autoria própria

Os resultados exibidos na Tabela 10 mostram que, assim como na heurística anterior, para as instâncias 1, 2 e 3, não se obteve a solução ótima global para o problema. Entretanto os resultados são consideravelmente melhores do que os da Heurística 1, se aproximando da solução ótima, por exemplo, na instância 1.

Para as instâncias 4, 5 e 6, os resultados seguem os mesmos padrões da primeira heurística construtiva, sendo as soluções neste caso, maiores do que os resultados obtidos no modelo matemático.

Já para as instâncias 7 e 8, os resultados também são melhores do que os obtidos na implementação do modelo matemático, reforçando que os valores obtidos em 3600 segundos não são ótimos.

Pode-se observar também no Quadro 7 que com exceção de um caso, na instância 5, a Heurística Construtiva 2 apresenta resultados melhores do que os resultados da Heurística Construtiva 1, inclusive nas instâncias mais complexas, com maiores números de tarefas e máquinas.

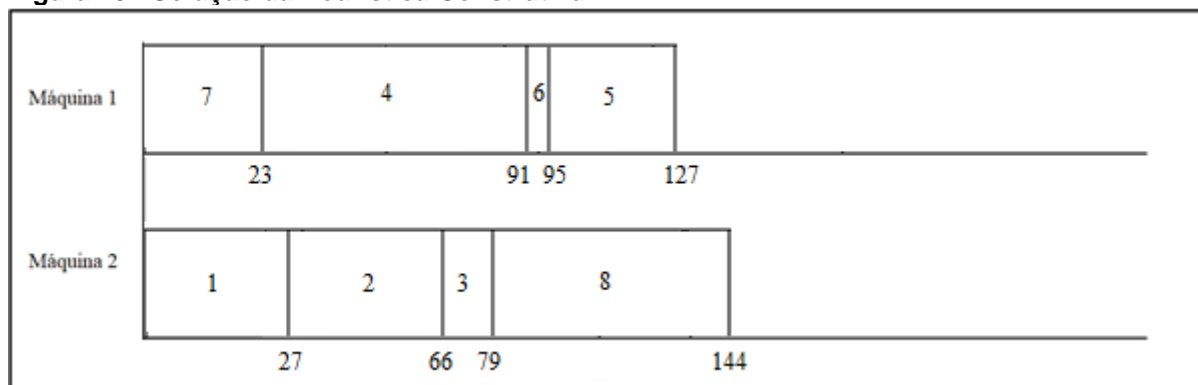
Isto pode ser explicado pelo fato de as tarefas serem ordenadas pela regra EDD, que de certa forma elimina atrasos desnecessários processando as tarefas com menor data de entrega primeiro.

Os resultados da Heurística Construtiva 1 eram considerados bons e, sendo os da Heurística Construtiva 2 melhores em 90% dos casos abordados, a torna muito mais atraente.

Observa-se ainda que os tempos de obtenção de resposta para a Heurística 2 não sofrem aumentos bruscos quando comparados aos tempos da Heurística 1, ainda que uma outra regra seja aplicada.

A Figura 18 traz o gráfico de Gantt com a solução proposta pela Heurística Construtiva 2 para a instância 3.

Figura 18 - Solução da Heurística Construtiva 2



Fonte: Autoria própria

Diferentemente do caso anterior, nesta heurística, as alocações em cada máquina seguem uma regra (EDD), então pode-se observar que não necessariamente as primeiras tarefas lidas serão as primeiras a serem processadas.

Aplica-se então a heurística de melhoria para verificar possíveis alterações nas soluções para cada instância.

Os novos resultados podem ser observados na Tabela 11.

Tabela 11 - Resultados da Heurística Construtiva 2 após melhoria

Instância	Resultado	Resultado após melhoria	Melhoria (%)	Tempo de resolução (s)
1	80	80	0,00	2,71
2	139	139	0,00	2,42
3	159	126	20,75	2,55
4	363	261	28,09	3,17
5	555	555	0,00	2,72
6	1063	659	38,01	2,71
7	757	721	4,76	2,72
8	556	517	7,01	2,27
9	2494	2316	6,74	4,21
10	8619	6922	19,69	7,60

Fonte: Autoria própria

Pode-se observar na Tabela 11, que assim como na Heurística Construtiva 1, resultados melhores foram alcançados após a heurística de melhoria ser acoplada ao método.

