



UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS
DEPARTAMENTO DE ENGENHARIA DE PRODUÇÃO
CURSO DE GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

LUISA FRAZÃO PRIMO

**ALGORITMOS HEURÍSTICOS E MODELAGENS MATEMÁTICAS PARA
OTIMIZAÇÃO DE SEQUENCIAMENTO EM AMBIENTES PERMUTATION FLOW
SHOP**

Recife

2024

LUISA FRAZÃO PRIMO

**ALGORITMOS HEURÍSTICOS E MODELAGENS MATEMÁTICAS PARA
OTIMIZAÇÃO DE SEQUENCIAMENTO EM AMBIENTES PERMUTATION FLOW
SHOP**

Trabalho de Conclusão de Curso
apresentado ao Curso de Graduação em
Engenharia de Produção da Universidade
Federal de Pernambuco, como requisito
parcial para obtenção do título de
Engenheira de Produção.

Orientador: Prof. Dr. Raphael Harry Frederico Ribeiro Kramer

Recife
2024

Ficha de identificação da obra elaborada pelo autor,
através do programa de geração automática do SIB/UFPE

Primo, Luisa Frazão.

Algoritmos Heurísticos e Modelagens Matemáticas para Otimização de Sequenciamento em Ambientes Permutation Flow Shop / Luisa Frazão Primo. - Recife, 2024.

48 p. : il., tab.

Orientador(a): Raphael Harry Frederico Ribeiro Kramer

(Graduação) - Universidade Federal de Pernambuco, Centro de Tecnologia e Geociências, , 2024.

1. Otimização. 2. Problemas de Sequenciamento. 3. Heurísticas. 4. Algoritmos. 5. Planejamento e Controle da Produção. I. Kramer, Raphael Harry Frederico Ribeiro . (Orientação). II. Título.

620 CDD (22.ed.)

LUISA FRAZÃO PRIMO

**ALGORITMOS HEURÍSTICOS E MODELAGENS MATEMÁTICAS PARA
OTIMIZAÇÃO DE SEQUENCIAMENTO EM AMBIENTES PERMUTATION FLOW
SHOP**

Trabalho de Conclusão de Curso
apresentado ao Curso de Graduação em
Engenharia de Produção da Universidade
Federal de Pernambuco, como requisito
parcial para obtenção do grau de Bacharel
em Engenharia de Produção.

Aprovado em: 14/03/2024.

BANCA EXAMINADORA

Prof. Dr. Raphael Harry Frederico Ribeiro Kramer (Orientador)
Universidade Federal de Pernambuco

Prof. Dr. Alexandre Ramalho Alberti (Examinador Interno)
Universidade Federal de Pernambuco

Prof. Dr. João Mateus Marques de Santana (Examinador Interno)
Universidade Federal de Pernambuco

AGRADECIMENTOS

Agradeço, primeiramente, aos meus pais Silvana e Marcos e à minha irmã Cecília, que me apoiam desde o meu primeiro dia de vida e sem os quais eu não estaria aqui hoje. Acompanharam de perto toda a minha trajetória de vida e acadêmica, estando ao meu lado em todos os momentos. Obrigada por nunca terem soltado a minha mão.

Às minhas amigas Ana Clara, Ana Maria, Eduarda, Gabriela, Nathalia e Melina: Grupo de amigas que se formou nos anos de colégio, batizados por nós com o nome *Madness*. Obrigada por sempre me motivarem a seguir os meus sonhos e fazer o que fosse necessário para ser feliz.

Obrigada, também, aos amigos que fiz ao longo da graduação, em especial a Bruna, Cinthia, Eduarda, Everton, Gustavo, Larissa e Mariana, que foram fundamentais para que eu conseguisse enfrentar os desafios da faculdade e tantas vezes me motivaram a continuar me dedicando aos estudos.

Agradeço ao corpo docente do Departamento de Engenharia de Produção, que me acolheu nos últimos cinco anos e sempre se mostrou extremamente comprometido com a excelência acadêmica e com o desenvolvimento de seus alunos. Em particular, agradeço ao Prof. Dr. Raphael Kramer, que coorientou minha Iniciação Científica e orientou o meu Trabalho de Conclusão de Curso, sendo um dos responsáveis por despertar o meu interesse por pesquisa operacional, consequentemente me auxiliando na minha escolha de carreira.

Gratidão, também, ao meu gestor no trabalho Daniel Castro, pela sua paciência enquanto eu conciliava meus estudos e o estágio, e cujos ensinamentos foram imprescindíveis para o meu desenvolvimento profissional e acadêmico.

Por fim, gostaria de agradecer a mim mesma, que é a única pessoa que compreende plenamente o que é estar no meu próprio corpo e mente e é a única que esteve comigo em absolutamente todos os momentos. Obrigada por não ter desistido mesmo quando essa parecia ser a única opção e obrigada por estar se permitindo, depois de tanto tempo, ser feliz.

RESUMO

A eficiência operacional para a otimização de recursos e redução de perdas é um dos grandes objetivos da indústria. O Planejamento e Controle da Produção (PCP) exerce um importante papel na obtenção desse objetivo, sendo responsável por coordenar atividades dentro do ambiente produtivo visando a eficiência do sistema. Dentro do PCP, o problema de *scheduling* se mostra bastante relevante uma vez que se trata do sequenciamento eficiente de tarefas dentro de um sistema produtivo com o intuito de otimizar algum recurso. Nesse sentido, esse trabalho explorou os diferentes métodos existentes para se resolver o problema de sequenciamento eficiente de tarefas em ambiente *permutation flow shop* analisou o desempenho de diferentes abordagens. Um dos métodos mais comuns de resolução do problema de *scheduling* em ambiente produtivo *permutation flow shop* é o algoritmo NEH (Nawaz-Enscore-Ham). Para averiguar a qualidade e aplicabilidade dessa heurística, realizou-se também uma implementação desse algoritmo juntamente com a implementação de duas modelagens matemáticas do problema de *scheduling* em ambiente *flow shop* a fim de comparar os diferentes métodos de solução tanto no que se diz a respeito da solução final quanto ao tempo necessário para se gerar uma solução. Foi possível observar através do experimento proposto que a heurística construtiva em questão é capaz de fornecer resultados para o problema de *scheduling* em ambiente *permutation flow shop* em um tempo significativamente menor, sendo sua aplicação relevante em contextos que exigem uma resposta rápida, como no chão de fábrica.

Palavras-chave: Otimização; Problemas de sequenciamento; Heurísticas; Algoritmos; Planejamento e controle da produção.

ABSTRACT

Operational efficiency to optimize resources and reduce losses is one of the manufacturing industry's main objectives. Production Planning and Control (PPC) plays an important role in achieving this objective, being responsible for coordinating activities within the production environment aiming at system efficiency. Within the field of PPC, the scheduling problem is quite relevant since it involves the efficient sequencing of tasks within a production system with the aim of optimizing some resource. In this sense, the following article investigated the different existing methods to solve the problem of efficient task sequencing in a permutation flow shop environment and analyzed the performance of different approaches. One of the most common methods for solving the scheduling problem in a permutation flow shop production environment is the NEH (Nawaz-Enscore-Ham) algorithm. To verify the quality and applicability of this heuristic, an implementation of this algorithm was also carried out along with the implementation of two mathematical models of the scheduling problem in a flow shop environment in order to compare the different solution methods both in terms of final solution in terms of the time required to generate a solution. It was possible to observe through the proposed experiment that the constructive heuristic in question is capable of providing results for the scheduling problem in a permutation flow shop environment in a significantly shorter time, proving its application to be relevant in contexts that require a quick response, such as on the factory floor.

Keywords: Optimization; Sequencing problems; Heuristics; Algorithms; Planning and production control.

