



**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE DESENVOLVIMENTO SUSTENTÁVEL DO SEMIÁRIDO
UNIDADE ACADÊMICA DE ENGENHARIA DE PRODUÇÃO
CURSO DE ENGENHARIA DE PRODUÇÃO**

MARIANE EMANUELLE PESSOA SANTOS

**UMA ABORDAGEM HEURÍSTICA DE BUSCA LOCAL PARA PROBLEMAS DE
SEQUENCIAMENTO DA PRODUÇÃO EM MÁQUINAS PARALELAS COM
RELEASE DATES E TEMPOS DE *SETUP***

**SUMÉ - PB
2023**

MARIANE EMANUELLE PESSOA SANTOS

UMA ABORDAGEM HEURÍSTICA DE BUSCA LOCAL PARA PROBLEMAS DE SEQUENCIAMENTO DA PRODUÇÃO EM MÁQUINAS PARALELAS COM *RELEASE DATES* E TEMPOS DE *SETUP*

Monografia apresentada ao Curso Superior de Engenharia de Produção do Centro de Desenvolvimento Sustentável do Semiárido da Universidade Federal de Campina Grande, como requisito parcial para obtenção do título de Bacharel em Engenharia de Produção.

Orientador: Professor Dr. Yuri Laio Teixeira Veras Silva.

**SUMÉ - PB
2023**



S237a Santos, Mariane Emanuelle Pessoa.

Uma abordagem heurística de busca local para problemas de sequenciamento da produção em máquinas paralelas com realease dates e tempos de setup. / Mariane Emanuelle Pessoa Santos. - 2023.

95 f.

Orientador: Professor Dr. Yuri Laio Teixeira Veras Silva.

Monografia - Universidade Federal de Campina Grande; Centro de Desenvolvimento Sustentável do Semiárido; Curso de Engenharia de Produção.

1. Sequenciamento da produção. 2. Máquinas paralelas idênticas - sequenciamento. 3. Abordagem de otimização. 4. Heurística de busca local. I. Silva, Yuri Laio Teixeira Veras. II. Título.

CDU: 658.5(043.1)

Elaboração da Ficha Catalográfica:

Johnny Rodrigues Barbosa
Bibliotecário-Documentalista
CRB-15/626

MARIANE EMANUELLE PESSOA SANTOS

UMA ABORDAGEM HEURÍSTICA DE BUSCA LOCAL PARA PROBLEMAS DE SEQUENCIAMENTO DA PRODUÇÃO EM MÁQUINAS PARALELAS COM *RELEASE DATES* E TEMPOS DE *SETUP*

Monografia apresentada ao Curso Superior de Engenharia de Produção do Centro de Desenvolvimento Sustentável do Semiárido da Universidade Federal de Campina Grande, como requisito parcial para obtenção do título de Bacharel em Engenharia de Produção.

BANCA EXAMINADORA:

**Professor Dr. Yuri Laio Teixeira Veras Silva.
Orientador - UAEP/CDSA/UFCG**

**Professora Dra. Maria Creuza Borges de Araújo.
Examinadora Interna - UAEP/CDSA/UFCG**

**Professor Me. Pablo Ramom Matias de Andrade.
Examinador Externo - PPGEM/UFPB**

Trabalho Aprovado em: 28 de novembro de 2023.

SUMÉ - PB

*Dedico este trabalho a minha família, por todo amor
e incentivo na realização dos meus sonhos.*

AGRADECIMENTOS

Sou grata, sobretudo, ao meu Deus, que permaneceu sempre ao meu lado não me deixando desistir, abençoando a minha vida com pessoas maravilhosas que me fizeram enxergar o mundo com outras perspectivas e que tanto me apoiaram.

Aos meus amados pais, Risoneide e Edvaldo, que representam o alicerce da minha vida. Suas lições e orientações moldaram a pessoa que sou hoje, serei eternamente grata pelos incentivos que me motivam diariamente a buscar tudo aquilo que almejo. Aos meus irmãos, Bruno e Pedro, pelo amor e companhia ao longo da minha jornada.

Aos meus queridos amigos que a faculdade me presenteou e que levarei para a vida, em especial Júnior, Andressa e Maítala por viverem juntos comigo todos os desafios da graduação e se fazerem presentes nos momentos mais especiais, vocês foram essenciais nessa caminhada, obrigada pelo apoio e pelas risadas compartilhadas. À minha amiga Laís, que o Cariritec me apresentou, tenho grande carinho e orgulho por tudo que vem conquistando.

Ao meu namorado e amigo, Mateus, que ao longo desta fase sempre me incentivou e me apoiou, sendo meu ponto de paz e conforto. Sou grata, por todo amor, carinho e dedicação, você trouxe alegria e leveza aos meus dias.

Ao meu professor e orientador Yuri, por tornar as aulas tão interessantes e cativantes. Agradeço pela paciência, pelos incentivos, pela disponibilidade e por toda contribuição, seus ensinamentos foram essenciais para a construção e conclusão deste trabalho.

Aos docentes do curso de Engenharia de Produção da UFCG – CDSA, que compartilharam seus conhecimentos e colaboraram para a minha formação acadêmica.

RESUMO

A crescente concorrência entre as empresas impulsiona a busca contínua por estratégias que priorizem o atendimento das expectativas dos clientes, fortalecendo a competitividade no mercado. O sucesso dessas estratégias está relacionado diretamente a implementação de um sequenciamento da produção eficiente, o qual desempenha um papel essencial na redução dos tempos de entrega e na otimização da utilização dos recursos, garantindo assim, não apenas a satisfação dos clientes, mas também o aumento da competitividade nas organizações. Diante disso, o objetivo deste trabalho consiste em propor uma abordagem heurística para otimizar o sequenciamento da produção no setor têxtil de uma empresa de tecelagem, com a finalidade de minimizar os atrasos totais. O cenário em estudo considera um ambiente composto por máquinas paralelas idênticas, com *release dates* distintos e tempos de *setup* dependentes da sequência de produção. A heurística desenvolvida emprega estratégias de busca local e perturbação, visando diversificar o espaço de busca e encontrar soluções de boa qualidade. O algoritmo foi testado em oito instâncias, cujas informações abrangem duas linhas de produção distintas em um período de quatro meses. Os resultados da aplicação da heurística nas diversas instâncias, demonstraram a capacidade dessa abordagem em produzir soluções eficientes, alcançando uma redução média dos atrasos totais de 27,4%, ao mesmo tempo em que mantém um custo computacional baixo, garantindo soluções de boa qualidade em menos de dois minutos.

Palavras-chave: Sequenciamento da produção. Máquinas paralelas idênticas. Abordagem de otimização. Heurística de busca local.

ABSTRACT

The growing competition among companies drives a continuous search for strategies that prioritize meeting customer expectations, thereby strengthening competitiveness in the market. The success of these strategies is directly related to the implementation of an efficient scheduling, which plays an essential role in reducing delivery times and optimizing resource utilization. This ensures not only customer satisfaction but also an increase in competitiveness within organizations. In light of this, the objective of this work is to propose a heuristic approach to optimize scheduling in the textile sector of a weaving company, aiming to minimize total the total tardiness. The studied scenario considers an environment composed of identical parallel machines with distinct release dates and sequence-dependent setup times. The developed heuristic employs local search and perturbation strategies to diversify the search space and find high-quality solutions. The algorithm was tested on eight instances, covering two different production lines over a four-month period. The results of applying the heuristic to various instances demonstrated the capability of this approach to generate efficient solutions, achieving an average reduction of total tardiness by 27.4%, while maintaining a low computational cost, ensuring high-quality solutions in less than two minutes.

Keywords: Scheduling. Identical parallel machines. Optimization approach. Local Search heuristic.

LISTA DE FIGURAS

Figura 1 – Estrutura do trabalho.....	17
Figura 2 – Horizontes temporais e suas respectivas atividades desenvolvidas pelo PCP	20
Figura 3 – Sequenciamento em máquinas paralelas idênticas.....	26
Figura 4 – Sequenciamento arbitrário das tarefas	29
Figura 5 – Classificação de heurísticas e metaheurísticas.....	30
Figura 6 – Gráfico representativo do funcionamento da busca local.....	32
Figura 7 – Etapas de heurísticas baseadas em busca local.....	40
Figura 8 – Caracterização da pesquisa científica	42
Figura 9 – Etapas de desenvolvimento do estudo	44
Figura 10 – Esquema ilustrativo das linhas de produção A e B	47
Figura 11 – Gráfico de Gantt para demonstrar o <i>release date</i>	49
Figura 12 – Exemplo de sequenciamento descrito na Tabela 2 e Quadro 3.....	53
Figura 13 – Etapas da abordagem heurística.....	55
Figura 14 – Representação de uma solução arbitrária.....	58
Figura 15 – Estratégia de <i>swap</i> intra máquinas.....	59
Figura 16 – Estratégia de <i>swap</i> inter máquinas.....	59
Figura 17 – Estratégia de <i>insertion</i>	62
Figura 18 – Comportamento da perturbação	67
Figura 19 – Resultados experimentais para a instância A1	68
Figura 20 – Resultados experimentais para a instância A2	70
Figura 21 – Resultados experimentais para a instância A3	71
Figura 22 – Resultados experimentais para a instância A4	73
Figura 23 – Resultados experimentais para a instância B1	74
Figura 24 – Resultados experimentais para a instância B2	76
Figura 25 – Resultados experimentais para a instância B3	77
Figura 26 – Resultados experimentais para a instância B4	79
Figura 27 – Representação do melhor sequenciamento da instância B4.....	82
Figura 28 – Comparação entre o atraso mínimo obtido com a heurística e o atraso obtido com o método adotado pela empresa	82
Figura 29 – Comparação entre o atraso máximo obtido com a heurística e o atraso obtido com o método empregado pela empresa	82

LISTA DE QUADROS

Quadro 1 – Métodos de solução para problemas de redução do atraso total	36
Quadro 2 – Variações de objetivos em problemas de sequenciamento de máquinas paralelas idênticas.....	38
Quadro 3 – Exemplo de matriz de tempos de <i>setup</i>	48
Quadro 4 – Parâmetros da heurística de busca local	63

LISTA DE TABELAS

Tabela 1 – Exemplo de instância para seis tarefas e duas máquinas	28
Tabela 2 – Exemplo de instância para seis tarefas.....	46
Tabela 3 – Resultados das combinações para calibração dos parâmetros	64
Tabela 4 – Dados das instâncias	65
Tabela 5 – Instâncias do problema.....	66
Tabela 6 – Resultados experimentais para a instância A1	66
Tabela 7 – Resultados experimentais para a instância A2	68
Tabela 8 – Resultados experimentais para a instância A3	69
Tabela 9 – Resultados experimentais para a instância A4	71
Tabela 10 – Resultados experimentais para a instância B1	72
Tabela 11 – Resultados experimentais para a instância B2	74
Tabela 12 – Resultados experimentais para a instância B3	75
Tabela 13 – Resultados experimentais para a instância B4	77
Tabela 14 – Resumo das oito instâncias	78
Tabela 15 – Resultados da simulação do método adotado pela empresa.....	80
Tabela 16 – Comparativo dos resultados obtidos pela heurística proposta e o método adotado pela empresa	81

LISTA DE ALGORITMOS

Algoritmo 1 – Avaliação da solução	56
Algoritmo 2 – Procedimento de busca local	60
Algoritmo 3 – Mecanismo de perturbação	61

LISTA DE SIGLAS

ABS	Acrilonitrila-Butadieno-Estireno
ACO	Ant Colony Optimization
ATCS	Adaptive Tabu Search
B&B	Branch-and-Bound
BRKGA	Biased Random-Ke Genetic Algorithm
BS _{OU}	Best Solution
EDD	Earliest Due Date
ERP	Planejamento dos Recursos da Empresa
GA	Genetic Algorithms
GRASP	Greedy Randomized Adaptive Search Procedure
HC	Heurística Construtiva
IG	Iterated Greedy
ILS	Iterated Local Search
LS	Local Search
MIGP	Programação Inteira Mista com Metas
MILP	Programação Linear Inteira Mista
PCP	Planejamento e Controle da Produção
PMP	Plano Mestre de Produção
PSO	Particle Swarm Optimization
RIG	Random Iterated Greedy
RSA	Restricted Simulated Annealing
SA	Simulated Annealing
SBC1	Shortest-Batch-Completion Time Heuristic 1
SLA	Smallest Laxity-Value
TPI	Total Processing Time
TS	Tabu Search
VBA	Visual Basic for Applications
VLNS	Very Large-scale Neighborhood Search
VNS	Variable Neighborhood Search

SUMÁRIO

1	INTRODUÇÃO.....	13
1.1	JUSTIFICATIVA.....	14
1.2	OBJETIVOS.....	16
1.2.1	Objetivos gerais.....	16
1.2.2	Objetivos específicos	16
1.3	ESTRUTURA DO TRABALHO.....	16
2	REVISÃO DA LITERATURA.....	19
2.1	PLANEJAMENTO E CONTROLE DA PRODUÇÃO.....	19
2.2	SEQUENCIAMENTO DA PRODUÇÃO.....	21
2.3	SEQUENCIAMENTO EM MÁQUINAS PARALELAS IDÊNTICAS.....	25
2.4	HEURÍSTICAS DE OTIMIZAÇÃO.....	29
2.4.1	Heurísticas de busca local.....	31
2.5	TRABALHOS RELACIONADOS.....	33
2.5.1	Trabalhos relacionados com enfoque em outros objetivos.....	37
3	METODOLOGIA.....	40
3.1	CARACTERIZAÇÃO DA PESQUISA.....	40
3.2	ETAPAS DA PESQUISA.....	41
4	CARACTERIZAÇÃO DO PROBLEMA.....	44
4.1	DESCRIÇÃO DA EMPRESA.....	44
4.2	DESCRIÇÃO DO PROBLEMA.....	45
4.2.1	Processamento das tarefas.....	45
4.2.2	Características das máquinas.....	47
4.3	MODELAGEM MATEMÁTICA DO PROBLEMA.....	49
5	RESULTADOS.....	52
5.1	ABORDAGEM HEURÍSTICA PROPOSTA.....	52
5.1.1	Representação da solução.....	54
5.1.2	Geração da solução inicial.....	55
5.1.3	Avaliação da solução.....	55
5.1.4	Procedimento de busca local.....	58
5.1.5	Mecanismo de perturbação.....	61
5.1.6	Calibração dos parâmetros.....	62
5.2	RESULTADOS COMPUTACIONAIS.....	65
5.2.1	Instâncias do problema.....	65
5.2.2	Resultados experimentais para as instâncias da linha A.....	66
5.2.3	Resultados experimentais para as instâncias da linha B.....	72
5.2.4	Resumo do resultado das instâncias.....	78
5.2.5	Simulação do método aplicado pela empresa.....	80
5.2.6	Comparação entre a heurística e o método utilizado pela empresa.....	81
6	CONSIDERAÇÕES FINAIS.....	84
	REFERÊNCIAS.....	86

1 INTRODUÇÃO

Em um ambiente empresarial altamente competitivo, que requer constantemente uma maior produção em prazos mais curtos e com a utilização mínima de recursos, as empresas são motivadas a alcançar níveis excepcionais de desempenho para se manterem competitivas no mercado (Ribeiro, 2020). De acordo com Pereira (2022), a crescente pressão que as empresas enfrentam para aprimorar a qualidade de seus serviços e processos tem aumentado o foco nas estratégias de programação de produção dentro das indústrias, principalmente por serem capazes de elevar a eficiência produtiva e lucratividade das organizações.

Segundo Slack, Johnston e Chambers (2009), a programação da produção é considerada uma das atividades mais desafiadoras no contexto do planejamento de produção, devido à complexidade envolvida na integração dos diversos recursos disponíveis na linha de produção. Oliveira (2015) destaca que um planejamento inadequado durante essa etapa pode resultar em excesso de produtos acabados em estoque, atrasos nos pedidos de venda, potenciais desperdícios de recursos e outros problemas similares.

Conforme mencionado por Assis (2014), a programação das tarefas de produção engloba a tomada de decisões acerca de quando, onde e por quem cada tarefa deverá ser realizada, com o objetivo de garantir entregas pontuais, minimizando o tempo de espera dos clientes. A determinação dos prazos e a definição da sequência de execução das tarefas são de responsabilidade do processo de sequenciamento da produção, que considera principalmente a disponibilidade dos recursos na empresa e as restrições do sistema (Lopes, 2008).

De acordo com Faêda (2015), o principal objetivo do sequenciamento é realizar a alocação de um conjunto de n tarefas em um grupo de m máquinas, estabelecendo a ordem de processamento na qual as tarefas serão executadas em cada máquina, com o intuito de alcançar o critério de otimização definido pela empresa. Esses critérios estão principalmente associados à diminuição dos custos, à minimização dos atrasos totais e à redução dos tempos de conclusão dos pedidos.

O sequenciamento da produção é considerado um dos desafios mais significativos e amplamente explorados no âmbito da Pesquisa Operacional (Silva, 2019). As possibilidades de investigação e aplicação nesse campo são importantes tanto para organizações de manufatura como para as que prestam serviços, o que tem motivado os pesquisadores a explorar problemas relacionados ao sequenciamento a partir de diversas abordagens ao longo das últimas décadas (Fuchigami; Rangel, 2018).

Dentre essas abordagens, algumas levam em consideração a existência de máquinas operando simultaneamente na mesma função, sendo denominadas na literatura como máquinas paralelas idênticas. Segundo Müller, Dias e Araújo (2002), as máquinas paralelas são consideradas idênticas quando há um único conjunto de tempos de execução das tarefas e esses tempos permanecem constantes, independentemente da máquina à qual uma tarefa é designada.

A resolução desses problemas de acordo com Lopes (2022), pode ser feita a partir da utilização de métodos exatos, como Programação Linear, *Branch and Bound* ou por meio de métodos aproximados, como heurísticas e metaheurísticas. Em situações de grande complexidade, quando existe a possibilidade de uma infinidade de combinações, encontrar uma solução viável ou ótima em um período de tempo computacional aceitável utilizando métodos exatos, pode se tornar uma tarefa extremamente desafiadora, chegando até a ser impraticável. Por isso, utiliza-se nesses cenários, algoritmos heurísticos para auxiliar na resolução dos problemas (Kaveh; Khayatazad, 2012).

Assim, optou-se neste trabalho pela utilização de uma heurística baseada em estratégias de busca local, para otimização do sequenciamento da produção de uma empresa de tecelagem, que opera com um sistema de máquinas paralelas idênticas. A heurística desenvolvida tem como principal finalidade a minimização dos atrasos totais, levando em consideração as variáveis e restrições do problema, incluindo tempos de *setup* e *release dates*.

1.1 JUSTIFICATIVA

Um dos desafios mais significativos da gestão operacional nas empresas, seja no setor de manufatura ou de serviços, é o sequenciamento da produção para atender às múltiplas demandas dos clientes (Silva; Riveros; Bolívar, 2013). A implementação prática do sequenciamento apresenta, em geral, um elevado nível de complexidade devido a quantidade de variáveis e restrições a serem consideradas. Além disso, segundo Guerrini, Belhot e Azzolini Júnior (2014), o número de possibilidades de soluções de otimização cresce exponencialmente à medida que as ordens a serem programadas aumentam, dificultando mais ainda a resolução desses problemas.

Uma das configurações dentro da área de sequenciamento que possui alta complexidade de resolução diz respeito as máquinas paralelas idênticas, a qual foi caracterizada por Garey e Johnson (1979) como NP-difícil, considerando apenas um único objetivo, isto é, não se tem conhecimento da existência de algoritmos que possam encontrar uma solução ótima em tempo

polinomial. Dessa forma, ao incrementar outras variáveis como tempos de *setups* e *release dates*, o problema se torna de maior complexidade computacional.

No entanto, de acordo com Ebrahiminy, Fatemi Ghomi e Karimi (2013), a maior parte dos problemas de sequenciamento não consideram os tempos de *setup* como uma variável de influência, adicionando-a como parte do tempo de processamento. Apesar de torná-los mais simples, a prática não condiz com a realidade operacional das empresas, como é o caso da indústria têxtil, em que nas operações de tecelagem os tempos de *setup* dependem da sequência das atividades. A desconsideração dessa variável pode resultar não apenas em sequenciamentos ineficazes, mas também provocar atrasos significativos na produção.

Devido à complexidade desse problema, a utilização de métodos heurísticos acabam sendo uma escolha mais viável, apesar de não garantirem a solução ótima global, as heurísticas possibilitam encontrar soluções de boa qualidade em um tempo computacional razoável quando aplicadas a situações reais. Muitas abordagens heurísticas empregadas nesse tipo de problema são fundamentadas em buscas locais, que fazem uso de estratégias de vizinhança para explorar o espaço de soluções (Pereira, 2019).

Outro ponto a ser destacado, refere-se ao fato de que a área de Planejamento e Controle da Produção da empresa em estudo não dispõe de um mecanismo ou software específico que facilite a geração dos sequenciamentos. A determinação da sequência de produção é feita de forma empírica pelos programadores de produção que consideram como critério a data de entrega mais próxima. Realizar o sequenciamento dessa forma, torna o processo mais difícil, especialmente quando se considera todas as variáveis e restrições do problema, como *setups* e *release dates*.

Diante do exposto, a motivação para a escolha do tema deste trabalho surgiu dos desafios enfrentados durante a etapa de sequenciamento de tarefas por uma empresa do setor têxtil e da complexidade do problema. Além disso, a questão em foco, embora tenha ampla relevância prática, não foi completamente explorada na literatura, especialmente no que diz respeito a variáveis cruciais como *release dates* e tempos de *setup*. Como destacado, as heurísticas de busca local apresentam um bom desempenho na definição do sequenciamento da produção, principalmente quando são desenvolvidas considerando as características específicas do problema.

1.2 OBJETIVOS

Com foco na resolução dos problemas identificados ao decorrer do trabalho, estabeleceu-se os objetivos a serem alcançados neste estudo.

1.2.1 Objetivos gerais

Desenvolver uma abordagem heurística eficiente, baseada em busca local para a resolução do problema de sequenciamento de tarefas em máquinas paralelas idênticas, com tempos de *setup* dependentes da sequência e *release dates* em uma indústria têxtil, visando a minimização dos atrasos totais.

1.2.2 Objetivos específicos

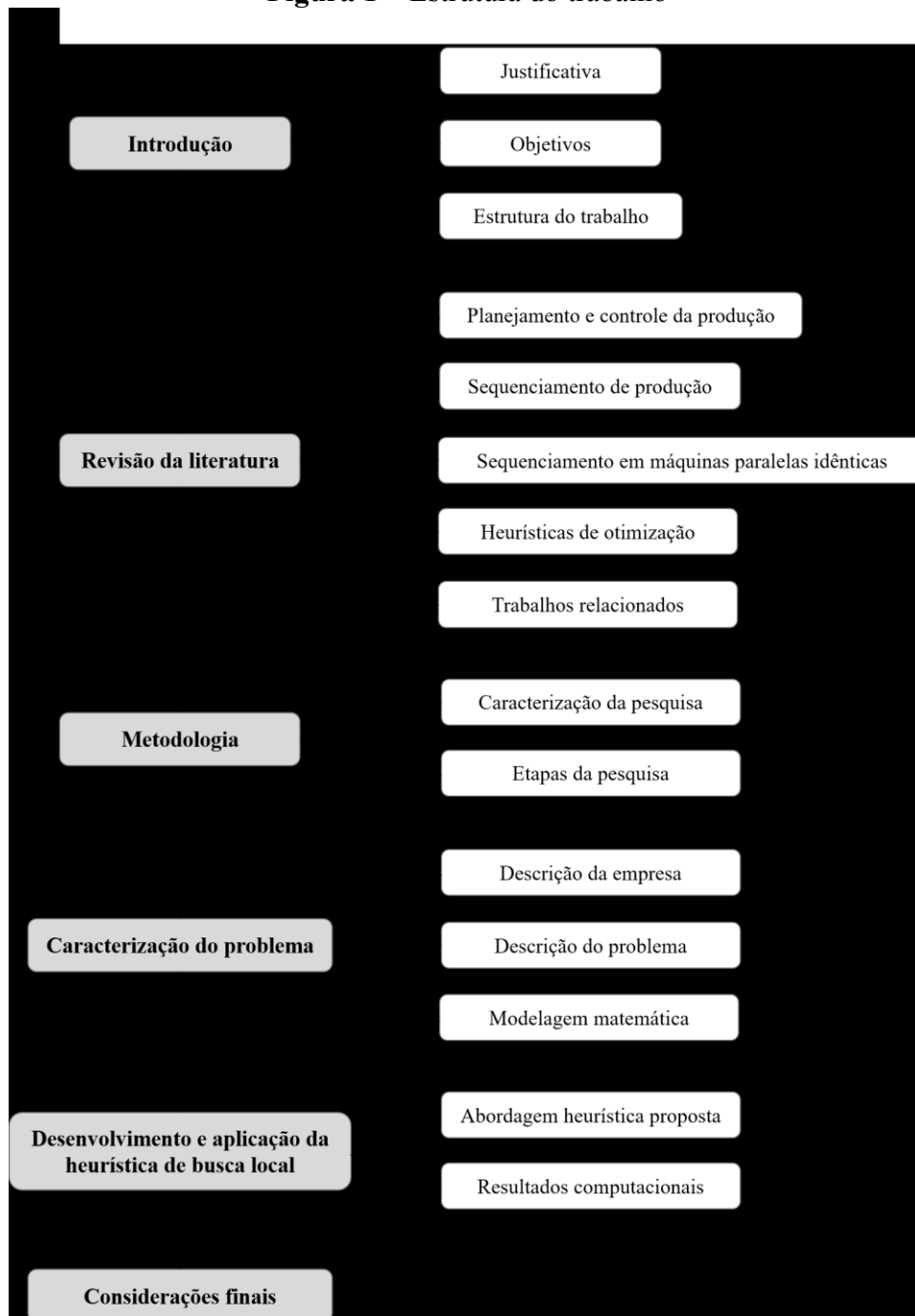
Para atingir o propósito geral mencionado, foram delineados os seguintes objetivos específicos:

- Estudar o problema prático enfrentado pela empresa, a fim de obter uma compreensão abrangente das variáveis, restrições e limitações existentes;
- Realizar uma revisão bibliográfica para identificar as metodologias empregadas na resolução dos problemas de sequenciamento em máquinas paralelas idênticas;
- Elaborar a modelagem matemática;
- Propor uma abordagem heurística de busca local;
- Analisar os resultados computacionais obtidos com a aplicação da heurística nas instâncias reais do problema;
- Comparar o desempenho da heurística com o método utilizado pela empresa.