Ao aplicar a heurística de melhoria a regra EDD deixa de ser seguida para toda a ordenação quando se realizam trocas, entretanto os resultados se tornam mais satisfatórios quando se encontra uma solução melhor do que a inicial, o que neste caso foi para 70% das instâncias testadas.

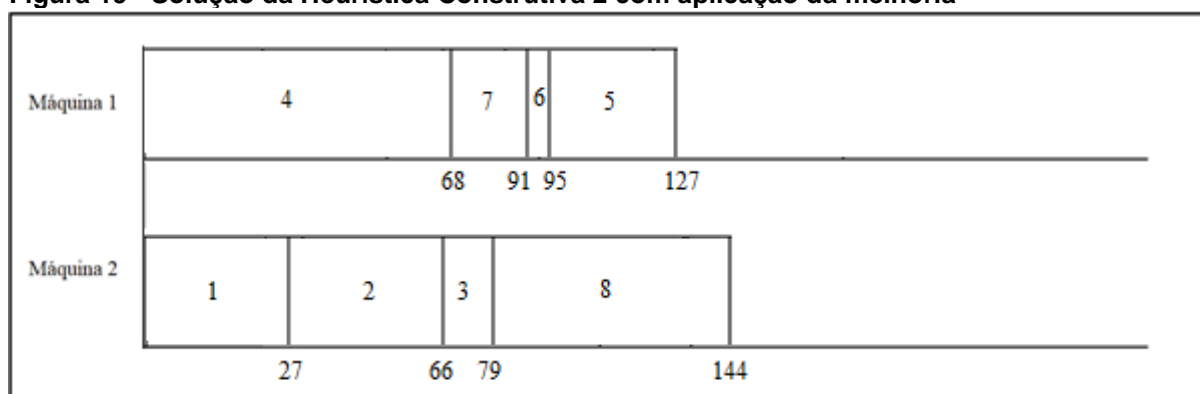
Com relação aos tempos de processamento, pode-se observar um aumento, principalmente nas instâncias com maiores números de tarefas, mas assim como nos casos anteriores, este tempo ainda é curto e compatível com aplicações práticas do problema.

A explicação para isto é a mesma do aumento dos tempos na aplicação da melhoria na primeira heurística construtiva. O maior número de tarefas possibilita um maior número de trocas para serem testadas, o que aumenta proporcionalmente o tempo de resposta do programa.

Ainda que o tempo seja maior, ainda é muito mais viável manter a combinação das heurísticas do que manter a programação matemática.

A Figura 19 traz o gráfico de Gantt para a mesma instância observada antes da aplicação do método de melhoria.

Figura 19 - Solução da Heurística Construtiva 2 com aplicação da melhoria



Fonte: Autoria própria

Para M1, pode-se observar apenas uma troca na posição inicial, que foi entre as tarefas 4 e 7.

Em M2 nenhuma troca foi efetivada, o que pode significar que nenhuma das tarefas da máquina atrasou ou adiantou, ou que a troca entre elas trouxe resultados piores do que os obtidos com a ordenação inicial.

Ainda que a alocação das tarefas seja a mesma nas máquinas considerando as heurísticas construtivas 1 e 2, dificilmente a ordenação das tarefas para as duas heurísticas será igual.

Isto se dá porque ao realizar uma troca entre tarefas atrasadas ou adiantadas com suas vizinhas, existe a restrição de que as tarefas atrasadas não podem ser as primeiras de cada máquina e que tarefas adiantadas não podem ser as últimas alocadas.

5.4 RESULTADOS PARA A HEURÍSTICA CONSTRUTIVA 3

Os resultados obtidos com a implementação da Heurística 3 são mostrados na Tabela 12.

