



UNIVERSIDADE FEDERAL DE PERNAMBUCO  
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS  
DEPARTAMENTO DE ENGENHARIA DE PRODUÇÃO  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

CLARISSA MARIA RODRIGUES DE OLIVEIRA

**IMPLEMENTAÇÃO DE MÉTODOS DE OTIMIZAÇÃO PARA A RESOLUÇÃO  
DE UM PROBLEMA DE SEQUENCIAMENTO EM MÁQUINAS PARALELAS  
COM CONSIDERAÇÕES AMBIENTAIS**

Recife  
2024

CLARISSA MARIA RODRIGUES DE OLIVEIRA

IMPLEMENTAÇÃO DE MÉTODOS DE OTIMIZAÇÃO PARA A RESOLUÇÃO DE UM  
PROBLEMA DE SEQUENCIAMENTO EM MÁQUINAS PARALELAS COM CONSIDERAÇÕES  
AMBIENTAIS

Dissertação de Mestrado apresentada ao Departamento de Engenharia de Produção da Universidade Federal de Pernambuco, como requisito parcial para obtenção do título de Mestre em Engenharia de Produção. Área de Concentração: Pesquisa Operacional.

Orientador: Prof. Dr. Raphael Harry Frederico Ribeiro Kramer.

RECIFE

2024

.Catalogação de Publicação na Fonte. UFPE - Biblioteca Central

Oliveira, Clarissa Maria Rodrigues de.

Implementação de métodos de otimização para a resolução de um problema de sequenciamento em máquinas paralelas com considerações ambientais / Clarissa Maria Rodrigues de Oliveira. - Recife, 2024.

141f. : il.

Dissertação (Mestrado), Universidade Federal de Pernambuco, Centro de Tecnologia e Geociências, Programa de Pós-Graduação em Engenharia de Produção, 2024.

Orientação: Raphael Harry Frederico Ribeiro Kramer.

1. Sequenciamento de Produção; 2. Sequenciamento Verde; 3. Máquinas Paralelas; 4. Otimização Multiobjetivo; 5. Consumo Energético; 6. Heurísticas. I. Kramer, Raphael Harry Frederico Ribeiro. II. Título.

UFPE-Biblioteca Central

CDD 658.5

CLARISSA MARIA RODRIGUES DE OLIVEIRA

IMPLEMENTAÇÃO DE MÉTODOS DE OTIMIZAÇÃO PARA A RESOLUÇÃO DE UM  
PROBLEMA DE SEQUENCIAMENTO EM MÁQUINAS PARALELAS COM CONSIDERAÇÕES  
AMBIENTAIS

Dissertação de Mestrado apresentada ao Departamento de Engenharia de Produção da Universidade Federal de Pernambuco, como requisito parcial para obtenção do título de Mestre em Engenharia de Produção. Área de Concentração: Pesquisa Operacional.

Aprovada em: 30/07/2024.

BANCA EXAMINADORA

---

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

---

Prof. Dra. Isis Didier Lins (Examinadora Interna)  
Universidade Federal de Pernambuco

---

Prof. Dr. Teobaldo Leite Bulhões Júnior (Examinador  
Externo)  
Universidade Federal da Paraíba

## AGRADECIMENTOS

À CAPES, pelo apoio financeiro com a manutenção da bolsa de auxílio.

À minha querida família, minha mãe Ana, meu pai Sandro e minha irmã Larissa, por acreditarem em mim, pelo incentivo e apoio incondicional em todos os momentos dessa jornada.

Ao Prof. Dr. Raphael Harry Frederico Ribeiro Kramer, pela excelente e sábia orientação, pelo ensino com muita dedicação e personalidade, que admiro e me inspira profundamente. Sua orientação foi crucial para o desenvolvimento deste trabalho e para o meu crescimento acadêmico e pessoal.

Ao meu companheiro e parceiro de laboratório, Matheus, por ser meu maior suporte nos desafios do mestrado e da vida. Agradeço por estar sempre disposto a me escutar, a qualquer hora e em qualquer lugar, seja sobre os problemas da pesquisa, sempre pronto a ajudar a resolvê-los, seja sobre as ideias, me auxiliando a maturá-las. Agradeço pelas inúmeras reflexões nas altas madrugadas, pelo constante incentivo nos momentos de baixa autoestima, e por ser o homem que tanto admiro e que me inspira a ser uma pessoa melhor. Agradeço também pela ajuda com dúvidas em programação Julia e  $\LaTeX$ . Sua colaboração foi essencial para a realização deste trabalho.

Aos professores participantes da banca examinadora, Prof. Dra. Isis Didier Lins e Prof. Dr. Teobaldo Leite Bulhões Júnior, pelo tempo dedicado, pelas valiosas colaborações e sugestões que enriqueceram este trabalho.

Aos colegas de laboratório, pelas reflexões, discussões produtivas e momentos de lazer que tornaram essa jornada mais leve e prazerosa.

A todos que, de alguma forma, contribuíram para a realização deste trabalho, meu muito obrigada.

## RESUMO

Esta dissertação aborda o problema de sequenciamento em máquinas paralelas com considerações ambientais em um contexto bi-objetivo. O objetivo é minimizar simultaneamente o *makespan* (maior tempo de término das tarefas) e o consumo total de energia, considerando tarifas que variam ao longo do dia, uma política conhecida como *Time-of-Use*. O estudo apresenta uma revisão sistemática sobre o tema, identifica lacunas na literatura e implementa cinco abordagens heurísticas, além de um método exato, para resolver o problema. O trabalho adapta o algoritmo proposto por Anghinolfi *et al.* (2021) e a meta-heurística *Two-Phase Pareto Local Search*. As adaptações envolvem a troca da política de refinamento, a definição de “EPS-K” e a implementação de uma busca em vizinhança que permite pioras no *makespan* até certo ponto, para explorar melhor o espaço de soluções. Dentre as quatro novas propostas, duas mostraram-se mais eficientes que a abordagem original, confirmando a superioridade das heurísticas desenvolvidas em termos de medidas de performance. Os indicadores de performance *Hipervolume*, *Pureza* e  $D_r$  foram considerados na avaliação dos conjuntos de soluções, estes medem a qualidade da fronteira em número de soluções, espaçamento destas e proximidade com o ponto ideal. A comparação das abordagens foi realizada por meio do Teste de Postos Sinalizados de Wilcoxon, que indicou a superioridade da Abordagem 5 sobre as demais em todas as medidas de performance, tanto para instâncias pequenas quanto grandes. Além disso, a Abordagem 2 também se destacou, mostrando melhorias significativas em um dos indicadores de performance. Por fim, esse trabalho contribui na ampliação do conhecimento científico e no suporte ao desenvolvimento de novas pesquisas, além de estimular a aplicação prática em sistemas de manufatura de forma a auxiliar gestores em tomadas de decisões eficientes e sustentáveis, otimizando o uso de recursos e reduzindo os impactos ambientais, de modo a promover competitividade e rentabilidade às empresas.

**Palavras-chave:** Sequenciamento de Produção. Sequenciamento Verde. Máquinas Paralelas. Otimização Multiobjetivo. Consumo Energético. Heurísticas. Revisão Sistemática.

## ABSTRACT

This dissertation addresses the problem of scheduling on parallel machines with environmental considerations in a bi-objective context. The goal is to simultaneously minimize the *makespan* (the maximum completion time of the jobs) and the total energy consumption, considering time-of-use tariffs that vary throughout the day. The study presents a systematic review on the topic, identifies gaps in the literature, and implements five heuristic approaches, as well as an exact method, to solve the problem. The work adapts the algorithm proposed by Anghinolfi *et al.* (2021) and the Two-Phase Pareto Local Search metaheuristic. The adaptations include changing the refinement policy, defining “EPS-K”, and implementing a neighborhood search that allows deteriorations in *makespan* up to a certain point to better explore the solution space. Among the four new proposals, two proved to be more efficient than the original approach, confirming the superiority of the developed heuristics in terms of performance measures. The performance indicators *Hypervolume*, *Purity*, and  $D_r$  were considered in evaluating the solution sets, measuring the quality of the Pareto front in terms of the number of solutions, their spacing, and proximity to the ideal point. The comparison of the approaches was conducted using the Wilcoxon Signed-Rank Test, which indicated the superiority of Approach 5 over the others in all performance measures for both small and large instances. Additionally, Approach 2 also stood out, showing significant improvements in one of the performance indicators. Finally, this work contributes to the expansion of scientific knowledge and supports the development of new research, as well as encourages practical application in manufacturing systems to assist managers in making efficient and sustainable decisions, optimizing resource use, and reducing environmental impacts, thereby promoting competitiveness and profitability for companies.

**Keywords:** Production Scheduling. Green Scheduling. Parallel Machines. Multi-objective Optimization. Energy Consumption. Heuristics. Systematic Review.

## LISTA DE ILUSTRAÇÕES

Figura 1 – Fronteira de Pareto, Pontos Extremos, Ideal e de Nadir . . . . .	26
Figura 2 – Pontos não-dominados encontrados utilizando-se o Método da Restrição- $\epsilon$ (MR) .	27
Figura 3 – Representação do cálculo do <i>Hipervolume</i> . . . . .	29
Figura 4 – Ano de publicação para cada classificação de quantidade de objetivos . . . . .	41
Figura 5 – Aspectos ou políticas da produção que influenciam no controle energético . . . . .	44
Figura 6 – Diferentes tipos de máquinas paralelas . . . . .	46
Figura 7 – Métodos de resolução . . . . .	47
Figura 8 – Etapas da pesquisa . . . . .	64
Figura 9 – Atribuição e sequenciamento em máquinas paralelas . . . . .	66
Figura 10 – Exemplo para unidades de tempo adjacentes e consecutivas . . . . .	72
Figura 11 – Exemplo de localização livre e livre dividida . . . . .	74
Figura 12 – Exemplo de conversão de um sequenciamento dividido em um viável . . . . .	74
Figura 13 – Exemplo de atualização da lista de localizações $L_{p\hat{h}}$ . . . . .	79
Figura 14 – Exemplo para identificar uma <i>Exchangeable Period Sequence</i> (EPS) . . . . .	83
Figura 15 – Exemplo para identificar troca e rearranjo . . . . .	84
Figura 16 – Exemplo para identificar um EPS-I e um EPS-J . . . . .	85
Figura 17 – Fronteiras MIP, Abordagem 1, 2 e 5 para instâncias pequenas 3 e 9 . . . . .	104
Figura 18 – Fronteiras Abordagem 1, 2 e 5 para instâncias grandes 40 e 50 . . . . .	110
Figura 19 – Fronteiras da instância 1 . . . . .	122
Figura 20 – Fronteiras da instância 2 . . . . .	122
Figura 21 – Fronteiras da instância 3 . . . . .	122
Figura 22 – Fronteiras da instância 4 . . . . .	123
Figura 23 – Fronteiras da instância 5 . . . . .	123
Figura 24 – Fronteiras da instância 6 . . . . .	123
Figura 25 – Fronteiras da instância 7 . . . . .	124
Figura 26 – Fronteiras da instância 8 . . . . .	124
Figura 27 – Fronteiras da instância 9 . . . . .	124
Figura 28 – Fronteiras da instância 10 . . . . .	125
Figura 29 – Fronteiras da instância 11 . . . . .	125
Figura 30 – Fronteiras da instância 12 . . . . .	125
Figura 31 – Fronteiras da instância 13 . . . . .	126