1.3 ESTRUTURA DO TRABALHO

O presente trabalho está estruturado em seis seções, conforme ilustrado na Figura 1. As seções subsequentes deste estudo são brevemente descritas a seguir.

Figura 1 – Estrutura do trabalho



Fonte: Autoria própria (2023).

A segunda seção fornece uma breve revisão da literatura, abordando os principais conceitos de sequenciamento, especialmente os problemas envolvendo máquinas paralelas idênticas, além de apresentar os métodos de otimização utilizados para a resolução desses problemas. Na terceira seção, está disposta a metodologia adotada para realização do estudo.

Já na quarta seção está detalhada a descrição da empresa, bem como a descrição do problema, mostrando os principais fatores e variáveis que devem ser consideradas no estudo.

Ao final da seção, apresenta-se a modelagem matemática para o problema, por meio da Programação Linear Inteira Mista.

Na quinta seção são apresentadas em detalhes as etapas para o desenvolvimento da abordagem heurística, os resultados computacionais obtidos com a implementação da heurística nas oito instâncias e a análise comparativa entre a heurística desenvolvida e o método utilizado pela empresa. Por fim, na sexta seção consta as considerações finais sobre esta pesquisa.

2 REVISÃO DA LITERATURA

Nesta seção, são descritos os principais fundamentos para o embasamento teórico da pesquisa relacionados ao papel do Planejamento e Controle da Produção nas organizações, aos conceitos e fatores que caracterizam os problemas de sequenciamento da produção, com foco especial em sistemas que incorporam máquinas paralelas idênticas e as heurísticas de otimização. Por fim, é apresentada uma breve revisão das principais contribuições relacionadas ao tema em questão.

2.1 PLANEJAMENTO E CONTROLE DA PRODUÇÃO

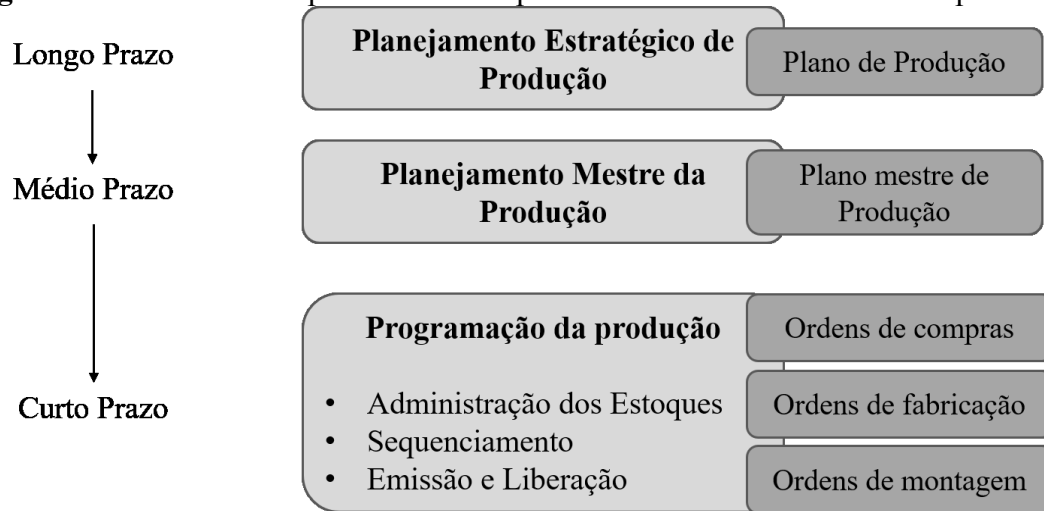
A competitividade empresarial impulsiona as organizações a buscarem constantemente ferramentas estratégicas que as destaquem em relação aos seus concorrentes. De acordo com Chiavenato (2014), uma das principais vantagens competitivas das empresas é alcançar a excelência na produção a partir de processos eficientes. No entanto, a fim de obter uma produção bem-sucedida e alcançar níveis desejáveis de eficiência e eficácia, é essencial que a produção se apoie em um sistema confiável de planejamento e controle, de modo a garantir o atingimento dos objetivos estabelecidos, contribuindo para o sucesso e competitividade das empresas.

Segundo Jacobs *et al.* (2018), a principal função do Planejamento e Controle da Produção (PCP) é gerir de forma eficiente a utilização de recursos humanos e equipamentos, bem como otimizar o fluxo de materiais no sistema produtivo, de modo que as organizações consigam atender às exigências dos clientes de maneira satisfatória a partir do uso adequado da capacidade dos seus fornecedores e das suas instalações internas.

Guerrini, Belhot e Azzolini Júnior (2014) afirmam que a agregação e o desenvolvimento das atividades de Planejamento e Controle de Produção ocorre por meio de um fluxo de informações que leva em consideração as etapas do processo produtivo, a hierarquia de planejamento, na qual busca-se desagregar as informações necessárias e, sobretudo, o horizonte de tempo que possibilitará uma melhor orientação dos objetivos da organização em relação à produção.

De modo geral, as atividades de apoio do sistema de PCP podem ser categorizadas em três horizontes temporais distintos: longo prazo, médio prazo e curto prazo, conforme apresentado na Figura 2. Estes prazos estão associados respectivamente aos níveis estratégico, tático e operacional das organizações (Jacobs *et al.*, 2018).

Figura 2 - Horizontes temporais e suas respectivas atividades desenvolvidas pelo PCP



Fonte: Adaptado de Tubino (2017).

Conforme Jacobs *et al.* (2018) e Tubino (2017), as atividades executadas pelo PCP são distribuídas em cada um dos prazos da seguinte forma:

- longo prazo: são estabelecidas as estratégias de longo prazo da empresa, e, nessa perspectiva, o setor de planejamento e controle da produção assume a responsabilidade de determinar a capacidade adequada de equipamentos, instalações e fornecedores para garantir o atendimento às futuras demandas do mercado. Portanto, a organização deverá desenvolver um Plano de Produção considerando as previsões de vendas nesse horizonte de planejamento, a fim de ajustar os recursos produtivos conforme a demanda esperada;
- médio prazo: nesse período, é elaborado o Plano Mestre de Produção (PMP), cuja função é planejar táticas para que o sistema produtivo opere de maneira mais eficiente, visando atender às previsões de vendas de médio prazo e/ou os pedidos em carteira já acordados com os clientes. O enfoque do médio prazo é direcionado para o planejamento das quantidades adequadas de materiais, para que cheguem no momento e local apropriados de modo a atender às necessidades dos clientes, além de identificar eventuais problemas que possam afetar a viabilidade do plano de produção, propondo medidas preventivas, como a necessidade de utilizar horas extras, recorrer à subcontratação e/ou terceirização;
- curto prazo: realiza-se minuciosamente a programação dos recursos para atender aos requisitos de produção no dia a dia, abrangendo aspectos como o gerenciamento de tempo, pessoal, materiais, equipamentos e instalações. Nesse contexto, o PCP assume a responsabilidade de programar a produção, sequenciando as etapas e os pedidos, gerenciando os estoques e emitindo as ordens de compras, fabricação e montagem, ao mesmo tempo que gera relatórios para acompanhar e controlar o processo produtivo. Com

base nas informações contidas no PMP, nos dados de engenharia e nos registros de controle de estoque, a programação da produção define, a curto prazo, a quantidade e o momento mais adequados para comprar, fabricar ou montar cada elemento necessário para a composição dos produtos finais.

2.2 SEQUENCIAMENTO DA PRODUÇÃO

O sequenciamento da produção consiste em determinar a sequência mais adequada para a execução das ordens de fabricação em cada uma das máquinas disponíveis, considerando os tempos de início e término de processamento em cada uma delas (Félix, 2019). O processo de tomada de decisão acerca do sequenciamento ocorre frequentemente em diversas indústrias de manufatura e prestação de serviços, visando principalmente otimizar um ou mais objetivos por meio da alocação de recursos para tarefas em períodos de tempo específicos (Pinedo, 2016).

Segundo Pereira (2022), o sequenciamento como processo de tomada de decisão é uma prática essencial para garantir que as atividades de produção ocorram de forma ordenada e eficiente, maximizando a utilização dos recursos disponíveis. Nesse contexto, o desenvolvimento de um cronograma de tarefas detalhado assume uma importância fundamental, pois ajuda a manter a eficiência e o controle das operações (Pinedo, 2016).

De acordo com Pinedo (2016), todos os problemas de sequenciamento envolvem um número finito de n tarefas e m máquinas, em que o índice j se refere a uma tarefa específica, enquanto o índice i se refere a uma máquina em particular. Segundo o autor, há diversos fatores que caracterizam os problemas de sequenciamento da produção, os mais frequentemente encontrados são:

- Tempo de processamento (p_{ij}) – representa o tempo necessário para completar o processamento da tarefa j na máquina i . Caso o tempo de processamento da tarefa j não dependa da máquina ou a tarefa j seja processada em apenas uma máquina específica, o índice i é omitido;
- *Release date* (r_j) – corresponde ao momento em que a tarefa j está liberada para iniciar o seu processamento no sistema de produção;
- *Due date* (d_j) – refere-se ao prazo de entrega da tarefa j prometido ao cliente. Em alguns cenários, são aplicadas penalizações caso o processamento do pedido seja concluído após o prazo acordado ou caso seja entregue antecipadamente. Ao considerar o conceito

de *due date* tem-se também o *deadline*, o qual corresponde a data limite que a tarefa j deverá ser entregue ao cliente, dessa forma, a tarefa não pode ser concluída após o prazo estabelecido pelo *deadline*;

- Peso (w_j) – trata-se de um indicador de prioridade que representa a relevância de uma determinada tarefa j em relação às outras tarefas do sistema. A prioridade pode estar associada ao custo de manutenção, estoque e até mesmo aos custos que envolvem as penalizações pelo não cumprimento dos prazos estabelecidos dessa tarefa específica.

Além de considerar os fatores mencionados anteriormente, Pinedo (2016) expressa os problemas de sequenciamento de acordo com três variáveis (α , β e γ), onde α corresponde a configuração das máquinas, β apresenta os detalhes sobre as características do processamento e/ou indica as restrições do problema e, por fim, γ que descreve o objetivo do problema a ser otimizado. Assim, as principais configurações de máquinas (α) encontradas nos processos produtivos são:

- *Single machine (I)* – refere-se ao cenário em que há apenas uma máquina responsável por todas as tarefas, é considerado o caso mais simples;
- *Identical machines in parallel (P_m)* – nessa circunstância existem m máquinas em paralelo no sistema, de modo que a tarefa j pode ser executada em qualquer uma das máquinas que estão em paralelo ou então em um subconjunto dessas máquinas M_j pré-determinado. Caso exista uma restrição no processamento da tarefa, ou seja, se a tarefa j puder ser processada em apenas uma máquina específica, essa informação será registrada na entrada M_j do campo β ;
- *Machines in parallel with diferente speeds (Q_m)* – há um total de m máquinas operando em paralelo, no entanto, cada uma delas possui velocidades diferentes. Nessa situação, a velocidade da máquina i é indicada por v_i e o tempo p_{ij} necessário para processar a tarefa j é calculado como p_j/v_i ;
- *Flow shop (F_m)* – nos problemas que envolvem o *flow shop* existem m máquinas em série, e todas as tarefas seguem a mesma sequência de produção, por isso, há uma ordem pré-definida de máquinas em que todas as tarefas devem ser processadas. Após a conclusão em uma máquina, as tarefas são inseridas na fila da próxima máquina para processamento.

- *Job shop* (J_m) – esse tipo de configuração ocorre quando há m máquinas e cada tarefa é processada em uma sequência de produção específica. Nesse caso, as tarefas podem ser processadas em cada máquina uma única vez ou mais, além de não haver a necessidade de ser processada em todas as m máquinas;
- *Open shop* (O_m) – existe um total de m máquinas disponíveis e cada tarefa pode ser executada em qualquer ordem nessas máquinas. Esse sistema proporciona uma maior flexibilidade, uma vez que não há uma sequência fixa de processamento em relação às máquinas, permitindo escolher a ordem em que as tarefas são processadas em cada uma delas.

No campo β são indicados as restrições e limitações referentes ao sistema de produção. Conforme mencionado por Pinedo (2016) as principais são:

- *Release date* (r_j) – como restrição indica que a tarefa não pode ter o seu processamento iniciado antes do prazo do *release date*, ou seja, a tarefa j não poderá ser processada no sistema antes de uma data específica de liberação. Caso r_j não seja mencionado no campo β , significa que não há restrições de data para iniciar o processamento da tarefa;
- *Preemptions* ($prmp$) – permite a interrupção do processamento de uma tarefa a qualquer momento, podendo essa ser retomada na mesma máquina ou em outra. Essa característica permite que a tarefa interrompida mantenha o progresso já realizado, garantindo que a quantidade de processamento não seja perdida;
- *Precedence constraints* ($prec$) – geralmente essa restrição está presente em situações com uma única máquina ou em configurações com máquinas paralelas, em que uma tarefa só pode iniciar seu processamento quando as tarefas anteriores forem concluídas;
- *Sequence dependent setup times* (s_{jk}) – nesta situação, pode haver tempos de preparação entre o processamento de diferentes tarefas, e esses tempos geralmente dependem da sequência em que as tarefas são realizadas. Portanto, s_{jk} representa o tempo de preparação entre o processamento das tarefas j e k . Quando k é a primeira tarefa da sequência, s_{0k} denota o tempo de preparação para essa tarefa, enquanto s_{j0} representa o tempo de preparação após a conclusão da tarefa j quando esta é a última da sequência. Além disso, se o tempo de preparação também é influenciado pela máquina onde o

processamento está ocorrendo, a notação s_{ijk} é empregada, sendo i o índice que representa a máquina específica;

- *Job families (fmls)* – neste caso as n tarefas pertencem a F diferentes famílias de tarefas. Embora as tarefas da mesma família possam ter tempos de processamento distintos, elas podem ser executadas em sequência em uma mesma máquina sem necessidade de realizar o *setup* entre elas. Contudo, ao ocorrer uma transição de uma família para outra na máquina, por exemplo, da família g para a família h , será necessária uma preparação prévia.

Por fim, o campo γ que se refere as possíveis funções objetivos que podem ser otimizadas. Pinedo (2016), destaca alguns exemplos que são comumente utilizados nos problemas de sequenciamento:

- *Makespan (C_{max})* – o *makespan* considera o momento em que a última tarefa é concluída no sequenciamento, sua definição é dada por $\max(C_1, \dots, C_n)$ e busca pela minimização do tempo de término;
- *Maximum lateness (L_{max})* – diz respeito ao atraso mais significativo observado em relação às datas de vencimento entre todas as tarefas, sendo seu principal objetivo a minimização da pior violação da data de entrega. O cálculo do atraso de cada tarefa pode ser expresso como $L_j = C_j - d_j$ e definido como $\max(L_1, \dots, L_n)$;
- *Total completion time ($\sum C_j$)* – trata-se do somatório dos tempos de término de todas as tarefas que foram processadas;
- *Total weighted completion time ($\sum w_j C_j$)* – refere-se a soma dos tempos de conclusão ponderados. Essa soma é comumente conhecida na literatura como *flow time* e é calculada a partir da atribuição de pesos em cada tarefa, que reflete sua importância ou prioridade;
- *Discounted total weighted completion time ($\sum w_j (1 - e^{-rC_j})$)* – os custos são reduzidos a uma taxa r por unidade de tempo, normalmente o valor de r está entre 0 e 1. Dessa forma, caso a tarefa j não seja concluída até o tempo t ocorrerá um custo adicional no período $[t, t + dt]$ definido por $w_j r e^{-rt} dt$. Por outro lado, se a tarefa j for

concluída no tempo t , o custo total incorrido durante o período $[0, t]$ é dado por $w_j(1 - e^{-rt})$;

- *Total weighted tardiness* ($\sum w_j T_j$) – refere-se à redução da soma dos atrasos de cada tarefa considerando a atribuição de pesos. O atraso ponderado de uma tarefa é calculado considerando a diferença entre o tempo real de conclusão e o prazo estabelecido, sendo esse valor multiplicado pelo peso atribuído à tarefa em questão. Os atrasos totais também podem ser calculados sem a utilização de pesos, nesses casos a notação adotada é $\sum T_j$;
- *Weighted number of tardy jobs* ($\sum w_j U_j$) – essa métrica busca avaliar o número total de tarefas atrasadas em um cenário de sequenciamento, em que cada tarefa atrasada é ponderada conforme seu impacto ou importância, tal como na métrica mencionada anteriormente.

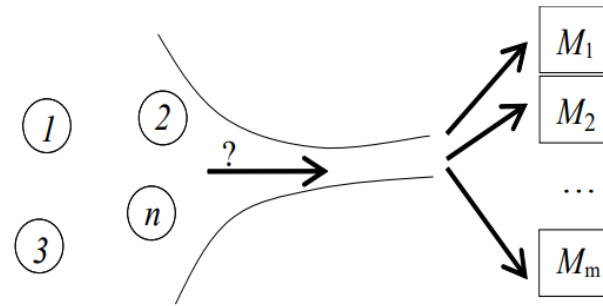
2.3 SEQUENCIAMENTO EM MÁQUINAS PARALELAS IDÊNTICAS

O planejamento de máquinas paralelas pode ser visualizado como um procedimento dividido em duas fases. Inicialmente, é necessário definir quais tarefas serão designadas para quais máquinas. Em seguida, é preciso estabelecer a ordem das tarefas atribuídas a cada máquina (Tocha, 2014).

Segundo Pinedo (2016), os problemas envolvendo máquinas paralelas idênticas são considerados NP-difíceis, isto é, não se tem conhecimento de um algoritmo capaz de resolver esses problemas em tempo polinomial, tornando a obtenção de soluções precisas mais complexa à medida que o número de tarefas aumenta. Um exemplo de um problema que se enquadra na classe NP-difícil é o sequenciamento em máquinas paralelas idênticas com o objetivo de minimizar os atrasos totais das tarefas, denotado por $P_m | \sum T_j$ (Mensendiek; Gupta; Herrmann, 2015).

De acordo com Biskup, Herrmann e Gupta (2008), o problema de minimização dos atrasos totais em máquinas paralelas idênticas pode ser delineado da seguinte forma: considere-se um conjunto $N = \{1, 2, \dots, n\}$ composto por n tarefas que precisam ser sequenciadas em um conjunto $M = \{1, 2, \dots, m\}$ de m máquinas paralelas idênticas, conforme ilustrado na Figura 3.

Figura 3 – Sequenciamento em máquinas paralelas idênticas



Fonte: Kaabi e Harrath (2014).

Assume-se que cada uma das tarefas $j \in N$ apresenta um processo determinístico, um *due date* d_j e um tempo de processamento p_j , em que, divisão, interrupção ou cancelamento não são permitidos e os valores de m e n permanecem fixos. Também se considera que todas as tarefas estão prontas para execução no instante zero, de modo que o processamento de uma tarefa começa imediatamente após o término da tarefa anterior. O tempo de atraso $T_j(S)$ de uma tarefa j em um sequenciamento S é determinado pela fórmula $T_j(S) = \max\{0, C_j(S) - d_j\}$, onde $C_j(S)$ representa o momento em que a tarefa j é finalizada no sequenciamento S . Adicionalmente, define-se como S_i a sequência das tarefas programadas na máquina $i \in M$ (Biskup; Herrmann; Gupta, 2008).

A partir das definições realizadas acima, os autores Biskup, Herrmann e Gupta (2008), representam o problema de sequenciamento de todas as atividades em múltiplas máquinas por

$$S = (S_1, S_2, \dots, S_m), \text{ cujo objetivo é minimizar o atraso total } T = \sum_{j=1}^n T_j(S).$$

Além disso, também propuseram um modelo de programação linear inteira mista que não exige a atribuição de índices de máquina às variáveis binárias para a problemática descrita acima. Para essa modelagem os autores utilizaram notações em que: C_h representa o tempo de conclusão da tarefa h ($h = 1, \dots, n$); T_h indica o atraso da tarefa h ($h = 1, \dots, n$); y_h é a variável binária que assume o valor um se a tarefa h for a primeira tarefa sequenciada em uma das m máquinas ($h = 1, \dots, n$); x_{jh} é caracterizada como a variável binária que tem o valor igual a um se a tarefa j for sequenciada imediatamente antes da tarefa h na mesma máquina ($j, h = 1, \dots, n, j \neq h$); $x_{j,n+1}$ é a variável binária que é igual a um se a tarefa j for a última tarefa em uma máquina ($j = 1, \dots, n$); e por fim, R que indica um número suficientemente grande.

A formulação matemática proposta por Biskup, Herrmann e Gupta (2008) encontra-se descrita a seguir.

$$\min Z = \sum_{h=1}^n T_h \quad (1)$$

Sujeito a:

$$\sum_{h=1}^n y_h \leq m \quad (2)$$

$$y_h + \sum_{j=1, j \neq h}^n x_{jh} = 1, \forall h = 1, \dots, n \quad (3)$$

$$\sum_{j=1, j \neq h}^{n+1} x_{jh} = 1, \quad \forall h = 1, \dots, n \quad (4)$$

$$C_h \geq p_h \cdot y_h, \quad \forall h = 1, \dots, n \quad (5)$$

$$C_h \geq C_j + p_h - R(1 - x_{jh}), \quad \forall j, h = 1, \dots, n, \quad j \neq h \quad (6)$$

$$T_h \geq C_h - d_h, \quad \forall h = 1, \dots, n \quad (7)$$

$$x_{jh} + x_{hj} \leq 1, \quad \forall j, h = 1, \dots, n, \quad j \neq h \quad (8)$$

$$\sum_{j=1}^n x_{jn+1} \leq m \quad (9)$$

$$y_j + \sum_{h=1, h \neq j}^{n+1} x_{jh} \leq 2, \quad \forall j = 1, \dots, n \quad (10)$$

$$\begin{aligned} x_{jh} \in \{0,1\}, \quad \forall j = 1, \dots, n, \quad h = 1, \dots, n+1, \quad j \neq h \\ y_h \in \{0,1\}, \quad \forall h = 1, \dots, n \\ T_h, C_h \geq 0, \quad \forall h = 1, \dots, n \end{aligned} \quad (11)$$

Na formulação mencionada acima, a função objetivo (1) minimiza o atraso total do sequenciamento. A restrição (2) assegura que, no máximo, m tarefas podem ser selecionadas

como a primeira a serem executadas em cada máquina. As restrições (3) garantem que cada tarefa seja alocada em pelo menos uma máquina ou precedida por outra tarefa. De maneira similar, o conjunto das restrições (4) asseguram que cada tarefa seja seguida por outra tarefa ou seja a última na sequência de execução de alguma máquina (Biskup; Herrmann; Gupta, 2008).

As restrições (5) asseguram que o tempo de conclusão da primeira tarefa em cada máquina seja pelo menos igual ao seu tempo de processamento. O tempo de conclusão C_h deve ser igual ou maior que o tempo de processamento p_h acrescido do tempo de conclusão de sua tarefa predecessora C_j para todas as tarefas restantes h . Caso j não seja o predecessor direto de h , a introdução de R torna o conjunto de restrições (6) menos restritivas. Conforme mostram as restrições (7), o cálculo do atraso é realizado a partir da diferença entre o tempo de conclusão e a data de vencimento (Biskup; Herrmann; Gupta, 2008).