LISTA DE FIGURAS

Figura 1: Solução de um problema utilizando abordagem matemática	19
Figura 2: Relação entre os problemas clássicos de <i>scheduling</i>	23
Figura 3: Sequenciamento de tarefas em ambientes <i>flow shop</i> não-permutacional e <i>flow shop</i> permutacional	24

LISTA DE QUADROS

Quadro 1: Tipos de modelagem do problema de scheduling	20
Quadro 2: Lista de possíveis valores para o campo α da notação de Graham et al. (1979)	21
Quadro 3: Lista de possíveis valores para o campo β da notação de Graham et al. (1979)	21
Quadro 4: Lista de possíveis valores para o campo γ da notação de Graham et al. (1979)	22
Quadro 5: Parâmetros e variáveis de decisão dos modelos matemáticos apresentados.....	31

LISTA DE TABELAS

Tabela 1: Instância 2×4 para o problema de scheduling em ambiente flow shop ..	24
Tabela 2: Resultados obtidos para instâncias 20×5	37
Tabela 3: Resultados obtidos para instâncias 20×10	38
Tabela 4: Resultados obtidos para instâncias 20×20	39
Tabela 5: Resultados obtidos para instâncias 50×5	39
Tabela 6: Resultados obtidos para instâncias 50×10	40
Tabela 7: Resultados obtidos para instâncias 50×20	40
Tabela 8: Resultados obtidos para instâncias 100×5	41
Tabela 9: Resultados obtidos para instâncias 100×10	41
Tabela 10: Resultados obtidos para instâncias 100×20	42
Tabela 11: Resultados obtidos para instâncias 200×10	42
Tabela 12: Resultados obtidos para instâncias 200×20	43
Tabela 13: Resultados obtidos para instâncias 500×20	43

LISTA DE ABREVIATURAS E SIGLAS

NP	<i>Nondeterministic Polynomial Time</i>
PCP	Planejamento e Controle da Produção
PEPS	Primeiro a Entrar, Primeiro a Sair

SUMÁRIO

1	INTRODUÇÃO	12
1.1	JUSTIFICATIVA E RELEVÂNCIA	13
1.2	OBJETIVOS.....	15
1.2.1	Geral.....	15
1.2.2	Específicos.....	15
1.3	METODOLOGIA.....	15
1.4	ORGANIZAÇÃO DO TRABALHO	16
2	FUNDAMENTAÇÃO TEÓRICA.....	18
2.1	MODELOS MATEMÁTICOS NA PESQUISA OPERACIONAL	18
2.2	PROBLEMAS DE SEQUENCIAMENTO.....	19
2.2.1	Permutation Flow Shop.....	23
2.3	HEURÍSTICAS E METAHEURÍSTICAS	24
2.3.1	Heurística NEH.....	25
3	REVISÃO DA LITERATURA	27
4	CONTEXTO E DESCRIÇÃO DO PROBLEMA.....	30
4.1	CONTEXTO DO PROBLEMA.....	30
4.2	DESCRIÇÃO DO PROBLEMA.....	30
5	EXPERIMENTOS COMPUTACIONAIS	35
5.1	RESULTADOS OBTIDOS.....	37
6	CONCLUSÃO	45
	REFERÊNCIAS.....	46

1 INTRODUÇÃO

No contexto dinâmico e acelerado da indústria, a coordenação eficaz das atividades do processo produtivo é imprescindível para atingir a eficiência operacional e minimizar perdas. O Planejamento e Controle da Produção é um conceito intrínseco a esse processo: Uma área que abrange a alocação estratégica de recursos, o sequenciamento de tarefas e a sincronização das operações dentro do ambiente fabril (THOMAS & MCCLAIN, 1993).

Segundo Thomas e McClain (1993), o Planejamento e Controle da Produção (ou PCP) contempla uma série de atividades voltada para o controle das diferentes esferas da produção, como previsão de demanda, planejamento de qualidade, gestão de estoques, gestão da qualidade, entre outros. Nesse sentido, trata-se de uma área que tem como um de seus pilares a busca pela eficiência no processo produtivo e valoriza, assim, técnicas e conceitos que promovam isso, como as técnicas existentes para resolver problemas de *scheduling*.

O problema de *scheduling* se trata da programação eficiente de tarefas que compõem o sistema de produção para otimizar algum recurso adotado na sua função objetivo como, por exemplo, tempo (GELDERS & VAN WASSENHOVE, 1981). O problema de *scheduling* parte do princípio de que, através da ordenação estratégica de tarefas, é possível melhorar a eficiência da produção, minimizar tempo ocioso e cumprir com prazos na indústria (GELDERS & VAN WASSENHOVE, 1981). Além disso, é importante ressaltar que o problema de *scheduling* nas indústrias se trata de um problema operacional e rotineiro, sendo importante não apenas a qualidade da solução gerada, mas também o tempo hábil para gerar a solução.

Apesar da facilidade para identificar as características do problema de *scheduling*, ele se trata de um problema de alta complexidade. Os problemas de *scheduling* são típicos problemas de otimização combinatória, sendo a grande maioria deles classificados como NP-difícil. Trata-se de uma classe de problemas de cujo tempo necessário para encontrar a solução ótima (i.e., a melhor dentre o conjunto de possíveis soluções) cresce exponencialmente com o tamanho da instância (problema a ser resolvido). Assim, o tempo necessário para se encontrar uma solução ótima para esse problema pode se tornar inviável para aplicação prática, visto que o número de possíveis soluções pode assumir dimensões extremamente grandes.

1.1 JUSTIFICATIVA E RELEVÂNCIA

A complexidade dos problemas de *scheduling* pode ser fundamentada por diversos aspectos. Um exemplo disso pode ser observado ao se analisar um caso simples envolvendo uma única máquina, com tempos de setup dependentes da ordem de processamento das tarefas e com o objetivo de minimizar o tempo total de processamento (*makespan*). Resolver este problema é equivalente a resolver um Problema do Caixeiro Viajante, o qual é amplamente reconhecido como um problema NP-difícil (APPLEGATE et al, 2007).

Por essa razão, frequentemente os problemas de sequenciamento são resolvidos utilizando algoritmos. Algoritmos são ferramentas essenciais na solução de problemas de escalonamento de tarefas, existindo diferentes abordagens para otimizar sequências de tarefas e utilização de máquinas. Existem diversos métodos de solução para esses tipos de problemas, incluindo métodos exatos, algoritmos heurísticos e técnicas metaheurísticas. Esses algoritmos como algoritmos genéticos, algoritmos de busca local, entre outros, fornecem informações importantes para o desenvolvimento de estratégias de programação eficientes, podendo se adaptar a objetivos e restrições específicos (YENISEY; YAGMAHAN, 2013).

No contexto das indústrias, é crucial que as atividades do processo de produção sejam executadas de forma a otimizar a utilização de recursos e minimizar as perdas do sistema. Realizar esse feito pode se mostrar desafiador, devido às diversas restrições presentes na natureza das operações fabris, como a quantidade de máquinas disponíveis para realizar processamentos, suas capacidades de processamento, a ordem de processamento das atividades, os tempos de setup, entre outras condições.

Assim, torna-se fundamental determinar a melhor sequência possível para a alocação de tarefas dentro da fábrica, a qual deve respeitar todas as restrições do sistema e maximizar a produtividade. Essa situação caracteriza a definição do problema de *scheduling* (PINEDO, 2016).

É válido investir em pesquisas sobre esse tema uma vez que existem uma série de aplicações práticas nos quais o sequenciamento eficiente de tarefas pode levar à economia de custos, maior produtividade e maior satisfação do cliente.

Além da redução de custos, a motivação por trás da busca pela eficiência no processamento de tarefas nas indústrias também satisfaz necessidades vinculadas à sustentabilidade ambiental. Os problemas de *green shop scheduling* são extensões dos problemas tradicionais de *scheduling* cuja função objetivo ou funções objetivo visam otimizar critérios ambientais como o consumo de energia ou emissão de poluentes (LI & WANG, 2022).

Os problemas de sequenciamento de tarefas são comuns na indústria e podem ter diferentes aplicações nos diferentes ambientes produtivos, podendo adotar diferentes lógicas de sequência de processamento. Os ambientes representam as diferentes características que os sistemas produtivos podem assumir e, assim, caracterizarão diferentes tipos de problema de *scheduling*. Alguns exemplos de ambientes produtivos incluem *job shop*, *flow shop*, *permutation flow shop* e *open shop* (PINEDO, 2016).

O ambiente *flow shop* refere-se a uma configuração comum em sistemas de manufatura onde um conjunto de atividades é processado por uma sequência de máquinas em uma ordem específica, tendo assim uma ordem fixa de processamento das atividades nas máquinas. Em um ambiente *flow shop*, cada trabalho segue a mesma “rota” de máquinas, começando pela máquina 1 e seguindo para a máquina 2, e assim por diante, até a última máquina da sequência (ROSSIT et al., 2017). O modelo *flow shop* é predominante em ambientes de fabricação, linhas de montagem e instalações de serviços de informação, representando uma parcela significativa dos sistemas de produção (ROSSIT et al., 2017), justificando assim a sua relevância.

O problema de *permutation flow shop*, um caso específico do problema de *flow shop* clássico, têm sido amplamente estudados e implementados devido à quantidade de oportunidades de aplicação no mundo real da indústria (ÖZTOP et al., 2020). Trata-se do problema de *scheduling flow shop* no qual a ordem das tarefas em todas as máquinas é a mesma (KALCZYNSKI & KAMBUROWSKI, 2007). Nesse trabalho, focaremos nos problemas de *scheduling* em ambientes *permutation flow shop*.

1.2 Objetivos

Esse trabalho investigou profundamente o problema de sequenciamento de tarefas em ambientes permutation *flow shop* juntamente com os métodos existentes para resolvê-lo com o intuito de alcançar os seguintes objetivos:

1.2.1 Geral

Estudar e aplicar algoritmos heurísticos capazes de resolver de forma eficiente problemas de sequenciamento da produção (*scheduling*), em ambientes permutation *flow shop*. Além disso, comparar a eficiência e eficácia da solução gerada por meio de algoritmos heurísticos com a execução de modelos matemáticos.