Figura 32 – Fronteiras da instância 14 . . . . .	126
Figura 33 – Fronteiras da instância 15 . . . . .	126
Figura 34 – Fronteiras da instância 16 . . . . .	127
Figura 35 – Fronteiras da instância 17 . . . . .	127
Figura 36 – Fronteiras da instância 18 . . . . .	127
Figura 37 – Fronteiras da instância 19 . . . . .	128
Figura 38 – Fronteiras da instância 20 . . . . .	128
Figura 39 – Fronteiras da instância 21 . . . . .	128
Figura 40 – Fronteiras da instância 22 . . . . .	129
Figura 41 – Fronteiras da instância 23 . . . . .	129
Figura 42 – Fronteiras da instância 24 . . . . .	129
Figura 43 – Fronteiras da instância 25 . . . . .	130
Figura 44 – Fronteiras da instância 26 . . . . .	130
Figura 45 – Fronteiras da instância 27 . . . . .	130
Figura 46 – Fronteiras da instância 28 . . . . .	131
Figura 47 – Fronteiras da instância 29 . . . . .	131
Figura 48 – Fronteiras da instância 30 . . . . .	131
Figura 49 – Fronteiras da instância 31 . . . . .	132
Figura 50 – Fronteiras da instância 32 . . . . .	132
Figura 51 – Fronteiras da instância 33 . . . . .	132
Figura 52 – Fronteiras da instância 34 . . . . .	133
Figura 53 – Fronteiras da instância 35 . . . . .	133
Figura 54 – Fronteiras da instância 36 . . . . .	133
Figura 55 – Fronteiras da instância 37 . . . . .	134
Figura 56 – Fronteiras da instância 38 . . . . .	134
Figura 57 – Fronteiras da instância 39 . . . . .	134
Figura 58 – Fronteiras da instância 40 . . . . .	135
Figura 59 – Fronteiras da instância 41 . . . . .	135
Figura 60 – Fronteiras da instância 42 . . . . .	135
Figura 61 – Fronteiras da instância 43 . . . . .	136
Figura 62 – Fronteiras da instância 44 . . . . .	136
Figura 63 – Fronteiras da instância 45 . . . . .	136
Figura 64 – Fronteiras da instância 46 . . . . .	137

Figura 65 – Fronteiras da instância 47 . . . . .	137
Figura 66 – Fronteiras da instância 48 . . . . .	137
Figura 67 – Fronteiras da instância 49 . . . . .	138
Figura 68 – Fronteiras da instância 50 . . . . .	138
Figura 69 – Fronteiras da instância 51 . . . . .	138
Figura 70 – Fronteiras da instância 52 . . . . .	139
Figura 71 – Fronteiras da instância 53 . . . . .	139
Figura 72 – Fronteiras da instância 54 . . . . .	139
Figura 73 – Fronteiras da instância 55 . . . . .	140
Figura 74 – Fronteiras da instância 56 . . . . .	140
Figura 75 – Fronteiras da instância 57 . . . . .	140
Figura 76 – Fronteiras da instância 58 . . . . .	141
Figura 77 – Fronteiras da instância 59 . . . . .	141
Figura 78 – Fronteiras da instância 60 . . . . .	141

## LISTA DE TABELAS

Tabela 1 – Artigos selecionados para a segunda fase da revisão sistemática . . . . .	36
Tabela 1 – Artigos selecionados para a segunda fase da revisão sistemática . . . . .	37
Tabela 1 – Artigos selecionados para a segunda fase da revisão sistemática . . . . .	38
Tabela 1 – Artigos selecionados para a segunda fase da revisão sistemática . . . . .	39
Tabela 1 – Artigos selecionados para a segunda fase da revisão sistemática . . . . .	40
Tabela 7 – Funções objetivos com considerações ambientais dos artigos avaliados . . . . .	50
Tabela 7 – Funções objetivos com considerações ambientais dos artigos avaliados . . . . .	51
Tabela 7 – Funções objetivos com considerações ambientais dos artigos avaliados . . . . .	52
Tabela 7 – Funções objetivos com considerações ambientais dos artigos avaliados . . . . .	53
Tabela 7 – Funções objetivos com considerações ambientais dos artigos avaliados . . . . .	54
Tabela 7 – Funções objetivos com considerações ambientais dos artigos avaliados . . . . .	55
Tabela 7 – Funções objetivos com considerações ambientais dos artigos avaliados . . . . .	56
Tabela 1 – Resultados para <i>Hipervolume</i> para instâncias pequenas. . . . .	97
Tabela 2 – Estatística de teste $p$ para os resultados de <i>Hipervolume</i> para as instâncias pequenas. . . . .	98
Tabela 3 – Resultados para <i>Pureza</i> para instâncias pequenas. . . . .	99
Tabela 4 – Estatística de teste $p$ para os resultados de <i>Pureza</i> para as instâncias pequenas. . . . .	100
Tabela 5 – Resultados para $D_r$ para instâncias pequenas. . . . .	101
Tabela 6 – Estatística de teste $p$ para os resultados de $D_r$ para as instâncias pequenas. . . . .	101
Tabela 7 – Resultados para tempo computacional para instâncias pequenas. . . . .	103
Tabela 8 – Estatística de teste $p$ para os resultados de tempo computacional para as instâncias pequenas. . . . .	103
Tabela 9 – Resultados para <i>Hipervolume</i> para instâncias grandes. . . . .	105
Tabela 10 – Estatística de teste $p$ para os resultados de <i>Hipervolume</i> para as instâncias grandes. . . . .	105
Tabela 11 – Resultados para <i>Pureza</i> para instâncias grandes. . . . .	106
Tabela 12 – Estatística de teste $p$ para os resultados de <i>Pureza</i> para as instâncias grandes. . . . .	107
Tabela 13 – Resultados para $D_r$ para instâncias grandes. . . . .	108
Tabela 14 – Estatística de teste $p$ para os resultados de $D_r$ para as instâncias grandes. . . . .	108
Tabela 15 – Resultados para tempo computacional para instâncias grandes. . . . .	109
Tabela 16 – Estatística de teste $p$ para os resultados de tempo computacional para as instâncias grandes. . . . .	109

## LISTA DE QUADROS

Quadro 2 – Periódicos de publicação . . . . .	41
Quadro 3 – Desempenhos operacionais vs. Classificação quanto a quantidade de funções objetivo . . . . .	42
Quadro 4 – Desempenhos ambientais vs. Classificação quanto a quantidade de funções objetivo	42
Quadro 5 – Modelagem de trabalhos classificados como mono-objetivo . . . . .	44
Quadro 6 – Restrições com considerações ambientais dos artigos avaliados . . . . .	49
Quadro 8 – Notação . . . . .	69

## LISTA DE ALGORITMOS

Algoritmo 1 – MR- $\epsilon$ . . . . .	71
Algoritmo 2 – Converter_sequenciamento . . . . .	75
Algoritmo 3 – Heurística_gananciosa_dividida . . . . .	76
Algoritmo 4 – Atualizar_Lista_Locais1 . . . . .	78
Algoritmo 5 – Atualizar_Lista_Locais2 . . . . .	81
Algoritmo 6 – Busca_local . . . . .	86
Algoritmo 7 – SGS_ES . . . . .	87
Algoritmo 8 – Fase_1 . . . . .	89
Algoritmo 9 – Heuristica_rekursiva . . . . .	89
Algoritmo 10 – Add_solução . . . . .	90
Algoritmo 11 – PLS . . . . .	91
Algoritmo 12 – Melhor_melhoria . . . . .	94

## LISTA DE ABREVIATURAS E SIGLAS

2PPLS	<i>Two-Phase Pareto Local Search</i>
ACO	<i>Ant Colony Optimization</i>
ALNS	<i>Adaptive Large Neighborhood Search</i>
DE	<i>Differential Evolution</i>
EDEMA	<i>Estimation of Distribution Evolution Memetic Algorithm</i>
EPS	<i>Exchangeable Period Sequence</i>
ES	<i>Exchange-search</i>
FOA	<i>Fruit Fly Optimization Algorithm</i>
GA	<i>Genetic Algorithm</i>
GRASP	<i>Greedy Randomized Adaptive Search Procedure</i>
ILS	<i>Iterative Local Search</i>
LA	<i>Learning Automata</i>
MILP	Modelo de Programação Linear Inteira Mista
MOEA/D	<i>Multiobjective Evolutionary Algorithm Based on Decomposition</i>
MQIP	Modelo de Programação Quadrática Inteira Mista
MR	Método da Restrição- $\epsilon$
MSP	Método da Soma Ponderada das funções objetivo
NSGA-II	<i>Non-dominated Sorting Genetic Algorithm-II</i>
NSGA-III	<i>Non-dominated Sorting Genetic Algorithm-III</i>
PLS	<i>Pareto Local Search</i>
POM	Problema de Otimização Multi-objetivo
PSMP	Problema de Sequenciamento em Máquinas Paralelas
PSMPCA	Problema de Sequenciamento em Máquinas Paralelas com Considerações Ambientais
PSO	<i>Particle Swarm Optimization</i>
SA	<i>Simulated Annealing</i>
SGH	<i>Split-Greedy Heuristic</i>
SGS	<i>Split-Greedy Scheduler</i>
SGS-ES	<i>Split-Greedy Scheduler with Exchange Search</i>
TEC	Custo Total Energético
TOU	<i>Time-Of-Use</i>

TS        *Tabu Search*  
VNS      *Variable Neighborhood Search*

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	16
1.1	JUSTIFICATIVA E PROBLEMÁTICA	19
1.2	OBJETIVOS	22
1.3	ESTRUTURA DO TRABALHO	22
<b>2</b>	<b>REFERENCIAL TEÓRICO</b>	23
2.1	PROGRAMAÇÃO DA PRODUÇÃO	23
2.2	PROBLEMAS DE OTIMIZAÇÃO MULTIOBJETIVO	25
2.2.1	<i>Método da Restrição-<math>\epsilon</math></i>	26
2.2.2	<i>Método da soma ponderada das funções objetivo</i>	27
2.2.3	<i>Indicadores de performance para soluções de problemas de otimização multiobjetivo</i>	28
2.3	HEURÍSTICAS E META-HEURÍSTICAS	30
2.4	TESTE DE POSTOS SINALIZADOS DE WILCOXON	32
<b>3</b>	<b>REVISÃO SISTEMÁTICA DA LITERATURA</b>	34
3.1	DEFINIÇÕES GERAIS DA PESQUISA	34
3.2	CATEGORIZAÇÃO DOS TRABALHOS	35
3.2.1	<i>Sumarização dos resultados</i>	35
3.2.2	<i>Modelagem com questões ambientais</i>	48
3.2.3	<i>Sequenciamento da produção em máquinas paralelas com considerações ambientais</i>	59
<b>4</b>	<b>METODOLOGIA</b>	63
<b>5</b>	<b>DESCRIÇÃO DO PROBLEMA</b>	65
5.1	O PROBLEMA DE MÁQUINAS PARALELAS	65
5.2	CONSIDERAÇÕES E SUPOSIÇÕES DO PROBLEMA EM ESTUDO	65
5.3	FORMULAÇÃO MATEMÁTICA	67
<b>6</b>	<b>ALGORITMOS IMPLEMENTADOS</b>	70
6.1	MÉTODO DA RESTRIÇÃO- $\epsilon$ EXATO	70
6.2	ALGORITMOS HEURÍSTICOS	71
6.2.1	<i>Sequenciamento Guloso Dividido</i>	72
6.2.1.1	<i>Heurística Gananciosa Dividida</i>	72
6.2.1.2	<i>Busca Local</i>	83
6.2.1.3	<i>Heurística Gananciosa Dividida com Busca Local</i>	87
6.2.2	<i>Two-Phase Pareto Local Search</i>	88