As restrições (8) têm por objetivo evitar que, se a tarefa j for a predecessora da tarefa h , a tarefa h não possa preceder a tarefa j . Já a restrição (9) estabelece um limite máximo de m tarefas que podem ser processadas por último, enquanto que as restrições (10) garantem que apenas um sucessor seja permitido para a primeira tarefa em uma máquina. Importante notar que as restrições (8) a (10), em si, não são essenciais para a formulação do problema, no entanto, elas restringem o espaço de solução e, assim, reduzem o esforço computacional para resolução do problema. Por fim, tem-se as restrições (11) que definem os limites dos valores de todas as variáveis (Biskup; Herrmann; Gupta, 2008).

Para melhor demonstrar o problema supracitado, apresenta-se a seguir um exemplo para uma instância do problema com $n = 6$ tarefas e $m = 2$ máquinas. A Tabela 1 apresenta os valores em dias referentes aos tempos de processamento (p_j) de cada tarefa j e as suas respectivas datas de entregas (d_j).

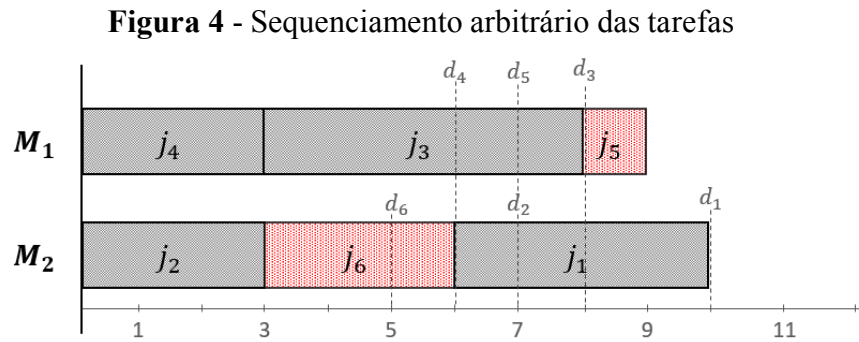
Tabela 1 – Exemplo de instância para seis tarefas e duas máquinas

j	p_j	d_j
1	4	10
2	3	7
3	5	8
4	3	6
5	1	7
6	3	5

Fonte: Autoria própria (2023).

Na Figura 4, observa-se um sequenciamento em que são alocadas aleatoriamente as tarefas 4, 3 e 5 para a máquina 1 e as tarefas 2, 6 e 1 para a máquina 2. Para determinar o atraso

total, é necessário calcular a diferença entre o instante final de conclusão de cada tarefa e o seu prazo de entrega, realizando ao final o somatório total dos atrasos por tarefa.



Fonte: Autoria própria (2023).

No caso ilustrado acima, nota-se que há duas tarefas atrasadas, as tarefas 5 e 6, tendo em vista, que a tarefa 5 concluiu o seu processamento dois dias após o período acordado para a entrega e a tarefa 6 após um dia, para esse cenário tem-se um atraso total de três dias.

2.4 HEURÍSTICAS DE OTIMIZAÇÃO

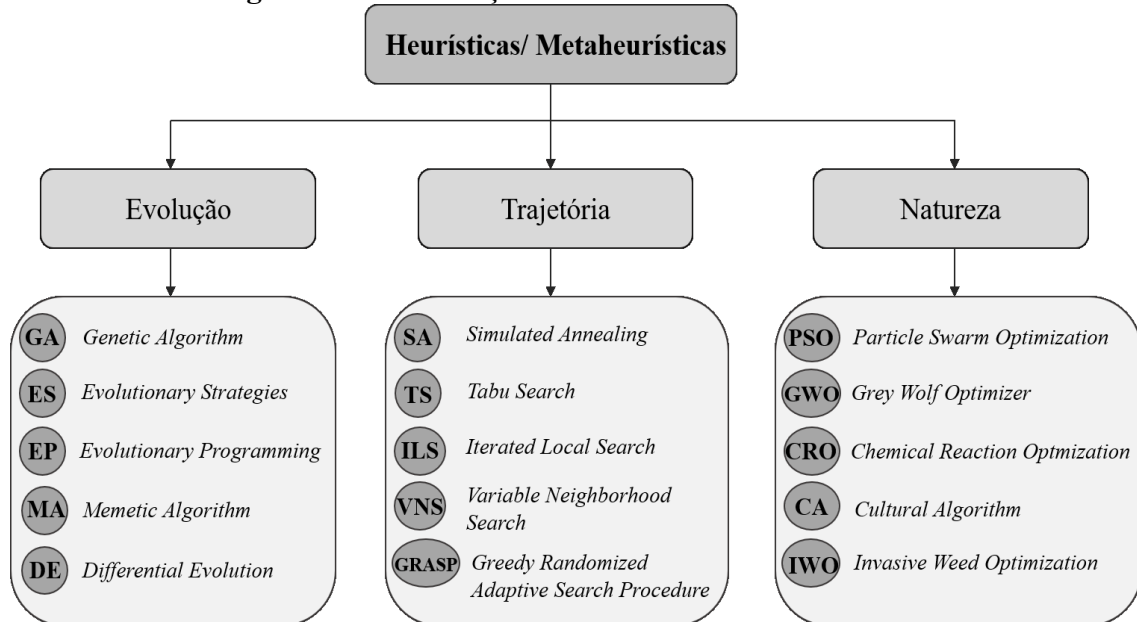
Para analisar a eficiência de um algoritmo ou determinar até que ponto ele continua sendo eficiente em um problema de sequenciamento da produção, é fundamental avaliar a complexidade da resolução no cenário específico do estudo (Vivian, 2010).

Segundo Santos (2016), dentro da área de Pesquisa Operacional há vários problemas para os quais não se tem conhecimento de algoritmos exatos que possam, em tempo polinomial, encontrar a solução ótima global. Esses desafios apresentam complexidade computacional classificada como NP-Difícil. Para estes casos, conforme ressaltado por Arroyo (2002), recorre-se a utilização de métodos aproximados, também referidos como métodos heurísticos e metaheurísticos.

Com o intuito de encontrar soluções de boa qualidade, Pereira (2011), destaca que uma diversidade de algoritmos avançados e técnicas de otimização são concebidas e aprimoradas, como Algoritmos Genéticos, *Tabu Search*, *Particle Swarm Optimization* (PSO), *Local Search* (LS), *Iterated Local Search* (ILS), *Greedy Randomized Adaptive Search Procedure* (GRASP), além de diversos outros métodos conforme ilustrado na Figura 5.

As heurísticas e metaheurísticas tratadas na Figura 5 são categorizadas e baseadas em evolução, trajetória e inspiradas na natureza. De acordo com Harifi *et al.* (2020), os modelos evolucionários simulam o processo de seleção natural e evolução biológica, na qual populações concorrem em busca de soluções, em que uma função objetivo orienta a seleção dos melhores indivíduos em cada geração.

Figura 5 - Classificação de heurísticas e metaheurísticas



Fonte: Adaptado de Harifi *et al.* (2020).

Os métodos baseados em trajetória operam com uma solução única, que passa por modificações em cada fase e é substituída por soluções superiores encontradas nas vizinhanças (Nesmachnow, 2014). Por outro lado, as técnicas inspiradas na natureza aderem à princípios naturais e abrangem categorias como métodos de enxame, biologia, física/química, seres humanos e plantas, explorando o comportamento coletivo desses elementos como uma estratégia para encontrar soluções (Harifi *et al.*, 2020).

De acordo com Arroyo (2002), essas abordagens têm a capacidade de oferecer soluções de boa qualidade em tempos razoáveis e compatíveis com as exigências de rapidez associadas aos problemas, desde que sejam devidamente desenvolvidas e ajustadas. Além disso, esses métodos são amplamente empregados para resolver problemas que envolvem a entrada de dados de grandes dimensões.

As heurísticas são abordagens que não demandam uma grande quantidade de requisitos para resolver um problema, porém requerem um profundo conhecimento sobre o próprio problema em questão (Santos, 2016). Portanto, conforme destacado por Goldberg e Luna

(2005), para cada problema de otimização, desenvolve-se estratégias aproximadas que considerem as singularidades do modelo, de modo a conceber heurísticas altamente especializadas.

De acordo com Goldbarg e Luna (2005), as heurísticas têm atraído um interesse crescente no meio acadêmico, não apenas por entregarem resultados satisfatórios em relação aos objetivos propostos, mas também devido à qualidade das soluções alcançadas, tornando-as uma alternativa cada vez mais atrativa para resolver muito dos desafios do mundo real envolvendo modelos combinatórios. Segundo Aguiar *et al.* (2005), na literatura, é possível identificar dois tipos principais de heurísticas: as heurísticas construtivas e as heurísticas de busca local.

As heurísticas construtivas são projetadas para solucionar um problema específico e têm como finalidade criar uma solução passo a passo, seguindo regras particulares associadas à estrutura desse problema (Arroyo, 2002). Os elementos candidatos costumam ser ordenados com base em uma abordagem gulosa, que avalia a inserção de cada elemento, e apenas o melhor elemento é adicionado em cada etapa. Normalmente, esse tipo de heurística é empregado para a criação da solução inicial do problema (Gomes Júnior, 2007).

Segundo Pinto (2014), as heurísticas de busca local são inicializadas a partir de uma solução viável e a cada iteração a solução é modificada por meio de movimentações realizadas nos seus elementos, caso alcance um ótimo local é substituída pela melhor solução encontrada em uma vizinhança pré-definida. Conforme destacado por Arroyo (2002), o desempenho das heurísticas de busca local está intrinsecamente ligado a dois fatores, o primeiro refere-se à seleção da solução inicial e o segundo à correta definição de uma vizinhança que estabeleça conexões entre as soluções no espaço de decisões.

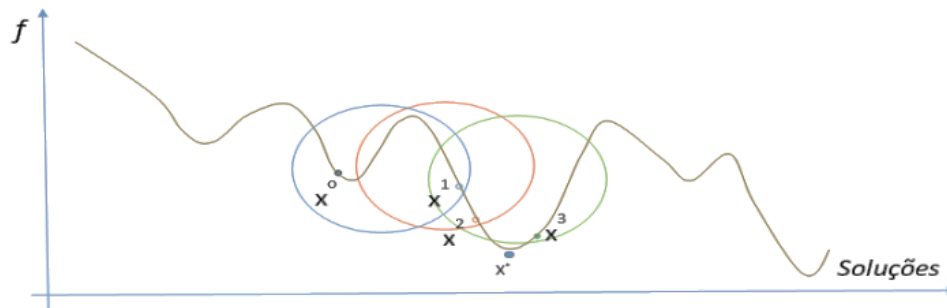
2.4.1 Heurísticas de busca local

Os métodos de busca local são fundamentados na ideia de vizinhança, dessa forma, as abordagens exploram o espaço de busca, movendo-se de forma iterativa na vizinhança de uma solução para outra que esteja próxima (Alvarenga; Rocha, 2006).

Para uma melhor compreensão do funcionamento da busca local, os autores Marques, Siqueira e Zacarias (2017) apresentaram o seguinte exemplo: ao considerar a minimização de uma função objetivo f , a solução inicial x_0 é avaliada e, por meio de múltiplas iterações com soluções vizinhas, uma solução aprimorada, x_1 , é encontrada e passa a ser a solução atual. Posteriormente, novas buscas são conduzidas, agora considerando uma nova vizinhança, o que

permite que a solução progrida até a obtenção de x_3 . A Figura 6 ilustra algumas das possibilidades de vizinhança para o exemplo supracitado.

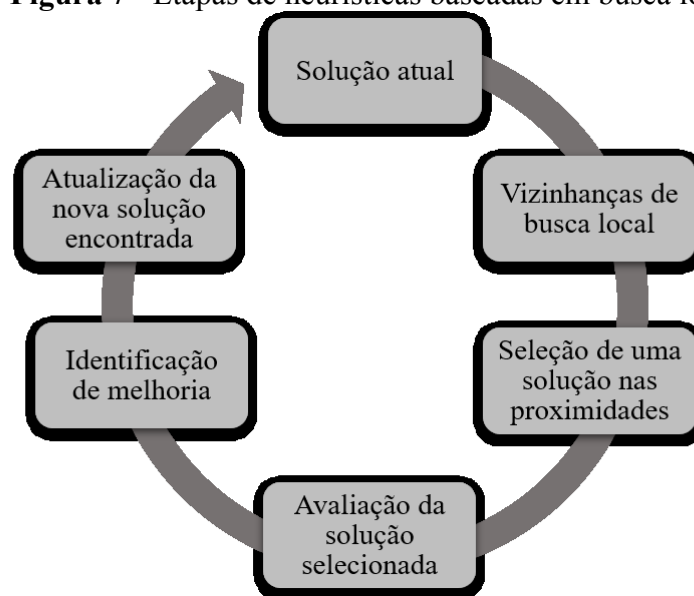
Figura 6 – Gráfico representativo do funcionamento da busca local



Fonte: Marques, Siqueira e Zacarias (2017).

De acordo com Coelho (2006), as heurísticas de busca local começam com uma solução inicial arbitrária e buscam aprimorá-la por meio de buscas na vizinhança com operações como troca, remoção ou inserção. Alvarenga e Rocha (2006), destacam que caso uma melhoria seja identificada, isto é, se o valor da função objetivo obtida for melhor, a solução atual é modificada e, em seguida, a vizinhança próxima dessa nova solução é explorada. Esse ciclo se repete até que não seja mais possível encontrar novas melhorias, todas as etapas do ciclo podem ser observadas na Figura 7.

Figura 7 - Etapas de heurísticas baseadas em busca local



Fonte: Autoria própria (2023).

Geralmente, a busca local é uma abordagem de otimização de fácil concepção e implementação, que oferece soluções de qualidade em um curto espaço de tempo, tornando-a amplamente empregada na prática. No entanto, uma das limitações mais significativas da busca local é sua tendência a convergir para ótimos locais, por isso, os pesquisadores fazem uso de técnicas de perturbação (Talbi, 2009). Segundo Menezes (2014), essas perturbações facilitam a saída de ótimos locais por meio de movimentos, abrindo oportunidade para explorar novas regiões do espaço de busca sem comprometer as características do ótimo local atual.

De acordo com Dias (2006), as técnicas heurísticas de busca local mais conhecidas incluem o método de Descida e o método da Subida, os quais podem ser executados a partir da aplicação de duas abordagens: completa ou estratégia de primeira melhora.

Haddad (2012), destaca que o método da Descida/Subida Completa inicia a partir de uma solução e realiza uma busca usando uma estrutura de vizinhança para procurar o melhor vizinho. Esse método é conhecido por empregar a estratégia *Best Improvement*, pois avalia todos os vizinhos disponíveis, dessa forma, o método só é concluído após encontrar o melhor vizinho em um ótimo local.

O método da Descida/Subida de Primeira Melhora foi desenvolvido para reduzir o custo computacional elevado associado ao processo de análise de toda a estrutura de vizinhança, por isso, utiliza da estratégia *First Improvement*, que interrompe a busca na vizinhança quando identifica um vizinho melhor que a solução atual, ainda que este não seja necessariamente o melhor (Cota, 2014).

Segundo Colherinhas (2016), há outras heurísticas de busca local mais refinadas, como a *Variable Neighborhood Descent* que foi originalmente apresentada por Mladenović e Hansen (1997). De acordo com o autor, essa heurística investiga o espaço de soluções por meio de trocas sistemáticas na configuração das vizinhanças, possibilitando uma abordagem mais flexível.

2.5 TRABALHOS RELACIONADOS

Nas últimas décadas, o interesse nos problemas de programação de tarefas tem aumentado significativamente, tanto entre profissionais quanto acadêmicos. Isso se deve, em grande parte às diversas aplicações na indústria e à complexidade computacional. Encontrar soluções ótimas ou quase ótimas para esses problemas é um desafio real enfrentado pelas empresas, por isso, os pesquisadores têm desenvolvido várias abordagens para solucionar esses

problemas, como formulações matemáticas, estratégias heurísticas e métodos de decomposição (Sáenz-Alanís *et al.*, 2016).

Em particular, tem-se a aplicação desses métodos nos problemas de sequenciamento envolvendo máquinas paralelas idênticas, que são estudados sob diversas restrições e considerando diferentes funções objetivas, incluindo a redução de *makespan*, dos atrasos totais e dos tempos de conclusão. Neste tópico, serão apresentados alguns trabalhos que tiveram como principal foco a redução dos atrasos totais, sendo este o objetivo a ser tratado no presente estudo.

Biskup, Herrmann e Gupta (2008) propuseram a utilização de três algoritmos heurísticos que exploram diferentes estratégias para problemas de redução dos atrasos totais. Os autores consideraram três regras para a construção da heurística BHG: *Smallest Laxity-Value* (SLA); *Total Processing Time* (TPI) e *Earliest Due Date* (EDD).

Na pesquisa desenvolvida por Rocholl e Mönch (2019) foi proposta uma formulação de Programação Linear Inteira Mista (MILP) para abordar tanto os problemas de sequenciamento de itens quanto de lotes com foco na redução de atrasos. Para isso, eles conceberam heurísticas de referência que se baseiam em escalonamento de listas, ILS e alguns algoritmos baseados em *Biased Random-Key Genetic Algorithm* (BRKGA), empregando diferentes estratégias de decomposição que demonstraram um ótimo desempenho em relação ao tempo de processamento e a qualidade das soluções.

Gharehgozli, Tavakkoli-Moghaddam e Zaerpour (2009), utilizaram um modelo de Programação Inteira Mista com Metas (MIGP) a partir da abordagem *fuzzy*, com o intuito de reduzir a soma dos atrasos totais ponderados e o tempo total de processamento.

Ferreira e Silva (2019), apresentaram uma versão da metaheurística ILS que incorpora as técnicas de busca *Very Large-scale Neighborhood Search* (VLNS) e *Dynasearch Swap*, com o intuito de minimizar a soma dos atrasos ponderados. O algoritmo desenvolvido mostrou-se aplicabilidade em problemas de grande escala em que são obtidas soluções de boa qualidade. Kim, Song e Jeong (2019), investigaram o problema de sequenciamento em máquinas paralelas com divisão de tarefas e tempos de *setup* a partir do desenvolvimento de abordagens metaheurísticas como *Simulated Annealing* (SA) e *Genetic Algorithms* (GA), os resultados mostraram que os algoritmos propostos conseguem resolver de forma eficiente problemas de grande porte.

Dois abordagens distintas foram utilizadas por Lee (2017) para resolver o problema de sequenciamento em uma fábrica de placas Acrilonitrila-Butadieno-Estireno, conhecidas como placas ABS. A primeira abordagem envolveu a criação de uma heurística *Adaptive Tabu Search* (ATCS) fundamentada na regra de despacho, enquanto a segunda consistiu na elaboração da

metaheurística *Random Iterated Greedy* (RIG). A partir das análises experimentais, nota-se que a metaheurística proposta demonstra um desempenho superior em relação aos algoritmos de busca tabu, obtendo soluções ótimas para problemas de pequeno porte, além de promover melhorias significativas em problemas de grande porte.

Shim e Kim (2008), propuseram a utilização de um algoritmo *Branch-and-Bound* (B&B) em problemas de sequenciamento onde uma tarefa é dividida em partes menores, permitindo que essas partes sejam processadas em diferentes máquinas. Os resultados das análises computacionais evidenciaram que o algoritmo proposto demonstrou a capacidade de alcançar soluções ótimas para cenários envolvendo até quatro máquinas e doze tarefas ou cinco máquinas e oito tarefas, dentro de um período de tempo razoável.

A técnica de otimização heurística baseada em GRASP em conjunto com o *path relinking*, foram utilizados por Armentano e França Filho (2007), com o intuito de encontrar melhores soluções para os problemas de máquinas paralelas idênticas envolvendo sequências dependentes dos tempos de *setup*.

Tanaka e Araki (2008) utilizaram em seu estudo um algoritmo B&B em conjunto com a técnica de Relaxamento Lagrangiano, os autores obtiveram ótimas soluções para problemas com instâncias de até 25 tarefas, independentemente do número de máquinas envolvidas. De maneira análoga, Lee e Kim (2015) também desenvolveram em seu estudo um algoritmo B&B para reduzir o atraso total de tarefas sequenciadas em duas máquinas paralelas idênticas. Os experimentos conduzidos por instâncias geradas aleatoriamente revelaram que o algoritmo B&B proposto conseguiu identificar soluções ideais para casos com até 28 tarefas, dentro de um tempo de computação considerado razoável.

Yalaoui (2012), propôs a aplicação de abordagens exatas e aproximadas para minimizar os atrasos totais em máquinas paralelas idênticas com *release dates* distintos. Para resolução dessa problemática, o autor utilizou um método exato por meio do software Cplex, o algoritmo *Ant Colony Optimization* (ACO), o método *Tabu Search* (TS), um conjunto de estratégias heurísticas que se fundamentam em regras de prioridade e uma versão modificada do método BHG de Biskup, Hermann e Gupta (2008). Os resultados mostraram que o método adaptado BHG apresentou o melhor desempenho, principalmente para instâncias maiores que 12 tarefas, garantindo a melhor solução em 50% dos casos. Já o método ACO obteve um bom funcionamento para instâncias de pequeno porte. Com relação ao método exato, embora ele forneça as melhores soluções, o software utilizado limita o tempo computacional em 1800 segundo, não permitindo a utilização em todos os tamanhos de instâncias definidas no estudo.

Como resumo desta pesquisa, apresenta-se a seguir o Quadro 1 que lista as principais informações dos 11 artigos escolhidos, destacando os métodos de solução utilizados pelos autores para resolução dos problemas de sequenciamento em máquinas paralelas idênticas com foco na minimização dos atrasos totais.

Quadro 1 – Métodos de solução para problemas de redução do atraso total

Método de solução	Trabalho
<i>Greedy Randomized Adaptive Search Procedure</i>	Armentano e França Filho (2007)
<i>Iterated Local Search</i>	Rocholl e Mönch (2019) Ferreira e Silva (2019)
<i>Tabu Search</i>	Lee (2017) Yalaoui (2012)
Algoritmo BHG	Biskup, Herrmann e Gupta (2008) Yalaoui (2012)
Algoritmo <i>Ant Colony Optimization</i>	Yalaoui (2012)
Algoritmo <i>Branch-and-Bound</i>	Shim e Kim (2008) Tanaka e Araki (2008) Lee e Kim (2015)
Algoritmos genéticos	Kim, Song e Jeong (2019) Rocholl e Mönch (2019)
Programação de Metas Inteira Mista Difusa	Gharehgozli, Tavakkoli-Moghaddam e Zaerpour (2009)
Metaheurísticas <i>Random Iterated Greedy</i>	Lee (2017)
Metaheurísticas <i>Simulated Annealing</i>	Kim, Song e Jeong (2019)

Fonte: Autoria própria (2023).

Com base no Quadro 1, observa-se que o Algoritmo *Branch-and-Bound* foi amplamente empregado nos problemas, no entanto, sua aplicação se limita a instâncias com menos de 30 tarefas. Por outro lado, os métodos *Iterated Local Search* e *Tabu Search* também são utilizados, destaca-se principalmente as heurísticas e metaheurísticas baseadas em ILS, que demonstraram um ótimo desempenho quando aplicadas em instâncias de grande escala, conforme mostra o estudo de Ferreira e Silva (2019). Diante disso, optou-se por utilizar as heurísticas LS no presente estudo.

2.5.1 Trabalhos relacionados com enfoque em outros objetivos

Além das abordagens para minimização dos atrasos totais em problemas de sequenciamento com máquinas paralelas idênticas, é importante destacar que há diversos outros objetivos amplamente explorados no campo de pesquisa.

Turker e Sel (2011), desenvolveram um algoritmo genético de permutação sequencial abrangente para resolver problemas de sequenciamento de duas máquinas paralelas com o objetivo de reduzir o *makespan*. O algoritmo genético funciona a partir de uma busca que rastreia todas as possibilidades utilizando um conjunto de dados aleatórios.

Behnamian, Zandieh e Ghomi (2009) propuseram um algoritmo híbrido o qual combina as metaheurísticas *Ant Colony Optimization* (ACO), SA e *Variable Neighborhood Search* (VNS) com o objetivo de minimizar o *makespan* em problemas de sequenciamento envolvendo tempos de *setup*. Para validar a abordagem, os pesquisadores realizaram comparações com dois algoritmos de um método híbrido baseado em VNS, o SA/VNS e o ACO/VNS, além de empregar o próprio algoritmo VNS proposto em literatura. Os resultados demonstraram que o algoritmo híbrido desenvolvido, apresentou um melhor desempenho entre todos diante dos conjuntos de instâncias estabelecidos na pesquisa.

Lin *et al.* (2011) desenvolveram uma heurística gulosa iterativa aprimorada para problemas de sequenciamento com a finalidade de minimizar o atraso máximo envolvendo restrições de tempo de *setup* e *release dates*, o método é caracterizado por uma progressiva redução da temperatura, um elemento-chave que intensifica a capacidade de localizar soluções melhores ou ótimas. A heurística IG_{ST} proposta combinou a simplicidade da heurística *Iterated Greedy* (IG) com o mecanismo de resfriamento do SA, mostrando-se superior às metaheurísticas *Best Solution* (BS_{OU}), *Restricted Simulated Annealing* (RSA) e IG_{CT}.