Nele também são exibidos os tempos de obtenção das soluções para cada instância.

Tabela 12 - Resultados da Heurística Construtiva 3

Instância	Resultado	Resultado do modelo matemático	Tempo de resolução (s)
1	124	72	2,15
2	194	57	2,03
3	173	66	2,10
4	417	[84]	1,98
5	495	[124]	2,54
6	903	[805]	2,40
7	989	[2670]	2,09
8	1088	[4083]	3,23
9	4556	-	3,96
10	24216	-	4,01

Fonte: Autoria própria

Conforme pode-se observar na Tabela 12, esta heurística traz para algumas instâncias, resultados semelhantes aos resultados obtidos na primeira heurística construtiva apresentada.

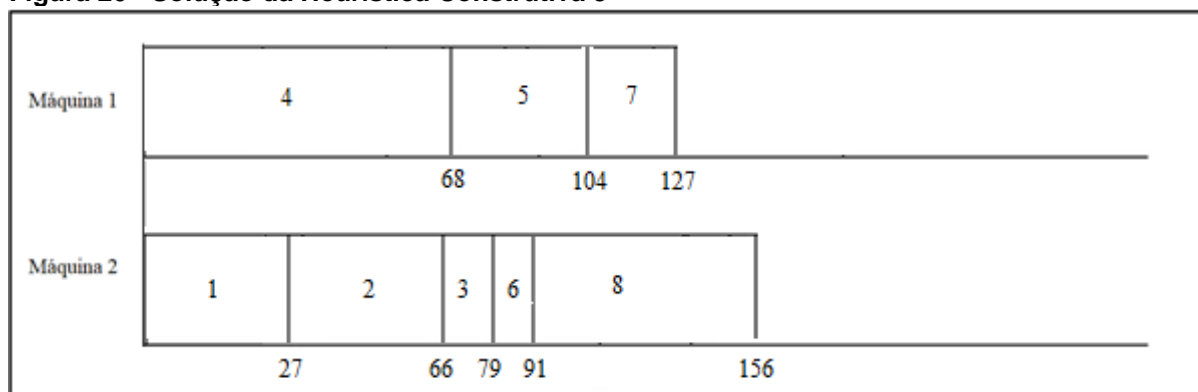
Isto se dá pela Heurística Construtiva 3 considerar inicialmente a mesma regra de alocação e como as tarefas são ordenadas que a primeira. O motivo das diferenças nos resultados se dá pelo fato de se considerar a ocupação das máquinas no momento de alocação das tarefas.

Em três instâncias houve melhorias nos resultados – a instância número 3, a instância número 6 e a instância número 7. Entretanto apenas o resultado da instância 6 é melhor também do que o resultado da Heurística Construtiva 2.

As instâncias com maiores números de tarefas – 8, 9 e 10 foram as que obtiveram os piores resultados quando se compara à Heurística 1 e também à Heurística 2.

Um exemplo da aplicação da heurística é dado na Figura 20, que traz a solução da heurística para a instância 3.

Figura 20 - Solução da Heurística Construtiva 3



Fonte: Autoria própria

Pode-se observar com base na Figura 20 que o menor tempo de processamento não foi o único critério de alocação na máquina, quando se olha para a tarefa 6 que possui menor tempo de processamento na máquina 1. Entretanto foi alocada à máquina 2 pela máquina 1 estar ocupada processando a tarefa 5.

Ao aplicar a heurística de melhoria na Heurística 3 obtiveram-se os resultados contidos na Tabela 13.