1.2.2 Específicos

- Estudar aspectos teóricos e práticos no tocante a modelagem matemática e algoritmos heurísticos;
- Estudar a teoria relacionada aos problemas de *scheduling*, em especial aos problemas de sequenciamento da produção em ambientes de *permutation flow shop*;
- Implementar algoritmos heurísticos capazes de resolver eficientemente problemas de sequenciamento da produção do permutation *flow shop*;
- Implementar modelos matemáticos criados para resolver problemas de sequenciamento de produção do tipo *permutation flow shop*;
- Realizar experimentos computacionais para comparar o desempenho da heurística e dos modelos matemáticos.

1.3 METODOLOGIA

Segundo Gil (2002), as pesquisas exploratórias são aquelas que tem como objetivo se familiarizar mais com o problema, trazendo-o à tona e estruturando

hipóteses a seu respeito. Assim, podemos dizer que essa pesquisa pode ser classificada com base em seu objetivo como uma pesquisa exploratória.

Também, pode-se dizer que o trabalho em questão apresenta uma abordagem quantitativa uma vez que, assim como a abordagem citada, tem como principais preocupações a sua mensurabilidade, causalidade, generalização e replicação (CAUCHICK, 2012).

Quanto ao método científico utilizado, pode-se dizer que o trabalho utilizou o método dedutivo pois, segundo Marconi & Lakatos (2003), o método dedutivo é caracterizado pela aceitação inicial de todas as premissas como verdadeiras e, então, a conclusão deve ser verdadeira. Tais aspectos estão presentes nessa pesquisa, justificando a sua classificação.

Pode-se, ainda, afirmar que se trata de uma pesquisa aplicada no que diz respeito à sua finalidade. Isso é possível visto que o trabalho foca na aplicação dos modelos matemáticos e algoritmos propostos.

Quanto aos dados, trata-se de uma modelagem. Cauchik (2012) define modelo como “uma representação de uma situação ou realidade, conforme vista por uma pessoa ou um grupo de pessoas, e construída de forma a auxiliar o tratamento daquela situação de uma maneira sistemática”. O uso de modelos matemáticos que modelam o ambiente produtivo *flow shop* para atingir os objetivos dessa pesquisa justificam a sua classificação.

1.4 ORGANIZAÇÃO DO TRABALHO

Esse trabalho está organizado em seis capítulos, sendo eles Introdução, Fundamentação Teórica, Revisão da Literatura, Contexto e Descrição do Problema, Experimentos Computacionais e Conclusão. O Capítulo 1 é dividido em quatro subseções: Justificativa e Relevância, Objetivos, Metodologia e Organização do Trabalho (essa seção), que juntas têm a função de apresentar o tema do trabalho, demonstrar a sua relevância e descrever a metodologia utilizada para desenvolver a pesquisa.

O Capítulo 2 tem o intuito de explicar ao leitor os conceitos fundamentais que sustentam o tema do trabalho e são de importante compreensão para o pleno entendimento da pesquisa.

O Capítulo 3 visa fazer um apanhado dos avanços e tendências no tema sendo abordado no trabalho. O Capítulo 4 desse trabalho faz uma contextualização e descrição formal do problema sendo investigado. O Capítulo 5 apresenta os experimentos computacionais realizados durante a pesquisa juntamente com os seus resultados e o Capítulo 6 apresenta as conclusões obtidas a partir do trabalho como um todo.

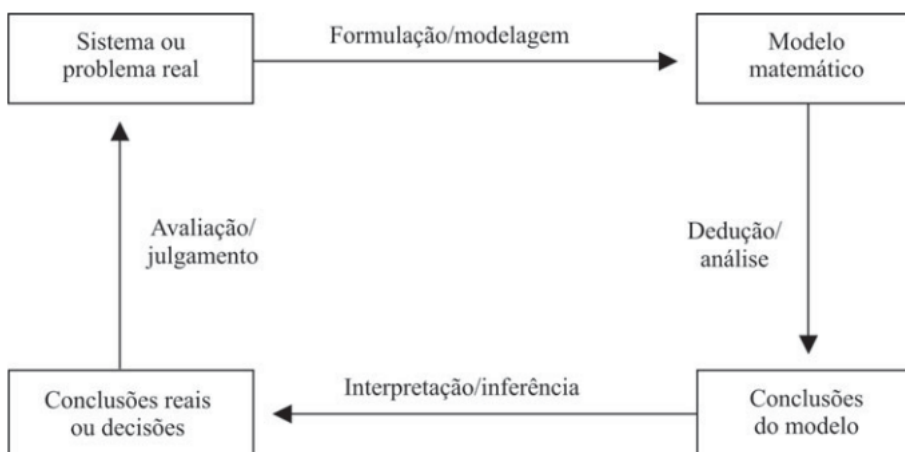
2 FUNDAMENTAÇÃO TEÓRICA

A fundamentação teórica supre, nesse trabalho, o papel de fornecer ao leitor as bases teóricas e conceituais necessárias para a compreensão do tema abordado nessa pesquisa. Assim, serão apresentados aqui os princípios e conceitos essenciais dos problemas de sequenciamento do tipo *permutation flow shop*, além dos principais métodos e abordagens adotados para a resolução desse tipo de problema.

2.1 MODELOS MATEMÁTICOS NA PESQUISA OPERACIONAL

A Pesquisa Operacional consiste na criação de métodos e sistemas para encontrar soluções ótimas para problemas complexos de planejamento e de decisão, considerando possíveis restrições e diferentes objetivos (ARENALES, 2007). Nesse sentido, a modelagem matemática exerce um papel fundamental na Pesquisa Operacional, pois ela busca criar uma representação simplificada da realidade que seja detalhada o suficiente para transmitir as características intrínsecas do problema, porém, simples o suficiente para que seja possível tratá-lo usando métodos de solução (ARENALES, 2007). Abaixo, é possível observar uma representação simplificada do processo de resolução de um problema utilizando a abordagem da modelagem matemática.

Figura 1: Solução de um problema utilizando abordagem matemática



Fonte: Arenales et al. (2007)

Como descrito por Arenales et al. (2007), a formulação (modelagem) é responsável por definir as variáveis e relações matemáticas existentes no problema real ou sistema com o intuito de descrever o seu comportamento relevante. Como resultado desse processo, é gerado o modelo matemático. A dedução/análise consiste na aplicação de tecnologia e técnicas matemáticas para solucionar o modelo matemático criado na etapa anterior e analisar as soluções geradas por ele, chegando-se a conclusões sobre o modelo. Na interpretação/inferência, verifica-se que as conclusões alcançadas sobre o modelo permitem afirmar que este é suficientemente equivalente ao problema ou sistema real para gerar conclusões reais ou decisões relevantes. Com isso, realiza-se uma avaliação/julgamento dessas conclusões reais e decisões para averiguar se essas de fato estão adequadas. Caso não estejam, o que frequentemente ocorre, o modelo matemático precisa ser revisto, assim reiniciando o ciclo.

2.2 PROBLEMAS DE SEQUENCIAMENTO

Dentre os diversos problemas que podem ser tratados na Pesquisa Operacional, existe o problema de sequenciamento de tarefas, ou *scheduling*. Esse consiste em um problema NP-difícil que tem o objetivo de definir, em um ambiente de manufatura,

quais tarefas devem ser executadas em quais máquinas e em que intervalo de tempo (PARUNAK, 1991).

Segundo Pinedo (2016), os problemas de *scheduling* podem ser modelados de diferentes formas. O Quadro 1 apresenta as mais comuns.

Quadro 1: Tipos de modelagem do problema de scheduling

Modelo de uma única máquina	Modelo no qual há apenas uma máquina disponível para processar um conjunto de tarefas.
Modelo avançado de uma única máquina	Acrescenta ao modelo de uma única máquina objetivos ou restrições adicionais, como setups dependentes de sequências específicas, prazos etc.
Modelos de máquinas paralelas	Modelos nos quais há várias máquinas idênticas disponíveis para processar um conjunto de tarefas.
<i>Flow Shops</i> e <i>Flow Shops</i> Flexíveis	As tarefas seguem uma sequência fixada de operações em uma série de máquinas. <i>Flow shops</i> flexíveis permitem que mais de uma máquina execute uma determinada operação.
<i>Job Shops</i>	Cada tarefa do conjunto de tarefas pode exigir uma sequência distinta de operações a ser executada pelas máquinas do conjunto de máquinas.
<i>Open Shops</i>	As tarefas podem ser processadas por qualquer máquina em qualquer ordem.