No estudo conduzido por Sáenz-Alanís *et al.* (2016), foi introduzida uma estratégia heurística em GRASP com o intuito de resolver um problema de sequenciamento de lotes em uma cervejaria no México, focando na otimização do *makespan*. A heurística elaborada demonstrou uma notável eficácia ao produzir soluções em menos de 19 segundos, ao mesmo tempo em que conseguiu reduzir, em média, até 44,8% do total de dias requeridos para satisfazer a demanda.

A pesquisa desenvolvida por Mellouli *et al.* (2009) propõe a utilização dos métodos de programação linear inteira mista, programação dinâmica e do algoritmo *Branch-and-Bound* para minimizar a soma dos tempos de conclusão em um sistema de máquinas paralelas idênticas.

Mundim e Fuchigami (2017), propuseram uma heurística robusta com o objetivo de otimizar o *makespan* em situações de sequenciamento que incorporam tempos de *setup* dependentes das ordens de produção. Os autores utilizaram um limitante inferior derivado de um modelo de programação linear inteira mista, obtendo um desvio percentual de 5,51% em relação a esse limite, o que demonstrou a eficácia da solução desenvolvida.

Najat *et al.* (2019), sugeriram uma abordagem heurística *Shortest-Batch-Completion Time Heuristic 1* (SBC1) e um modelo matemático para minimizar o número de tarefas atrasadas, levando em consideração a manutenção preventiva periódica das máquinas. Os resultados obtidos com a implementação da abordagem heurística mostraram que, em 19 de 54 instâncias, a solução ótima foi alcançada.

Diaz-Cazañas, Paz-Martínez e Delgado-Sobrinó (2019) desenvolveram um algoritmo híbrido denominado HCSA com o intuito de minimizar o *makespan* em uma indústria de plásticos. O algoritmo foi gerado utilizando uma Heurística Construtiva (HC) em combinação com a técnica SA, destacando-se consideravelmente em relação a outras abordagens previamente descritas na literatura e aplicadas ao mesmo contexto.

Os objetivos dos problemas abordados nos estudos previamente apresentados juntamente com os autores e ano estão detalhados no Quadro 2.

Quadro 2 - Variações de objetivos em problemas de sequenciamento de máquinas paralelas idênticas

Objetivo do problema	Trabalho
<i>Makespan</i>	Behnamian, Zandieh e Ghomi (2009) Turker e Sel (2011) Sáenz-Alanís <i>et al.</i> (2016) Mundim e Fuchigami (2017) Diaz-Cazañas, Paz-Martínez e Delgado-Sobrinó (2019) Mellouli <i>et al.</i> (2009)
Atraso máximo	Lin <i>et al.</i> (2011)
Número de tarefas atrasadas	Najat <i>et al.</i> (2019)

Fonte: Autoria própria (2023).

A partir do Quadro 2, torna-se evidente que o *makespan* é o objetivo mais estudado nas pesquisas dedicadas ao problema de sequenciamentos em máquinas paralelas idênticas. esse fator se deve a sua relevância prática nos ambientes de manufatura, em que a minimização do tempo de produção é crucial para reduzir custos e melhorar a eficiência operacional, além de

ser um objetivo versátil amplamente aplicável. No entanto, é relevante destacar que, embora o *makespan* seja o objetivo predominante, outras métricas de desempenho como a redução do atraso máximo e do número de tarefas atrasadas, também corroboram para atingir as metas das organizações.

3 METODOLOGIA

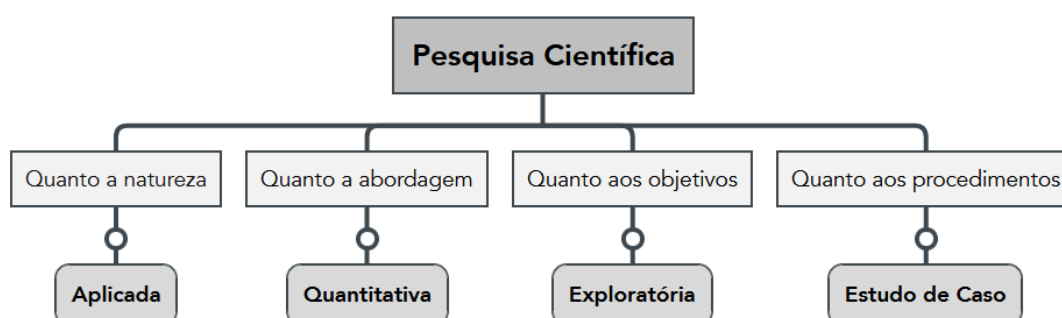
Este capítulo descreve os procedimentos metodológicos empregados para a elaboração do presente trabalho. Inicialmente é apresentada a caracterização da pesquisa, em seguida, são expostas as etapas a serem realizadas no processo de desenvolvimento e aplicação do modelo heurístico.

3.1 CARACTERIZAÇÃO DA PESQUISA

A pesquisa científica pode ser definida como uma série de ações planejadas, que busca encontrar uma resolução para um problema e se fundamenta em métodos racionais e sistemáticos. Normalmente, a pesquisa é conduzida quando um problema surge e não se dispõe de informações para resolvê-lo (Silva; Menezes, 2005).

Segundo Gil (2017), há várias maneiras de categorizar as pesquisas, permitindo uma compreensão mais abrangente e detalhada. Dessa forma, a classificação da pesquisa é realizada mediante alguns aspectos, como: natureza, abordagens metodológicas, objetivos e procedimentos técnicos utilizados (Nascimento, 2016). Na Figura 8, encontra-se a categorização na qual este estudo se enquadra.

Figura 8 - Caracterização da pesquisa científica



Fonte: Autoria própria (2023).

Referente à natureza, este trabalho é caracterizado como uma pesquisa aplicada, pois sua principal finalidade consiste na utilização de conhecimentos práticos para conceber uma sequência de produção otimizada, por meio da implementação de um método heurístico, cujo principal objetivo é a redução dos atrasos totais dos pedidos. Conforme enfatizado por Prodanov e Freitas (2013), a pesquisa aplicada tem como seu objetivo fundamental a geração de

conhecimento direcionado para aplicação prática, visando a resolução de problemas específicos encontrados no cotidiano.

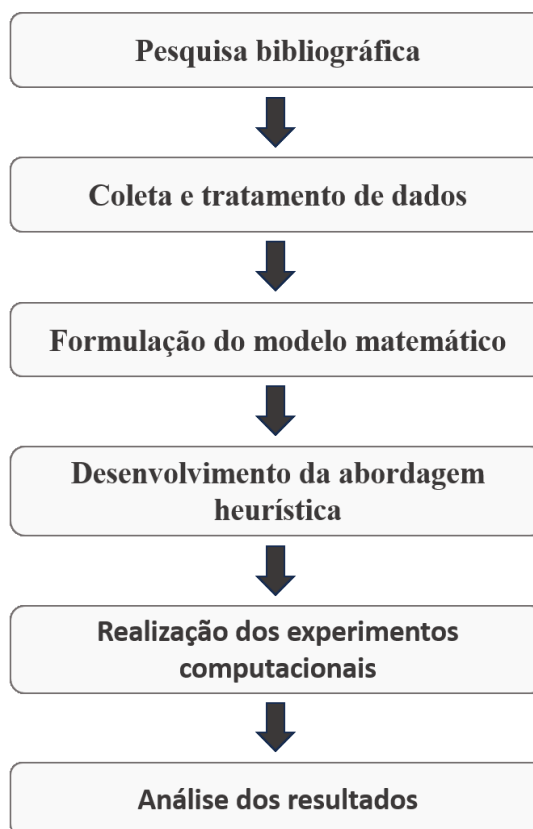
Segundo Silva e Menezes (2005), a pesquisa quantitativa pressupõe que tudo é mensurável, de modo a transformar opiniões e informações em valores numéricos para fins de categorização e análise por meio de recursos estatísticos. Zanella (2011) complementa, destacando que esse tipo de pesquisa é caracterizado pela adoção de métodos matemáticos e estatísticos, que se preocupam com a quantificação dos resultados. Com base nas evidências apresentadas, o estudo em questão demonstra uma abordagem quantitativa, uma vez que se utiliza de dados numéricos para aplicação do método heurístico e recorre a recursos estatísticos para uma quantificação mais precisa dos resultados alcançados.

Com relação ao objetivo, ele é classificado como de caráter exploratório, tendo em vista, que se realizou pesquisas bibliográficas para familiarização e melhor entendimento do problema abordado, além de analisar as abordagens heurísticas já utilizadas pelos pesquisadores, estimulando a compreensão acerca do objetivo da pesquisa, o qual busca propor melhorias no sequenciamento de tarefas em um ambiente produtivo com foco na redução dos atrasos totais, por meio da aplicação de uma abordagem heurística. De acordo com Gil (2017), as pesquisas de natureza exploratória buscam criar um ambiente de maior familiaridade com o problema em questão, tornando-o mais explícito ou elaborando hipóteses. Ainda segundo o autor, essas pesquisas são conhecidas por sua flexibilidade, já que buscam abranger diversos aspectos relacionados ao fenômeno em estudo.

Por fim, no que diz respeito aos procedimentos, esta pesquisa classifica-se como um estudo de caso, que é definido por Cauchick-Miguel (2018), como uma pesquisa empírica que se dedica à investigação minuciosa de um fenômeno específico, o qual ocorre em um cenário real e contemporâneo, além de envolver análises mais detalhadas. Portanto, este trabalho busca obter uma análise mais abrangente e detalhada do problema e dos objetivos da pesquisa, destacando o contexto em que a pesquisa foi conduzida, a fim de aprimorar a aplicação da abordagem heurística.

3.2 ETAPAS DA PESQUISA

O desenvolvimento do estudo em questão é dividido em seis etapas essenciais. Para uma melhor compreensão, a Figura 9 ilustra o fluxo metodológico do trabalho.

Figura 9 - Etapas de desenvolvimento do estudo

Fonte: Autoria própria (2023).

A primeira etapa consiste em adquirir embasamento teórico sobre o problema, por essa razão realizou-se uma pesquisa bibliográfica abrangente em diversas fontes, incluindo livros, teses, bases de dados de artigos científicos, além de periódicos nacionais e internacionais. A seleção dos materiais foi baseada na sua relevância para o trabalho, resultando em pesquisas sobre temas como planejamento e controle da produção, sequenciamento de tarefas, otimização de sequenciamentos de máquinas paralelas idênticas e métodos de resolução de problemas de sequenciamento.

Para realizar a coleta dos dados, estabeleceu-se o contato direto com a equipe responsável pela área de planejamento e controle da produção da organização. Os dados fornecidos já estavam disponíveis no sistema de Planejamento de Recursos da Empresa (ERP) e abrangem um intervalo de quatro meses, compreendendo o período de junho/23 a setembro/23. Devido a extensa quantidade de dados fornecidos, foi necessário efetuar o tratamento dos mesmos, o que motivou a realização de uma seleção, com o propósito de identificar quais dados eram verdadeiramente cruciais para o desenvolvimento da abordagem

heurística. Como resultado, os dados selecionados compreendem informações pertinentes aos pedidos, como: tempo de processamento, *release date*, tempo de *setup* e data de entrega.

A terceira etapa do processo envolve a elaboração do modelo matemático, começando pela delimitação do escopo do problema relacionado ao sequenciamento para a redução dos atrasos totais. Essa definição é orientada pela identificação de três elementos fundamentais:

- Variáveis de decisão;
- Função objetivo;
- Restrições que afetam o funcionamento do modelo.

As decisões que podem ser quantificadas são representadas por meio de variáveis de decisão e as limitações que essas variáveis de decisão estão sujeitas são expressas por meio de equações e inequações, que são denominadas restrições e desempenham um papel fundamental na concepção do modelo.

A quarta etapa consistiu no desenvolvimento da abordagem heurística. Para a sua elaboração empregou-se a linguagem de programação *Visual Basic for Applications* (VBA) e o auxílio de planilhas eletrônicas no *Microsoft Excel*, que continham os dados das instâncias para validar o funcionamento da heurística. O processo de desenvolvimento inicia com a geração da solução inicial, seguida da avaliação dessa solução de acordo com a função objetivo. Posteriormente, a abordagem emprega estratégias de busca local iterativa juntamente com perturbações, com a finalidade de refinar a solução e reduzir os atrasos totais. Para finalizar o desenvolvimento da heurística, realizou-se a calibração dos parâmetros utilizados na estratégia de busca local, de modo a otimizar o desempenho da heurística e alcançar soluções mais eficazes.

Os experimentos foram feitos em um computador Intel Core i5 com 2,3 GHz, 8 GB de memória RAM e sistema operacional Windows 10. Para a execução dos experimentos, utilizou-se um total de oito instâncias, sendo quatro delas relacionadas a uma linha de produção para o período de junho a setembro, e as outras quatro restantes foram referentes a outra linha de produção durante o mesmo período.

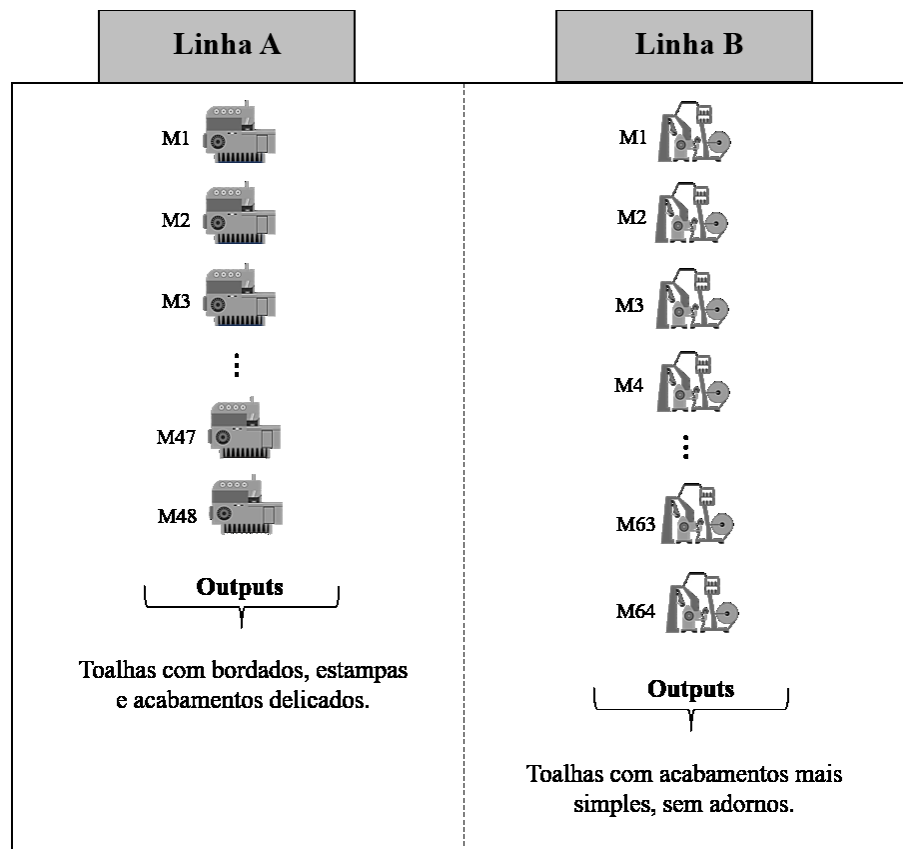
Por fim, na quinta etapa analisou-se os resultados das experimentações computacionais conduzidas para as oito instâncias, considerando principalmente parâmetros relacionados ao atraso total e ao tempo de execução da heurística para cada cenário específico.

4 CARACTERIZAÇÃO DO PROBLEMA

4.1 DESCRIÇÃO DA EMPRESA

O foco da investigação concentra-se na otimização do sequenciamento da produção de uma empresa inserida no setor têxtil, localizada nas cidades de João Pessoa - PB e Natal - RN. A organização é responsável pelo processo de tecelagem de produtos pertencentes ao segmento de banho, principalmente toalhas, operando em duas linhas de produção distintas, as quais serão referenciadas nesse estudo como linhas A e B. A Figura 10 oferece uma representação visual da distribuição das linhas de produção e seus respectivos *outputs*.

Figura 10 – Esquema ilustrativo das linhas de produção A e B



Fonte: Autoria própria (2023).

A linha A é composta por um conjunto de 48 teares que operam de forma paralela, realizando as mesmas funções. O que diferencia a linha A em relação a linha B, é o nível de tecnologia empregado em suas máquinas, que incorporam técnicas de tecelagem avançadas. Em virtude disso, os produtos processados por essa linha tendem a apresentar detalhes mais refinados e complexos, tais como bordados delicados, estampas e acabamentos especiais.

Por sua vez, a linha B opera com um total de 64 teares idênticos, também dispostos em paralelo. No entanto, os produtos resultantes dessa linha tendem a possuir um acabamento mais simples, principalmente devido ao fato de que as máquinas dessa linha são de uma geração mais antiga em termos de tecnologia de tecelagem.

4.2 DESCRIÇÃO DO PROBLEMA

Em ambientes industriais com prazos de entregas desafiadores, é crucial que os programadores de produção estejam preparados com as estratégias mais eficazes para lidar com os problemas de sequenciamento e programação da produção, garantindo que os produtos sejam entregues dentro dos prazos acordados. A capacidade de otimizar o sequenciamento de tarefas e as datas de entrega é essencial para garantir o atendimento às necessidades dos clientes de forma eficiente e satisfatória.

Durante a pesquisa, identificou-se que a empresa não dispõe de um software específico para sequenciar suas ordens de fabricação. Dessa forma, a programação é conduzida por meio de planilhas eletrônicas no *Excel*, principalmente com base nas informações fornecidas pelo sistema ERP da empresa, como os itens que devem ser produzidos, suas quantidades, matérias-primas necessárias, tempos de processamento e prazos de produção, além de se basear na experiência do programador de produção, que prioriza as ordens de fabricação com prazos de entrega mais iminentes. No entanto, essa abordagem torna difícil a adaptação rápida a mudanças nos prazos de entrega de determinadas tarefas, o que pode ser um desafio.

Com base no que foi exposto, a questão central que é abordada neste trabalho é o problema de sequenciamento de tarefas em máquinas paralelas idênticas, considerando a existência de *release date* e sequências dependentes de *setup*, com objetivo principal na minimização dos atrasos totais. O problema estudado possui algumas características que precisam ser destacadas, como aspectos referentes ao processamento das tarefas e ao funcionamento das máquinas, os quais estão descritos nas próximas subseções.

4.2.1 Processamento das tarefas

Para uma descrição mais abrangente destaca-se as seguintes características relacionadas ao processamento das tarefas neste problema em estudo:

- a) para cada linha de produção há um conjunto específico de tarefas n , o que significa que os pedidos da linha A não são processados na linha B;

- b) as tarefas possuem um tempo de processamento p_i e um prazo de entrega d_i específicos;
- c) os pedidos podem ser processados em qualquer máquina da linha de produção que estão designados. Além disso, o tempo de processamento das tarefas não se modificam, são independentes da máquina;
- d) cada tarefa só pode ser processada uma vez;
- e) existe uma variedade de cinco tipos de produtos, cada um requer configurações e materiais específicos para sua produção;
- f) alguns pedidos estão sujeitos a *release date* r_i , isso significa que nem todos os pedidos estão imediatamente disponíveis no instante $t = 0$. Essa situação pode ocorrer devido à falta de disponibilidade de matéria-prima, o que implica que ela só estará acessível apenas em um momento específico.

Para melhor compreensão dos aspectos mencionados anteriormente é fornecido a seguir um exemplo. Considere uma instância com $n = 6$ tarefas, em que i representa os pedidos, p_i denota o tempo de processamento de cada tarefa, r_i indica o *release date* individual de cada pedido, k_i representa o tipo de produto e d_i corresponde à data de entrega dessas tarefas. A Tabela 2 mostra os valores específicos em dias da instância apresentada.

Tabela 2 – Exemplo de instância para seis tarefas

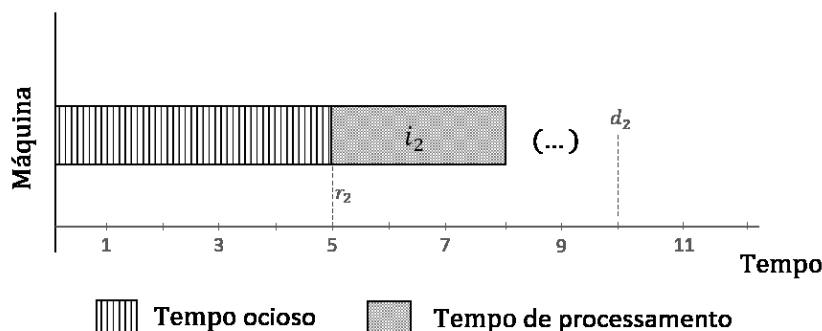
i	p_i	r_i	k_i	d_i
1	4	3	2	10
2	3	5	1	10
3	5	0	1	9
4	3	5	3	13
5	1	0	3	5
6	2	3	2	7

Fonte: Autoria própria (2023).

A partir da instância apresentada na Tabela 2, nota-se que há seis pedidos para serem sequenciados. Cada um desses pedidos é categorizado como sendo do tipo 1, 2 ou 3, sendo essa distinção de extrema importância no processo de sequenciamento, pois a natureza do pedido determina o tempo de *setup* necessário entre eles e em que momento o próximo pedido pode ser iniciado. Além disso, cada pedido possui seu próprio *release date*. Isso implica que, por exemplo, a tarefa $i = 2$ só pode iniciar a partir da data 5, mesmo que ela seja a primeira na

sequência, o processo terá um período de ociosidade que não permitirá que ela inicie em $t = 0$, o que afetará no cálculo do atraso total, a situação descrita pode ser visualizada na Figura 11.

Figura 11 – Gráfico de Gantt para demonstrar o *release date*



Fonte: Autoria própria (2023).

Ao analisar a Figura 11, fica evidente que a tarefa $i = 2$ foi a primeira a ser sequenciada, no entanto, devido ao fato de possuir uma *release date* definido para o dia cinco, essa tarefa só pode iniciar no próprio dia cinco ou posteriormente. Além disso, nota-se que, dado o tempo de processamento da tarefa, que é de três dias, ela só será concluída no dia oito, deixando apenas dois dias de margem em relação à sua data de entrega.

O período de ociosidade pode ser aproveitado para a execução de tarefas que possuem *release date* igual a zero. Entretanto, para que essa utilização eficiente ocorra, é necessário estabelecer um sequenciamento em que o pedido $i = 2$ não seja posicionado como o primeiro da sequência. Essa abordagem possibilitará uma distribuição mais equilibrada do tempo disponível, resultando em uma utilização mais eficiente dos recursos à disposição.

4.2.2 Características das máquinas

Para construção da heurística é essencial compreender as características operacionais das máquinas e como elas influenciam no sequenciamento. No contexto do problema em questão deve-se considerar que cada máquina é capaz de processar apenas uma tarefa por vez e ao iniciar o processamento não se admite interrupções.

Considera-se também que os tempos de *setup* são dependentes da sequência adotada e estão intrinsecamente relacionados à configuração dos teares. Isso ocorre devido à necessidade de ajustar a arcada da máquina conforme o tipo de produto, podendo modificar o fio/tecido para elaboração da toalha, sua quantidade por metro cúbico, entre outros ajustes.

Assim, quando uma tarefa j é sequenciada logo após uma tarefa i , sendo elas pertencentes a diferentes tipos de pedidos, há um tempo de preparação de máquina s_{ij} . Já quando não existe tempos de preparação de máquinas, isto é, $s_{ij} = 0$, isso indica que ambas as tarefas são pertencentes ao mesmo tipo de pedido. Essa distinção pode ser claramente visualizada no exemplo fornecido no Quadro 3. Além disso, assume-se que a máquina não requer tempo de preparação inicial, ou seja, o tempo necessário para preparar a máquina para processar a primeira tarefa na sequência é nulo.

Quadro 3 – Exemplo de matriz de tempos de *setup*

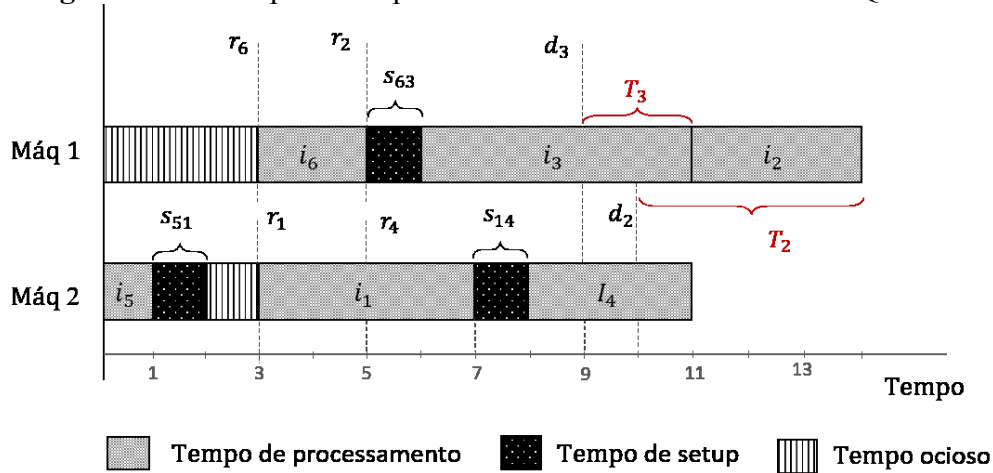
Tipo (k_i)	1	2	3
1	0	1	1
2	1	0	1
3	1	1	0

Fonte: Autoria própria (2023).