Tabela 13 - Resultados da Heurística Construtiva 3 após melhoria

Instância	Resultado	Resultado após melhoria	Melhoria (%)	Tempo de resolução (s)
1	124	124	0,00	2,42
2	194	91	53,09	2,26
3	173	103	40,46	2,19
4	417	293	29,74	2,42
5	495	487	1,62	2,65
6	903	511	43,41	2,62
7	989	553	44,08	2,29
8	1088	647	40,53	2,82
9	4556	3077	32,46	6,78
10	24216	9423	61,09	9,42

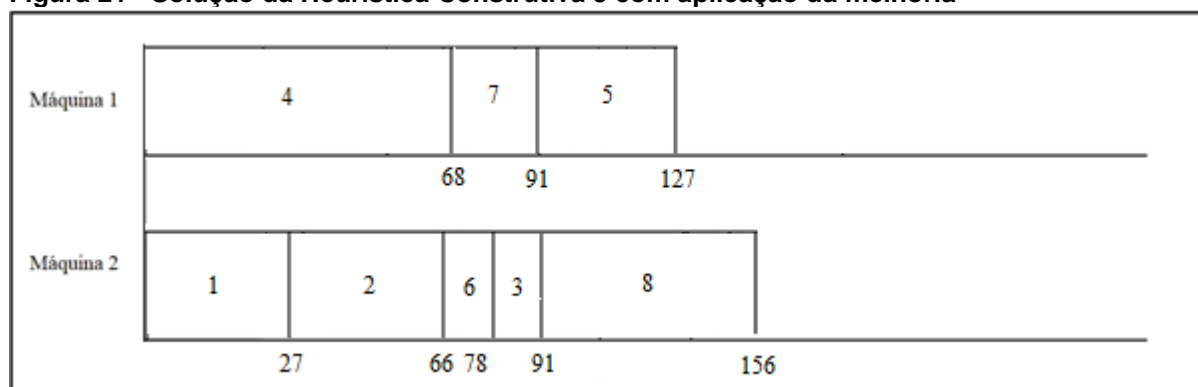
Fonte: Autoria própria

Pode-se observar que para esta heurística também houve melhoria na maioria das instâncias testadas, totalizando 90% dos casos.

Os tempos de processamento também aumentaram consideravelmente no caso da obtenção das melhorias com as trocas, entretanto continuam sendo muito mais aplicáveis do que os da resolução exata.

A Figura 21 traz o gráfico de Gantt para a solução obtida com a aplicação da melhoria para a mesma instância observada na Figura 20.

Figura 21 - Solução da Heurística Construtiva 3 com aplicação da melhoria



Fonte: Autoria própria

Na máquina 1, observa-se que as tarefas 7 e 5 trocaram de posição e foram as únicas na máquina a sofrer ação do método de melhoria. Na máquina 2, a única troca observada foi entre as tarefas 6 e 3. Estas duas trocas realizadas nas duas

máquinas totalizaram uma redução de 70 unidades de soma de adiantamentos e atrasos na solução final para a instância.

5.5 ANÁLISES COMPARATIVAS

A Tabela 14 traz todos os resultados para o modelo matemático e para as heurísticas construtivas.

Tabela 14 - Comparação entre os resultados obtidos

Modelo matemático	Heurística Construtiva 1	Heurística Construtiva 2	Heurística Construtiva 3
72	124	80	124
57	194	139	194
66	206	159	173
[84]	417	363	417
[124]	495	555	495
[805]	1124	1063	903
[2670]	1227	757	989
[4083]	997	556	1088
-	4276	2494	4556
-	18505	8619	24216

Fonte: Autoria própria

Pode-se observar que dentre todas as heurísticas analisadas neste trabalho, a Heurística Construtiva 2 é a que tem os melhores resultados para o problema de estudado, seguida de um empate entre as Heurísticas Construtivas 1 e 3.

Entretanto, observa-se ainda, que para os resultados exatos obtidos, apenas em um caso o valor da Heurística Construtiva 2 se aproximou. As outras heurísticas não se aproximaram dos valores exatos em nenhum caso.

Mesmo que esta aproximação não tenha ocorrido para todos os casos, ainda é válido confiar nas heurísticas, principalmente a Construtiva 2, pois para maiores números de tarefas, estas ganharam do modelo matemático com relação ao valor da função objetivo.

Após aplicar a Heurística de Melhoria, observou-se que houve melhorias nas heurísticas construtivas conforme mostra a Tabela 15.