Fonte: Pinedo (2016)

Os problemas de sequenciamento podem ser classificados usando uma notação de três campos, também conhecida como notação $\alpha | \beta | \gamma$ (alfa | beta | gama)

(GRAHA,M et al., 1979). Cada campo é responsável por classificar o ambiente de produção descrito quanto a diferentes aspectos.

O campo α classifica o ambiente de produção de acordo com o número de máquinas ou estágios. O campo β realiza a classificação do problema de acordo com as propriedades (ou restrições) dos seus recursos e tarefas. Por fim, o campo γ classifica o problema de acordo com a sua medida de desempenho, ou seja, aquela que será otimizada na função objetivo (FUCHIGAMI, 2021).

Os Quadros 1, 2 e 3 abaixo exibem possíveis valores para cada um desses campos.

Quadro 2: Lista de possíveis valores para o campo α da notação de Graham et al. (1979)

1:	Máquina única
P:	Máquinas paralelas idênticas
Q:	Máquinas paralelas uniformes
R:	Máquinas paralelas não relacionadas
J:	<i>Job shop</i>
O:	<i>Open shop</i>
F:	<i>Flow shop</i>
HF:	<i>Flexible flow shop</i> ou <i>flexible flow line</i>

Fonte: Fuchigami (2021)

Quadro 3: Lista de possíveis valores para o campo β da notação de Graham et al. (1979)

$\emptyset$:	Quando não houver propriedades específicas ou restrições
r_j :	Presença de datas de liberação (<i>release dates</i> ou <i>ready times</i>)
d_j :	Presença de prazos de entrega (<i>due dates</i>)
$s_{ij}(s_{ijk})$:	Presença de tempos de <i>setup</i> (preparação) dependentes da sequência
$s_j(s_{jk})$:	Presença de tempos de <i>setup</i> (preparação) independentes da sequência
prmp:	As ações podem ser interrompidas (<i>preemption</i>)
prmu:	A solução deve ser uma programação permutacional

<i>brkdwn</i> :	As máquinas podem ficar indisponíveis (<i>breakdown</i>)
<i>split</i> :	As operações podem ser subdivididas (<i>splitting</i>)
<i>block</i> :	Bloqueio (<i>blocking</i>); indica que existe uma capacidade finita de armazenamento temporário (<i>buffer</i>) entre duas máquinas
M_j :	Elegibilidade da máquina; conjunto de máquinas que podem processar a operação da tarefa J_j
<i>no-waits</i> :	As tarefas não podem esperar entre duas máquinas sucessivas (<i>no-wait</i>)
<i>recrc</i> :	As tarefas podem visitar uma máquina mais de uma vez (<i>recirculation</i>)
<i>prec</i> :	Presença de restrições de precedência geral

Fonte: Fuchigami (2021)

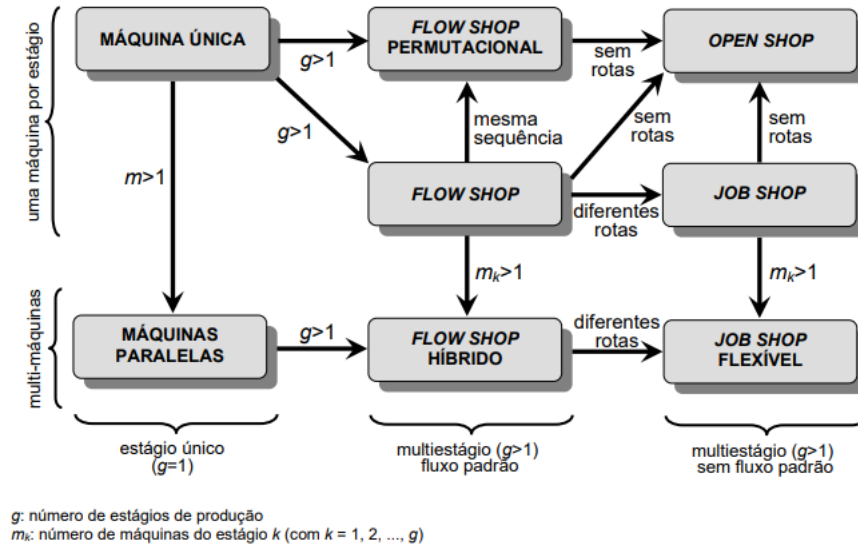
Quadro 4: Lista de possíveis valores para o campo γ da notação de Graham et al. (1979)

C_{max} :	Duração total da programação (<i>makespan</i>)
$\bar{F}$:	Tempo médio de fluxo (<i>flow time</i>) das tarefas
F (ou $\sum F_j$):	Tempo total de fluxo (<i>total flow time</i>) das tarefas
F_w (ou $\sum w_j F_j$):	Fluxo ponderado (<i>weighted flow time</i>) das tarefas
T (ou $\sum T_j$):	Atraso total (<i>total tardiness</i>) das tarefas
T_w (ou $\sum w_j T_j$):	Atraso ponderado (<i>weighted tardiness</i>) das tarefas
T_{max} :	Maior atraso (<i>tardiness</i>) das tarefas
n_T (ou $\sum U_j$):	Número de tarefas atrasadas (<i>tardy jobs</i>)
E_{max} :	Maior adiantamento (<i>earliness</i>) de tarefas
E (ou $\sum E_j$):	Adiantamento total (<i>total earliness</i>) das tarefas
L_{max} :	Maior desvio de pontualidade (<i>lateness</i>) das tarefas
ET (ou $\sum E_j + T_j$):	Soma dos adiantamentos e atrasos das tarefas
I (ou $\sum I_j$):	Total de tempo ocioso das máquinas (<i>idle time</i>)
Os problemas também podem ser multicritério	

Fonte: Fuchigami (2021)

A Figura 2 ilustrada a relação entre os problemas clássicos de *scheduling*:

Figura 2: Relação entre os problemas clássicos de *scheduling*



Fonte: Fuchigami (2021)

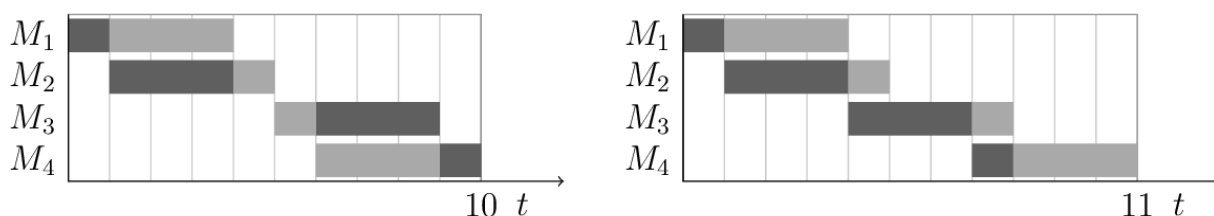
2.2.1 Permutation Flow Shop

O problema de *flow shop* permutacional é uma versão simplificada do problema tradicional de *flow shop*, porém, com grande aplicabilidade prática na indústria (FUCHIGAMI, 2015, p. 56). Trata-se de um caso de *flow shop* no qual, além de ser necessário que cada atividade siga a mesma ordem de processamento nas máquinas, cada máquina processe todas as atividades seguindo a mesma sequência. Nesse caso, o número de possíveis soluções para o problema reduz para $n!$, visto que a mesma sequência de tarefas deve ser mantida em todas as máquinas (FUCHIGAMI, 2015, p. 56). Na Figura 4, é possível observar um exemplo visual da diferença entre o *flow shop* clássico (não-permutacional) e o *flow shop* permutacional, onde são apresentadas soluções distintas para uma mesma instância (conforme os dados apresentados na Tabela 1).

Tabela 1: Instância 2×4 para o problema de scheduling em ambiente flow shop

Tarefas	Operações			
	M_1	M_2	M_3	M_4
J_1	1	3	3	1
J_2	3	1	1	3

Fonte: Benavides Roja (2015)

Figura 3: Sequenciamento de tarefas em ambientes *flow shop* não-permutacional e *flow shop* permutacional

Fonte: Benavides Roja (2015)

2.3 HEURÍSTICAS E METAHEURÍSTICAS

Hillier (2005) define um método heurístico como um procedimento que encontrará, provavelmente, uma solução viável e de boa qualidade, não necessariamente sendo a solução ótima do problema. Apesar de não ser possível provar a qualidade da solução encontrada pelo método heurístico, caso se trate de um método robusto, a solução deve estar ao menos próxima da ótima.

Para resolver o problema proposto, as heurísticas podem adotar técnicas diferentes para estruturar suas soluções, podendo elas serem construtivas ou de refinamento, por exemplo. As heurísticas construtivas são aquelas que visam construir uma solução elemento por elemento, enquanto as heurísticas de refinamento (também conhecidas como técnicas de busca local) são aquelas que partem de uma solução inicial e visa aprimorá-la a cada iteração (SOUZA, 2011).