Ao examinar o Quadro 3, que oferece um exemplo dos tempos de preparação de máquina em dias, torna-se evidente que esses tempos estão intrinsecamente ligados à sequência e que essa dependência está diretamente relacionada ao tipo de produto, dessa forma, nota-se que quando se trata do mesmo tipo de produto, não há tempo de preparação de máquina, o que pode ser observado a partir da diagonal principal da matriz. Vale ressaltar, que a matriz apresentada aqui foi construída com base nas informações referentes aos três tipos de produtos mencionados no exemplo da Tabela 2.

Com base nas informações fornecidas sobre o processamento das tarefas, as especificações das máquinas e as informações dos tempos de *setup*, é possível exemplificar esse cenário por meio de um gráfico de Gantt apresentado na Figura 12. Este gráfico ilustra a sequência de execução dos pedidos listados na Tabela 2 em duas máquinas paralelas idênticas, considerando também os tempos de *setup* indicados no Quadro 3.

Figura 12 – Exemplo de sequenciamento descrito na Tabela 2 e Quadro 3



Fonte: Autoria própria (2023).

A sequência de produção exibida na Figura 12 foi determinada de maneira aleatória, resultando na alocação dos pedidos 2, 3 e 6 à máquina 1, enquanto os pedidos 1, 4 e 5 foram alocados à máquina 2. No sequenciamento adotado, observa-se que a máquina 1 só inicia seu processamento no instante $t = 3$, isso se deve ao fato de que o pedido 6 é o primeiro a ser sequenciado e o início do seu processamento só pode ocorrer a partir desse instante. O mesmo cenário ocorre com o pedido 1 na máquina 2.

Além disso, nota-se a ocorrência de tempos de preparação de máquinas, uma vez que alguns pedidos sequenciados pertencem a tipos diferentes, como no caso dos pedidos 1 e 4, onde o pedido 1 é do tipo 2 e o pedido 4 é do tipo 3, resultando em um tempo de *setup* $s_{14} = 1$.

No caso ilustrado acima, percebe-se também que na máquina 2 há dois pedidos atrasados, especificamente os pedidos 3 e 2. A tarefa 3 foi concluída dois dias após o prazo de entrega acordado d_3 , enquanto a tarefa 2 sofreu um atraso de quatro dias em relação ao seu prazo, para esse cenário tem-se um atraso total de seis dias. Esse cálculo é realizado pela somatória da diferença entre o instante final de conclusão de cada tarefa e o seu prazo de entrega.

Na seção seguinte apresenta-se o modelo matemático para representar o problema.

4.3 MODELAGEM MATEMÁTICA DO PROBLEMA

O modelo matemático proposto para esse problema foi desenvolvido por meio da Programação Linear Inteira Mista (PLIM) e pode ser representado como $P_m | r_i, s_{ij} | \sum T_i$. Para essa modelagem considera-se o número de tarefas n a serem sequenciadas e o número de

máquinas m , de modo que $n \geq m$. As variáveis de decisão e parâmetros utilizados nesse modelo seguem as seguintes notações:

- p_i : tempo de processamento da tarefa i , ($i = 1, \dots, n$);
- d_i : data de entrega da tarefa i , ($i = 1, \dots, n$);
- s_{ij} : sequência dependente do tempo de *setup* entre a tarefa i e a tarefa j , $i, j \in \{0, 1, \dots, n, n + 1\}$;
- r_i : *release date* da tarefa i , ($i = 1, \dots, n$);
- c_i : tempo de conclusão da tarefa i , ($i = 1, \dots, n$);
- x_{ij} : caracterizada como variável binária. Se a tarefa i for programada imediatamente anterior à tarefa j , $x_{ij} = 1$. Caso contrário, $x_{ij} = 0$;
- T_i : indica o atraso da tarefa i ($i = 1, \dots, n$).

A modelagem matemática proposta em PLIM está descrita na sequência:

$$\min Z = \sum_{i=1}^n T_i \quad (12)$$

Sujeito a:

$$\sum_{j=1}^n x_{0j} = m \quad (13)$$

$$\sum_{j=1}^{n+1} x_{ij} = 1, \forall i = 1, \dots, n \quad (14)$$

$$\sum_{i=0}^n x_{ij} = 1, \quad \forall j = 1, \dots, n \quad (15)$$

$$c_0 = 0 \quad (16)$$

$$c_j \geq p_j + \sum_{i=0}^n s_{ij}(s_{ij} + c_i), \quad \forall j = 1, \dots, n \quad (17)$$

$$c_j \geq r_j + p_j + \sum_{i=0}^n x_{ij} s_{ij}, \quad \forall j = 1, \dots, n \quad (18)$$

$$T_i \geq c_i - d_i, \quad \forall i = 1, \dots, n \quad (19)$$

$$T_i \geq 0, \quad \forall i = 1, \dots, n \quad (20)$$

$$\begin{aligned} x_{ij} &\in \{0,1\}, \quad \forall i, j \in \{0,1, \dots, n, n+1\} \\ x_{ij} &= 0, \quad i = j \\ c_j &\geq 0, \quad \forall j = 0, 1, \dots, n \end{aligned} \quad (21)$$

A equação (12) define a função objetivo, que é responsável pela minimização do atraso total das tarefas sequenciadas. A restrição (13) garante que o escalonamento ocorra em um conjunto de m máquinas. O conjunto de restrições (14) e (15) asseguram que cada tarefa seja selecionada e sequenciada em apenas uma máquina. A restrição (16) estabelece o instante de conclusão da tarefa fictícia 0, permitindo assim a utilização das restrições recursivas (17) e (18) para determinar os tempos de conclusão das tarefas reais.

As restrições (17) e (18) garantem que as tarefas somente iniciem seu processamento após a finalização da tarefa anterior, posterior ou simultâneo ao *release date* da tarefa em questão e após a conclusão da configuração requerida, que dependerá das particularidades da tarefa anterior e da tarefa atual. O atraso referente a cada tarefa é calculado considerando a diferença entre o tempo de conclusão e a data de entrega, conforme indicado nas restrições (19). Por fim, os conjuntos de restrições (20) e (21) estabelecem os limites para os valores das variáveis de decisão.

5 RESULTADOS

Ao longo dessa seção, é apresentado em detalhes o processo de construção da abordagem heurística, os resultados computacionais obtidos por meio da aplicação da heurística nas oito instâncias fornecidas pela empresa de tecelagem e a análise comparativa entre a heurística desenvolvida e o método utilizado pela empresa.

5.1 ABORDAGEM HEURÍSTICA PROPOSTA

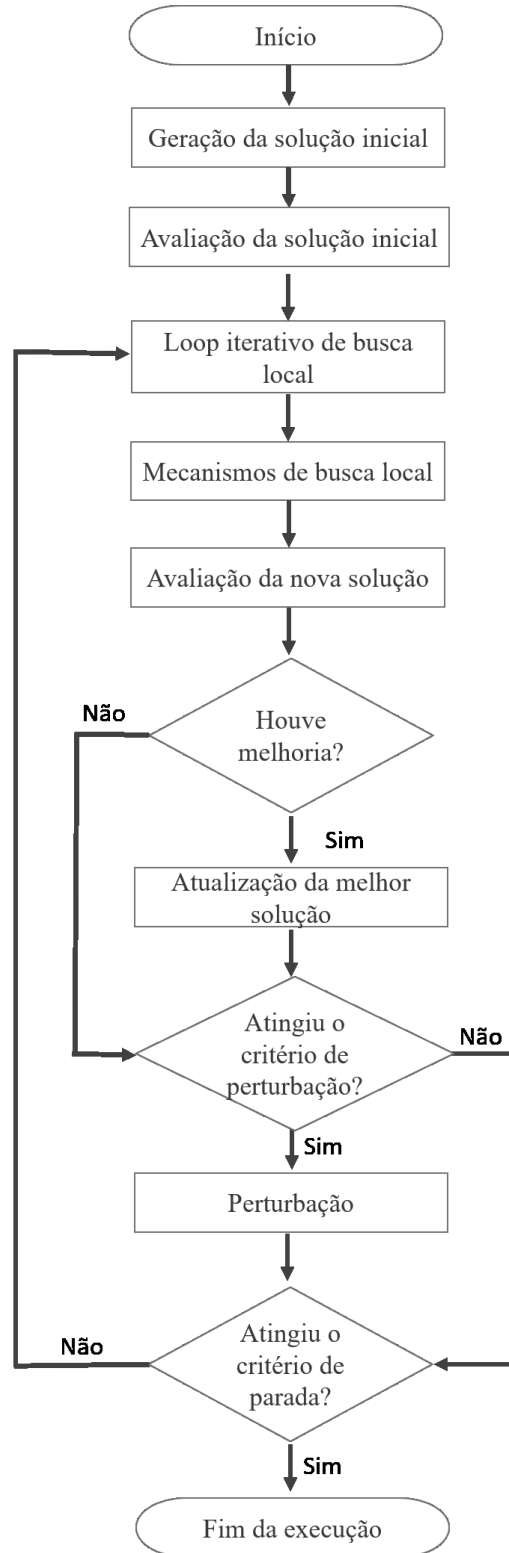
Em problemas de grande dimensão, os métodos exatos de otimização podem não ser a escolha mais adequada, principalmente devido à complexidade computacional envolvida. Esse fato se deve à necessidade de examinar minuciosamente todas as possíveis soluções, o que se torna impraticável quando o espaço de busca é extenso.

Os métodos heurísticos e metaheurísticos, por outro lado, demonstram eficácia ao encontrar soluções de boa qualidade em um tempo relativamente curto, tornando-se uma abordagem mais viável para lidar com problemas complexos de grande escala, o que pode ser visto em trabalhos como o de Ferreira e Silva (2019) e Rocholl e Mönch (2019).

A abordagem heurística deste estudo foi desenvolvida na linguagem de programação VBA e utiliza técnicas de busca local e perturbação para minimizar os atrasos totais em sequenciamentos de máquinas paralelas idênticas, levando em consideração os tempos de *setup* dependentes da sequência e *releases dates*.

Segundo Pereira 2019, a aplicação de heurísticas que se baseiam na busca local para melhorar a eficiência de problemas de programação em máquinas paralelas é algo recorrente, especialmente quando envolvem os tempos de preparação. O desenvolvimento da heurística envolveu algumas etapas, que estão ilustradas na Figura 13.

Figura 13 – Etapas da abordagem heurística



Fonte: Autoria própria (2023).

Primeiramente, realiza-se a geração de uma solução inicial, a qual prioriza o sequenciamento dos pedidos com a data mais próxima, além de considerar uma probabilidade de alocação das tarefas em máquinas com os menores tempos de processamento. Em seguida,

essa solução inicial é avaliada, considerando a função objetivo de minimização dos atrasos totais. Após essa avaliação, a heurística entra na fase do *loop* iterativo de busca local, no qual aplica-se três estratégias de vizinhança para encontrar uma nova solução. Cada nova solução gerada durante o processo é novamente avaliada da mesma forma que a solução inicial. Se o algoritmo identificar alguma melhoria na nova solução encontrada, ela então é armazenada como a melhor solução.

Após a verificação de melhoria, o algoritmo avalia se um determinado critério para a perturbação das soluções foi atendido, em caso afirmativo, a heurística aplica uma técnica de perturbação para diversificar as soluções e explorar diferentes regiões do espaço de busca, evitando a estagnação em ótimos locais e segue para verificar o critério de parada, que depende da definição do número máximo de iterações. Caso o critério de perturbação não seja satisfeito, o algoritmo prossegue diretamente para a avaliação do critério de parada. Quando o critério de parada é atingido, o código é encerrado, e a melhor solução encontrada até o momento é reportada como resultado final.

Entretanto, se o critério de parada não for satisfeito, o código retorna ao início do loop iterativo de busca local e repete todo o procedimento, buscando continuamente melhorar a solução. Esse processo é repetido até que o critério de parada seja atendido. Vale ressaltar, que ao obter uma solução por meio das estratégias de busca local, na próxima iteração, essas mesmas estratégias são aplicadas a solução corrente para gerar uma nova solução.

Nos próximos tópicos, serão detalhadas as etapas da heurística proposta, além da representação da solução e a calibração dos parâmetros.

5.1.1 Representação da solução

A solução é representada por uma matriz bidimensional, em que as quantidades de máquinas e tarefas irão variar de acordo com a instância, que depende do mês e da linha de produção. Em cada posição da matriz, identificada como (i, j) , o valor i representa a máquina à qual a tarefa está alocada, enquanto que o valor j indica a posição sequencial da tarefa naquela máquina para o seu processamento. Essa estrutura de dados possibilita a visualização clara e organizada das atribuições das tarefas às máquinas e sua ordem de execução, um exemplo ilustrativo dessa representação pode ser visualizado na Figura 14.

Figura 14 – Representação de uma solução arbitrária

		1	2	3	4	
Máquinas	1	5	8	10	2	
	2	11	9	1	3	→ Tarefa
	3	7	12	4	6	

Fonte: Autoria própria (2023).

Com base na Figura 14, observa-se a alocação de doze pedidos em três máquinas. Por exemplo, na máquina 1, os pedidos a serem processados são [5, 8, 10, 2], seguindo a mesma lógica para as outras máquinas. Nesse contexto, a posição (1, 1) na matriz indica que a tarefa 5 é a primeira a ser processada na máquina 1, enquanto a posição (1, 4) indica que a tarefa 2 é a última a ser processada na mesma máquina.

5.1.2 Geração da solução inicial

A geração da solução inicial é construída priorizando os pedidos com as datas de entrega mais próximas, além de direcionar a alocação para máquinas com tempos de processamento reduzidos. Inicialmente, aloca-se as tarefas com datas de entrega mais iminentes na primeira posição de cada máquina disponível. Em seguida, para alocar as demais tarefas verifica-se qual máquina detém o menor tempo de processamento e se o critério de probabilidade de 20% foi alcançado. Essa probabilidade de alocação é utilizada para ampliar a diversificação das soluções iniciais, dessa forma, as tarefas podem ou não ser alocadas as máquinas que possuem o menor tempo de processamento, garantindo que a cada execução do algoritmo uma solução inicial diferente seja gerada. O processo de geração da solução é concluído quando não há mais pedidos a serem alocados.

5.1.3 Avaliação da solução

A função objetivo $f(s)$ deste problema consiste na minimização dos atrasos totais. Para alcançar esse objetivo, é essencial realizar a avaliação não apenas da solução inicial, mas também de todas as soluções subsequentes que surgirão ao longo das iterações. O Algoritmo 1 fornece o procedimento utilizado para determinar o valor da função objetivo da solução corrente.

Por meio do Algoritmo 1, percebe-se que o cálculo do atraso individual T_i das tarefas depende da determinação do tempo de conclusão de cada tarefa C_i e da sua respectiva data de entrega d_i . Entretanto, o tempo de conclusão individual das tarefas é algo que vai se modificar conforme o sequenciamento selecionado, por isso, deve-se levar em consideração variáveis como *release date* r_i , tempos de *setup* $s_{(i-1, i)}$ entre a tarefa anterior e a atual, além dos tempos de processamento de cada uma das tarefas P_i .

Algoritmo 1 – Avaliação da solução

```

1: Procedimento Avaliação
2: entrada:  $s, C_i, d_i, r_i, s_{(i-1, i)}, P_i, T_i$ 
3: saída:  $f(s)$ 
4: para  $i = 1, \dots, Q_m$  faça
5:   para  $j = 1, \dots, Q_t$  faça
6:     se  $C_{i-1} \geq r_i$  então
7:        $C_i = C_{i-1} + s_{(i-1, i)} + P_i$ 
8:     senão
9:        $C_i = r_i + s_{(i-1, i)} + P_i$ 
10:    fim se
11:    se  $C_i \leq d_i$  então
12:       $T_i = 0$ 
13:    senão
14:       $T_i = C_i - d_i$ 
15:    fim se
16:  fim para
17: fim para
18: Retorne  $f(s)$ 
19: fim Avaliação

```

Fonte: Autoria própria (2023).

A fim de calcular o atraso total, o algoritmo precisa percorrer todas as posições da matriz bidimensional. Portanto, ele deve ser executado repetidamente, abrangendo o número total de máquinas na instância, representado por Q_m , bem como o número de tarefas por máquina, indicado por Q_t .

Existem dois cenários que devem ser considerados ao calcular o tempo de conclusão. O primeiro ocorre quando o tempo de processamento da tarefa anterior é igual ou superior ao *release date* da tarefa atual, como descrito na linha 6, isso implica que a próxima tarefa pode começar após o tempo de preparação, caso exista, uma vez que já estará pronta para iniciar seu processamento. Nesse caso, o tempo de conclusão dessa tarefa pode ser calculado usando a seguinte equação:

$$C_i = C_{i-1} + s_{(i-1, i)} + P_i \quad (22)$$

A Equação (22) calcula o tempo de conclusão da tarefa atual levando em consideração três componentes principais: o tempo de conclusão da tarefa anterior C_{i-1} , o tempo de preparação da máquina da tarefa anterior para a tarefa atual $s_{(i-1, i)}$ e o tempo de processamento do pedido atual P_i . Esses elementos combinados fornecem uma estimativa precisa do momento em que a tarefa atual será concluída. Caso a tarefa seja a primeira a ser sequenciada, o tempo de conclusão será igual ao tempo do seu próprio processamento.

No entanto, há um segundo cenário a ser considerado, no qual a tarefa anterior $i - 1$ pode ser concluída antes do *release date* da próxima tarefa. Nesse caso, o tempo de conclusão da tarefa atual deve considerar a ociosidade, incluindo o *release date* ao cálculo, assim a equação se torna:

$$C_i = r_i + s_{(i-1, i)} + P_i \quad (23)$$

Vale ressaltar que para aquelas tarefas que ocupam a primeira posição do sequenciamento, considera-se na Equação (23) apenas o *release date* e o tempo de processamento dessa tarefa. Uma vez calculado o tempo de conclusão de cada tarefa, é possível determinar o atraso individual de cada uma, que é dado pela diferença entre o tempo de conclusão da tarefa e sua data de entrega, como mostra a Equação (24):

$$T_i = C_i - d_i \quad (24)$$

Observa-se na linha 11 e 12 do Algoritmo 1, que se a tarefa for concluída antes da sua data de entrega não há atraso, então $T_i = 0$. Finalmente, o atraso total $f(s)$ é obtido a partir da Equação (25), em que se soma todos esses atrasos individuais:

$$f(s) = \sum_{i=1}^n T_i \quad (25)$$

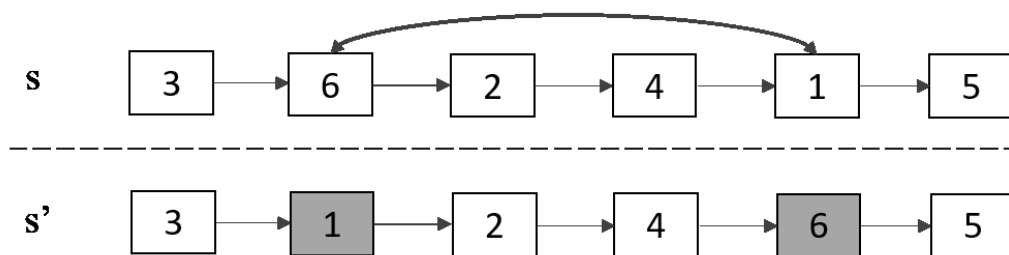
Os valores dos atrasos totais são armazenados para permitir comparações e substituições durante as iterações, sempre que melhorias forem identificadas. Isso garante que a heurística busque constantemente otimizar a solução encontrada.

5.1.4 Procedimento de busca local

A eficácia de um algoritmo de busca local está intrinsecamente ligada à escolha apropriada da vizinhança, a qual desempenha papel importante ao estabelecer conexões entre as soluções disponíveis no espaço de decisão, sendo sua configuração frequentemente determinada pelas características específicas do problema em análise. No contexto do problema de máquinas paralelas idênticas, foram aplicadas três estratégias diferentes para a geração de vizinhanças, sendo elas: *swap* intra máquinas, *swap* inter máquinas e *insertion*.

- **Swap intra máquinas:** consiste em selecionar duas posições distintas em uma mesma máquina e, em seguida, permutar as tarefas contidas nessas posições entre si, ou seja, o valor que originalmente ocupava a primeira posição selecionada passa a ocupar a segunda posição, e vice-versa. A ilustração desses movimentos é apresentada na Figura 15, na qual ocorre a troca de posições entre as tarefas 6 e 1.

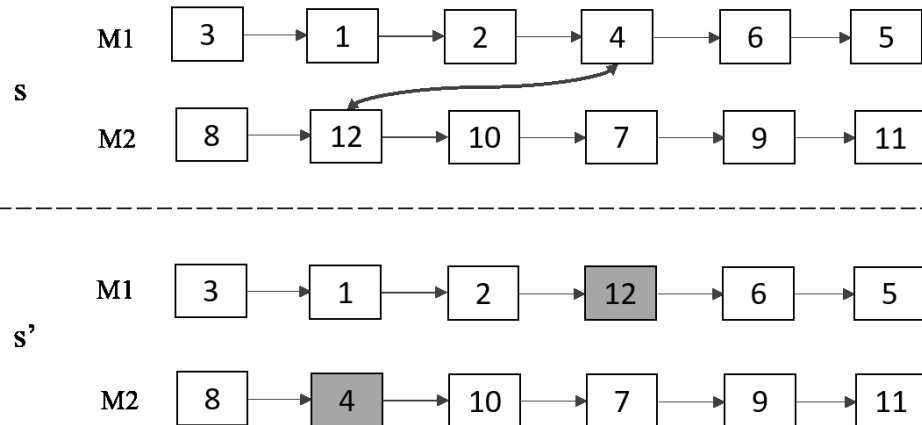
Figura 15 – Estratégia de *swap* intra máquinas



Fonte: Autoria própria (2023).

- **Swap inter máquinas:** os movimentos realizados nessa estratégia são similares ao *swap* intra máquina, no entanto, a troca acontece entre posições de máquinas distintas. Nesse processo, são escolhidas duas máquinas e uma posição referente a cada máquina selecionada, as tarefas contidas nessas posições são trocadas, uma pela outra. A Figura 16 demonstra a aplicação dessa estratégia ao realizar o movimento de troca de posições entre a tarefa 4 pertencente a máquina 1 e a tarefa 12 alocada a máquina 2.

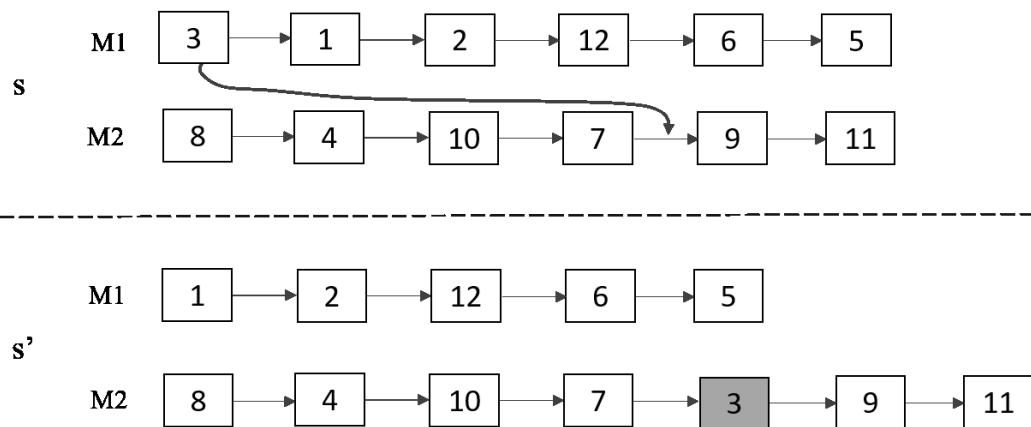
Figura 16 – Estratégia de *swap* inter máquinas



Fonte: Autoria própria (2023).

- **Insertion:** para essa estratégia, uma tarefa é retirada de sua posição atual em uma máquina, em seguida, é inserida em uma nova posição em outra máquina distinta da qual foi retirada. A representação visual dessa estratégia pode ser observada na Figura 17, em que a tarefa 3 sequenciada na máquina 1 é inserida na máquina 2 assumindo a posição da tarefa 9.

Figura 17 – Estratégia de *insertion*



Fonte: Autoria própria (2023).

No Algoritmo 2 é descrito um pseudocódigo da busca local iterativa, mostrando as etapas necessárias para a obtenção de soluções vizinhas a partir da aplicação das estratégias de busca local e perturbação. Essas etapas visam aprimorar e refinar as soluções encontradas.