Tabela 15 - Melhorias por heurística

Heurística	Instâncias melhoradas (%)
1	90
2	70
3	70

Fonte: Autoria Própria

As heurísticas que mais obtiveram melhorias com a aplicação da melhoria proposta foram a 1 e a 3, as duas heurísticas com os piores resultados para a ordenação inicial.

Apesar de a Heurística 2 ter obtido um resultado menor de instâncias melhoradas, o seu resultado ainda é satisfatório, pois representa 70% das instâncias melhoradas com a aplicação da heurística de melhoria.

A Tabela 16 traz os resultados de todas as heurísticas após a aplicação da melhoria e os compara com os resultados do modelo matemático.

Tabela 16 - Comparação entre os resultados após aplicação da melhoria

Modelo matemático	Heurística Construtiva 1 com melhoria	Heurística Construtiva 2 com melhoria	Heurística Construtiva 3 com melhoria
72	124	80	124
57	91	139	91
66	126	126	103
[84]	293	261	293
[124]	487	555	487
[805]	678	659	511
[2670]	732	721	553
[4083]	564	517	647
-	2855	2316	3077
-	11116	6922	9423

Fonte: Autoria própria

Após a aplicação da melhoria, a Heurística Construtiva 3 obteve maior notoriedade, obtendo cinco melhores resultados que a Heurística 2 e quatro melhores resultados que a Heurística Construtiva 1.

Observa-se ainda, que na maioria, as melhorias entre as heurísticas 1 e 3 acontecem juntas. Isto se dá pelo fato de a Heurística 3 ser proveniente da Heurística 1.

A comparação dos resultados das heurísticas após a melhoria com o modelo matemático agora pode ser mais explorada, visto que a melhoria fez com que o valor dos resultados se aproximasse ainda mais dos valores exatos e dos limitantes superiores obtidos.

A Heurística de Melhoria proporcionou resultados melhores do que apenas as Heurísticas Construtivas para o problema em um tempo de resposta menor do que o modelo exato poderia propor.

6 CONCLUSÕES

O problema de minimização da soma de adiantamentos e atrasos pode ser considerado como um *trade off*, ou seja, deseja-se minimizar objetivos opostos. Neste caso, minimizando o atraso, pode-se obter o avanço dos adiantamentos e o mesmo funciona em situações opostas.

Com base nos resultados obtidos, pode-se considerar que apesar de ainda se observar valores diferentes dos ótimos para as instâncias em que foi possível obtê-los pela programação matemática, a heurística de melhoria desenvolvida, quando combinada com as heurísticas construtivas, é eficiente para a alocação e ordenação de tarefas em máquinas paralelas não relacionadas visando minimizar a soma de adiantamentos e atrasos.

Os melhores pontos de partida para aplicar a heurística de melhoria foram as soluções iniciais fornecidas por uma ordenação nas máquinas utilizando a regra EDD, que é o caso da Heurística Construtiva 2.

Pode-se observar também a partir da Heurística Construtiva 3 que, com estas instâncias, olhar para o nível de ocupação das máquinas no momento de alocação não trouxe resultados tão bons quanto utilizar as regras da Heurística Construtiva 2.

Isto pode ser observado na prática em ambientes produtivos em que são consideradas máquinas com tempos de processamento muito diferentes, como por exemplo um torno mecânico operado por uma pessoa comparado a um CNC programado para desempenhar as tarefas.

A heurística de melhoria desenvolvida ainda pode ser melhorada quando se considera a possibilidade de explorar as vizinhanças de soluções entre máquinas diferentes e não apenas entre tarefas alocadas a uma mesma máquina.

Utilizar esta observação pode aumentar a gama de soluções possíveis e trocas a serem feitas para melhorar a solução final.

Apesar de os resultados obtidos com as heurísticas serem diferentes dos resultados ótimos e em alguns casos apresentarem grande diferença, as heurísticas podem ser consideradas positivas, pois faz-se necessário o seu uso para obter os resultados para maiores números de tarefas pelo tempo de resposta do programa ser muito menor do que os da solução ótima.