6.2.3	<i>Abordagem proposta</i> . . . . .	91
7	<b>RESULTADOS E DISCUSSÃO</b> . . . . .	95
7.1	Instâncias Utilizadas . . . . .	95
7.2	Resultados Computacionais . . . . .	96
7.2.1	<i>Instâncias Pequenas</i> . . . . .	96
7.2.2	<i>Instâncias Grandes</i> . . . . .	104
8	<b>CONCLUSÕES</b> . . . . .	111
	<b>REFERÊNCIAS</b> . . . . .	114
	<b>APÊNDICE A –FRONTEIRAS DE PARETO PARA INSTÂNCIAS PEQUENAS</b>	122
	<b>APÊNDICE B –FRONTEIRAS DE PARETO PARA INSTÂNCIAS GRANDES</b>	132

## 1 INTRODUÇÃO

Os problemas na área de planejamento e programação da produção podem ser categorizados hierarquicamente em três níveis: estratégico, tático e operacional. O nível estratégico representa o mais elevado, onde são estabelecidas metas e objetivos de longo prazo, envolvendo decisões que requerem altos investimentos. Por outro lado, o nível tático lida com o planejamento de atividades, incluindo o planejamento agregado da produção, que decide sobre aspectos como mão de obra, horas extras e subcontratações para atender a uma demanda prevista, tipicamente em um horizonte de meses a um ano. Também engloba o planejamento das quantidades de produção, determinando quando e quanto produzir para cada produto dentro de um prazo que geralmente varia de semanas a seis meses. Por fim, o nível operacional envolve a atribuição de tarefas às máquinas e a programação (*scheduling*) das operações em cada máquina disponível, estabelecendo uma sequência ou agenda de processamento para essas tarefas, com definição dos tempos de início e término de cada uma (Arenales *et al.*, 2015).

Segundo Afzalirad e Shafipour (2018), um dos problemas mais estudados na área de otimização combinatória é o sequenciamento da produção, que busca determinar a ordem ideal para um conjunto de atividades. Para estabelecer essa sequência de maneira eficaz, é essencial compreender os diferentes tipos de ambientes de manufatura existentes, garantindo que o sequenciamento escolhido reflita adequadamente os processos analisados. Nesse sentido, Pinedo (2016) classifica os ambientes de manufatura em categorias como máquina única, máquinas paralelas, *flow shop*, *job shop* e *open shop*. Essa classificação é fundamental para adequar as estratégias de sequenciamento às características específicas de cada ambiente produtivo.

O Problema de Sequenciamento em Máquinas Paralelas (PSMP), objeto de estudo deste projeto de pesquisa, apresenta uma configuração comum e aplicação prática em diversas indústrias, como embalagens, impressão, alimentícia, têxtil, química, entre outras, onde várias máquinas são utilizadas para o processamento de tarefas. Essas máquinas podem diferir em velocidade, capacidade de produção, consumo de energia, custos, entre outros fatores (Zandi *et al.*, 2020).

Nesse ambiente, uma quantidade específica de tarefas deve ser processada em um conjunto de máquinas disponíveis para otimizar uma medida de desempenho operacional. Duas medidas de desempenho têm recebido destaque na literatura: a minimização do atraso ponderado em relação a um prazo e a minimização do makespan, que corresponde ao instante de término do processamento de todas as tarefas (Yin *et al.*, 2017).

O gerenciamento da agenda de fabricação permite a incorporação de questões ambientais visando a redução de custos. Por exemplo, Giret *et al.* (2015) realizaram uma revisão do estado da

arte sobre sustentabilidade na programação de sistemas de manufatura e identificaram que práticas sustentáveis na manufatura são impulsionadas por preocupações ambientais, redução de recursos não renováveis, questões legais, economia de custos, entre outros fatores. No entanto, a literatura tem focado principalmente no nível estratégico, onde há um direcionamento de esforços para o desenvolvimento sustentável em áreas como o design da cadeia de suprimento, layout e processos de reciclagem. Contudo, segundo os mesmos autores, há também uma grande oportunidade no nível operacional, através da programação, para buscar eficiência operacional considerando aspectos ambientais.

De maneira geral, a consideração de perspectivas ambientais em problemas de sequenciamento da produção pode ser realizada através de indicadores ambientais, como o consumo total de energia (Zandi *et al.*, 2020) e emissões totais de carbono (Mansouri *et al.*, 2016). No entanto, o foco principal do problema reside na atribuição e ordenação das tarefas em cada máquina, visando otimizar/minimizar uma medida de desempenho de qualidade do programa de produção. De acordo com Arenales *et al.* (2015), essas principais medidas incluem o *makespan* (tempo total de processamento), tempo total de fluxo, atraso máximo, total de atrasos, soma de atrasos e antecipações, e número de tarefas atrasadas.

Otimizar essas medidas muitas vezes entra em conflito com a busca pela otimização dos objetivos ambientais. Por exemplo, aumentar a velocidade das máquinas para reduzir o tempo de processamento das atividades, e conseqüentemente o *makespan* (uma medida de qualidade operacional), pode resultar em maior consumo de energia e, conseqüentemente, em aumento das emissões de carbono. Portanto, a programação da produção deve considerar esses *trade-offs*.

Nesse sentido, as pesquisas voltadas para a otimização do sequenciamento da produção com considerações ambientais geralmente utilizam modelagens matemáticas que abordam o problema com duas funções objetivo ou mais (bi-objetivo ou multiobjetivo). Essas abordagens permitem encontrar múltiplas soluções, que quando ótimas, pertencem à fronteira de Pareto (como em Zeng *et al.* (2018)). No entanto, também existem abordagens que optam por selecionar apenas uma medida de desempenho para minimização, enquanto a outra é incorporada como restrição do problema (como visto em Li *et al.* (2016a) e Saberi-Aliabad *et al.* (2020)), ou ainda consideram uma soma ponderada entre múltiplas funções de desempenho (como em Liang *et al.* (2015) e Wu *et al.* (2022)).

O Problema de Sequenciamento em Máquinas Paralelas com Considerações Ambientais (PSMPCA) é uma extensão do PSMP, classificado como NP-difícil. Neste tipo de problema, a resolução demanda um tempo de processamento computacional não polinomial para encontrar a solução ótima (Yin *et al.*, 2017). Portanto, apenas instâncias de pequena escala podem ser resolvidas em tempo

computacional viável utilizando métodos matemáticos exatos. Diante desse cenário, pesquisadores desenvolveram métodos aproximados para resolver esse tipo de problema. Essas abordagens permitem encontrar soluções de qualidade em tempo computacional razoável, embora não garantam a otimalidade nem quão próximas das soluções ótimas elas podem estar (Abdel-Basset *et al.*, 2018). Esses métodos são classificados como heurísticas.

As heurísticas constituem técnicas de busca inteligente para encontrar soluções aproximadas com menores custos computacionais. Elas são divididas em três categorias principais: heurísticas construtivas, de refinamento e metaheurísticas.

As heurísticas construtivas são responsáveis por criar uma solução inicial. Quando essas heurísticas incorporam a ideia de algoritmos gulosos, constroem a solução adicionando elementos um a um, onde a seleção do próximo elemento é baseada em uma função de avaliação específica, visando a inserção do elemento que proporciona o melhor benefício local (ou seja, desconsiderando o impacto global na solução como um todo). Por outro lado, as heurísticas de refinamento implementam técnicas de busca dentro de uma estrutura chamada vizinhança, com o objetivo de procurar por soluções melhores a partir de uma solução pré-existente. Por fim, as meta-heurísticas são mais complexas e aplicam heurísticas subordinadas para buscar soluções de forma mais inteligente, utilizando procedimentos para escapar de ótimos locais (Souza, 2022).

Para Abdel-Basset *et al.* (2018), as meta-heurísticas possuem um robusto mecanismo de busca por soluções baseado no equilíbrio entre exploração (diversificação) e intensificação. A exploração é responsável por procurar a melhor solução em diversas regiões do espaço de soluções, enquanto a intensificação foca na busca em regiões específicas.

As contribuições desta pesquisa são as seguintes:

- (1) **Apresentação de uma Revisão Sistemática:** O artigo apresenta uma revisão sistemática que mapeia o estado da arte referente às pesquisas sobre sequenciamento da produção em máquinas paralelas com considerações ambientais. Esta revisão oferece uma visão abrangente e atualizada das variantes do problema estudado e das abordagens para resolvê-lo na literatura.
- (2) **Propostas de Novas Heurísticas:** O artigo introduz duas novas propostas heurísticas que são capazes de melhorar o desempenho dos indicadores de performance de uma proposta atual de qualidade. Essas heurísticas oferecem uma abordagem inovadora e eficiente para resolver o PSMPCA.

Essas duas contribuições permitem estudar um problema relevante e em ascensão na literatura de sequenciamento da produção. As novas heurísticas de desempenho superior auxiliam gestores de produção a alcançar seus objetivos econômicos e ambientais, proporcionando soluções

práticas e eficientes para o setor manufatureiro.

Em suma, devido ao crescimento de pesquisas na área de sequenciamento sobre essa temática nos últimos anos, esta pesquisa tem como objetivo principal propor algoritmos heurísticos para resolver problemas de sequenciamento em máquinas paralelas que incorporem aspectos ambientais, gerando soluções melhores em um tempo razoável. Esse trabalho visa permitir uma redução dos impactos ambientais gerados pelo ambiente manufatureiro, promovendo uma produção mais sustentável e eficiente.

## 1.1 JUSTIFICATIVA E PROBLEMÁTICA

No contexto da alta competitividade das organizações, o planejamento de sistemas de manufatura desempenha um papel fundamental para o bom desempenho de uma empresa, devido à capacidade de adotar estratégias que potencializem os recursos disponíveis, como máquinas, mão de obra, matéria-prima, tempo e dinheiro. Diante disso, a pesquisa operacional utiliza técnicas de estruturação e resolução de modelos que representam a realidade de um determinado problema ou sistema estudado. Essas técnicas exploram requisitos de qualidade e buscam soluções que maximizem a produtividade, auxiliando na tomada de decisões estratégicas (Goldbarg e Luna, 2005).

Uma das atividades mais cruciais no planejamento de sistemas é o sequenciamento (*scheduling*) de tarefas, que visa designar e ordenar as atividades para um conjunto de máquinas, buscando otimizar a utilização dos recursos das organizações (Afzalirad e Shafipour, 2018). É evidente que há um impacto significativo no desempenho do sistema quando se dedica tempo e esforço para buscar uma programação eficiente, em vez de adotar um agendamento aleatório (Pinedo, 2016). Em virtude disso, a programação da produção exerce um importante papel na indústria de manufatura. Um dos ambientes de produção atualmente em destaque é o de máquinas paralelas, objeto central deste trabalho. Estudos têm sido conduzidos em diversas indústrias, como usinagem e soldagem (Sharma *et al.* (2015), Wang *et al.* (2018)), além da indústria química (Wu *et al.* (2022)).