Algoritmo 2 – Procedimento de busca local

```

1: Procedimento BuscaLocal (swap intra, swap inter, insertion)
2: entrada:  $s, C_i, d_i, r_i, s_{ij}, P_i$ 
3: saída:  $s'$ 
4: para Iter = 1, ..., MaxIter faça
5:   para  $i = 1, \dots, Q_{si}$  faça
6:     se  $v > \text{Prob}$  então
7:       Selecione duas tarefas diferentes
8:       Realize o swap intra entre as duas tarefas selecionadas
9:     fim se
10:  fim para
11:  para  $i = 1, \dots, Q_{se}$  faça
12:    se  $v > \text{Prob}$  então
13:      Selecione duas tarefas de máquinas diferentes
14:      Realize o swap inter entre as duas tarefas
15:    fim se
16:  fim para
17:  para  $i = 1, \dots, Q_{in}$  faça
18:    se  $v > \text{Prob}$  então
19:      Selecione duas posições ( $P_1, P_2$ ) em máquinas ( $M_1, M_2$ ) diferentes
20:      Insira a tarefa associada a  $P_1$  da  $M_1$  na posição  $P_2$  da  $M_2$ 
21:    fim se
22:  fim para
23:  Avalie a função objetivo para  $s'$ 
24:    se  $f(s') < f(s)$  então
25:       $s \leftarrow s'$ 
26:    fim se
27:  se critério de perturbação for atingido então
28:    Realize o procedimento de perturbação
29:  fim se
30: fim para
31: Retorne  $s'$ 
32: fim BuscaLocal

```

Fonte: Autoria própria (2023).

A variável s representa a melhor solução corrente que irá ser utilizada para encontrar novas vizinhanças, enquanto s' indica a solução obtida por meio das estratégias de busca local. Os parâmetros C_i, s_{ij}, P_i representam, respectivamente, os tempos de conclusão, de *setup* e de processamento de cada tarefa. O termo d_i indica a data de entrega er_i o *release date*. Todas essas variáveis são fundamentais para realizar a avaliação das funções objetivos $f(s')$ e $f(s)$, que quantificam o atraso total.

O procedimento de busca local inicia definindo a quantidade de iterações que serão realizadas para aplicação das estratégias de vizinhança e perturbação. Portanto, as iterações continuam sendo executadas até que o limite máximo, indicado por MaxIter, seja alcançado. A

partir do algoritmo 2, nota-se que as estratégias de vizinhança só serão executadas se o valor v for superior à probabilidade predefinida indicada por Prob. Dessa forma, a primeira estratégia empregada é o *swap* intra máquinas, que realiza uma quantidade de movimentações de trocas de posição denotadas por Q_{si} . A segunda, refere-se ao *swap* inter máquinas, que também possui uma quantidade específica de trocas Q_{se} . Por fim, tem-se a terceira estratégia que realiza uma quantidade de Q_{in} inserções de tarefas gerando uma nova solução s' .

Após a aplicação das estratégias, a solução s' é avaliada considerando a função objetivo descrita na subseção 5.1.3. Caso a nova solução gerada apresente um atraso total menor em comparação com a melhor solução existente, isto é, se $f(s') < f(s)$, então s' é adotada como a nova melhor solução ($s \leftarrow s'$).

Para assegurar que as soluções subsequentes não permaneçam estagnadas nos mesmos ótimos locais sem demonstrar melhorias, é introduzido, no término de cada iteração, o procedimento de perturbação. No entanto, essa execução é condicionada ao cumprimento do critério preestabelecido para a perturbação.

5.1.5 Mecanismo de perturbação

A técnica de perturbação aplicada ao processo iterativo de busca local pode ser visualizada a partir do Algoritmo 3. A finalidade principal da perturbação é introduzir alterações na solução atual, possibilitando que o algoritmo explore diferentes regiões e evite ficar retido em ótimos locais sem identificar novas melhorias.

Algoritmo 3 – Mecanismo de perturbação

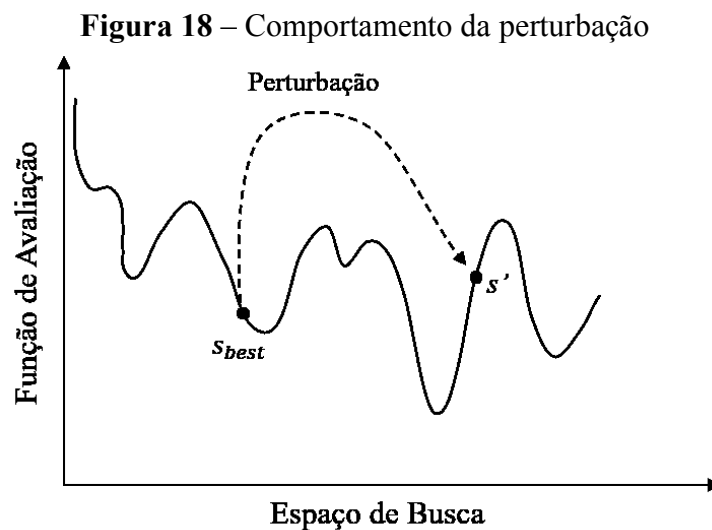
- 1: **Procedimento** Perturbação
 - 2: **entrada:** $s, s_{best}, f(s), f(s_{best}), c_m$
 - 3: **saída:** s'
 - 4: **se** C_p for satisfeito **então**
 - 5: **para** $i = 1, \dots, Q_m$ **faça**
 - 6: Realize a estratégia de *insertion*
 - 7: **fim para**
 - 8: $c_m \leftarrow 0$
 - 9: **fim se**
 - 10: Retorne s'
 - 11: **fim** Perturbação
-

Fonte: Autoria própria (2023).

O procedimento de perturbação só é inicializado quando o critério C_p é alcançado. Esse critério de perturbação, analisa se o atraso total da função atual $f(s)$ é 20% maior que o atraso

total da melhor solução vigente $f(s_{best})$, além de considerar se a quantidade de melhorias registradas c_m é maior que 1% do valor máximo de iterações definidas na busca local. Se essas condições forem satisfeitas, então, aplica-se a perturbação utilizando a estratégia *insertion* descrita na seção 5.1.4, a qual se repetirá até a quantidade total de máquinas Q_m da instância.

Ao realizar a estratégia de *insertion* na melhor solução corrente s_{best} , gera-se uma nova solução s' , que servirá como ponto de partida na próxima iteração de busca local, evitando revisitar ótimos locais já acessados. Esse procedimento pode ser visualizado na Figura 18.



Fonte: Autoria própria (2023).

Por fim, a variável c_m é zerada para inicializar o processo de perturbação sem considerar a quantidade de melhorias alcançadas até o momento. Vale ressaltar, que os movimentos de perturbação não visam melhorar a solução corrente, apenas diversificar o espaço de busca local para encontrar soluções que ainda não foram exploradas, podendo encontrar soluções melhores ou piores.

Coelho (2006), destaca que as perturbações devem ser suficientemente fortes para viabilizar a exploração de diversas soluções pela busca local, ao mesmo tempo em que precisam ser suaves o bastante para evitar uma reinicialização completa e aleatória.

5.1.6 Calibração dos parâmetros

Com o objetivo de aprimorar a eficiência dos resultados, realizou-se ajustes nos parâmetros da heurística, com foco especial naqueles que estão diretamente ligados às estratégias de busca local. Especificamente, procedeu-se à calibração dos seguintes parâmetros:

a probabilidade de execução de cada estratégia, o número de *swaps* intra e inter, a quantidade de *insertion* e o limiar de melhoria para reiniciar a busca local com a melhor solução atual. Esses ajustes são fundamentais para garantir que a heurística seja capaz de se adaptar de forma precisa às particularidades do problema, resultando em soluções de maior qualidade.

Para fornecer uma visão detalhada dessas configurações, os parâmetros calibrados e seus respectivos valores podem ser observados no Quadro 4.

Quadro 4 – Parâmetros da heurística de busca local

Parâmetros	Valores
Probabilidade de inicialização das estratégias de busca local	Rnd () > 0,20 Rnd () > 0,35 Rnd () > 0,50
Quantidade de <i>swap</i> intra	Q_m
Quantidade de <i>swap</i> inter	$Q_m/12$ $Q_m/4$
Quantidade de <i>Insertion</i>	$Q_m/12$ $Q_m/4$
Critério de retorno à melhor solução	$f(s) > 1,20 \cdot f(s_{best})$ $f(s) > 1,35 \cdot f(s_{best})$ $f(s) > 1,50 \cdot f(s_{best})$

Fonte: Autoria própria (2023).

Os valores dos parâmetros contidos no Quadro 4 foram definidos baseados nos estudos presentes na revisão da literatura deste trabalho e nos experimentos conduzidos no decorrer do estudo. Os resultados dos experimentos demonstraram que ao utilizar esses valores obtém-se soluções mais promissoras com relação ao desempenho da heurística.

Para a calibração adequada da heurística, é essencial conduzir cenários que explorem várias combinações de parâmetros, avaliando as médias dos atrasos, os tempos médios de execução e o desvio padrão dos atrasos. Nos experimentos, as combinações de parâmetros foram executadas 10 vezes em um *loop* de 5.000 iterações, sendo selecionada a combinação que apresentou os melhores resultados com relação ao atraso total. A Tabela 3 apresenta os resultados das combinações realizadas.

Tabela 3 – Resultados das combinações para calibração dos parâmetros

Combinação	Atraso médio (dias)	Desvio padrão (dias)	Tempo médio (s)
1	219,81	16,128	13,72
2	149,63	7,874	14,39
3	163,82	10,541	14,31
4	174,66	10,987	14,21
5	200,27	9,281	13,72
6	200,17	13,634	14,22
7	210,12	22,598	14,00
8	173,25	20,408	13,42
9	184,26	18,409	13,04
10	185,63	15,713	13,44
11	172,60	12,848	14,20
12	181,07	17,469	13,50
13	188,77	22,496	12,72
14	205,98	17,711	13,90
15	210,95	16,327	13,66
16	233,89	14,220	13,46
17	195,72	16,478	14,09
18	187,73	19,794	12,58
19	201,04	21,461	13,63
20	196,08	15,878	14,37
21	197,79	22,674	14,54
22	212,43	21,537	14,40
23	213,78	13,509	14,99
24	236,31	20,318	13,91
25	223,99	26,904	14,88
26	190,76	13,610	19,04
27	211,08	17,509	18,92
28	218,48	19,239	18,70

Fonte: Autoria própria (2023).

Para o estudo em questão, a segunda combinação demonstrou o melhor desempenho nos experimentos de calibração, conforme observa-se na Tabela 3, apresentando um atraso médio de 149,63 dias e um desvio padrão de 7,874. Os valores dos parâmetros utilizados nessa combinação envolvem uma probabilidade de 0,5 para a execução das três estratégias de vizinhança, um número de *swap* intra igual a quantidade de máquinas da instância denotada por Q_m , uma quantidade de *swap* inter e *insertion* igual a $Q_m/12$ e uma condição de retorno à melhor solução definida como $f(s) > 1,20 \cdot f(s_{best})$. Essa última indica que se o atraso total da solução corrente for maior em 20% com relação a melhor solução, então deve-se retornar a melhor solução como a solução atual.

5.2 RESULTADOS COMPUTACIONAIS

Esta seção apresenta os resultados obtidos com a implementação da abordagem heurística em VBA nas instâncias pertencentes a empresa de tecelagem. Os testes experimentais foram executados em um computador equipado com um processador Intel Core i5 de 2,3 GHz, 8 GB de memória RAM e com um sistema operacional Windows 10.

5.2.1 Instâncias do problema

A abordagem heurística de busca local foi empregada em oito instâncias fornecidas pela empresa de tecelagem em estudo. As informações dessas instâncias estão apresentadas na Tabela 4.

Tabela 4 – Dados das instâncias

Linha de produção	Nº de máquinas	Nº de tarefas/ mês			
		Junho	Julho	Agosto	Setembro
Linha A	48	240	183	264	297
Linha B	64	319	372	344	415

Fonte: Autoria própria (2023).

A partir das informações expostas na Tabela 4, observa-se que cada instância é determinada por sua linha de produção, a qual possui uma quantidade específica de máquinas disponíveis, e pelas quantidades de tarefas, que irão depender do mês em que se deseja realizar o sequenciamento.

No presente estudo utiliza-se nos experimentos os dados relativos aos meses de junho, julho, agosto e setembro de 2023, resultando na criação de quatro instâncias para cada linha de produção, uma vez que cada mês envolve quantidades diferentes de tarefas, cada uma com características distintas, como tempos de processamento, datas de entregas e *release dates*. Assim, as quatro primeiras instâncias correspondem à linha de produção A, que abrange 48 teares, enquanto as quatro subsequentes pertencem à linha B, composta por 64 teares. As oito instâncias podem ser visualizadas na Tabela 5.

Tabela 5 – Instâncias do problema

Instância	Nº de máquinas	Nº de tarefas
A1	48	240
A2	48	183
A3	48	264
A4	48	297
B1	64	319
B2	64	372
B3	64	344
B4	64	415

Fonte: Autoria própria (2023).

Conduziu-se 10 execuções para cada uma das oito instâncias, totalizando 80 experimentos. Em cada execução, foi estabelecido um limite máximo de 100.000 iterações dentro do *loop* de busca local, visando obter uma boa solução. Nas próximas subseções serão apresentados os resultados e as suas respectivas análises para as instâncias das duas linhas de produção.

5.2.2 Resultados experimentais para as instâncias da linha A

Na Tabela 6 consta os resultados das 10 execuções para a instância A1, contendo 48 máquinas e 240 tarefas referentes ao mês de junho. Os valores exibidos na Tabela 6 incluem os atrasos totais em dias indicado por $\sum T_i$, o tempo de execução, a média e o desvio padrão.

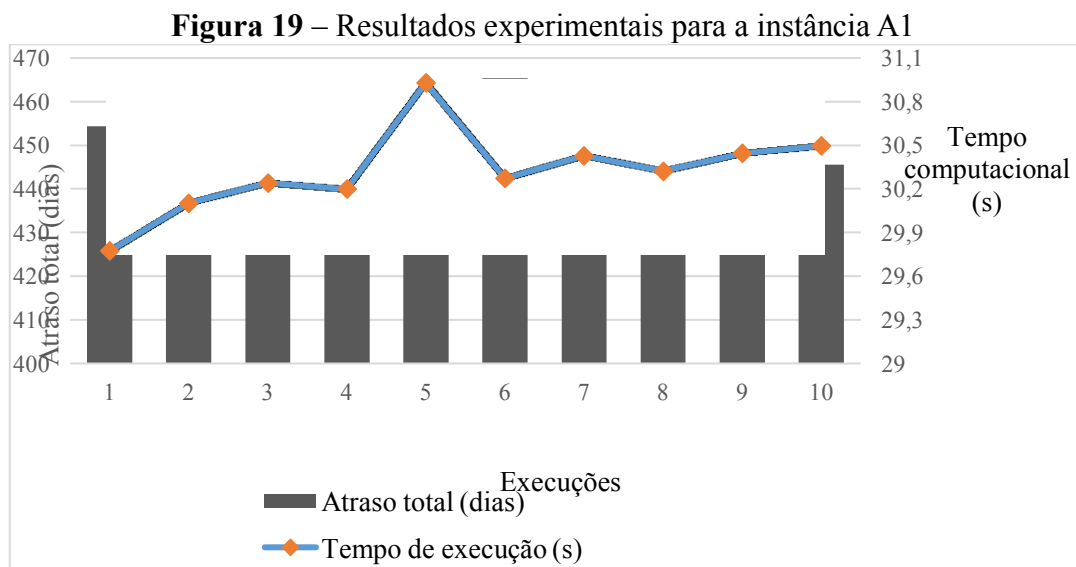
Tabela 6 – Resultados experimentais para a instância A1

Execução	$\sum T_i$ (dias)	Tempo de execução (s)
1	454,4	29,773
2	450,0	30,101
3	438,0	30,242
4	430,2	30,199
5	434,7	30,929
6	465,4	30,273
7	437,2	30,429
8	444,8	30,321
9	461,7	30,445
10	445,5	30,496
Média	446,2	30,321
Desvio Padrão	11,688	0,298

Fonte: Autoria própria (2023).

A partir da Tabela 6, constata-se que a média dos atrasos obtida considerando todas as execuções, corresponde a 446,2 dias e o tempo médio de execução da heurística para essa instância é de 30,321 segundos. Além disso, observa-se que os desvios padrões para os atrasos totais e os tempos de execução são respectivamente 11,688 e 0,298.

A partir desses valores, nota-se que o desvio padrão dos atrasos totais apresenta uma maior variabilidade, indicando que a dispersão dos atrasos entre as execuções com relação a média é alta. A Figura 19 oferece uma melhor visualização dos dados expostos da Tabela 6.



Fonte: Autoria própria (2023).

Dentre as 10 execuções, a quarta se destacou com o menor atraso total, registrando 430,2 dias. Isso representa uma diferença significativa de 35,2 dias em relação à sexta execução, que detém o maior atraso observado, além de uma diferença de 16 dias em relação à média geral dos atrasos.

No que diz respeito aos tempos de execução, observa-se que não houve variações significativas. Esse fato se confirma pelo baixo desvio padrão de 0,298, conforme apresentado na Tabela 6. Além disso, nota-se a partir da Figura 19, que a primeira execução obteve o menor tempo de execução, registrando 29,773 segundos, enquanto que a quinta apresentou o maior tempo de 30,929 segundos, representando uma diferença de apenas 1,156 segundos.

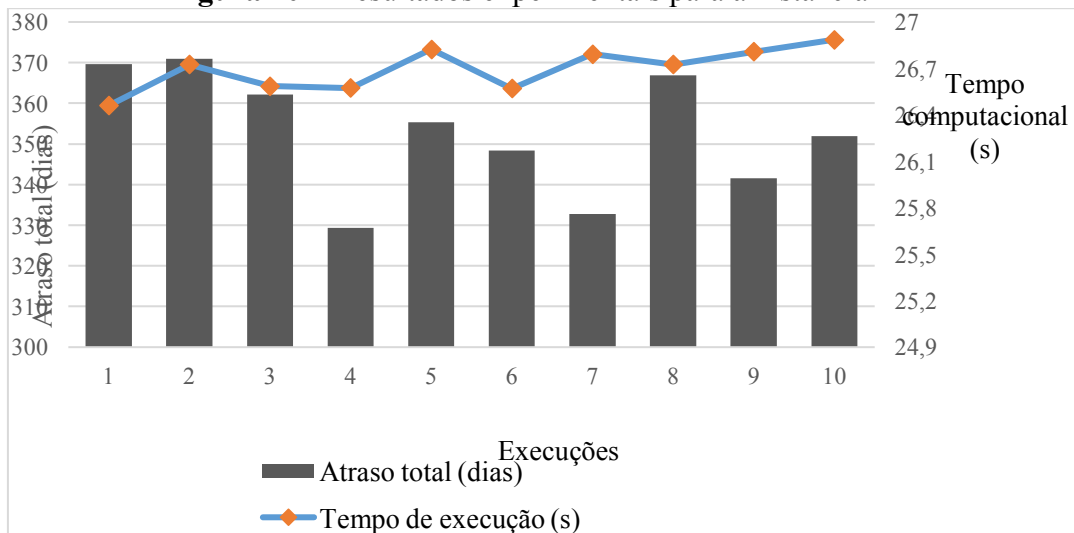
Os resultados da segunda instância da linha A, que compreende 48 máquinas e 183 tarefas para o mês de julho, são apresentados na Tabela 7.

Tabela 7 – Resultados experimentais para a instância A2

Execução	$\sum T_i$ (dias)	Tempo de execução (s)
1	369,6	26,460
2	371,0	26,726
3	362,1	26,585
4	329,3	26,574
5	355,3	26,824
6	348,4	26,570
7	332,8	26,792
8	366,9	26,726
9	341,5	26,808
10	351,9	26,886
Média	352,8	26,695
Desvio Padrão	14,926	0,139

Fonte: Autoria própria (2023).

Conforme mostra a Tabela 7, a média dos atrasos totais nas 10 execuções corresponde a 352,8 dias, enquanto que o tempo computacional médio é de 26,695 segundos. Nota-se, que os tempos de execuções foram menores em relação à primeira instância, devido à menor quantidade de tarefas para a instância A2. Ao considerar o desvio padrão para esses dois parâmetros, observa-se uma alta variação para o atraso total, com um desvio de 14,926 e uma baixa variação para o tempo de execução, com desvio de 0,139. A partir da Figura 20 torna-se evidente a variação dos atrasos entre as execuções.

Figura 20 – Resultados experimentais para a instância A2

Fonte: Autoria própria (2023).

A quarta execução apresentou o menor atraso, totalizando 329,3 dias, o que representa uma diferença de 41,7 dias em comparação com o maior atraso, registrado na segunda execução, e uma diferença de 23,58 dias em relação à média das execuções. Os resultados com relação aos tempos de execução foram bem similares, com uma variação mínima entre os valores, como observa-se na Figura 20. Tal fato é mais evidenciado pela diferença de 0,426 segundos entre o menor tempo obtido na primeira execução, correspondente a 26,460 segundos, e o maior tempo, que foi de 26,886 segundos e encontra-se quinta execução.

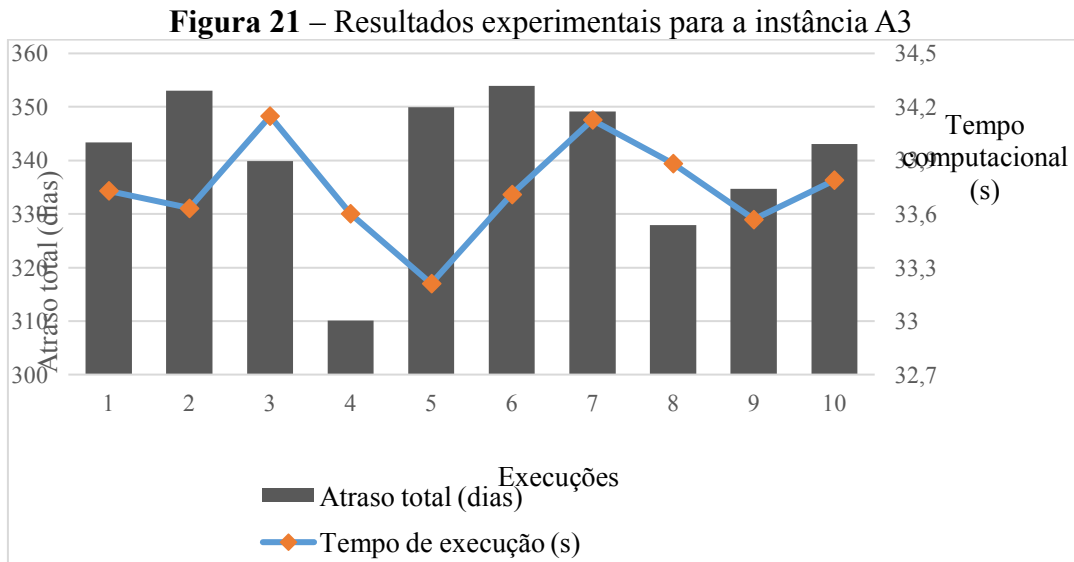
A Tabela 8 exibe os resultados acerca do atraso total e tempos de execução referentes aos experimentos realizados na instância A3, composta por 48 máquinas e 264 tarefas a serem processadas.

Tabela 8 – Resultados experimentais para a instância A3

Execução	ΣT_i (dias)	Tempo de execução (s)
1	343,4	33,730
2	353,0	33,632
3	339,9	34,148
4	310,1	33,601
5	349,9	33,210
6	353,9	33,710
7	349,1	34,128
8	327,9	33,882
9	334,7	33,570
10	343,1	33,789
Média	340,5	33,740
Desvio padrão	13,459	0,275

Fonte: Autoria própria (2023).

De acordo com os dados da Tabela 8, a média dos atrasos totais é de 340,5 dias e o tempo computacional médio é de 33,740 segundos. Ao examinar o desvio padrão, destaca-se uma variação significativa, especialmente com relação ao atraso total, que apresenta um desvio de 13,459. Por outro lado, o tempo de execução apresentou um baixo desvio de apenas 0,275, indicando pouca variabilidade em torno da média. A Figura 21 ilustra os dados acerca das execuções, a partir dela é possível perceber as execuções que apresentam os maiores atrasos bem como os menores atrasos totais.



A quarta execução destacou-se com a solução de menor atraso, totalizando 310,1 dias em um tempo computacional de 33,601 segundos. Ao comparar o valor do menor atraso com o maior atraso de 353,9 dias, encontrado na sexta execução, nota-se uma diferença de 43,8 dias. Quando se compara a melhor solução com a média dos atrasos, a diferença passa a ser de 30,4 dias, que apesar de ser um valor menor é altamente significativo para a escolha de um sequenciamento que atenda às necessidades da empresa e do cliente.