Com isso pode-se afirmar que os objetivos deste trabalho foram alcançados, ainda que os métodos heurísticos não tenham obtido as soluções ótimas.

Pode-se ainda aplicar as heurísticas utilizadas a problemas reais em trabalhos futuros.

REFERÊNCIAS

- ABUABARA, A.; MORABITO, R. Modelos de programação inteira mista para o planejamento do corte unidimensional de tubos metálicos na indústria aeronáutica agrícola. **Gestão & Produção**, [s.l.], v. 15, n. 3, p.605-617, dez. 2008. FapUNIFESP (SciELO). <http://dx.doi.org/10.1590/s0104-530x2008000300014>.
- ARENALES, M. et al. **Pesquisa Operacional: para cursos de engenharia**. Rio de Janeiro: Elsevier, 2011. 542 p.
- BERTRAND, J. Will M.; FRANSOO, Jan C. Operations management research methodologies using quantitative modeling. **International Journal Of Operations & Production Management**, [s.l.], v. 22, n. 2, p.241-264, fev. 2002. Emerald. <http://dx.doi.org/10.1108/01443570210414338>.
- BUZZO, W. R.; MOCCELLIN, J. V. 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). <http://dx.doi.org/10.1590/s0104-530x2000000300012>.
- CELINI JÚNIOR, M. R.; SILVA, M. C. R. **Projeto e implementação de heurísticas para um problema de programação da produção**. Ponta Grossa, 2016.
- COLIN, E. C.; SHIMIZU, T. Algoritmo de programação de máquinas individuais com penalidades distintas de adiantamento e atraso. **Pesquisa Operacional**, [s.l.], v. 20, n. 1, p.19-30, jun. 2000. FapUNIFESP (SciELO). <http://dx.doi.org/10.1590/s0101-74382000000100004>.
- CORMEN, T. H. et al. **Algoritmos: teoria e prática**. 3. ed. Rio de Janeiro: Elsevier, 2015. 944 p.
- CUNHA, C. B. **Heurísticas de melhoria**. São Paulo: Imagem, 2015. 15 slides, color.
- DING, J. et al. A hybrid evolutionary approach for the single-machine total weighted tardiness problem. **Computers & Industrial Engineering**, [s.l.], v. 108, p.70-80, jun. 2017. Elsevier BV. <http://dx.doi.org/10.1016/j.cie.2017.04.006>.

DOMBROWSKI, U.; WAGNER, T. Mental strain as field of action in the 4th industrial revolution. **Procedia Cirp**, [s.l.], v. 17, p.100-105, 2014. Elsevier BV. <http://dx.doi.org/10.1016/j.procir.2014.01.077>.

GAREY, M. R.; JOHNSON, D. S.; SETHI, R. 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). <http://dx.doi.org/10.1287/moor.1.2.117>.

GUPTA, J. N. D.; HO, J. C.; WEBSTER, S. Bicriteria optimisation of the makespan and mean flowtime on two identical parallel machines. **Journal Of The Operational Research Society**, [s.l.], v. 51, n. 11, p.1330-1339, nov. 2000. Springer Nature. <http://dx.doi.org/10.1057/palgrave.jors.2601016>.

HAMZADAYI, A.; YILDIZ, G. Modeling and solving static m identical parallel machines scheduling problem with a common server and sequence dependent setup times. **Computers & Industrial Engineering**, [s.l.], v. 106, p.287-298, abr. 2017. Elsevier BV. <http://dx.doi.org/10.1016/j.cie.2017.02.013>.

HILLIER, F. S.; LIEBERMAN, G. J. **Introdução à pesquisa operacional**. 8. ed. São Paulo: Amgh, 2010. 852 p.

JOAQUIM JUNIOR, C. F. et al. Aplicação de pesquisa operacional no planejamento de produção de bens de capital sob encomenda: um estudo de caso. **Tékhnē e Lógos**, Botucatu, v. 1, n. 3, p.96-112, jun. 2010.