Segundo Che *et al.* (2017b), o uso descontrolado da energia tem agravado as emissões de gases nocivos do efeito estufa, prejudicando o meio ambiente e alterando o clima. Portanto, a utilização eficiente dos recursos energéticos pode contribuir para o crescimento econômico e para a proteção ambiental. Por exemplo, de acordo com a Balanço Energético Nacional 2023 (2023), que aborda dados e informações sobre a matriz energética brasileira em 2022, 65% do consumo de energia no país foi destinado aos setores de transporte (33%) e industrial (32%) no mesmo ano, com um aumento de 2,9% no uso de energia em relação ao ano anterior. Especificamente nas indústrias, esse aumento foi de 1,6%. Em relação à energia elétrica no Brasil em 2022, houve um aumento de disponibilidade em

1,6%, mas o consumo final cresceu 2,3%. Apesar de o uso da eletricidade não gerar diretamente CO<sub>2</sub>, uma grande quantidade é liberada na geração de energia, principalmente pelas termoelétricas (Zhou *et al.*, 2018).

Além disso, o total de emissões de CO<sub>2</sub> atingiu 423 milhões de toneladas no Brasil, sendo 49,74% correspondente ao setor de transporte, seguido pelo setor industrial, responsável por 18,13%. Contudo, apesar da diminuição verificada das emissões em relação a 2021, ocasionada pela redução da geração térmica devido ao favorecimento do regime hídrico causado pela maior precipitação de chuvas em 2022, ainda se projeta um crescimento anual de emissões de CO<sub>2</sub> de 2,6% até 2030, atingindo 518,2 milhões de toneladas (Balanço Energético Nacional 2023, 2023).

Sequenciamentos ineficientes causam desperdícios nas organizações, seja em relação ao tempo operacional desperdiçado, reduzindo a produtividade e a utilização dos recursos, como a energia, o que ocasiona maiores impactos ambientais (Yang *et al.*, 2019). Conforme Pan *et al.* (2022), as indústrias manufatureiras estão cada vez mais pressionadas a cumprir rigorosas regulamentações em relação ao consumo energético e à poluição ambiental. Portanto, a busca por métodos práticos para solucionar esses problemas é de extrema relevância, e um suporte para isso consiste na programação eficiente, visando reduzir os impactos ambientais.

Ainda, impulsionadas pela escassez de recursos de combustíveis e grandes quantidades de emissões de CO<sub>2</sub>, as empresas devem adotar medidas para reduzir impactos ambientais ao mesmo tempo que atende seus pedidos com eficiência (Wu *et al.*, 2022).

Tais fatores gradualmente incentivam a conscientização ambiental dentro das organizações e abrem espaço para as práticas da manufatura verde, visando minimizar o impacto ambiental ao mesmo tempo em que se maximiza a utilização e disponibilidade dos recursos. Isso envolve atuação direta no planejamento de processos, seleção de matérias-primas, utilização de equipamentos e na programação da produção (Li *et al.*, 2016a).

Diante disso, Wu e Che (2019) destacam a viabilidade de utilizar máquinas energeticamente eficientes na manufatura, embora essa opção geralmente demande altos investimentos por parte das empresas. Assim, concentrar esforços na programação de tarefas com o objetivo de reduzir custos energéticos emerge como uma alternativa viável, capaz de gerar ganhos significativos com investimentos relativamente baixos.

Verifica-se, portanto, a importância de implementar métodos que minimizem os impactos ambientais da manufatura, promovendo uma utilização mais eficiente dos recursos energéticos para reduzir o consumo de energia e, conseqüentemente, as emissões de gases poluentes. Nesse contexto, este estudo foca no ambiente de manufatura de máquinas paralelas, onde os aspectos ambientais podem

ser incorporados tanto nas restrições quanto nos objetivos do modelo.

Estudos que desenvolvem modelos matemáticos e algoritmos para resolver problemas de programação da produção, focados no sequenciamento, consideram questões ambientais de diversas maneiras. Por exemplo, há abordagens que visam reduzir a emissão total de carbono (como estudado por Zheng e Wang (2016) e Xue *et al.* (2019)), além da avaliação do custo do consumo de energia, variando conforme os preços da energia em diferentes horários do dia. Essas simulações refletem a dinâmica prática do mercado de eletricidade, abordando períodos de pico, vales e horários normais (conforme discutido por Wang *et al.* (2018), Ding *et al.* (2015), Che *et al.* (2017b) e Fang *et al.* (2016)).

Adicionalmente, o controle energético pode ser efetuado através do desligamento das máquinas nos momentos de ociosidade (como explorado em Liang *et al.* (2015), Che *et al.* (2017a) e Antoniadis *et al.* (2020)), assim como pelo ajuste da velocidade das máquinas para regular tanto o consumo de energia quanto o tempo de processamento das tarefas. Geralmente, quanto maior a velocidade da máquina, maior a demanda energética (conforme discutido por Li *et al.* (2016b), Wu e Che (2019), Sharma *et al.* (2015) e Fang *et al.* (2016)).

Devido ao alto tempo computacional exigido pelos algoritmos exatos na resolução de problemas NP-difíceis, os métodos heurísticos surgem como uma alternativa viável para obter soluções de forma mais eficiente. Eles utilizam um conjunto de técnicas e algoritmos eficientes que exploram soluções aproximadas, as quais são consideradas de boa qualidade, embora não garantam a otimalidade (Goldbarg e Luna, 2005). Assim, os métodos heurísticos têm o objetivo de encontrar soluções viáveis e de boa qualidade por meio de uma busca que permite analisar um número menor de soluções. Além de serem mais rápidos em comparação aos métodos exatos, frequentemente são fáceis de implementar, possibilitando sua aplicação em casos reais.

Portanto, visto que este estudo se concentra na análise teórica e experimental de algoritmos para resolver problemas de PSMPCA, ele contribui significativamente para a expansão do conhecimento nesta área de pesquisa, com uma abordagem sustentável e um estímulo à aplicação prática em sistemas reais para verificar a eficácia do modelo e do algoritmo propostos.

Diante do exposto, os problemas que definem um agendamento para a produção incorporando questões ambientais podem ser estudados por meio de uma modelagem matemática que busca minimizar um único objetivo, sendo ele de desempenho operacional ou ambiental, no qual o outro é incluído nas restrições do problema. Outra estratégia é adotar uma modelagem mono-objetivo que pondera ambos os aspectos mencionados por meio de uma soma ponderada. Por fim, é possível otimizar mais de um critério simultaneamente, apresentando um modelo matemático multiobjetivo.

Logo, em virtude da problemática apresentada surge o seguinte questionamento: **como**

**estruturar um algoritmo que seja capaz de resolver instâncias de um problema de sequenciamento de máquinas paralelas considerando a minimização do custo total energético e que retorne soluções melhores ou competitivas quando comparadas com as melhores da literatura?**

## 1.2 OBJETIVOS

Essa pesquisa apresenta como objetivo geral propor um algoritmo heurístico para resolver um problema de máquinas paralelas com considerações ambientais.

Os seguintes objetivos específicos são indicados:

- Identificar os métodos heurísticos utilizados e as características consideradas nos problemas sobre o tema abordado por meio da revisão sistemática;
- Selecionar um problema de máquinas paralelas incorporando aspectos ambientais da literatura recente;
- Desenvolver e implementar um novo algoritmo heurístico em linguagem de programação para resolver o problema selecionado;
- Validar o algoritmo implementado e verificar seu desempenho em termos de função objetivo e tempo computacional por meio de testes computacionais com instâncias já consideradas na literatura relacionada;
- Comparar os resultados obtidos pelo algoritmo heurístico proposto com os resultados provenientes da resolução exata do problema para pequenas instâncias e com outros algoritmos da literatura para instâncias grandes;
- Discutir os resultados, indicando soluções que minimizem impactos ambientais.

## 1.3 ESTRUTURA DO TRABALHO

O restante do trabalho é estruturado conforme segue: o Capítulo 2 apresenta o referencial teórico com conceitos importantes relacionados à temática de sequenciamento da produção, otimização multiobjetivo e métodos de otimização de problemas de otimização. O Capítulo 3 descreve uma revisão sistemática que mapeia o estado da arte referente aos problemas de máquinas paralelas com considerações ambientais. O Capítulo 4 descreve as características metodológicas e as etapas de desenvolvimento desta pesquisa. O Capítulo 5 apresenta uma descrição do problema, apontando as considerações, suposições e formulação matemática do mesmo. O Capítulo 6 apresenta os algoritmos implementados e a abordagem proposta para abordar o problema deste estudo. O Capítulo 7 apresenta os resultados computacionais e análise das respostas. Por fim, o Capítulo 8 conclui o trabalho e apresenta perspectivas para pesquisas futuras.

## 2 REFERENCIAL TEÓRICO

Nesse capítulo, busca-se expor alguns conceitos importantes sobre o tema. A Seção 2.1 trata dos principais conceitos referentes ao problema de sequenciamento da produção. Seguidamente, a Seção 2.2 esclarece alguns conceitos relevantes para o entendimento de modelos matemáticos multiobjetivo. A Seção 2.3 aborda as meta-heurísticas mais comuns para a resolução de problemas de otimização combinatória. Por fim, a Seção 2.4 explica o teste estatístico de Wilcoxon utilizado para apoiar a análise de significância dos resultados obtidos.

### 2.1 PROGRAMAÇÃO DA PRODUÇÃO

O sequenciamento trata de um processo de tomada de decisão com o intuito de alocar recursos para tarefas em determinados períodos de tempo apresentando um ou mais objetivos para a otimização. Neste estudo, os recursos e as tarefas correspondem às máquinas disponíveis para o processamento de operações em um sistema de produção. Esses problemas podem ser representados de maneira estruturada através de modelos matemáticos frequentemente empregados na Programação Matemática. O objetivo é facilitar a tomada de decisão, selecionando entre várias alternativas a mais adequada conforme o objetivo desejado (Goldbarg e Luna, 2005).

Assim, considerando a grande quantidade e variedade de modelos matemáticos na literatura, Graham *et al.* (1979) desenvolveu uma notação que identifica, sucintamente, a configuração destes. Portanto, o problema de sequenciamento é definido por um trio,  $\alpha|\beta|\gamma$ , no qual, o campo  $\alpha$  refere-se ao ambiente de máquina, o  $\beta$  define características e restrições de processamento, por fim,  $\gamma$  configura o objetivo de otimização. Dado um conjunto de tarefas definido por  $N = \{1, \dots, n\}$  e um conjunto de máquinas  $M = \{1, \dots, m\}$ , os principais ambientes de máquinas especificados no campo  $\alpha$  e descritos por Fuchigami (2021) são:

- Máquina única (1): Caso que todas as tarefas pertencentes a  $N$  devem ser processadas em uma única máquina, i.e., existe apenas um estágio com apenas uma máquina;
- Máquinas paralelas ( $Pm$ ): Existem  $m$  máquinas pertencentes a  $M$ , em paralelo, que podem processar qualquer uma das  $n$  tarefas, i.e., existe um único estágio com mais de uma máquina;
- *Job shop* ( $J$ ): As tarefas apresentam um fluxo linear de processamento, onde cada tarefa  $n$  pertencente ao conjunto  $N$  possui um fluxo individual ou uma rota fixa específica, passando exatamente por  $m$  máquinas. Cada máquina é um estágio de produção.
- *Job shop* flexível ( $FJ$ ): É uma variação do *job shop* onde cada estágio de produção pode conter várias máquinas disponíveis, porém, cada tarefa é processada por apenas uma máquina em cada

estágio;

- *Flow shop (F)*: É uma extensão do *job shop*, porém, nesse caso, todas as tarefas seguem a mesma rota padrão através das máquinas, com cada máquina representando um estágio de produção;
- *Flow shop híbrido (HF)*: É o caso do *flow shop*, porém, os estágios de produção podem conter várias máquinas disponíveis. Cada tarefa é processada por apenas uma máquina em cada estágio.
- *Open shop (O)*: Similar ao *job shop*, porém sem uma rota pré-definida para as tarefas. A ordem de processamento pode variar e não é obrigatório que todas as máquinas sejam utilizadas.