Com relação aos tempos de execução, o maior tempo foi registrado na terceira execução, com 34,148 segundos, enquanto o menor tempo foi obtido na quinta execução, com 33,210 segundos. A diferença entre esses dois valores é de 0,938 segundos, não representando uma variação de grande impacto.

Por fim, na Tabela 9, apresentam-se os resultados experimentais referentes à última instância da linha de produção A, que abrange 48 máquinas e 297 tarefas, representando a maior instância da linha.

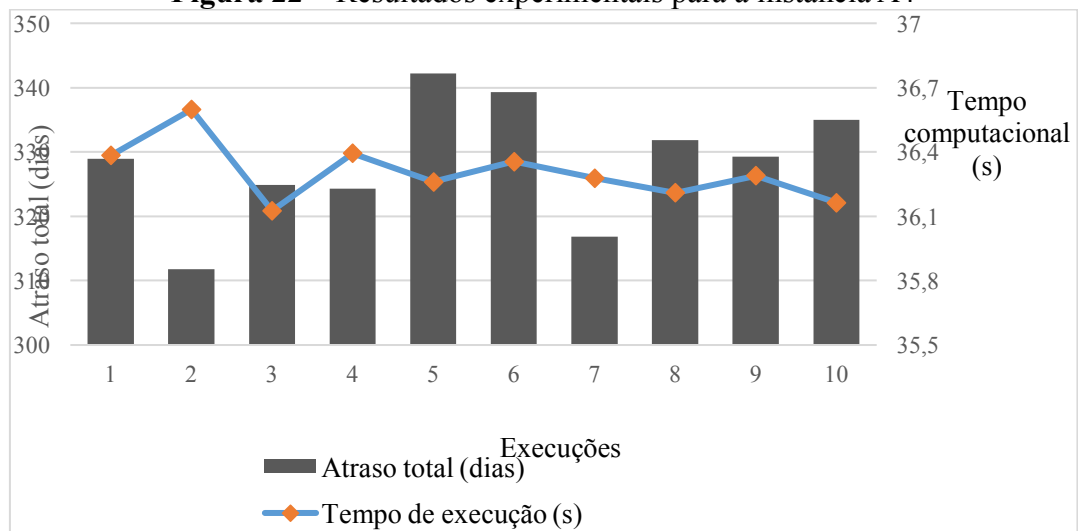
O aumento no número de tarefas tem um impacto direto no tempo de execução dos experimentos, uma vez que o algoritmo despende de mais tempo para coletar as informações da instância e para alocar cada tarefa em máquinas específicas. Essa influência é comprovada ao observar a terceira coluna da Tabela 9, que corresponde aos tempos de execuções e que quando comparada com as instâncias anteriores da linha A apresenta os maiores tempos computacionais.

Tabela 9 – Resultados experimentais para a instância A4

Execução	$\sum T_i$ (dias)	Tempo de execução (s)
1	328,9	36,384
2	311,8	36,598
3	324,9	36,126
4	324,3	36,394
5	342,2	36,260
6	339,3	36,355
7	316,8	36,278
8	331,8	36,210
9	329,3	36,289
10	335,0	36,164
Média	328,4	36,306
Desvio padrão	9,455	0,136

Fonte: Autoria própria (2023).

Os dados apresentados na Tabela 9 mostram que, ao considerar todas as 10 execuções, a média dos atrasos totais foi de 328,4 dias e o tempo médio de execução da heurística para essa instância totalizou 36,306 segundos. Além disso, destaca-se que os desvios padrões para os atrasos totais e os tempos de execução foram de 9,455 e 0,136, respectivamente, sendo considerados os menores desvios das instâncias da linha A. A Figura 22 apresenta de forma mais clara os resultados da última instância dessa linha.

Figura 22 – Resultados experimentais para a instância A4

Fonte: Autoria própria (2023).

A partir da Figura 22, é possível observar que a segunda execução obteve a melhor solução quando se trata dos atrasos totais, correspondendo a 311,8 dias, com o maior tempo de execução equivalente a 36,598 segundos. Ao comparar esse atraso com o maior atraso obtido, percebe-se que a diferença é de 30,4 dias, essa diferença torna-se menor quando comparada com a média geral dos atrasos, sendo igual a 16,6 dias.

Quanto aos tempos computacionais para a obtenção das soluções, o valor mais baixo foi de 36,126 segundos, ocorrendo na terceira execução. A partir da Figura 22, percebe-se que há uma baixa variabilidade entre os dados de tempo, de forma que 90% das execuções ocorreram entre 36,1 e 36,4 segundos. A variação entre os tempos é tão pequena que ao comparar o menor tempo com o maior, a diferença não ultrapassa 0,5 segundos, indicando boa estabilidade no desempenho do algoritmo em termos de tempos de execuções.

5.2.3 Resultados experimentais para as instâncias da linha B

A primeira instância da linha B, é composta por 64 máquinas e 319 tarefas, representando a menor instância dessa linha de produção. O resultado das execuções acerca dessa instância pode ser visualizado na Tabela 10.

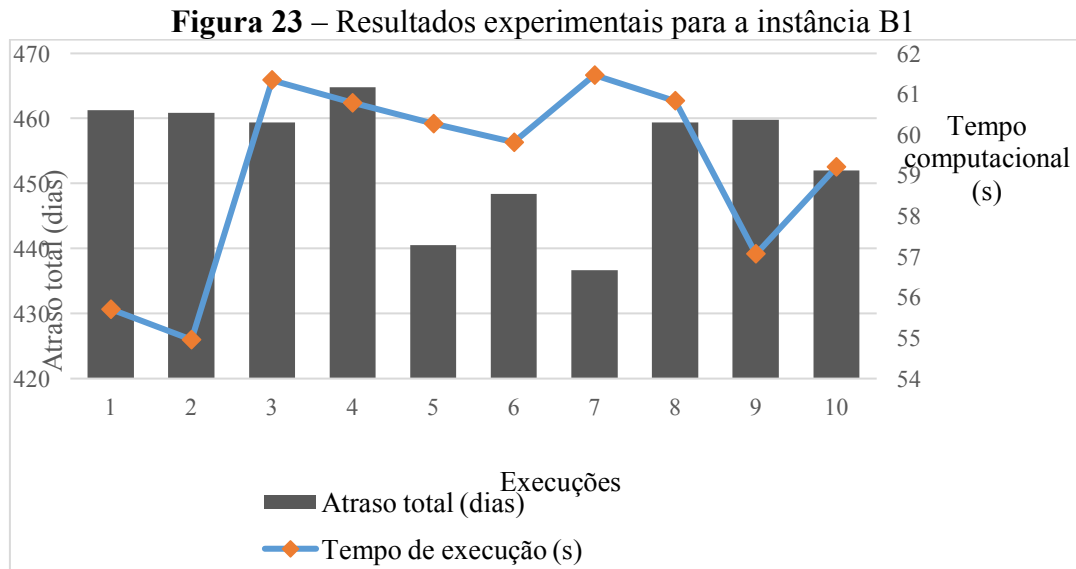
Tabela 10 – Resultados experimentais para a instância B1

Execução	$\sum T_i$ (dias)	Tempo de execução (s)
1	461,3	55,703
2	460,9	54,957
3	459,4	61,343
4	464,8	60,781
5	440,5	60,269
6	448,4	59,812
7	436,7	61,468
8	459,4	60,828
9	459,8	57,062
10	452,0	59,205
Média	454,3	59,143
Desvio padrão	9,576	2,384

Fonte: Autoria própria (2023).

Conforme apresentado na Tabela 10, a média com relação aos atrasos totais e tempos de execução são 454,3 dias e 59,143 segundos, respectivamente. As execuções realizadas para essa

instância obtiveram um desvio padrão considerável, especialmente quando se trata do tempo de execução computacional, que atingiu um desvio correspondente a 2,384, enquanto o desvio do atraso total foi de 9,576. A Figura 23 permite uma melhor análise dos dados de execução dessa instância.



Observa-se, a partir da Figura 23, que a sétima execução apresentou a solução com o menor atraso total, correspondente a 436,7 dias, no entanto, a mesma obteve o maior tempo computacional de 61,468 segundos. Quando comparada com a solução de maior atraso que está registrada na quarta execução, nota-se uma discrepância superior a 28 dias. Ao realizar essa diferença com a média geral, o valor reduz para 17,62 dias.

Em relação aos tempos computacionais, o menor valor foi de 54,957 segundos, apresentando uma diferença de 6,511 segundos ao realizar o comparativo com o maior tempo registrado, e 4,186 segundos quando comparado com a média. Essa variação entre os tempos é maior em comparação com as execuções da linha A, no entanto, torna-se mais frequente na linha B, como mostra a Figura 23.

A Tabela 11 apresenta os resultados das 10 execuções referentes à instância B2, que envolve 64 máquinas e 372 tarefas. De acordo com a Tabela 11, a média e o desvio obtidos para as 10 execuções são respectivamente de 503,36 dias e 12,493 com relação ao atraso total e de 58,646 segundos e 2,212 com relação ao tempo de execução do algoritmo. Nota-se, que a dispersão dos atrasos em torno da média é significativamente alta, o mesmo pode ser observado ao considerar os tempos de execução.

Tabela 11 – Resultados experimentais para a instância B2

Execução	ΣT_i (dias)	Tempo de execução (s)
1	504,1	60,691
2	509,8	62,734
3	503,4	59,4726
4	512,3	59,097
5	515,6	56,546
6	512,8	56,441
7	485,9	57,937
8	510,6	55,332
9	501,9	58,363
10	477,2	59,847
Média	503,3	58,646
Desvio padrão	12,493	2,212

Fonte: Autoria própria (2023).

As informações presentes na Tabela 11 podem ser visualizadas com mais clareza a partir da Figura 24.



Fonte: Autoria própria (2023).

Entre todas as execuções dessa instância, a décima alcançou a solução mais eficiente em termos de atraso, totalizando 477,2 dias, essa solução foi obtida com o tempo de execução de 59,847 segundos. O maior atraso foi registrado na quinta execução, com 515,6 dias, representando uma diferença de 38,4 dias em relação a melhor solução e 12,2 dias no que diz respeito a média.

No que se refere aos tempos computacionais de execução, observa-se uma variação considerável entre as execuções, o maior valor obtido encontra-se na segunda execução, com 62,734 segundos e o menor, na oitava execução, correspondendo a 55,332 segundos. A diferença entre esses valores é de 7,402 segundos.

Na Tabela 12, estão registrados os resultados obtidos nas execuções relacionadas à instância B3, composta por 64 máquinas e 344 tarefas.

Tabela 12 – Resultados experimentais para a instância B3

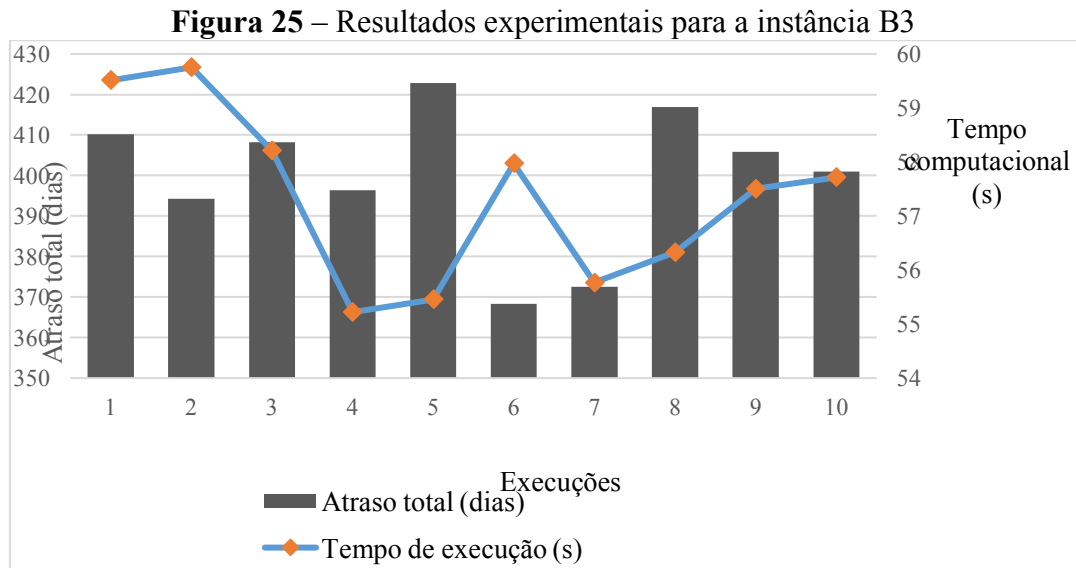
Execução	ΣT_i (dias)	Tempo de execução (s)
1	410,2	59,515
2	394,2	59,753
3	408,2	58,21
4	396,4	55,222
5	422,9	55,457
6	368,3	57,976
7	372,6	55,765
8	416,9	56,332
9	405,8	57,507
10	401,0	57,718
Média	399,6	57,346
Desvio padrão	17,692	1,612

Fonte: Autoria própria (2023).

Os dados apresentados na Tabela 12 mostram que a média dos atrasos totais e o tempo médio de execução da heurística para essa instância foi de 399,6 dias e 57,346 segundos, respectivamente. Destaca-se também a dispersão em torno dessas médias a partir dos valores do desvio padrão para os dois parâmetros citados. O desvio para o atraso foi de 17,692, indicando uma alta variabilidade entre a média e as execuções. Já para os tempos de execução, o desvio correspondeu a 1,612. A partir da Figura 25 é possível visualizar as melhores soluções com relação ao atraso total, bem como o desempenho dos tempos de execução.

Ao observar a Figura 25, nota-se que a sexta execução apresentou a solução com o menor atraso total, sendo este igual a 368,3 dias com o tempo de execução de 57,976 segundos. O maior atraso está registrado na quinta execução e corresponde a 422,9 dias, representando uma diferença de 54,6 dias com relação ao menor atraso encontrado. Quando se analisa a média e o menor atraso, essa diferença passa a ser 31,3 dias. Essas discrepâncias são altamente

significativas, especialmente ao levar em consideração a necessidade de cumprir os prazos de entrega com os clientes.



Fonte: Autoria própria (2023).

Os resultados referentes aos tempos de execução demonstraram uma certa variação. Esse fato é comprovado ao analisar a discrepância existente entre o menor valor obtido que foi de 55,222 segundos, encontrado na quarta execução, e o maior valor que está registrado na segunda execução, atingindo 59,753 segundos, a diferença chega a 4,5 segundos.

A Tabela 13 contém os resultados da última instância da linha de produção B, que compreende 64 máquinas e 415 tarefas, destacando-se como a maior configuração dentre as analisadas.

Por ser a maior instância do estudo, os tempos de execução obtidos foram os maiores, apresentando uma média de 71,551 segundos, com um desvio padrão de 0,550. Ao analisar os dados acerca do atraso total, observa-se que a instância alcançou um atraso médio de 222,8 dias com um desvio padrão de 7,497, o menor entre as oito instâncias analisadas.

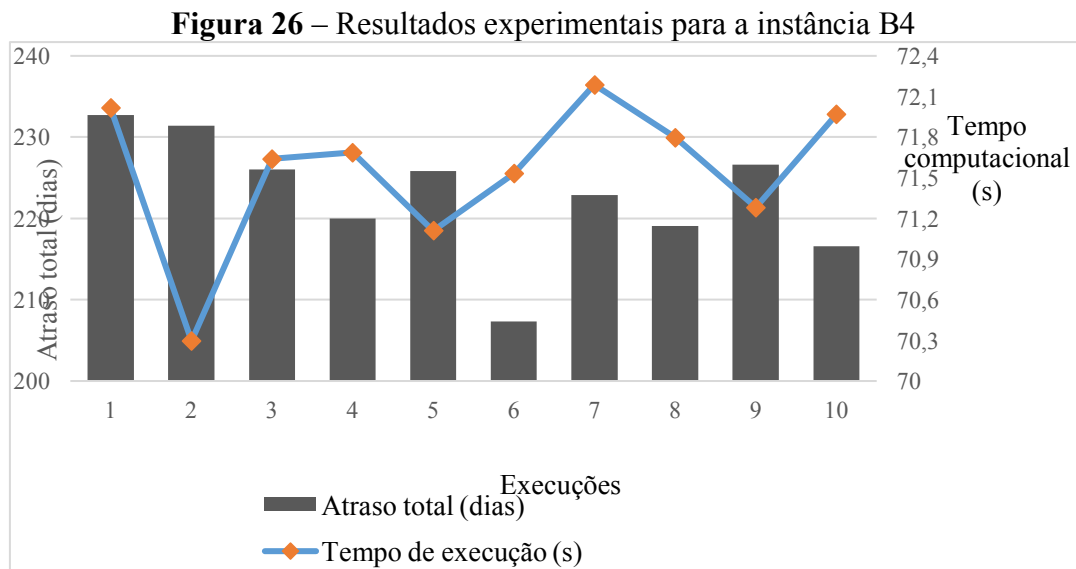
Além disso, é relevante destacar que a instância B4 apresentou as soluções com os menores atrasos em comparação com as demais instâncias. Essa distinção está intrinsecamente ligada à ordem de grandeza das informações dos pedidos vinculados a essa instância, especialmente no que diz respeito aos dados de tempos de processamento, os quais possuem valores inferiores em comparação com as outras instâncias da mesma linha de produção.

Tabela 13 – Resultados experimentais para a instância B4

Execução	ΣT_i (dias)	Tempo de execução (s)
1	232,7	72,015
2	231,4	70,296
3	226,0	71,640
4	220,0	71,687
5	225,8	71,109
6	207,3	71,531
7	222,9	72,187
8	219,1	71,796
9	226,6	71,281
10	216,6	71,968
Média	222,8	71,551
Desvio padrão	7,497	0,550

Fonte: Autoria própria (2023).

A Figura 26 permite a visualização do desempenho das soluções encontradas levando em consideração o atraso total e o tempo de execução para cada experimento.



Fonte: Autoria própria (2023).

A partir da Figura 26, constata-se que a sexta execução apresentou a melhor solução com relação aos atrasos totais, uma vez que ela demonstra um atraso de 207,3 dias, sendo considerado o menor dentre todos. Vale ressaltar, que o tempo de execução obtido para o menor atraso foi de 71,531 segundos. A primeira execução apresentou o maior atraso, com 232,7 dias,

diferindo da melhor solução em aproximadamente 25,4 dias e do atraso médio em cerca de 9,9 dias. Ao analisar a diferença entre o atraso da melhor solução e o atraso médio, o valor encontrado é de 15,5 dias.

No que diz respeito aos tempos de execução, o menor tempo foi identificado na segunda execução, com 70,296 segundos, enquanto que o maior, encontra-se na sétima execução e correspondeu a 72,187 segundos, representando uma diferença menor que 2 segundos entre os tempos dessas execuções.

5.2.4 Resumo do resultado das instâncias

Os aspectos mais significativos relacionados aos resultados dos sequenciamentos nas linhas de produção A e B, tais como os valores mínimos, média e desvio padrão, estão apresentados na Tabela 14.

Tabela 14 – Resumo das oito instâncias

Instância	$\sum T_i$ (dias)			Tempo de execução		
	Mínimo	Média	Desvio Padrão	Mínimo	Média	Desvio Padrão
A1	430,2	446,2	11,688	29,773	30,321	0,298
A2	329,3	352,8	14,926	26,460	26,695	0,139
A3	310,1	340,5	13,459	33,210	33,740	0,275
A4	311,8	328,4	9,455	36,126	36,306	0,136
B1	436,7	454,3	9,576	54,957	59,143	2,384
B2	477,2	503,3	12,493	55,332	58,646	2,212
B3	368,3	399,6	17,692	55,222	57,346	1,612
B4	207,3	222,8	7,497	70,296	71,551	0,550

Fonte: Autoria própria (2023).

Ao analisar a Tabela 14, observa-se que a heurística obteve os maiores tempos computacionais, quando executada com as informações das instâncias referentes a linha de produção B. Esse fenômeno ocorre devido ao maior número de máquinas e tarefas nessas instâncias, o que naturalmente demanda um período mais extenso para a execução do algoritmo e análise das soluções. Com relação aos desvios, nota-se que eles foram menores para a linha A, demonstrando maior consistência, uma vez que os dados possuem baixa variabilidade em torno da média.

A partir da Figura 27, torna-se possível visualizar a distribuição das 415 tarefas entre as 64 máquinas disponíveis. A sequência apresentada em cada máquina reflete precisamente a ordem em que as tarefas devem ser processadas para alcançar o atraso total de 207,3 dias, que representa em média um atraso de 0,5 dias por tarefa.

5.2.5 Simulação do método aplicado pela empresa

Para realizar análises comparativas entre a heurística de busca local desenvolvida e o método atualmente empregado pela empresa para determinar o sequenciamento, foi desenvolvido um algoritmo que simula o processo utilizado pela empresa. Esse algoritmo sequencia as tarefas seguindo o critério da data de entrega mais próxima e a ordem do pedido, além de considerar na alocação as máquinas com os menores tempos de processamento, de modo que os tempos de processamentos entre as máquinas não sejam tão discrepantes. Os resultados das simulações para as oito instâncias podem ser encontrados na Tabela 15. Dentre os resultados obtidos, observa-se que a instância B4 obteve o menor atraso e a instância B2 o maior.

Tabela 15 – Resultados da simulação do método adotado pela empresa

Instância	$\sum T_i$ (dias)
A1	585,7
A2	468,5
A3	587,0
A4	426,8
B1	522,1
B2	685,2
B3	634,2
B4	315,9

Fonte: Autoria própria (2023).

Vale ressaltar, que a determinação do sequenciamento da produção é realizada de forma manual pelo setor de PCP da empresa, por meio de planilhas eletrônicas, podendo levar horas ou até dias para ser feita, principalmente devido as dimensões das instâncias e as análises constantes para alocação. Além disso, considera-se também a possibilidade de falha humana, o que pode tornar o atraso total das tarefas ainda maior.

5.2.6 Comparação entre a heurística e o método utilizado pela empresa

Os resultados evidenciam que a Heurística de Busca Local (LS) proposta, supera em qualidade os desempenhos alcançados pelo método atualmente adotado pela empresa, como destacado na Tabela 16.

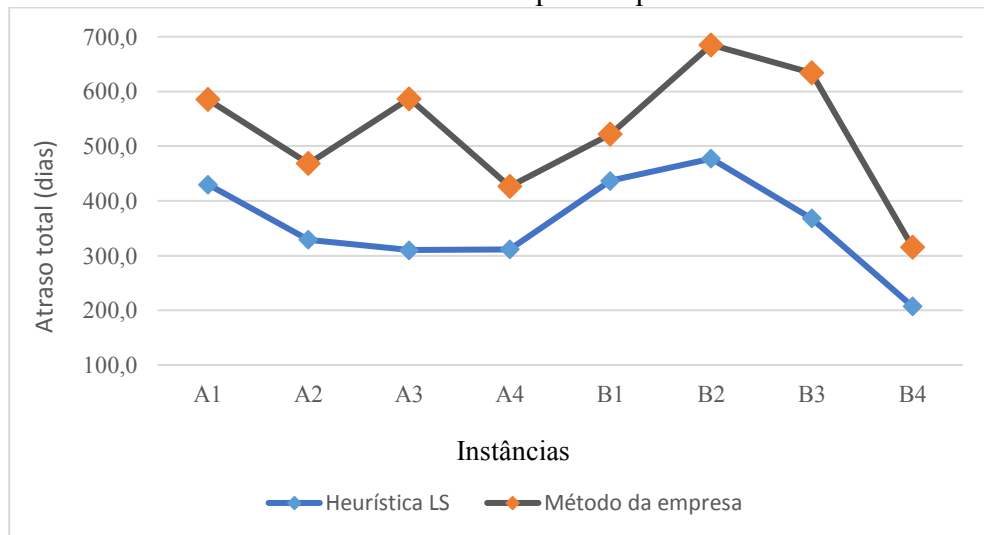
Tabela 16 – Comparativo dos resultados obtidos pela heurística proposta e o método adotado pela empresa

Instância	Heurística LS			Método da empresa	Percentual de melhora		
	$\sum T_i$ (dias)				($\%$)		
	Min	Avg	Max	-	Min	Avg	Max
A1	430,2	446,2	465,4	585,7	26,5	23,8	20,5
A2	329,3	352,8	371,0	468,5	29,7	24,7	20,8
A3	310,1	340,5	353,9	587,0	47,2	42,0	39,7
A4	311,8	328,4	342,2	426,8	26,9	23,0	19,8
B1	436,7	454,3	464,8	522,1	16,4	13,0	11,0
B2	477,2	503,3	515,6	685,2	30,4	26,5	24,8
B3	368,3	399,6	422,9	634,2	41,9	37,0	33,3
B4	207,3	222,8	232,7	315,9	34,4	29,5	26,3
Avg	358,9	381,0	396,0	528,2	31,7	27,4	24,5

Fonte: Autoria própria (2023).