Já a metaheurística consiste em um método geral de resolução que tem como output uma estrutura juntamente com diretrizes estratégicas para desenvolver um

método heurístico específico para um determinado problema (HILLIER, 2005). As metaheurísticas utilizam mecanismos para evitar que se fique “preso” em ótimos locais que podem estar longe dos ótimos globais. Elas podem ser classificadas de acordo com os mecanismos que elas adotam para fugir desses ótimos locais, existindo duas principais classificações: busca local e busca populacional (SOUZA, 2011).

2.3.1 Heurística NEH

O algoritmo NEH (Nawaz-Enscore-Ham) é uma heurística construtiva usada para resolver problemas do tipo *permutation flow shop* amplamente aplicada no contexto de PCP em indústrias devido à sua simplicidade e eficácia ao gerar rapidamente soluções próximas à ótima (STAFFORD & GUPTA, 2006).

Segundo Nawaz, Enscore e Ham (1983), o algoritmo proposto pelos mesmos é sustentado pela premissa de que, em um ambiente do tipo *flow shop* permutacional, tarefas com tempos de processamento maiores devem receber prioridade em detrimento às atividades com tempos de processamento mais curtos. Dessa forma, o algoritmo realiza a alocação das tarefas seguindo a ordem decrescente de seus tempos de processamento, de maneira a minimizar o *makespan*.

Seja m o número de máquinas e n o número de atividades, o algoritmo é resumido nas seguintes etapas:

1. Para cada atividade i , é calculado o tempo total de processamento em todas as máquinas;
2. São ordenadas todas as atividades por ordem decrescente do somatório de tempos de processamento nas máquinas T_i ;
3. São selecionadas as duas primeiras atividades da lista citada no item 2 (ou seja, as duas atividades com maior tempo total de processamento);
4. É calculado o *makespan* das duas possíveis sequências dessas duas atividades e é selecionada a sequência com menor *makespan*. Uma vez encontrada a melhor sequência dessas duas tarefas, a posição relativa delas duas não poderá ser alterada;
5. Agora, considera-se $i = 3$ e os seguintes passos são seguidos até que $i = n$:

- i. Seleciona-se a i^{a} atividade da lista ordenada citada no item 2;
- ii. A atividade é inserida em todas as possíveis i posições da sequência parcial encontrada no item 4, sem mudar a posição relativa das atividades que já compõem essa sequência. É fixada então na posição que garante o menor *makespan*;
- iii. Soma-se 1 a i e repete-se o processo até $i = n$.

3 REVISÃO DA LITERATURA

O problema de sequenciamento de tarefas em máquinas nos ambientes de produção no contexto de ambientes do tipo *flow shop* é o ponto central pesquisas científicas há várias décadas. Uma das primeiras aparições do problema de sequenciamento na literatura aconteceu em 1954 com a publicação da regra de Johnson, que trazia um método simples, porém eficaz para identificar a sequência ótima de atividades em um ambiente de produção do tipo *permutation flow shop* com duas máquinas (JOHNSON, 1954).

Uma das heurísticas mais conhecidas na área de *flow shop scheduling* é o algoritmo NEH (Nawaz-Enscore-Ham). Publicado em 1983, o algoritmo NEH tem sido amplamente aplicada a problemas de *scheduling* e é, há tempos, considerada a principal heurística de comparação para a resolução de problemas de sequenciamento do tipo *flow shop* (YENISEY & YAGMAHAN, 2013). Essa heurística é conhecida pela sua alta eficácia e eficiência, sendo capaz de gerar soluções próximas da ótima.

Em um estudo avaliando o desempenho de algoritmos aplicados ao contexto de sequenciamento *flowshop*, Kalczyński e Kamburowski (2007) conduziram uma análise abrangente para avaliar a eficiência de algoritmos que afirmam superar o algoritmo NEH. Sua pesquisa concluiu que, apesar do surgimento de vários algoritmos concorrentes, o algoritmo NEH continua sendo uma escolha robusta e confiável para resolver problemas de $F | pmu | C_{max}$. A capacidade do algoritmo NEH de fornecer soluções competitivas de forma consistente ressalta sua relevância e eficácia duradouras no domínio da programação de *flow shop*.

Apesar da grande aceitação do algoritmo NEH como o melhor algoritmos para problemas de *flow shop scheduling*, novas pesquisas foram realizadas e novos métodos de resolução desse problema foram desenvolvidos obtendo desempenhos semelhantes ou até melhores que o do NEH. Um exemplo disso é o algoritmo descrito no trabalho de Liu, Jin e Price (2017), o NEHLJP1, que utiliza um novo critério de priorização, levando em conta a assimetria dos tempos de processamento. Essa heurística apresentou um desempenho superior à famosa heurística NEH em experimentos computacionais realizados.

Outra abordagem comumente utilizada por pesquisadores é a utilização da solução gerada pelo NEH como uma solução inicial e, depois dela, partir para uma heurística de refinamento. Taillard (1990), utilizou essa abordagem quando propôs o método NEHT, o qual melhora significativamente a performance de execução do NEH. A solução gerada pelo NEHT é então aprimorada utilizando uma abordagem metaheurística tabu de busca local, o que melhorou significativamente a qualidade da solução.

Rios-Mercado & Bard (1998) propuseram, também, uma extensão do algoritmo NEH chamada de NEHT-RB que utiliza a modificação do algoritmo original NEH realizada por Taillard (1990) com o algoritmo NEHT citado no parágrafo anterior. O NEHT-RB apresentou uma menor complexidade do que o NEH e é capaz de contemplar também tempos de setup (não contemplados pelo algoritmo original NEH), provando ser superior nesses quesitos.

Quanto a avanços na área de heurísticas para problemas de *scheduling* em ambientes *flow shop* nos últimos anos, podemos destacar o algoritmo NEH-NGA proposto por Liang et al. (2022), que parte de uma solução gerada pelo NEH e a aprimora com a utilização de um algoritmo genético que aplica sobre ela três operadores de cruzamento. São esses os Operadores de Cruzamento Linear LOX (*linear order crossover*), Operador de Cruzamento Baseado em Posição POX (*position based crossover*) e o Operador de Mutação, Swap, Inversão e Inserção.

Outro recente exemplo de uma extensão do algoritmo NEH é encontrado no trabalho de Puka et al. (2024), que apresenta o algoritmo heurístico construtivo vN - NEH. Essa heurística utiliza a técnica da lista vN na fase de inserção, que consiste no ajuste dinâmico da lista de possíveis candidatos a serem inseridos na ordenação de tarefas a cada iteração.

O trabalho de Öztop et al. (2020) também se trata de uma heurística para problemas de *scheduling* em ambientes *permutation flow shop* que parte de uma solução inicial gerada pelo algoritmo NEH. Essa heurística se diferencia dos exemplos citados acima, no entanto, ao integrar objetivos relacionados a eficiência energética, tornando o problema de *permutation flow shop* um problema bi-objetivo.

Gmys et al. (2020) aborda o problema de *permutation flow shop* através de um algoritmo *branch-and-bound* que se diferencia de outros algoritmos ao realizar as ramificações de forma dinâmica e realizar refinamento de limites inferiores.

Tomando como base a revisão da literatura, percebe-se a relevância do estudo de problemas de *scheduling* em ambientes *flow shop* tendo em vista a sua vasta aplicabilidade no contexto das indústrias e a necessidade da geração de soluções rápidas nesses cenários. Os avanços na área permitem a obtenção de soluções para problemas de sequenciamento utilizando cada vez menos esforço computacional, sem perdas significativas na qualidade das soluções e, muitas vezes, até aprimorando a qualidade dessas soluções. Além disso, percebe-se uma expansão dos modelos de solução dos problemas de *scheduling* para englobar outros objetivos além de redução do *makespan*, ampliando assim a sua aplicabilidade.

4 CONTEXTO E DESCRIÇÃO DO PROBLEMA

Essa Seção está dividida em duas partes que, em conjunto, têm o objetivo de esclarecer para o leitor o problema sendo tratado nesse trabalho. A primeira parte é responsável por apresentar o contexto no qual o problema de *scheduling* em ambientes *flow shop* está inserido enquanto a segunda parte tem o intuito de descrever mais profundamente o problema juntamente com a abordagem a ele adotada nesse trabalho.

4.1 CONTEXTO DO PROBLEMA

Tendo em vista a relevância do sequenciamento eficiente de tarefas em ambientes industriais para o contexto do PCP pela sua capacidade de otimizar recursos na produção, o problema de *scheduling* é notavelmente comum a diversas indústrias/sistemas de produção. Além disso, o ambiente *permutation flow shop* é capaz de representar várias dessas indústrias, sendo esse o problema *flow shop* acrescido da restrição *pmu* (permutação) fazendo as atividades respeitarem uma lógica PEPS (primeiro que entra, primeiro que sai) no seu processamento nas máquinas. Nesse sentido, a aplicabilidade dessa classe de problemas em casos reais bastante vasta.