As máquinas paralelas podem ser classificadas em três tipos principais: idênticas ( $Pm$ ), caracterizada pelo fato de cada tarefa apresentar um tempo de processamento que independente da máquina (explanado em Wang *et al.* (2018), Wu *et al.* (2022) e Anghinolfi *et al.* (2021)); uniformes ( $Q$ ), onde os tempos de processamento variam de acordo com um fator multiplicativo (abordado por Safarzadeh e Niaki (2019)); e não relacionadas ( $R$ ), onde cada tarefa possui um tempo de processamento específico para cada máquina sem correlação entre elas (visto em Cota *et al.* (2019), Che *et al.* (2017b) e Xue *et al.* (2019)).

No caso das restrições do modelo, abordado no campo  $\beta$ , algumas características associadas às atividades podem ser inseridas no modelo, como:

- Data de liberação ( $r_j$ ): Corresponde ao tempo a partir do qual a tarefa  $j$  pode inicializar o processamento (considerado em Antoniadis *et al.* (2020) e Liang *et al.* (2015));
- Data de vencimento ( $d_j$ ): Representa o prazo de conclusão da tarefa  $j$  (visto em Antoniadis *et al.* (2020) e Liang *et al.* (2015));
- Peso ( $w_j$ ): Define um peso/prioridade para a tarefa  $j$  em relação às outras (considerado em Nanthapodej *et al.* (2021b));
- Preempção ( $Prmp$ ): Permite a interrupção do processamento de qualquer tarefa  $j$  com a sua retomada posteriormente (abordado em Antoniadis *et al.* (2020));
- Preparação de máquina ou *setup* ( $s_{ij}$ ): Determina que existe um tempo de preparação de máquina no processamento da tarefa  $j$  depois da tarefa  $i$  (observado em Bittencourt *et al.* (2023), Cota *et al.* (2021) e Rego *et al.* (2022)).

Por fim, o campo  $\gamma$  trata da medida de desempenho que se deseja otimizar, no caso do sequenciamento, comumente pretende-se minimizar medidas de performance como *makespan* ( $C_{max}$ ) (exposto em Gaggero *et al.* (2023)), atraso total (visto em Liang *et al.* (2015)), consumo energético (em Cheng *et al.* (2017) e Abikarram *et al.* (2019)), dentre outros.

Usando a notação de três campos para configurar um modelo, em Wu *et al.* (2022) estuda-se o problema ( $Pm|SD, PE|TTC + TEC$ ). No campo de ambiente de produção,  $Pm$  representa máquinas

paralelas idênticas, já no de caracterização das tarefas e do processo, *SD* é a sigla que simboliza a consideração de tarefas que deterioram com o tempo de espera para processamento na máquina e *PE* representa a associação do tempo de processamento a um consumo de energia, por fim, no campo de desempenho *TTC* representa o custo total de atraso das tarefas e *TEC* significa o custo total de consumo extra de energia devido ao fator de deterioração das tarefas.

## 2.2 PROBLEMAS DE OTIMIZAÇÃO MULTIOBJETIVO

Considerar apenas um objetivo em alguns problemas não é suficiente, sendo importante lidar com a otimização de múltiplos objetivos, no qual, para os problemas de sequenciamento com aspectos ambientais, busca-se um equilíbrio entre medidas de economia energética com desempenhos, normalmente, associados ao tempo, sendo esses conflitantes.

Conforme Xue *et al.* (2019), um Problema de Otimização Multi-objetivo (POM) pode ser definido como: considerando um problema de minimização, para uma dada solução  $x \in B$ , i.e.,  $x$  é um vetor de decisão pertencente ao espaço de soluções viáveis  $B$ , deseja-se minimizar  $f(x) = f_1(x), f_2(x), \dots, f_q(x)$ , sendo  $f_1, f_2, \dots, f_q$  objetivos conflitantes. Diante disso, uma solução viável  $a$  domina a solução viável  $b$  (com notação  $a \succ b$ ), se  $f_l(a) \leq f_l(b)$  para todo  $l \in \{1, 2, \dots, q\}$  e se  $f_l(a) < f_l(b)$  para pelo menos uma função objetivo  $l \in \{1, 2, \dots, q\}$ .

Portanto, no POM busca-se encontrar soluções que minimizem mais de uma função de desempenho, e toda solução  $x \in B$  que não é dominada por outra solução do espaço de soluções  $B$ , apresenta sua imagem pertencente ao conjunto Pareto-ótimo ou fronteira de Pareto. Logo, não existe uma única solução ótima, como nos modelos de objetivo único, mas sim, um conjunto de soluções que consideram os *trade-offs* entre as medidas de desempenho conflitantes. Ainda, quando essas soluções são obtidas por métodos aproximados (heurísticas), as imagens são pertencentes a fronteira aproximada de Pareto.

De acordo com Bérubé *et al.* (2009), alguns conceitos importantes são apresentados:

**Definição 2.2.1 (Solução eficiente)** Diz-se que uma solução  $x \in B$  é (Pareto) eficiente se não existe uma outra solução  $x' \in B$  de tal modo que  $x' \succ x$ .

**Definição 2.2.2 (Conjunto eficiente)** Agrupa todas as soluções ditas eficientes, isto é,  $A = \{x \in B : x \text{ é eficiente em } B\}$ .

**Definição 2.2.3 (Fronteira de Pareto)** Agrupa as imagens do conjunto  $A$  no espaço dos objetivos, ou seja,  $P = \{f_l(x) \forall l \in \{1, 2, \dots, q\} : x \in A\}$ .

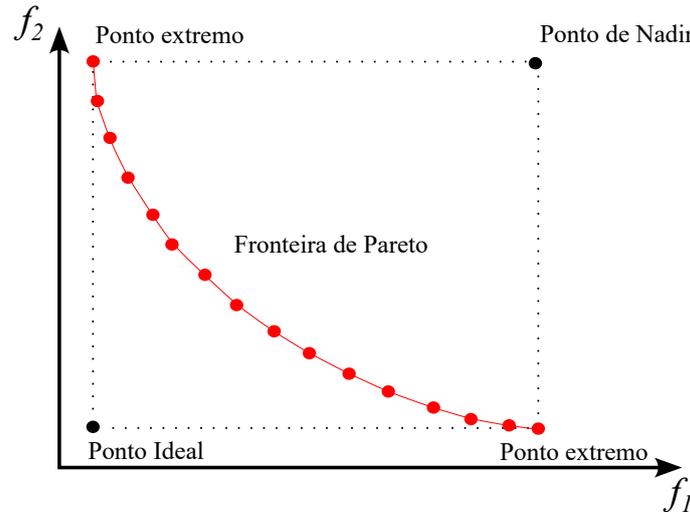
**Definição 2.2.4 (Ponto Ideal)** Para um problema bi-objetivo, o ponto ideal  $p^I = (f_1^I, f_2^I)$ , no qual,  $f_1^I = \min\{f_1(x)\}$  e  $f_2^I = \min\{f_2(x)\}$ ,  $x \in A$ .

**Definição 2.2.5 (Ponto de Nadir)** Para um problema bi-objetivo, o ponto de Nadir é entendido como  $p^N = (f_1^N, f_2^N)$ , no qual,  $f_1^N = \{f_1(x) : f_2(x) = f_2^I\}$  e  $f_2^N = \{f_2(x) : f_1(x) = f_1^I\}$ ,  $x \in A$ .

Assim, o conjunto de soluções eficientes  $A$  contém todas as soluções viáveis e não dominadas do problema e a fronteira de Pareto  $P$  contém todos os pontos não dominados no espaço dos objetivos. Além disso, os pontos extremos de uma fronteira de duas dimensões são obtidos pelos pares formados pelo melhor e pior valor de desempenho para ambos os critérios, i.e,  $p_1^A = (f_1^I, f_2^N)$  e  $p_2^A = (f_1^N, f_2^I)$ .

Por exemplo, na Figura 1 é mostrada a fronteira de Pareto  $P$ , que representa o conjunto de soluções eficientes no espaço de objetivos para um problema bi-objetivo (os pontos estão destacados em vermelho, evidenciando os extremos). Os pontos Ideal e de Nadir são marcados em preto.

Figura 1 – Fronteira de Pareto, Pontos Extremos, Ideal e de Nadir



Fonte: Elaborado pela autora (2024)

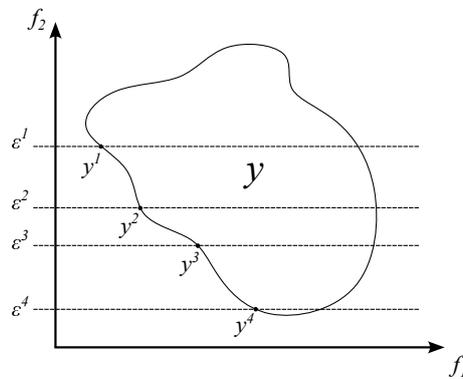
As próximas subseções apresentam dois métodos de resolução exata para POMs, o MR e o Método da Soma Ponderada das funções objetivo (MSP), nos quais estes são reformulados para conter apenas um único objetivo de desempenho.

### 2.2.1 Método da Restrição- $\varepsilon$

Essa abordagem, introduzida por Jaszkievicz (2002), não utiliza a agregação de funções objetivo, o método trabalha com a otimização de apenas um objetivo, enquanto que os demais são transformados em restrições através da imposição de *upper-bounds* (para objetivos de minimização;

ou *lower-bounds*, para objetivos de maximização), denotados por  $\varepsilon$ . Nessa técnica, com o objetivo de encontrar a Fronteira de Pareto, os valores de  $\varepsilon$  são diminuídos gradativamente ( $\varepsilon^1, \varepsilon^2, \varepsilon^3$  e  $\varepsilon^4$ ) e então novos problemas são resolvidos ( $y^1, y^2, y^3$  e  $y^4$ ), como observado na Figura 2 (Costa, 2015).

Figura 2 – Pontos não-dominados encontrados utilizando-se o MR



Fonte: Adaptado de Costa (2015)

Mavrotas (2009) apontam que essa abordagem é mais vantajosa do que o MSP. Pois, uma escolha não refinada do conjunto de pesos utilizados na junção das funções objetivos pode levar à geração de soluções redundantes, além disso, não é possível produzir todas as soluções eficazes em problemas cujo espaço de soluções não é convexo. Já o MR, a cada nova iteração gera uma nova solução eficiente (i.e., não dominada).

### 2.2.2 Método da soma ponderada das funções objetivo

O MSP é amplamente utilizado na literatura para resolver problemas com múltiplos objetivos. Essa abordagem envolve a escalarização do problema, criando uma função objetivo composta pela soma das funções objetivo, ponderadas por meio de um conjunto de pesos, como evidenciado na Equação (2.1) (Costa *et al.*, 2015).

$$\min \left\{ \sum_{l=1}^q \lambda_l \times f_l(x) : x \in B \right\} \quad (2.1)$$

Esses pesos representam a importância atribuída a cada um dos objetivos durante o processo de tomada de decisão. A principal vantagem dessa metodologia reside na simplicidade de sua execução, uma vez que os pesos modificam apenas os valores, permitindo a obtenção de várias soluções (Gandibleux, 2006). Após a obtenção da função ponderada, técnicas de otimização mono-objetivo podem ser aplicadas para resolver o problema.