Kahn, K. et al. Perspective: Establishing an NPD best practices framework. **Journal of Product Innovation Management**, USA, v. 23, n. 2, p.106-116, 2006.

LEE, W.; WANG, J. A three-agent scheduling problem for minimizing the makespan on a single machine. **Computers & Industrial Engineering**, [s.l.], v. 106, p.147-160, abr. 2017. Elsevier BV. <http://dx.doi.org/10.1016/j.cie.2017.02.001>.

LIN, J.; ZHANG, S. An effective hybrid biogeography-based optimization algorithm for the distributed assembly permutation flow-shop scheduling problem. **Computers & Industrial Engineering**, [s.l.], v. 97, p.128-136, jul. 2016. Elsevier BV. <http://dx.doi.org/10.1016/j.cie.2016.05.005>.

MACHLINE, C. Compras, estoques e inflação. **Revista de Administração de Empresas**, [s.l.], v. 21, n. 2, p.7-15, jun. 1981. FapUNIFESP (SciELO). <http://dx.doi.org/10.1590/s0034-75901981000200001>.

MOREIRA, J. M. M. A. P.; RODRIGUEZ, L. C. E. A incorporação de corredores de conectividade no manejo de florestas industriais utilizando a heurística da razão R. **Revista de Economia e Sociologia Rural**, [s.l.], v. 48, n. 2, p.255-282, jun. 2010. FapUNIFESP (SciELO). <http://dx.doi.org/10.1590/s0103-20032010000200001>.

MELO, E. L. **Instâncias**. Ponta Grossa: UTFPR, 2016. Notas de aula.

MÜLLER, F. M.; DIAS, O. B.; ARAÚJO, O. C. B. Algoritmo para o problema de seqüenciamento em máquinas paralelas não-relacionadas. **Production**, [s.l.], v. 12, n. 2, p.6-17, 2002. FapUNIFESP (SciELO). <http://dx.doi.org/10.1590/s0103-65132002000200002>.

OLIVEIRA, L. K.; MORABITO, R. Métodos exatos baseados em relaxações lagrangiana e surrogate para o problema de carregamento de paletes do produtor. **Pesquisa Operacional**, [s.l.], v. 26, n. 2, p.403-432, ago. 2006. FapUNIFESP (SciELO). <http://dx.doi.org/10.1590/s0101-74382006000200010>.

OLIVEIRA, R. A. M. **Importância do planejamento e controle da produção (PCP) dentro de uma organização**: O caso de uma empresa têxtil. 2014. 30 f. TCC (Graduação) - Curso de Administração, Faculdade Cenecista de Capivari - Facecap, Capivari, 2014.

PINEDO, M. L. **Scheduling**: Theory, algorithms, and systems. 3. ed. New York: Springer, 2008. 671 p.

REYNA, Y. C. F. et al. Influencia de los parámetros principales de un algoritmo genético para el flow shop scheduling. **Revista Cubana de Ciencias Informáticas**, La Habana, v. 8, n. 1, p.99-111, jan. 2014.

RONCONI, D. P.; KAWAMURA, M. S. The single machine earliness and tardiness scheduling problem: lower bounds and a branch-and-bound algorithm. **Computational & Applied Mathematics**, [s.l.], v. 29, n. 2, p.107-124, jun. 2010. Brazilian Society for Computational and Applied Mathematics (SBMAC). <http://dx.doi.org/10.1590/s1807-03022010000200002>

SANTOS, H. C. M.; FRANÇA, P. M. Meta-heurística para programação da produção com tempos de preparação dependentes da seqüência. **Gestão & Produção**, [s.l.], v. 2, n. 3, p.228-243, dez. 1995. FapUNIFESP (SciELO). <http://dx.doi.org/10.1590/s0104-530x1995000300001>.

SANTOS, V. L. A. **Sequenciamento de tarefas em máquinas paralelas com desgastes dependentes da seqüência**: Resolução heurística. 2016. 79 f. Dissertação (Mestrado) - Curso de Ciência da Computação, Universidade Federal de Viçosa, Viçosa, 2016.