A eficiência e organização desses ambientes geram um impacto direto na sua produtividade, qualidade e, conseqüentemente, competitividade das empresas, sendo então prioridade para elas encontrarem soluções de otimização da produção, como as heurísticas de problemas de sequenciamento.

4.2 DESCRIÇÃO DO PROBLEMA

Ao longo dos anos, pesquisadores abordaram diversos tipos de problemas de *scheduling*, como o *job shop*, *open shop*, *permutation flow shop*, entre outros. Desses, o problema de *flow shop scheduling* tem ganhado relevância devido à sua

complexidade combinatória possibilidade de aplicação prática em sistemas de produção reais (LIANG, Z., et al; 2022).

O problema de sequenciamento no ambiente *permutation flow shop*, um caso simplificado do *flow shop*, tem como objetivo minimizar o tempo de término do processamento da última atividade na última máquina, em um contexto no qual um conjunto de atividades precisa ser processada por um conjunto de máquinas respeitando uma sequência determinada, sendo necessário que todas as máquinas processem todas as tarefas na mesma ordem. (KALCZYNSKI & KAMBUROWSKI; 2007).

A seguir, são apresentadas duas possíveis formulações baseadas em programação matemática do *flow shop permutacional* abordadas no trabalho de Stafford, Tseng & Gupta (2005), ambas com o objetivo de minimizar o *makespan* (i.e., tempo de conclusão da última tarefa).

Abaixo, no Quadro 5, estão detalhados os parâmetros e variáveis de decisão considerados pelos modelos descritos no parágrafo anterior.

Quadro 5: Parâmetros e variáveis de decisão dos modelos matemáticos apresentados

Parâmetros	
N	Número de tarefas a serem processadas
M	Número de máquinas
Z_{ri}	Tempo de processamento da atividade i na máquina r
Variáveis de decisão	
C_{max}	Variável utilizada para cálculo do <i>makespan</i>
Z_{ij}	Variável binária que define a sequência das tarefas. A variável assume valor 1 se a tarefa i é alocada à posição da sequência j , ou 0, caso contrário.
B_{rj}	Tempo de início da tarefa na posição da sequência j na máquina r
D_{ik}	Variável binária. Assume o valor 1 se a tarefa i estiver programada para ocorrer antes da atividade k ; Caso contrário, 0; $i < k$
E_{rj}	Tempo de término da tarefa na posição j na máquina r
S_{ri}	Tempo de início da tarefa i na máquina r
X_{rj}	Tempo ocioso na máquina r antes do início da tarefa na posição da sequência j
Y_{rj}	Tempo de espera da tarefa na posição da sequência j depois que ela termina de ser processada pela máquina r

O primeiro modelo matemático, determinado WST, define o *makespan* como o somatório dos tempos de processamento T de todas as atividades na última máquina (M) mais o somatório dos tempos ociosos X entre um processamento e outro na última máquina (M):

$$\begin{aligned} \min C_{max} \\ Makespan = C_{max} = \sum_{i=1}^N T_{Mi} + \sum_{p=1}^N X_{Mp} \end{aligned} \quad (1) \quad i = 1, \dots, N$$

$$\sum_{j=1}^N Z_{ij} = 1 \quad (1 \leq i \leq N) \quad (2)$$

$$\sum_{i=1}^N Z_{ij} = 1 \quad (1 \leq j \leq N) \quad (3)$$

$$\sum_{i=1}^N T_{ri} Z_{i,j+1} + X_{r,j+1} + Y_{r,j+1} = \sum_{i=1}^N T_{r+1,i} Z_{i,j} + X_{r+1,j+1} + Y_{r,j} \quad (1 \leq r \leq M-1; 1 \leq j \leq N-1) \quad (4)$$

$$X_{r+1,1} = X_{r,1} + Y_{r,1} + \sum_{i=1}^N T_{ri} Z_{i1} \quad (1 \leq r \leq M-1) \quad (5)$$

$$Y_{r1} = 0 \quad (1 \leq r \leq M-1) \quad (6)$$

A função objetivo do modelo WST (1) busca a minimização do *makespan* e está sujeita às restrições 2 a 6, detalhadas acima. As restrições dessa modelagem matemática determinam que:

(2): Garante que cada atividade i será alocada a exatamente uma posição na sequência de tarefas. Essa restrição soma todas as posições j e garante que cada atividade está ocupando apenas uma posição na sequência de tarefas;

(3): Garante que cada posição j na sequência será ocupada por exatamente uma atividade. Essa restrição soma todas as atividades i e garante que apenas uma delas está designada à posição j na sequência de tarefas;

(4): Garante que a atividade na posição $j + 1$ não poderá começar a ser processada na máquina r até que a atividade na posição j tenha terminado de ser processada naquela máquina. Além disso, ela garante que a atividade na posição j não poderá iniciar seu processamento na máquina $r + 1$ até que ela tenha completado o seu processamento na máquina r ;

(5): Garante que a atividade na posição 1 irá começar a ser processada imediatamente nas máquinas 2 até M assim que tenha sido terminada de ser

processada na máquina anterior. Essa restrição liga os tempos de processamento, os tempos de espera e os tempos obsoletos entre máquinas consecutivas;

(6): Garante que o tempo de espera da atividade na posição 1 para a máquina 2 até a máquina M é zero. Essa restrição garante que a tarefa na posição 1 começa a ser processada sem tempo de espera nessas máquinas.

O segundo modelo matemático, denominado TS2, representa o *makespan* pelo tempo de término E da última atividade (N) na última máquina (M). A modelagem matemática respeita a seguinte estrutura:

$$\min C_{max} \quad (1)$$

$$Makespan = C_{max} = E_{MN}$$

$$\sum_{j=1}^N Z_{ij} = 1 \quad (1 \leq i \leq N) \quad (2)$$

$$\sum_{i=1}^N Z_{ij} = 1 \quad (1 \leq j \leq N) \quad (3)$$

$$E_{r,j} + \sum_{i=1}^N T_{ri} Z_{i,j+1} \leq E_{r,j+1} \quad \begin{matrix} (1 \leq r \leq M-1; \\ 1 \leq j \leq N-1) \end{matrix} \quad (12)$$

$$E_{r,j} + \sum_{i=1}^N T_{r+1,i} Z_{i,j} \leq E_{r+1,j} \quad \begin{matrix} (1 \leq r \leq M-1; \\ 1 \leq j \leq N-1) \end{matrix} \quad (13)$$

$$E_{11} \geq \sum_{i=1}^N T_{1i} Z_{i,1} \quad (14)$$

Essa modelagem matemática considera as restrições (2) e (3) do modelo WST, já detalhadas acima. Além disso, apresenta as seguintes restrições:

(12): Garante que a atividade na posição $j + 1$ da sequência de processamento não será finalizada em qualquer máquina até que seu predecessor na sequência tenha finalizado seu processamento naquela máquina. Além disso, garante que a atividade na posição $j + 1$ também será processada naquela máquina. Em suma, essa restrição reforça a sequência de término das tarefas nas máquinas;

(13): Garante que uma dada atividade não será capaz de ser finalizada na máquina $r + 1$ até que tenha ao menos terminado de ser processada na máquina anterior e, depois, tenha sido totalmente processada na máquina $r + 1$. Em suma, essa restrição controla o término das atividades em máquinas consecutivas;

(14): Garante que a atividade na posição 1 da sequência de processamento não poderá ser finalizada na máquina 1 até que tenha sido processada naquela máquina. Em suma, essa restrição garante o término da primeira atividade na primeira máquina.

5 EXPERIMENTOS COMPUTACIONAIS

Visto que o problema de sequenciamento de tarefas se trata de um problema NP-difícil e que a sua aplicabilidade se dá em um nível bastante operacional das indústrias, torna-se necessário que a otimização ocorra de forma rápida. Encontrar a solução em tempo hábil é de suma importância, o que nem sempre é possível de se fazer na busca pela solução ótima. Nesse contexto, as heurísticas se mostram como importantes ferramentas da indústria, em particular a heurística NEH que se destaca pela sua simplicidade e baixo esforço computacional necessário para ser executada.

As heurísticas, contudo, além de rápidas precisam ser capazes de gerar soluções de boa qualidade para justificar a sua utilização. Propõe-se então a comparação do algoritmo NEH com duas modelagens matemáticas distintas do problema de *permutation flow shop*: WST e TS2. Buscou-se avaliar, além da qualidade da solução final gerada, o tempo de execução dos modelos, visto que esse é um critério importante no contexto prático

Através da realização de testes computacionais com implantações da heurística e de ambas as modelagens matemáticas utilizando a mesma instância para todos, será possível realizar uma comparação direta entre as três soluções e mensurar os seus desempenhos segundo parâmetros estabelecidos.