Em POMs, um espaço de soluções  $B$  é definido como  $B = \{x \in R^n : h(x) = 0, g(x) \leq 0\}$ , onde  $n$  é a dimensão do espaço. No entanto, alguns POMs podem ser modelados utilizando variáveis discretas; nesse caso, o espaço de soluções é definido como  $B = \{x \in N^n : h(x) = 0, g(x) \leq 0\}$  (Arenales

*et al.*, 2015). No primeiro caso, o espaço de soluções é convexo, o que implica que todas as soluções eficientes do problema são suportadas. Uma solução suportada é obtida por meio da resolução do problema com funções objetivos agregadas, onde  $\lambda \geq 0 \in R^l$  ( $\lambda_i \geq 0, \forall i \in 1, \dots, l$ ) (Geoffrion, 1968).

Por outro lado, quando o espaço de soluções inclui variáveis inteiras, ele se torna não convexo. Nesse caso, o conjunto de soluções eficientes inclui tanto soluções suportadas quanto não-suportadas, sendo que estas últimas não podem ser obtidas por meio da resolução de problemas ponderados conforme a Equação (2.1) (Costa *et al.*, 2015).

No contexto do modelo estudado neste trabalho, o espaço de soluções é caracterizado por variáveis binárias, o que o torna não convexo. Conseqüentemente, o conjunto de soluções eficientes inclui tanto soluções suportadas quanto não-suportadas. Soluções suportadas podem ser obtidas através da resolução de problemas ponderados, enquanto soluções não-suportadas não podem ser alcançadas por este método.

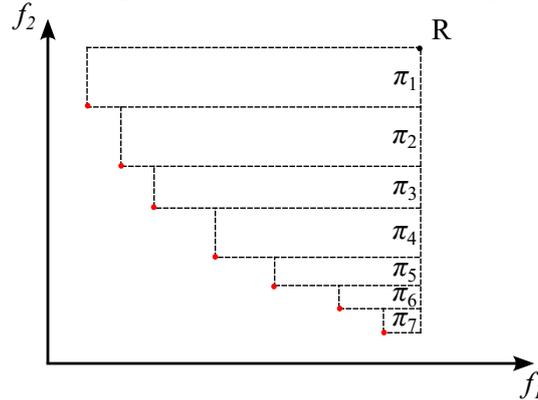
O conjunto de soluções encontradas pelos métodos de otimização multiobjetivo podem ser comparados conforme métricas de desempenho que são apresentadas na seção seguinte.

### 2.2.3 Indicadores de performance para soluções de problemas de otimização multiobjetivo

Um dos principais desafios associados às soluções obtidas por métodos multiobjetivo é a comparação entre as fronteiras de pontos não dominados. Em problemas mono-objetivos, as soluções podem ser facilmente comparadas pelo valor das funções objetivo. No entanto, essa abordagem não se aplica aos problemas com múltiplos objetivos, uma vez que a saída não é um único valor, mas sim um conjunto de soluções (Costa, 2015).

Uma das formas de avaliar a eficiência dos algoritmos multiobjetivo é pela posição da fronteira não-dominada. Segundo Zitzler *et al.* (2000), idealmente, os métodos devem procurar minimizar as distâncias entre o conjunto de pontos não-dominados obtidos e a fronteira de Pareto ótima (o conjunto de referência). Além disso, é desejável que as soluções ao longo da fronteira estejam bem distribuídas e que a quantidade de pontos não-dominados seja maior possível.

Muitos estudos têm se dedicado a estudar a comparação entre esses métodos, criando diversas métricas de desempenho para comparar os resultados de métodos de resolução de problemas com múltiplos objetivos (Costa, 2015). Diante disso, Zitzler (1999) apresenta a medida de desempenho *Hipervolume*, que avalia o volume (ou área, no caso de problemas bi-objetivo) sob a curva formada pelo conjunto de soluções não dominadas em relação a um ponto de referência (ou seja, um ponto que seja ao menos dominado por todos os pontos da fronteira). Como observado na Figura 3, os volumes ( $\pi_1, \pi_2, \pi_3, \dots, \pi_7$ ) são calculados a partir do ponto de referência  $R$ .

Figura 3 – Representação do cálculo do *Hipervolume*

Fonte: Adaptado de Costa (2015)

No trabalho de Bandyopadhyay *et al.* (2004), é apresentada outra medida de desempenho, a *Pureza* ( $P_i$ ), que mede a quantidade de soluções presentes nos outros métodos de resolução quando comparadas à fronteira exata. Suponha que existam  $L \geq 2$  estratégias aplicadas para a resolução do problema multiobjetivo, onde  $r_i, i \in \{1, \dots, L\}$ , representa a quantidade de soluções presentes na fronteira de Pareto de análise  $i$ , e  $r_i^*$  representa o número de soluções comuns da fronteira de Pareto  $i$  e de uma fronteira de referência, que pode ser obtida pela união das fronteiras dos  $L$  métodos avaliados. A *Pureza* pode ser calculada pela Equação (2.2).

$$P_i = \frac{r_i^*}{r_i} \quad (2.2)$$

No trabalho de Ishibuchi *et al.* (2003), é apresentado a medida de desempenho  $D_r$ , que consiste em uma métrica usada na avaliação da dispersão mínima das soluções encontradas por uma fronteira  $F$  em relação à fronteira de Pareto de referência ( $S^*$ ), como observado na Equação (2.3).

$$D_r = \frac{1}{S^*} \sum_{y \in S^*} \min\{d_{xy} : x \in F\} \quad (2.3)$$

$S^*$  é definido como o conjunto de soluções ótimas não dominadas ou pela união das fronteiras dos métodos avaliados. A métrica mede a distância entre todas as soluções encontradas na fronteira  $F$  e as soluções do conjunto de Pareto  $S^*$ . A medida é calculada usando a distância euclidiana, conforme demonstrado na Equação (2.4).

$$d_{xy} = \sqrt{\sum_{l=1}^q (f_l^*(x) - f_l^*(y))^2} \quad (2.4)$$

Sendo  $q$  a quantidade de objetivos de otimização do problema e  $f_l^*(a) = \frac{f_l(a) - f_l^{\min}}{f_l^{\max} - f_l^{\min}}$  é o valor da  $l$ -ésima função objetivo normalizada.

## 2.3 HEURÍSTICAS E META-HEURÍSTICAS

Métodos de solução exata retornam a solução ótima (melhor solução encontrada para uma certa medida de desempenho dentro do espaço de soluções viáveis, ou seja, todas as possíveis soluções que satisfazem as restrições do problema) por meio da enumeração sistemática do espaço de soluções. Exemplos incluem o algoritmo *Branch-and-bound*, programação dinâmica e busca exaustiva (Fuchigami, 2021). No entanto, esses métodos apresentam limitações em relação ao tempo computacional, especialmente para problemas de grande dimensão, como os de sequenciamento, definidos pelo número de máquinas e tarefas.

As heurísticas são úteis em casos onde não existe um algoritmo que retorne a solução ótima para um problema de grandes instâncias em um tempo razoável, quando uma solução boa já é adequada, quando há limitações de tempo e recursos, ou quando não se dispõe de dados confiáveis. Nesse contexto, encontrar uma solução boa é suficiente, visto que não faz sentido despender esforços em busca da solução ótima, uma vez que essa é apenas uma aproximação da realidade e não o valor exato (Fuchigami, 2021).

Por exemplo, em Oliveira *et al.* (2023b), os autores demonstraram a eficiência de algoritmos heurísticos baseados em métodos de busca evolucionária para estimar parâmetros que melhor se adequassem ao modelo do processo de falhas de sistemas reparáveis. Esses métodos apresentaram vantagens na resolução de problemas complexos, com uma implementação simplificada e encontrando soluções de alta qualidade. Esse problema é contínuo, pois as variáveis de decisão pertencem ao domínio dos números reais, resultando em um espaço de soluções infinito. Diferente dos problemas de sequenciamento da produção, que são, em geral, tratados com variáveis discretas e possuem um espaço de soluções finito, problemas contínuos apresentam desafios específicos devido à infinitude de soluções possíveis. A abordagem com algoritmos heurísticos, portanto, é eficiente para tratar tanto problemas discretos quanto contínuos, permitindo a exploração eficaz de espaços de soluções complexos em ambas as categorias.

Nesse mesmo sentido, muitos problemas de sequenciamento não possuem algoritmos de resolução exata em tempo razoável para grandes instâncias; esses problemas são conhecidos como NP-difíceis (Yang *et al.*, 2019). Para esses casos, métodos heurísticos e meta-heurísticos são apropriados para resolução em tempo computacional razoável, proporcionando boas respostas. Diante disso, para discutir as heurísticas e meta-heurísticas, faz-se necessário compreender os conceitos de ótimo local e global, além do conceito de vizinhança.

Por meio de uma solução  $s$  pertencente a um conjunto de soluções viáveis  $B$  (também

chamado de espaço de busca), é possível obter um conjunto  $N(s)$ , definido como vizinhança, a partir de um *movimento* aplicado a  $s$ . Esse *movimento* corresponde a uma função que transforma  $s$  em  $s'$ . Portanto, toda solução  $s' \in N(s)$  constitui em um vizinho de  $s$ , onde  $N(s) \subset B$  (Kramer, 2014).

Logo, uma solução  $s$  é denominada ótimo local se sua função objetivo  $f(s)$  é melhor ou igual à de todas as soluções  $s' \in N(s)$ . Por outro lado, é definida como ótimo global se  $s$  obtiver uma  $f(s)$  melhor ou igual à de qualquer outra solução em  $B$ . Portanto, um ótimo global também é um ótimo local, mas o contrário nem sempre é verdadeiro.

As heurísticas podem ser classificadas como construtivas e de refinamento, onde os métodos construtivos geram uma solução inicial viável para um problema, enquanto os métodos de refinamento buscam melhorar uma solução existente utilizando o conceito de busca local ou em vizinhança (Kramer, 2014). Os algoritmos de busca local exploram, a partir de um certo ponto atual no espaço de busca, outros pontos na vizinhança utilizando operadores específicos. Esses operadores permitem alcançar soluções “próximas”, avaliadas para uma certa função de custo/objetivo. Assim, caracteriza-se como um algoritmo iterativo que explora a vizinhança da solução atual e a atualiza caso uma solução melhor seja encontrada (Gendreau e Potvin, 2010). Em geral, o critério de parada pode ser o número máximo de iterações, tempo computacional ou a estagnação da melhoria da solução corrente.