SBICCA, A. Heurísticas no estudo das decisões econômicas: contribuições de Herbert Simon, Daniel Kahneman e Amos Tversky. **Estudos Econômicos (São Paulo)**, [s.l.], v. 44, n. 3, p.579-603, set. 2014. FapUNIFESP (SciELO). <http://dx.doi.org/10.1590/s0101-41612014000300006>.

SELEN, W. J.; HOTT, D. D. A Mixed-Integer Goal-Programming Formulation of the Standard Flow-Shop Scheduling Problem. **Journal Of The Operational Research Society**, [s.l.], v. 37, n. 12, p.1121-1128, dez. 1986. Springer Nature. <http://dx.doi.org/10.1057/jors.1986.197>.

SENTHILKUMAR, P.; NARAYANAN, S. GA Based heuristic to minimize makespan in single machine scheduling problem with uniform parallel machines. **Intelligent Information Management**, [s.l.], v. 03, n. 05, p.204-214, 2011. Scientific Research Publishing, Inc. <http://dx.doi.org/10.4236/iim.2011.35025>.

SILVA, F. M.; FERNANDES, F. C. F. Proposta de um sistema de controle da produção para fabricantes de calçados que operam sob encomenda. **Gestão & Produção**, São Carlos, v. 15, n. 3, p.523-538, dez. 2008.

SILVA, G. C.; OCHI, L. S.; MARTINS, S. L. Proposta e avaliação de heurísticas grasp para o problema da diversidade máxima. **Pesquisa Operacional**, [s.l.], v. 26, n. 2, p.321-360, ago. 2006. FapUNIFESP (SciELO). <http://dx.doi.org/10.1590/s0101-74382006000200007>.

SILVA, R. J.; MORABITO, R. Otimização da programação de cargas de forno em uma fábrica de fundição em aço-inox. **Gestão & Produção**, [s.l.], v. 11, n. 1, p.135-151, abr. 2004. FapUNIFESP (SciELO). <http://dx.doi.org/10.1590/s0104-530x2004000100012>.

SOUZA, M. D.; BORGONHONI, P. A consolidação dos três níveis de planejamento e controle de produção. **Caderno de Administração**, Maringá, v. 15, n. 2, p.19-28, jul. 2007.

SLACK, N.; CHAMBERS, S.; JOHNSTON, R. **Administração da Produção**. 2. ed. São Paulo: Atlas, 2002. 735 p.

SOLER, E. M. et al. Otimização dos custos de energia elétrica na programação da captação, armazenamento e distribuição de água. **Production**, [s.l.], v. 26, n. 2, p.385-401, jun. 2016. FapUNIFESP (SciELO). <http://dx.doi.org/10.1590/0103-6513.146113>.

SOSA, N. G. M.; GALVÃO, R. D.; GANDELMAN, D. A. **Algoritmo de busca dispersa aplicado ao problema clássico de roteamento de veículos**. Pesquisa Operacional, [s.l.], v. 27, n. 2, p.293-310, ago. 2007. FapUNIFESP (SciELO). <http://dx.doi.org/10.1590/s0101-74382007000200006>.

STEBEL, Sérgio Leandro. **Técnicas de otimização aplicadas em problemas de scheduling dos recursos de estocagem**. 2006. 132 f. Tese (Doutorado) - Curso de Ciências, Universidade Tecnológica Federal do Paraná, Curitiba, 2006.

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

TURRIONI, J. B.; MELLO, C. H. P. Metodologia de pesquisa em engenharia de produção. 1st ed. Itajubá: UNIFEI, 2012.

VIEIRA, C. E. C. **Heurísticas para o problema das p-medianas conectadas**. 2006. 191 f. Tese (Doutorado) - Curso de Informática, Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2006.

WANG, S.; YU, J. An effective heuristic for flexible job-shop scheduling problem with maintenance activities. **Computers & Industrial Engineering**, [s.l.], v. 59, n. 3, p.436-447, out. 2010. Elsevier BV. <http://dx.doi.org/10.1016/j.cie.2010.05.016>.