Foi realizada a implementação da heurística NEH juntamente com a implementação de duas formas distintas de modelagem matemática do problema de *scheduling* em ambiente *permutation flow shop*: WST e TS2. A heurística e os modelos matemáticos foram implementados utilizando linguagens de programação e executados tendo as mesmas instâncias como *input* a fim de comparar os resultados obtidos por cada um.

Uma vez que em aplicações reais o problema de *scheduling* tem uma natureza operacional e, por isso, exige uma ação rápida, estabeleceu-se o tempo limite de 1 hora para a execução dos modelos.

As três implementações foram executadas utilizando as instâncias de E. Taillard (1993), que variam pelo número de tarefas e de máquinas. Os modelos matemáticos WST e TS2 foram implementados utilizando a linguagem de programação Julia e a biblioteca de modelagem JuMP, enquanto a heurística NEH foi implementada em C++.

Os experimentos computacionais foram executados em um computador com sistema operacional *Windows 10*, processador *Intel Core i3* da 7ª geração e com 8 GB de memória RAM.

As instâncias utilizadas nesse trabalho podem ser instaladas através do seguinte link:

<http://mistic.heig-vd.ch/taillard/problemes.dir/ordonnancement.dir/ordonnancement.html>.

Essas consistem em matrizes dos tempos de processamento de cada tarefa em cada máquina, sendo as linhas referentes às máquinas e as colunas referentes às tarefas.

5.1 RESULTADOS OBTIDOS

Foi possível, então, comparar diretamente os resultados obtidos a partir das três implementações, as quais estão descritas nas Tabelas 2-13. Nas tabelas, as colunas n e m representam, respectivamente, os números de tarefas e de máquinas relacionados com as instâncias.

Na coluna “Obj” está sinalizada a solução gerada por cada implementação, ou seja, o valor encontrado para a função objetivo do problema. A coluna “T(s)” indica o tempo computacional necessário para resolver o problema. As colunas WTS e TS2 apresentam os resultados dos respectivos modelos, enquanto a coluna NEH corresponde aos resultados obtidos pela heurística.

Tabela 2: Resultados obtidos para instâncias 20 x 5

Instance	n	m	WTS		TS2		NEH		Gap
			Obj	T(s)	Obj	T(s)	Obj	T(s)	
SDST10_ta001	20	5	1278	4,3	1278	0,5	1278	0,5	0,00%
SDST10_ta002			1359	0,9	1359	1,3	1365	0,0	0,44%
SDST10_ta003			1081	4,3	1081	3,5	1140	0,0	5,18%
SDST10_ta004			1293	8,7	1293	6,9	1325	0,0	2,42%
SDST10_ta005			1235	36,8	1235	25,2	1305	0,0	5,36%
SDST10_ta006			1195	1,5	1195	1,2	1228	0,0	2,69%
SDST10_ta007			1234	3,0	1234	5,0	1279	0,0	3,52%
SDST10_ta008			1206	40,9	1206	4,9	1223	0,0	1,39%
SDST10_ta009			1230	9,7	1230	3,8	1291	0,0	4,73%
SDST10_ta010			1108	8,9	1108	2,6	1151	0,0	3,74%

Ao analisar os tempos de computação de cada experimento, podemos concluir que o modelo TS2 demonstrou uma melhoria em relação ao modelo WTS quando tratamos de tempo necessário para encontrar a solução. Isso se fundamenta no fato de que o modelo TS2 apresentou tempos computacionais menores que os tempos computacionais alcançados pelo modelo WTS em todos os experimentos. Também é

notável que a heurística NEH tende a produzir resultados com uma taxa de divergência em relação ao valor da solução ótima de em torno de 5% no pior dos casos. Dessa forma, pode-se dizer que essa se trata de uma heurística simples que pode ser aprimorada, por exemplo, através dos métodos de refinamento citados nesse trabalho como algoritmos de busca local.

Os modelos matemáticos abordados não foram capazes de encontrar resultados dentro de um tempo computacional inferior a 1 hora para os problemas com instâncias que continham 20 ou mais tarefas e mais de 5 máquinas. Desta forma, não foram reportadas soluções dos modelos para esses problemas. A heurística NEH foi, então, aplicada a mais instâncias para fins de teste. Os resultados estão apresentados nas Tabelas 3 a 13.

Tabela 3: Resultados obtidos para instâncias 20 x 10

Instance	n	m	NEH	
			Obj	T(s)
SDST10_ta011	20	10	1680	0,0
SDST10_ta012			1786	0,0
SDST10_ta013			1557	0,0
SDST10_ta014			1450	0,0
SDST10_ta015			1502	0,0
SDST10_ta016			1453	0,0
SDST10_ta017			1562	0,0
SDST10_ta018			1609	0,0
SDST10_ta019			1647	0,0
SDST10_ta020			1653	0,0

Tabela 4: Resultados obtidos para instâncias 20 x 20

Instance	n	m	NEH	
			Obj	T(s)
SDST10_ta021	20	20	2410	0,0
SDST10_ta022			2150	0,0
SDST10_ta023			2429	0,0
SDST10_ta024			2262	0,0
SDST10_ta025			2397	0,0
SDST10_ta026			2349	0,0
SDST10_ta027			2362	0,0
SDST10_ta028			2249	0,0
SDST10_ta029			2306	0,0
SDST10_ta030			2277	0,0

Tabela 5: Resultados obtidos para instâncias 50 x 5

Instance	n	m	NEH	
			Obj	T(s)
SDST10_ta031	50	5	2733	0,0
SDST10_ta032			2882	0,0
SDST10_ta033			2643	0,0
SDST10_ta034			2782	0,0
SDST10_ta035			2868	0,0
SDST10_ta036			2850	0,0
SDST10_ta037			2736	0,0
SDST10_ta038			2703	0,0
SDST10_ta039			2574	0,0
SDST10_ta040			2790	0,0

Tabela 6: Resultados obtidos para instâncias 50 x 10

Instance	n	m	NEH	
			Obj	T(s)
SDST10_ta041	50	10	3155	0,0
SDST10_ta042			3021	0,0
SDST10_ta043			3009	0,0
SDST10_ta044			3198	0,0
SDST10_ta045			3128	0,0
SDST10_ta046			3178	0,0
SDST10_ta047			3277	0,0
SDST10_ta048			3123	0,0
SDST10_ta049			3002	0,0
SDST10_ta050			3272	0,0

Tabela 7: Resultados obtidos para instâncias 50 x 20

Instance	n	m	NEH	
			Obj	T(s)
SDST10_ta041	50	20	4038	0,0
SDST10_ta042			3921	0,0
SDST10_ta043			3923	0,0
SDST10_ta044			3952	0,0
SDST10_ta045			3822	0,0
SDST10_ta046			3940	0,0
SDST10_ta047			3952	0,0
SDST10_ta048			3930	0,0
SDST10_ta049			3952	0,0
SDST10_ta050			4079	0,0

Tabela 8: Resultados obtidos para instâncias 100 x 5

Instance	n	m	NEH	
			Obj	T(s)
SDST10_ta041	100	5	5519	0,0
SDST10_ta042			5284	0,0
SDST10_ta043			5214	0,0
SDST10_ta044			5023	0,0
SDST10_ta045			5267	0,0
SDST10_ta046			5139	0,0
SDST10_ta047			5277	0,0
SDST10_ta048			5132	0,0
SDST10_ta049			5489	0,0
SDST10_ta050			5345	0,0

Tabela 9: Resultados obtidos para instâncias 100 x 10

Instance	n	m	NEH	
			Obj	T(s)
SDST10_ta041	100	10	5870	0,0
SDST10_ta042			5451	0,0
SDST10_ta043			5810	0,0
SDST10_ta044			5973	0,0
SDST10_ta045			5758	0,0
SDST10_ta046			5393	0,0
SDST10_ta047			5695	0,0
SDST10_ta048			5705	0,0
SDST10_ta049			6027	0,0
SDST10_ta050			5947	0,0

Tabela 10: Resultados obtidos para instâncias 100 x 20

Instance	n	m	NEH	
			Obj	T(s)
SDST10_ta041	100	20	6597	0,0
SDST10_ta042			6511	0,0
SDST10_ta043			6623	0,0
SDST10_ta044			6619	0,0
SDST10_ta045			6674	0,0
SDST10_ta046			6748	0,0
SDST10_ta047			6614	0,0
SDST10_ta048			6783	0,0
SDST10_ta049			6650	0,0
SDST10_ta050			6670	0,0

Tabela 11: Resultados obtidos para instâncias 200 x 10

Instance	n	m	NEH	
			Obj	T(s)
SDST10_ta041	200	10	10983	0,1
SDST10_ta042			10634	0,1
SDST10_ta043			11070	0,1
SDST10_ta044			11057	0,1
SDST10_ta045			10698	0,1
SDST10_ta046			10489	0,1
SDST10_ta047			10983	0,1
SDST10_ta048			10864	0,1
SDST10_ta049			10600	0,1
SDST10_ta050			10809	0,1