Os algoritmos de busca local intensificam a exploração em uma vizinhança para convergir para um ótimo local (não necessariamente o ótimo global). Por outro lado, as meta-heurísticas permitem transições no espaço de soluções para reduzir momentaneamente o valor de desempenho, com o objetivo de escapar de possíveis ótimos locais (Gendreau e Potvin, 2010). Estas podem ser divididas em meta-heurísticas de busca local e populacional. Dentre as primeiras, algumas mais comuns explanadas por Blum e Roli (2003) e Souza (2022) são:

- *Multi-start*: Utiliza uma técnica que procura uma nova solução (em geral, melhor) a partir de uma inicial por meio do procedimentos chamado de busca local (i.e., busca em vizinhança);
- Procedimento de busca adaptativa aleatória gulosa (*Greedy Randomized Adaptive Search Procedure* (GRASP)): É uma variante do *multi-start*, o qual aplica um procedimento quase guloso e explora mais vizinhanças;
- Busca Tabu (*Tabu Search* (TS)): Baseada em procedimentos de busca local o qual aplicam uma estratégia de manipulação de memória, que armazena uma lista de movimentos proibidos (lista tabu) com a intenção de evitar que soluções já examinadas sejam encontradas novamente;
- Busca local iterada (*Iterative Local Search* (ILS)): A partir de uma certa solução aplica-se uma busca local de forma a encontrar um mínimo local, em seguida, essa nova solução é perturbada e reinicia-se novamente a busca local;

- Busca em vizinhança variável (*Variable Neighborhood Search (VNS)*): Dispõe de um conjunto de estruturas de vizinhança para aplicar uma estratégia de mudança dinâmica de estruturas de vizinhança durante o refinamento;
- Recozimento simulado (*Simulated Annealing (SA)*): Realiza uma analogia do processo de resfriamento de metais aplicando um procedimento de busca local que utiliza de uma técnica de aceitação de soluções de menor desempenho definida por uma probabilidade determinada pela “temperatura” (com a intenção de diversificar a busca no espaço de soluções de maneira eficiente).

Dentre as meta-heurísticas de busca populacional, ainda conforme os mesmos autores, algumas mais comuns são:

- Algoritmo genético (*Genetic Algorithm (GA)*): Aplica métodos e estratégias de busca realizando uma analogia com a seleção e evolução natural, no qual os indivíduos mais adaptados apresentam maiores probabilidades de sobrevivência e replicam suas características por meio da reprodução;
- Otimização por nuvem de partículas (*Particle Swarm Optimization (PSO)*): Inspirado no comportamento social de pássaros e cardumes de peixes, no qual estes consideram o conhecimento coletivo e individual na busca por alimento;
- Otimização por colônia de formigas (*Ant Colony Optimization (ACO)*): Inspirado no comportamento social de formigas, no qual quando estas caminham em busca de alimento depositam feromônio no solo, logo, quando escolhem uma direção a seguir, os caminhos com maior quantidade de feromônio apresentam maiores probabilidades de seleção.

## 2.4 TESTE DE POSTOS SINALIZADOS DE WILCOXON

A comparação de desempenhos dos algoritmos pode ser analisada por meio de testes estatísticos. O Teste de Postos Sinalizados de Wilcoxon, também conhecido como Teste de Wilcoxon, é uma técnica estatística não paramétrica utilizada para comparar duas amostras relacionadas ou pareadas, não assumindo que os dados seguem uma distribuição normal (Woolson, 2007).

Neste trabalho, o Teste de Wilcoxon é empregado, uma vez que desejamos comparar duas abordagens em que as observações em uma estão relacionadas às observações na outra.

O teste de Wilcoxon avalia se há diferença significativa entre as medianas das duas amostras relacionadas. Ele utiliza as diferenças entre os pares de observações para calcular a estatística de teste, em vez de usar os próprios valores das observações.

Para um teste de Wilcoxon bilateral, as hipóteses são:

- $H_0 : M_d = 0$

- $H_1 : M_d \neq 0$

Onde  $M_d$  é a mediana populacional da variável *diferença*. Logo, a hipótese nula ( $H_0$ ) afirma que não há diferença entre as medianas das duas amostras relacionadas, e a hipótese alternativa ( $H_1$ ) afirma que há uma diferença entre as medianas.

O *valor-p* é calculado com base na distribuição de Wilcoxon e define a probabilidade da diferença entre os dois ser dada pelo acaso. É usado para determinar se rejeitamos ou não a hipótese nula. Se o *valor-p* for menor que um nível de significância pré-definido, geralmente 0,05, então rejeitamos a hipótese nula, indicando que há evidências estatísticas de que há uma diferença entre as medianas das duas amostras. O *valor-p* é a probabilidade da diferença entre os dois ser dada pelo acaso (Seward e Doane, 2014).

### 3 REVISÃO SISTEMÁTICA DA LITERATURA

Este capítulo oferece uma revisão sistemática do tema de pesquisa, permitindo uma análise das tendências e lacunas existentes na literatura. Ele compreende a Seção 3.1, que trata das definições do protocolo de pesquisa, a primeira etapa da revisão, e a Seção 3.2, que discute a sumarização dos dados coletados, a segunda etapa do processo.

A revisão sistemática é um método para avaliar o estado atual do conhecimento sobre um tema específico. Ela estabelece critérios bem definidos e uma abordagem lógica para identificar um conjunto relevante de estudos relacionados ao contexto da pesquisa. Isso envolve a definição clara das bases de dados e estratégias de busca utilizadas, bem como os procedimentos para selecionar, excluir e analisar os estudos encontrados, diante disso, é caracterizada por sua condição de reprodutibilidade por outros pesquisadores (Galvão e Ricarte, 2019). Essa técnica é mais adequada quando o conjunto de dados é pequeno em magnitude e o escopo da revisão é mais específico em sua natureza (Donthu *et al.*, 2021).

#### 3.1 DEFINIÇÕES GERAIS DA PESQUISA

A primeira etapa da revisão sistemática consiste no planejamento detalhado da revisão, que inclui a definição das condições do protocolo. Isso envolve estabelecer o objetivo da revisão, as questões a serem respondidas, os tipos de estudos a serem analisados, as palavras-chave a serem pesquisadas, as bases de dados a serem consultadas, os idiomas, os anos de publicação e os critérios de inclusão e exclusão dos estudos obtidos. É essencial que essas definições estejam alinhadas com o propósito da pesquisa. E na segunda etapa ocorre a execução do protocolo e a categorização das bibliografias.

O objetivo dessa revisão é avaliar o estado da arte acerca do problema de otimização de sequenciamento da produção no ambiente de máquinas paralelas com considerações ambientais. As questões a serem respondidas são: (1) Quais são os critérios de desempenho considerados no problema; (2) Como ocorre a interação entre os aspectos ambientais e operacionais; (3) Quais e como os aspectos ambientais são incorporados no problema; e (4) Quais os métodos de resolução do problema.

Decidiu-se optar por artigos exclusivamente, visto que são meios de comunicação científica que apresentam embasamento técnico devido a aplicação de um método, o que lhes confere confiabilidade e credibilidade. Além disso, optou-se pela base de dados *Web of Science*, base multidisciplinar que indexa os periódicos mais citados de cada área, de acordo com Silva e Menezes (2005).

Por fim, com base em um estudo preliminar em documentos obtidos por buscas no Google

Acadêmico, algumas palavras-chaves foram estrategicamente selecionadas para a revisão. Para atender ao ambiente de produção estudado a palavra-chave “*parallel machine*” foi definida, já para atender ao problema de sequenciamento a expressão “*sched\**” foi estabelecida, o qual o caractere “\*”, conhecido como operador de truncagem, permite incluir palavras variantes a partir de *sched* como *schedule*, *schedules*, *scheduled*, *scheduling* ou outros. Por fim, a abrangência dos aspectos ambientais deu-se por meio da expressão “*(green) OR (sustainab\*) OR (tariff\*) OR (energ\*) OR (electric\*) OR (carbon\*) OR ("time of use") OR (pollut\*)*”.

As palavras-chave foram combinadas estrategicamente para realizar buscas que possibilitassem filtrar apenas os trabalhos mais relevantes para o tema em estudo. Diante disso, a *string* final de busca utilizada para pesquisa foi a seguinte: *(parallel machine\*) AND (sched\*) AND ((green) OR (sustainab\*) OR (tariff\*) OR (energ\*) OR (electric\*) OR (carbon\*) OR ("time of use") OR (pollut\*))*. Incluindo apenas o título no campo de busca e sem limitações quanto ao período de busca ou idioma. A busca foi realizada na data 22/01/2024, retornando 68 resultados.

Os artigos resultantes foram lidos na íntegra, sendo selecionados para a revisão aqueles que abordam diretamente o problema de pesquisa. E foram excluídos com base nos seguintes critérios: (1) ausência de abordagem sobre sequenciamento da produção; (2) falta de relevância para o contexto de máquinas paralelas; (3) ausência de considerações ambientais na modelagem; (4) o documento não é de acesso livre.

## 3.2 CATEGORIZAÇÃO DOS TRABALHOS

Na Subseção 3.2.1, são apresentados e avaliados de forma sumarizada os artigos obtidos a partir da busca definida no protocolo. Já na Subseção 3.2.2, é realizado um estudo mais específico sobre a maneira como as questões ambientais são abordadas na literatura, especialmente em relação às máquinas paralelas. Por fim, a Subseção 3.2.3, discute alguns artigos voltados para os problemas de sequenciamento verde de máquinas paralelas e as soluções propostas para eles.

### 3.2.1 Sumarização dos resultados

Dos 68 trabalhos identificados nos resultados, 12 foram excluídos por diversos motivos. 4 trabalhos foram desconsiderados por não abordarem o tema de sequenciamento da produção. 3 não tratavam diretamente do ambiente de máquinas paralelas, sendo 1 deles sobre máquina única e 2 sobre *flowshop*. Outros 4 não abordavam questões ambientais ou não apresentavam considerações ambientais que influenciam nas decisões de programação. Além disso, 1 documento não estava

disponível gratuitamente, impossibilitando sua análise. Os 56 trabalhos restantes foram selecionados para análise e lidos na íntegra. A lista completa dos trabalhos selecionados é apresentada a seguir:

Quadro 1 – Artigos selecionados para a segunda fase da revisão sistemática

<b>ID</b>	<b>Referência</b>	<b>Título do artigo</b>
1	Che <i>et al.</i> (2017c)	Energy-conscious unrelated parallel machine scheduling under time-of-use electricity tariffs
2	Saberi-Aliabad <i>et al.</i> (2020)	Energy-efficient scheduling in an unrelated parallel-machine environment under time-of-use electricity tariffs
3	Zhang <i>et al.</i> (2021)	Two-stage parallel speed-scaling machine scheduling under time-of-use tariffs
4	Sanati <i>et al.</i> (2023)	Unrelated parallel machine energy-efficient scheduling considering sequence-dependent setup times and time-of-use electricity tariffs
5	Cheng <i>et al.</i> (2017)	An Improved Model for Parallel Machine Scheduling Under Time-of-Use Electricity Price
6	Ding <i>et al.</i> (2015)	Parallel Machine Scheduling Under Time-of-Use Electricity Prices: New Models and Optimization Approaches
7	Kurniawan <i>et al.</i> (2020)	Triple-chromosome genetic algorithm for unrelated parallel machine scheduling under time-of-use tariffs
8	Tan <i>et al.</i> (2018)	Economic batch sizing and scheduling on parallel machines under time-of-use electricity pricing
9	Feng e Peng (2024)	Robust identical parallel machine scheduling with two-stage time-of-use tariff and not-all-machine option
10	Gaggero <i>et al.</i> (2023)	Exact and heuristic solution approaches for energy-efficient identical parallel machine scheduling with time-of-use costs
11	Li <i>et al.</i> (2016a)	Parallel machine scheduling problems in green manufacturing industry
12	Safarzadeh e Nikaki (2019)	Bi-objective green scheduling in uniform parallel machine environments
13	Kurniawan <i>et al.</i> (2017)	A Genetic Algorithm for Unrelated Parallel Machine Scheduling Minimizing Makespan Cost and Electricity Cost Under <i>Time-of-Use</i> (TOU) Tariffs with Job Delay Mechanism