A heurística LS obteve reduções significativas, principalmente com relação aos atrasos mínimos obtidos nas instâncias A3 e B3, alcançando respectivamente uma melhoria de 47,2% e 41,9%, em comparação com o método utilizado pela empresa. Ao analisar os resultados de todas as instâncias, destaca-se uma melhoria média do atraso de 27,4%. Considerando os atrasos mínimos, a melhoria média é de 31,7%, enquanto para os atrasos máximos é de 24,5%. A Figura 28 ilustra os atrasos totais mínimos alcançados com a utilização da heurística LS em comparação com o método aplicado pela empresa.

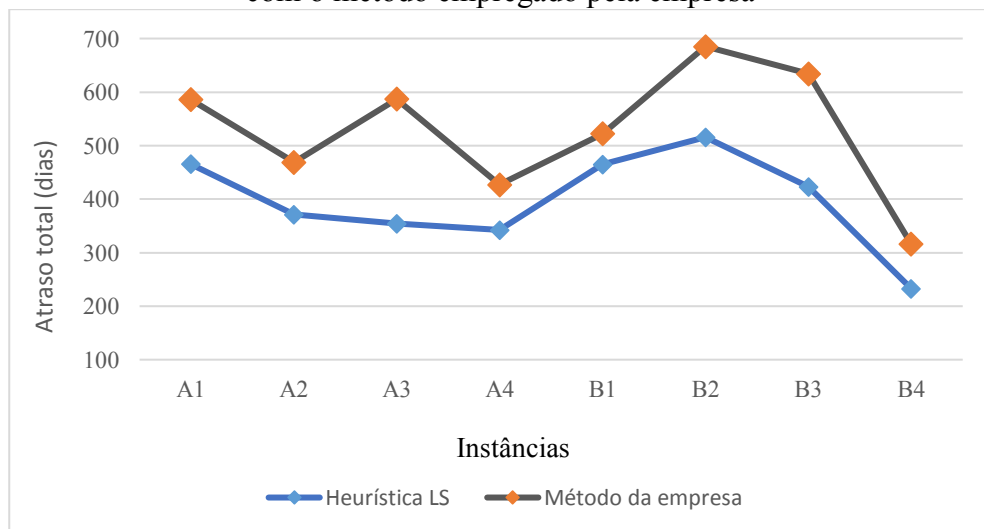
Figura 28 – Comparação entre o atraso mínimo obtido com a heurística e o atraso obtido com o método adotado pela empresa



Fonte: Autoria própria (2023).

Em todas as instâncias, a heurística LS demonstrou resultados superiores, uma vez que conseguiu encontrar soluções com atrasos inferiores ao método aplicado pela empresa. Percebe-se que a diferença entre as abordagens é mais acentuada nas instâncias A3, B2 e B3. A Figura 29 proporciona uma representação visual dos atrasos totais máximos obtidos pela heurística LS em contraste com o método utilizado pela empresa.

Figura 29 – Comparação entre o atraso máximo obtido com a heurística e o atraso obtido com o método empregado pela empresa



Fonte: Autoria própria (2023).

Embora a Figura 29 apresente o desempenho das soluções com os maiores atrasos totais obtidos pela heurística, ela ainda evidencia uma superioridade em termos de qualidade. A discrepância entre as soluções permanece significativa nas instâncias A3, B2 e B3. Por outro

lado, a instância B1 registrou um resultado próximo ao obtido pelo método empresarial, com uma diferença de 57,3 dias.

6 CONSIDERAÇÕES FINAIS

Devido à complexidade de resolução, os problemas de sequenciamento da produção tornam-se de grande interesse entre os pesquisadores e, sobretudo, nas indústrias que buscam melhorar continuamente seus processos e atender de forma satisfatória as demandas dos clientes. Entre as principais prioridades dessas organizações, destaca-se a pontualidade na entrega dos produtos, que além de minimizar os custos, fortalece a sua credibilidade no mercado. Diante disso, a necessidade de métodos de resolução eficazes e ágeis se torna ainda mais relevante. Um dos métodos que vem sendo bastante utilizados são as heurísticas, que embora não garantam a identificação da melhor solução no espaço de busca, elas se destacam por encontrar soluções viáveis e de boa qualidade.

Nesse contexto, o principal objetivo deste estudo consistiu em desenvolver uma abordagem heurística, fundamentada em técnicas de busca local, para otimizar o problema de sequenciamento de tarefas em máquinas paralelas idênticas, que se aplica ao cenário operacional de uma empresa de tecelagem. O enfoque dessa abordagem se concentrou na minimização dos atrasos totais, levando em conta a complexidade introduzida por variáveis como *release dates* distintos e tempos de *setup* que variam de acordo com a sequência de produção estabelecida.

Para isso, foram utilizadas no algoritmo três estratégias de busca local: *swap* intra máquinas, *swap* inter máquinas e *insertion*. Essas estratégias foram selecionadas com o intuito de explorar diversas possibilidades de soluções nas vizinhanças, e assim, encontrar resultados mais promissores. Além dessas estratégias, também se aplicou um mecanismo de perturbação com a finalidade de evitar a estagnação em ótimos locais e ampliar o espaço de busca.

A abordagem heurística foi aplicada em oito instâncias reais fornecidas pela empresa em estudo. As instâncias contêm informações de pedidos referentes a quatro meses de duas linhas de produção diferentes, designadas como linha A e linha B. Os resultados alcançados demonstraram que a heurística de Busca Local (LS) superou o desempenho do método utilizado pela empresa em todas as oito instâncias analisadas, apresentando uma melhoria média do atraso total de 27,4%. Ao considerar as melhores soluções obtidas, a melhoria média passa a ser de 31,7%.

Além disso, destaca-se que a instância B4 apresentou o menor atraso com 207,3 dias, enquanto a linha de produção A, na instância A3 apresentou o menor atraso, correspondendo a 310,1 dias. No entanto, apesar dessas instâncias apresentarem os menores atrasos, ressalta-se que a abordagem desenvolvida conseguiu alcançar soluções viáveis e eficientes para o restante

das instâncias das duas linhas de produção, em um tempo computacional baixo, inferior a dois minutos. Portanto, a heurística de busca local proposta mostra-se uma ótima alternativa para otimizar sequenciamentos de produção para problemas semelhantes, principalmente em ambientes industriais que priorizam a pontualidade nas entregas e buscam soluções de sequenciamentos mais ágeis.

A limitação central identificada nesta pesquisa está relacionada à complexidade de encontrar na literatura estudos que abordem, de maneira integrada, as variáveis de *release dates* e tempos de *setup* nos problemas de sequenciamento, especialmente no contexto da minimização de atrasos em máquinas paralelas idênticas.

Como sugestão para trabalhos futuros, sugere-se a utilização de outras abordagens heurísticas e metaheurísticas, em especial as heurísticas híbridas, visando integrar diferentes técnicas a fim de obter uma maior eficiência na solução do problema em análise.

REFERÊNCIAS

AGUIAR, F. N. de.; HONORATO, G.D.S.C.; SANTOS, H. G.; OCHI, L. S. Metaheurística busca tabu para o problema de coloração de grafos. *In: Simpósio Brasileiro de Pesquisa Operacional (SBPO)*, 37., 2005, Gramado. **Anais[...]**, Gramado, 2005, p. 2497-2504.

ALVARENGA, F. V. de.; ROCHA, M. L. Uma Metaheurística GRASP para o problema da árvore geradora de custo mínimo com grupamentos utilizando Grafos Fuzzy. **INFOCOMP Journal of Computer Science**, v. 5, n. 1, p. 66-75, 2006.

ARMENTANO, V. A.; FRANÇA FILHO, M. F. de. Minimizing total tardiness in parallel machine scheduling with setup times: An adaptive memory-based GRASP approach. **European Journal of Operational Research**, Elsevier, v. 183, n. 1, p. 100-114, 2007. Disponível em: <<https://doi.org/10.1016/j.ejor.2006.09.077>>.

ARROYO, J. E. C. **Heurísticas e metaheurísticas para otimização combinatória multiobjetivo**. 2002. 227 f. Dissertação (Doutorado em Engenharia Elétrica) - Universidade Estadual de Campinas, São Paulo, 2002.

ASSIS, A. N. de. **Um estudo sobre estratégias de PCP e SCO e uma aplicação de um modelo para o sequenciamento de produção**. 2014. 75 f. Dissertação (Mestrado em Engenharia de Produção e Sistemas) – Universidade Católica de Goiás, Goiânia, 2014.

BEHNAMIAN, J.; ZANDIEH, M.; GHOMI, S. M. T. F. Parallel-machine scheduling problems with sequence-dependent setup times using an ACO, SA and VNS hybrid algorithm. **Expert Systems with Applications**, Elsevier, v. 36, n. 6, p. 9637-9644, 2009. Disponível em: <<https://doi.org/10.1016/j.eswa.2008.10.007>>.

BISKUP, D.; HERRMANN, J.; GUPTA, J. N. D. Scheduling identical parallel machines to minimize total tardiness. **International Journal of Production Economics**, Elsevier, v. 115, n. 1, p. 134-142, 2008. Disponível em: <<https://doi.org/10.1016/j.ejor.2005.09.038>>.

CAUCHICK-MIGUEL, P.A. (org.). **Metodologia de pesquisa em engenharia de produção e gestão de operações**. 3. ed. Rio de Janeiro: Elsevier, 2018.

CHIAVENATO, I. **Gestão da produção: uma abordagem introdutória**. 3. ed. Barueri: Manole, 2014.

COELHO, A. M. **Uma abordagem via algoritmos meméticos para a solução do problema de horário escolar**. 2006. 117 f. Dissertação (Mestrado em Modelagem Matemática e Computacional) - Centro Federal de Educação Tecnológica de Minas Gerais, Belo Horizonte, 2006.

COLHERINHAS, G. B. **Ferramenta de otimização via algoritmos genéticos com aplicações em engenharia**. 2016. 103 f. Dissertação (Mestrado em Ciências Mecânicas) – Universidade de Brasília, Brasília, 2016.

COTA, L. P. **Novos Algoritmos para o Problema de Sequenciamento em Máquinas Paralelas Não-Relacionadas com Tempos de Preparação Dependentes da Sequência**. 2014. 134 f. Dissertação (Mestrado em Ciência da Computação) - Universidade Federal de Ouro Preto, Ouro Preto, 2014.

DIAS, T. C. S. **Algoritmos heurísticos e metaheurísticas híbridas aplicadas ao planejamento de uma rede de telecomunicações com topologia anel-estrela**. 2006. 106 p. Dissertação (Mestrado em Ciências em Engenharia de Sistemas e Computação) – Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2006.

DIAZ-CAZANAS, R.; PAZ-MARTINEZ, E. M. de la.; DELGADO-SOBRINO, D. R. Algoritmo para la programación integrada producción-mantenimiento preventivo en máquinas paralelas idénticas. **Ingeniería Industrial**, v. 40, n. 3, p. 260-271, 2019.

EBRAHIMINY, S. M. T.; FATEMI GHOMI, S. M. T.; KARIMI, B. Hybrid flow shop scheduling with sequence dependent family setup time and uncertain due dates. **Applied Mathematical Modelling**, Elsevier, v. 38, n. 09-10, 2013, p. 2490-2504. Disponível em: <<https://doi.org/10.1016/j.apm.2013.10.061>>.

FAÊDA, F. M. **Métodos de resolução do problema de sequenciamento em máquinas paralelas não-relacionadas com restrições de precedência e tempos de preparação**. 2015. 113 f. Dissertação (Mestrado em Ciência da Computação) – Universidade Federal de Viçosa, Viçosa, 2015.

FÉLIX, J. A. M. **Estudo e Melhoria do Sequenciamento da Produção numa Fábrica de Produção de Revestimentos em Cortiça**. 2019. 87 f. Dissertação (Mestrado Integrado em Engenharia e Gestão Industrial) - Faculdade de Engenharia da Universidade do Porto, Porto, 2019.

FERREIRA, E. O.; SILVA, G.P. O uso de técnicas de busca em vizinhança de grande porte para o problema de programação de máquinas paralelas. **Gestão da Produção, Operações e Sistemas**, v. 14, n. 5, p. 48 - 66, 2019. Disponível em: <<https://doi.org/10.15675/gepros.v14i5.2370>>.

FUCHIGAMI, H. Y.; RANGEL, S. A survey of case studies in production scheduling: Analysis and perspectives. **Journal of Computational Science**, v. 25, p. 425-436, 2018. Disponível em: <<https://doi.org/10.1016/j.jocs.2017.06.004>>.

GAREY, M. R.; JOHNSON, D. S. **Computers and intractability: A guide to the theory of np-completeness**. San Francisco: W.H. Freeman & Co, 1979.

GHAREHGOZLI, A. H.; TAVAKKOLI-MOGHADDAM, R.; ZAERPOUR, N. A fuzzy-mixed-integer goal programming model for a parallel-machine scheduling problem with sequence-dependent setup times and release dates. **Robotics and Computer-Integrated Manufacturing**, Elsevier, v. 25, n. 4-5, p. 853-859, 2009. Disponível em: <<https://doi.org/10.1016/j.rcim.2008.12.005>>.

GIL, A. C. **Como elaborar projetos de pesquisa**. 6. ed. São Paulo: Atlas, 2017.

GOLDBARG, M. C.; LUNA, H. P. L. **Otimização combinatória e programação linear: modelos e algoritmos**. 2. ed. Rio de Janeiro: Elsevier, 2005.

GOMES JÚNIOR, A. C. **Problema de Sequenciamento em uma Máquina com Penalidades por Antecipação e Atraso: Modelagem e Resolução**. 2007. 86 f. Dissertação (Mestrado em Engenharia de Produção) - Universidade Federal de Minas Gerais, Belo Horizonte, 2007.

GUERRINI, F. M.; BELHOT, R. V.; AZZOLINI JÚNIOR, W. **Planejamento e controle da produção: projeto e operação de sistemas**. 1. ed. Rio de Janeiro: Elsevier, 2014.

HADDAD, M. N. **Algoritmos heurísticos híbridos para o problema de sequenciamento em máquinas paralelas não-relacionadas com tempos de preparação dependentes da sequência**. 2012. 89 f. Dissertação (Mestrado em Ciência da Computação) - Universidade Federal de Ouro Preto, Ouro Preto, 2012.

HARIFI, S.; MOHAMMADZADEH, J.; KHALILIAN, M.; EBRAHIMNEJAD, S. Giza Pyramids Construction: an ancient-inspired metaheuristic algorithm for optimization. **Evolutionary Intelligence**, Springer, v. 14, p. 1743-1761, 2020. Disponível em: <<https://doi.org/10.1007/s12065-020-00451-3>>.

JACOBS, R. F.; BERRY, W. L.; WHYBARK, D. C.; VOLLMANN, T. E. **Manufacturing Planning and Control for Supply Chain Management: the CPIM reference**. 2. ed. McGraw-Hill Education, 2018.

KAABI, J.; HARRATH, Y. A survey of parallel machine scheduling under availability constraints. **International Journal of Computer and Information Technology**, v. 3, n. 2, p. 238-245, 2014.

KAVEH, A.; KHAYATAZAD, M. A new meta heuristic method: Ray Optimization. **Computers and Structures**, Elsevier, p. 283–294, 2012.

KIM, J. G.; SONG, S.; JEONG, B. Minimizing total tardiness for the identical parallel machine scheduling problem with splitting jobs and sequence-dependent setup times. **International Journal of Production Research**, Taylor & Francis, v. 58, n. 6, p. 1628-1643, 2019. Disponível em: <<https://doi.org/10.1080/00207543.2019.1672900>>.

LEE, C. H. A dispatching rule and a random iterated greedy metaheuristic for identical parallel machine scheduling to minimize total tardiness. **International Journal of Production Research**, Taylor & Francis, v. 56, n. 6, p. 2292-2308, 2017. Disponível em: <<https://doi.org/10.1080/00207543.2017.1374571>>.

LEE, J. Y.; KIM, Y. D. A branch and bound algorithm to minimize total tardiness of jobs in a two identical-parallel-machine scheduling problem with a machine availability constraint. **Journal of the Operational Research Society**, Taylor & Francis, v. 66, n. 9, p. 1542-1554, 2015. Disponível em: <<https://doi.org/10.1057/jors.2014.122>>.

LIN, S. W.; LEE, Z. J.; YING, K. C.; LU, C. C. Minimization of maximum lateness on parallel machines with sequence-dependent setup times and job release dates. **Computers & Operations Research**, Elsevier, v. 38, n. 5, p. 809-815, 2011. Disponível em: <<https://doi.org/10.1016/j.cor.2010.09.020>>.

LOPES, J. D. S. **Análise e otimização do sequenciamento de produção de uma empresa de médio porte de embalagens plástica**. 2008. 49 f. Monografia (Bacharelado em Engenharia de Produção) – Universidade Federal de Juiz de Fora, Juiz de Fora, 2008.

LOPES, L. H. **Heurística híbrida para sequenciamento de tarefas em Máquinas paralelas heterogêneas**. 2022. 28 f. Monografia (Bacharelado em Ciências e Tecnologia) – Universidade Federal de Santa Catarina, Joinville, 2022.

MARQUES, T. B.; SIQUEIRA, L. S.; ZACARIAS, R. O. Busca local com permute all pairs aplicada ao problema de alocação de facilidades. In: Computer on the Beach, 8., 2017, Florianópolis. **Anais [...]**, Florianópolis, 2017, p. 446-455.

MELLOULI, R.; SADFI, C.; CHU, C.; KACEM, I. Identical parallel-machine scheduling under availability constraints to minimize the sum of completion times. **European Journal of Operational Research**, Elsevier, v. 197, n. 3, p. 1150-1165, 2009. Disponível em: <<https://doi.org/10.1016/j.ejor.2008.03.043>>.

MENEZES, W. W. D. **Metaheurísticas GRASP e ILS aplicadas ao problema da variabilidade do tempo de resposta**. 2014. 63 f. Dissertação (Mestrado em Informática) - Universidade Federal da Paraíba, João Pessoa, 2014.

MENSENDIEK, A.; GUPTA, J. N. D.; HERRMANN, J. Scheduling identical parallel machines with fixed delivery dates to minimize total tardiness. **European Journal of Operational Research**, Elsevier, v. 243, n. 2, p. 514-522, 2014. Disponível em: <<https://doi.org/10.1016/j.ejor.2014.12.002>>.

MLADENOVIC, N.; HANSEN, P. Variable neighborhood search. **Computers & operations research**, v. 24, n. 11, p. 1097-1100, 1997.

MÜLLER, F. M.; DIAS, O. B.; DE ARAÚJO, O. C. B. Algoritmo para o problema de sequenciamento em máquinas paralelas não-relacionadas. **Production**, v. 12, p. 6-17, 2002.

MUNDIM, L. R.; FUCHIGAMI, H. Y. Uma heurística robusta para programação de máquinas paralelas com tempos de setup dependentes da sequência. **Revista Produção Online**, v. 17, n. 2, p. 463-481, 2017. Disponível em: <<https://doi.org/10.14488/1676-1901.v17i2.2375>>.

NAJAT, A.; YUAN, C.; GURSEL, S.; TAO, Y. Minimizing the Number of Tardy Jobs on Identical Parallel Machines Subject to Periodic Maintenance. **Procedia Manufacturing**, Elsevier, v. 38, p. 1409-1416, 2019. Disponível em: <<https://doi.org/10.1016/j.promfg.2020.01.147>>.

NASCIMENTO, F. P.; SOUSA, F. L. L. **Metodologia da Pesquisa Científica: teoria e prática – como elaborar TCC**. Brasília: Thesaurus, 2016.

NESMACHNOW, S. An overview of metaheuristics: accurate and efficient methods for optimisation. **International Journal of Metaheuristics**, v. 3, n. 4, p. 320-347, 2014.

OLIVEIRA, M. M. C. de. **Reformulações para o problema integrado de dimensionamento e sequenciamento da produção**. 2015. 93 f. Dissertação (Doutorado em Matemática) – Universidade Estadual Paulista, São José do Rio Preto, 2015.

PEREIRA, A. A. S. **Metaheurísticas para o problema de flowshop flexível com penalidades de adiantamento e atraso**. 2011. 71 f. Dissertação (Mestrado em Ciência da Computação) – Universidade Federal de Viçosa, Viçosa, 2011.

PEREIRA, L. M. **Análise de estruturas de vizinhança para o problema de sequenciamento de máquinas paralelas não relacionadas com tempos de preparação**. 2019. 78 f. Dissertação (Mestrado em Engenharia Elétrica) – Universidade Federal de Minas Gerais, Belo Horizonte, 2019.

PEREIRA, S. M. **Aplicação de Simulação para a resolução de um problema de escalonamento da produção em máquinas paralelas e redução do tempo total de processamento**. 2022. 113 f. Dissertação (Mestrado em Engenharia Industrial) - Universidade do Minho, Portugal, 2022.

PINEDO, M. L. **Scheduling: Theory, Algorithms, and Systems**. 5. ed. New York: Springer, 2016.

PINTO, A. F. **Uma heurística baseada em busca local iterada para o problema de clusterização de módulos de software**. 2014. 96 f. Dissertação (Mestrado em Informática) – Universidade Federal do Estado do Rio de Janeiro, Rio de Janeiro, 2014.

PRODANOV, C. C.; FREITAS, E. C. de. **Metodologia do trabalho científico: métodos e técnicas da pesquisa e do trabalho acadêmico**. 2. ed. Novo Hamburgo: Feevale, 2013.

RIBEIRO, K. D. O. **Algoritmos para o problema de sequenciamento de tarefas em máquinas paralelas idênticas**. 2020. 60 f. Dissertação (Mestrado em Instrumentação, Controle e Automação de Processos de Mineração) – Universidade Federal de Ouro Preto, Ouro Preto, 2020.

ROCHOLL, J.; MÖNCH, L. Decomposition heuristics for parallel machine multiple orders per job scheduling problems with a common due date. **Journal of the Operational Research Society**, Taylor & Francis, v. 72, n. 8, p. 1737-1753, 2019. Disponível em: <<https://doi.org/10.1080/01605682.2019.1640589>>.

SÁENZ-ALANÍS, C. A.; DEVASSIA, J. V.; SALAZAR-AGUILAR, M. A.; BOYER, V. A parallel machine batch scheduling problem in a brewing Company. **The International Journal of Advanced Manufacturing Technology**, Springer, v. 87, p. 65-75, 2016. Disponível em: <<https://doi.org/10.1007/s00170-016-8477-8>>.

SANTOS, V. L. A. **Sequenciamento de tarefas em máquinas paralelas com desgastes dependentes da sequência: resolução heurística**. 2016, 92 f. Dissertação (Mestrado em Ciência da Computação) - Universidade Federal de Viçosa, Minas Gerais, 2016.

SHIM, S. O.; KIM, Y. D. A branch and bound algorithm for an identical parallel machine scheduling problem with a job splitting property. **Computers & Operations Research**,

Elsevier, v. 35, n. 3, p. 863-875, 2008. Disponível em: <<https://doi.org/10.1016/j.cor.2006.04.006>>.

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

SILVA, J. M. P. **Abordagens exata e heurística para o problema de escalonamento em máquinas paralelas idênticas com servidor único de setup**. 2019. 79 f. Dissertação (Mestrado em Engenharia de Produção) – Universidade Federal da Paraíba, João Pessoa, 2019.

SILVA, P. P. B.; RIVEROS, D. P. B.; BOLÍVAR, J. E. B. Aplicación de una heurística constructiva en programación secuencial para asignación de varios trabajos a varias máquinas en paralelo. **Scientia et Technica**, v. 18, n. 1, p. 120-125, 2013.

SLACK, N.; CHAMBERS, S.; JOHNSTON, R. **Administração de produção**. 3. ed. São Paulo: Atlas, 2009.

TALBI, E.G. **Metaheuristics: from design to implementation**. John Wiley & Sons, 2009.

TANAKA, S.; ARAKI, M. A branch-and-bound algorithm with Lagrangian relaxation to minimize total tardiness on identical parallel machines. **International Journal of Production Economics**, Elsevier, v. 113, n. 1, p. 446-458, 2008. Disponível em: <<https://doi.org/10.1016/j.ijpe.2007.10.006>>.

TOCHA, C.A. **Desenvolvimento de ferramentas computacionais para o sequenciamento da produção**. 2014. 109 f. Dissertação (Mestrado em Engenharia de Produção) - Universidade Federal do Paraná, Curitiba, 2014.

TUBINO, D. F. **Planejamento e controle da produção: teoria e prática**. 3. ed. São Paulo: Atlas, 2017.

TURKER, A. K.; SEL, C. Scheduling Two Parallel Machines with Sequence dependent Setups and A Single Server. **Gazi University Journal of Science**, v. 24, n. 1, p. 113-123, 2011.

VIVIAN, C. J. **Aplicação do método simulated annealing em um problema real de sequenciamento da produção**. 2010. 128 f. Dissertação (Mestrado em Ciências). Programa de Pós-Graduação em Métodos Numéricos em Engenharia - Universidade Federal do Paraná, Curitiba, 2010.

YALAOUI, F. Minimizing total tardiness in parallel-machine scheduling with release dates. **International Journal of Applied Evolutionary Computation**, v. 3, n. 1, p. 21-46, 2012.

ZANELLA, L. C. H. **Metodologia da pesquisa**. 2. ed. Santa Catarina: UFSC, 2013.