Tabela 12: Resultados obtidos para instâncias 200 x 20

Instance	n	m	NEH	
			Obj	T(s)
SDST10_ta041	200	20	11575	0,1
SDST10_ta042			11694	0,1
SDST10_ta043			11852	0,1
SDST10_ta044			11703	0,1
SDST10_ta045			11608	0,1
SDST10_ta046			11558	0,1
SDST10_ta047			11807	0,1
SDST10_ta048			11802	0,1
SDST10_ta049			11668	0,1
SDST10_ta050			11829	0,1

Tabela 13: Resultados obtidos para instâncias 500 x 20

Instance	n	m	NEH	
			Obj	T(s)
SDST10_ta041	500	20	26648	1,6
SDST10_ta042			27200	1,6
SDST10_ta043			26862	1,6
SDST10_ta044			26983	1,9
SDST10_ta045			26895	2,0
SDST10_ta046			27132	2,0
SDST10_ta047			26750	2,0
SDST10_ta048			27139	2,0
SDST10_ta049			26616	2,2
SDST10_ta050			27027	2,1

Apesar de não ter sido possível comparar o algoritmo NEH com as duas formulações matemáticas WST e TS2 nos casos das instâncias maiores, é nítido o

bom desempenho da heurística quanto ao tempo necessário para gerar uma solução, que mesmo nas maiores instâncias com 500! possíveis soluções foi capaz de gerar resultados em no máximo 2,2 segundos.

6 CONCLUSÃO

Após um estudo aprofundado realizado sobre os problemas de sequenciamento de tarefas em ambientes produtivos do tipo *permutation flow shop* juntamente com os métodos existentes para solucioná-los, pode-se concluir que se trata de um tema de extrema relevância no contexto industrial com muitas oportunidades de avanço.

O algoritmo NEH se prova como uma solução bastante eficiente e de boa qualidade para o problema proposto, porém, com grandes oportunidades de aprimoramento seja através da modificação de alguns de seus aspectos estruturais ou de seu uso como solução inicial a ser refinada posteriormente através de outras heurísticas.

Além disso, a heurística pode ser adaptada para contemplar outras possíveis características do problema como, por exemplo, tempos de setup, ambientes de produção dinâmicos, entre outros. Também, é possível identificar oportunidades de se considerar mais de um objetivo, podendo-se considerar outras variáveis a serem otimizadas além do *makespan*. Essa possível flexibilização da heurística ampliaria a sua aplicabilidade por permitir a sua adequação a contextos diferentes, o que aumenta a sua importância.

REFERÊNCIAS

1. APPLEGATE, D. L., BIXBY, R. E., CHVÁTAL, V. e Cook, W. J. The Traveling Salesman Problem: A Computational Study, 2007
2. ARENALES, M. N. et al. Pesquisa Operacional. São Paulo - SP: LTC, 2007.
3. BENAVIDES ROJAS, A. J. Heurísticas para Escalonamento em *Flow Shops*: Considerando Escalonamentos Não-Permutacionais e Trabalhadores Heterogêneos. Tese (Doutorado em Ciência da Computação) - Universidade Federal do Rio Grande do Sul, Instituto de Informática, Porto Alegre, 2015
4. CAUCHICK MIGUEL, P. A. Metodologia de Pesquisa em Engenharia de Produção e Gestão de Operações. Editora Elsevier, 2012.
5. DING, Jian-Ya; SONG, Shiji; WU, Cheng. Carbon-efficient *scheduling* of *flow shops* by multi-objective optimization. European Journal of Operational Research, v. 248, n. 3, p. 758-771, 2016.
6. FUCHIGAMI, H. Y. Sequenciamento da produção em sistemas *flow shop*. Goiânia–GO: UFG, 2015.
7. FUCHIGAMI, H.Y. Introdução ao sequenciamento da produção. Aparecida de Goiânia: UFG, 2021. Material didático, versão 8.0.
8. GELDERS, Ludo F.; VAN WASSENHOVE, Luk N. Production planning: a review. European Journal of Operational Research, v. 7, n. 2, p. 101-110, 1981.
9. GIL, A.C. Como elaborar projetos de pesquisa. 4 ed. SP: Atlas, 2002.
10. GMYS, Jan et al. A computationally efficient Branch-and-Bound algorithm for the permutation *flow-shop scheduling* problem. European Journal of Operational Research, v. 284, n. 3, p. 814-833, 2020.
11. GRAHAM, Ronald Lewis et al. Optimization and approximation in deterministic sequencing and *scheduling*: a survey. In: Annals of discrete mathematics. Elsevier, 1979. p. 287-326.
12. GUPTA, J. N. D.; STAFFORD, E. F. *Flow shop scheduling* research after five decades. European Journal of Operational Research, v. 169, p. 699-711, 2006
13. HILLIER, F. S.; LIEBERMAN, G. J. Introdução à Pesquisa Operacional. 8. ed. São Paulo: McGraw-Hill, 2005.
14. JOHNSON, S. M. Optimal two- and three-stage production schedules with setup times included. Naval Research Logistics Quarterly, 1(1), 61-68, 1954.
15. KALCZYNSKI, P. J., & KAMBUROWSKI, J. On the NEH heuristic for minimizing the *makespan* in permutation *flow shops*. Omega, 35, 53–60, 2007.
16. LI, Mei; WANG, Gai-Ge. A review of green *shop scheduling* problem. Information Sciences, v. 589, p. 478-496, 2022.
17. LIANG, Zhongyuan et al. A computational efficient optimization of *flow shop scheduling* problems. Scientific Reports, v. 12, n. 1, p. 845, 2022.

18. LIU, Weibo; JIN, Yan; PRICE, Mark. A new improved NEH heuristic for permutation *flowshop scheduling* problems. *International Journal of Production Economics*, v. 193, p. 21-30, 2017.
19. MANSOURI, S. Afshin; AKTAS, Emel; BESIKCI, Umut. Green *scheduling* of a two-machine *flowshop*: Trade-off between *makespan* and energy consumption. *European journal of operational research*, v. 248, n. 3, p. 772-788, 2016.
20. MARCONI, M. A., LAKATOS, E. V. Fundamentos de Metodologia Científica. 5. ed. São Paulo: Atlas, 2003.
21. NAWAZ, Muhammad; ENSCORE JR, E. Emory; HAM, Inyong. A heuristic algorithm for the m-machine, n-job *flow-shop* sequencing problem. *Omega*, v. 11, n. 1, p. 91-95, 1983.
22. ÖZTOP, Hande et al. An energy-efficient permutation *flowshop scheduling* problem. *Expert systems with applications*, v. 150, p. 113279, 2020.
23. PARUNAK, H. Van Dyke. Characterizing the manufacturing *scheduling* problem. *Journal of manufacturing systems*, v. 10, n. 3, p. 241-259, 1991.
24. PINEDO, Michael L. *Scheduling: Theory, Algorithms, and Systems*. 5. ed. New York, NY, USA: Springer, 2016. 670 p. ISBN 978-3-319-26578
25. PUKA, Radosław et al. Deterministic constructive vN-NEH+ algorithm to solve permutation flow shop scheduling problem with makespan criterion. *Computers & Operations Research*, v. 162, p. 106473, 2024.
26. RÍOS-MERCADO, Roger Z.; BARD, Jonathan F. Heuristics for the flow line problem with setup costs. *European Journal of Operational Research*, v. 110, n. 1, p. 76-98, 1998.
27. ROSSIT, D. A. et al. The Non-Permutation *Flow-Shop scheduling* problem: A literature review. *Omega*, v. 77, p. 143-153, 2018.
28. SOUZA, Marcone Jamilson Freitas. Inteligência Computacional para Otimização. Universidade Federal de Ouro Preto, 2011/1.
29. TAILLARD, Eric. Some efficient heuristic methods for the *flow shop* sequencing problem. *European journal of Operational research*, v. 47, n. 1, p. 65-74, 1990.
30. TAILLARD, Eric. Benchmarks for basic *scheduling* problems. *European journal of operational research*, v. 64, n. 2, p. 278-285, 1993.
31. THOMAS, L. Joseph; MCCLAIN, John O. An overview of production planning. *Handbooks in operations research and management science*, v. 4, p. 333-370, 1993.
32. WU, Xiuli; SUN, Yangjun. A green *scheduling* algorithm for flexible job shop with energy-saving measures. *Journal of cleaner production*, v. 172, p. 3249-3264, 2018.
33. XU, Jiuping; ZHOU, Xiaoyang. A class of multi-objective expected value decision-making model with birandom coefficients and its application to *flow shop scheduling* problem. *Information Sciences*, v. 179, n. 17, p. 2997-3017, 2009.
34. YENISEY, Mehmet Mutlu; YAGMAHAN, Betul. Multi-objective permutation *flow shop scheduling* problem: Literature review, classification and current trends. *Omega*, v. 45, p. 119-135, 2014.