Continua na página seguinte

Quadro 1 – Artigos selecionados para a segunda fase da revisão sistemática

<b>ID</b>	<b>Referência</b>	<b>Título do artigo</b>
14	Anghinolfi <i>et al.</i> (2021)	A bi-objective heuristic approach for green identical parallel machine scheduling
15	Jiang <i>et al.</i> (2023)	Approximation algorithms for bi-objective parallel-machine scheduling in green manufacturing
16	Rego <i>et al.</i> (2022)	A mathematical formulation and an NSGA-II algorithm for minimizing the makespan and energy cost under time-of-use electricity price in an unrelated parallel machine scheduling
17	Xue <i>et al.</i> (2019)	Estimation of distribution evolution memetic algorithm for the unrelated parallel-machine green scheduling problem
18	Tan <i>et al.</i> (2019)	Genetic algorithms with greedy strategy for green batch scheduling on non-identical parallel machines
19	Cota <i>et al.</i> (2021)	Bi-criteria formulation for green scheduling with unrelated parallel machines with sequence-dependent setup times
20	Salimifard <i>et al.</i> (2021)	A multi objective volleyball premier league algorithm for green scheduling identical parallel machines with splitting jobs
21	Hidri <i>et al.</i> (2021)	Green Scheduling of Identical Parallel Machines with Release Date, Delivery Time and No-Idle Machine Constraints
22	Zandi <i>et al.</i> (2020)	Green parallel machines scheduling problem: A bi-objective model and a heuristic algorithm to obtain Pareto frontier
23	Zheng e Wang (2016)	A Collaborative Multiobjective Fruit Fly Optimization Algorithm for the Resource Constrained Unrelated Parallel Machine Green Scheduling Problem
24	Xiao <i>et al.</i> (2021)	A branch and bound algorithm for a parallel machine scheduling problem in green manufacturing industry considering time cost and power consumption
25	Angel <i>et al.</i> (2012)	Energy Aware Scheduling for Unrelated Parallel Machines
26	Antoniadis <i>et al.</i> (2020)	Parallel Machine Scheduling to Minimize Energy Consumption

Continua na página seguinte

Quadro 1 – Artigos selecionados para a segunda fase da revisão sistemática

<b>ID</b>	<b>Referência</b>	<b>Título do artigo</b>
27	Pei <i>et al.</i> (2020)	An Approximation Algorithm for Unrelated Parallel Machine Scheduling Under TOU Electricity Tariffs
28	Módos <i>et al.</i> (2021)	On parallel dedicated machines scheduling under energy consumption limit
29	Hu <i>et al.</i> (2023)	Parallel machine scheduling with linearly increasing energy consumption cost
30	Li <i>et al.</i> (2016b)	Unrelated parallel machine scheduling problem with energy and tardiness cost
31	Zeng <i>et al.</i> (2018)	Bi-objective scheduling on uniform parallel machines considering electricity cost
32	Wu e Che (2019)	A memetic differential evolution algorithm for energy-efficient parallel machine scheduling
33	Wang <i>et al.</i> (2023)	Energy-efficient unrelated parallel machine scheduling with general position-based deterioration
34	Heydar <i>et al.</i> (2022)	Approximate dynamic programming for an energy-efficient parallel machine scheduling problem
35	Rocholl <i>et al.</i> (2018)	Electricity power cost-aware scheduling of jobs on parallel batch processing machines
36	Qi <i>et al.</i> (2020)	Multi-objective Parallel-machine Scheduling Problem Considering Energy Consumption and Deteriorating Effect
37	Pan <i>et al.</i> (2018)	A New Imperialist Competitive Algorithm for Multiobjective Low Carbon Parallel Machines Scheduling
38	Lu e Qiao (2017)	An Improved Genetic Algorithm for a Parallel Machine Scheduling Problem with Energy Consideration
39	Wang <i>et al.</i> (2018)	Bi-objective identical parallel machine scheduling to minimize total energy consumption and makespan
40	Lu e Qiao (2020)	A Sustainable Parallel-machine Scheduling Problem with Time Constraint Based on Hybrid Metaheuristic Algorithm

Continua na página seguinte

Quadro 1 – Artigos selecionados para a segunda fase da revisão sistemática

<b>ID</b>	<b>Referência</b>	<b>Título do artigo</b>
41	Pan <i>et al.</i> (2020)	A Knowledge-Based Two-Population Optimization Algorithm for Distributed Energy-Efficient Parallel Machines Scheduling
42	Bok <i>et al.</i> (2024)	The production scheduling problem employing non-identical parallel machines with due dates considering carbon emissions and multiple types of energy sources
43	Abikarram <i>et al.</i> (2019)	Energy cost minimization for unrelated parallel machine scheduling under real time and demand charge pricing
44	Liang <i>et al.</i> (2015)	An Ant Optimization Model for Unrelated Parallel Machine Scheduling with Energy Consumption and Total Tardiness
45	Cheng <i>et al.</i> (2019)	Mixed-integer programming for unrelated parallel machines scheduling problem considering electricity cost and makespan penalty cost
46	Zhang <i>et al.</i> (2020)	A new unrelated parallel machine scheduling problem with tool changes to minimise the total energy consumption
47	Zhu e Tianyu (2019)	A Novel Multi-Objective Scheduling Method for Energy Based Unrelated Parallel Machines With Auxiliary Resource Constraints
48	Zhou <i>et al.</i> (2018)	A multi-objective differential evolution algorithm for parallel batch processing machine scheduling considering electricity consumption cost
49	Wu <i>et al.</i> (2022)	Decomposition approaches for parallel machine scheduling of step-deteriorating jobs to minimize total tardiness and energy consumption
50	Fallahi <i>et al.</i> (2023)	Unrelated parallel batch processing machine scheduling for production systems under carbon reduction policies: NSGA-II and MOGWO metaheuristics
51	Nattaf <i>et al.</i> (2015)	A batch sizing and scheduling problem on parallel machines with different speeds, maintenance operations, setup times and energy costs
52	Wang e Qi (2023)	Scheduling an Energy-Aware Parallel Machine System with Deteriorating and Learning Effects Considering Multiple Optimization Objectives and Stochastic Processing Time
53	Gao <i>et al.</i> (2023)	A tabu search algorithm for the unrelated parallel machine scheduling problem with varied carbon emission constraints in different time intervals

Continua na página seguinte

Quadro 1 – Artigos seleccionados para a segunda fase da revisão sistemática

ID	Referência	Título do artigo
54	Nanthapodej <i>et al.</i> (2021b)	Variable Neighborhood Strategy Adaptive Search to Solve Parallel-Machine Scheduling to Minimize Energy Consumption While Considering Job Priority and Control Makespan
55	Bektur (2022)	An NSGA-II-Based Memetic Algorithm for an Energy-Efficient Unrelated Parallel Machine Scheduling Problem with Machine-Sequence Dependent Setup Times and Learning Effect
56	Nanthapodej <i>et al.</i> (2021a)	Hybrid Differential Evolution Algorithm and Adaptive Large Neighborhood Search to Solve Parallel Machine Scheduling to Minimize Energy Consumption in Consideration of Machine-Load Balance Problems

Fonte: Elaborado pela autora (2024)

Dos 56 artigos analisados, 47 (84%) foram publicados em revistas acadêmicas, enquanto 9 (16%) foram apresentados em conferências. Os 47 artigos publicados em revistas estão distribuídos em um total de 31 periódicos distintos, apresentados no Quadro 2. Dos quais os 9 primeiros se destacam por conter 53,19% dos artigos avaliados, dentre eles estão o *Journal of cleaner production* e o *European Journal of operational research*. O primeiro é um periódico altamente conceituado no campo da sustentabilidade e produção limpa e o segundo é uma das principais revistas internacionais da área de pesquisa operacional e é conhecida por sua alta qualidade editorial e rigor acadêmico.

Quanto à classificação relacionada à quantidade de funções objetivas consideradas nos modelos matemáticos dos artigos avaliados, observou-se que 24 (43%) trabalhos foram classificados como multiobjetivo (com dois critérios de desempenho), enquanto 32 (57%) foram classificados como mono-objetivo, conforme apresentado na Figura 4, distribuídos por ano de publicação.

Embora a abordagem do tema seja relativamente recente, datando desde 2012, a análise de problemas com mais de uma função objetivo é ainda mais (desde 2018), devido à complexidade envolvida na resolução de modelos multiobjetivo. Contudo, quando lidamos com objetivos conflitantes, uma abordagem que considera uma modelagem multiobjetivo pode ser mais vantajosa, pois proporciona um conjunto de soluções que promovem uma competição entre os critérios de desempenho (objetivos), facilitando assim o processo de tomada de decisão.

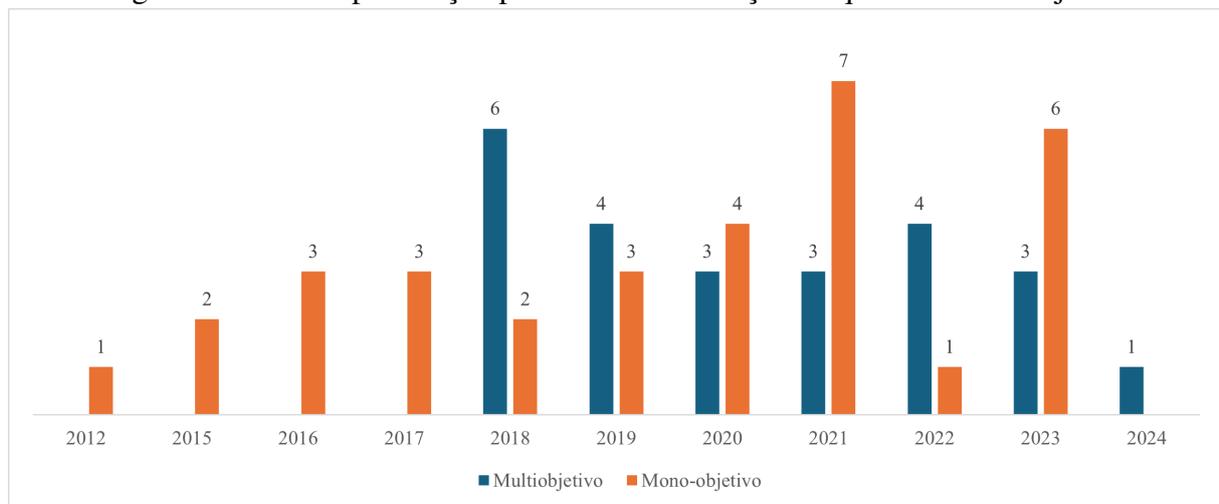
Para isso, entre os critérios de desempenho operacionais apresentados no Quadro 3, tanto os trabalhos classificados como mono-objetivo quanto os multiobjetivo, frequentemente recorrem ao *makespan* na modelagem, seguido por métricas relacionadas ao atraso e tempo total de conclusão. O

Quadro 2 – Periódicos de publicação

Revistas	Total
Journal of cleaner production	6
European journal of operational research	3
IEEE transactions on automation science and engineering	3
International journal of production research	3
Computers & industrial engineering	2
Journal of the operational research society	2
Mathematical problems in engineering	2
Memetic computing	2
Sustainability	2
Annals of mathematics and artificial intelligence	1
Applied intelligence	1
Applied sciences-basel	1
Arabian journal for science and engineering	1
CMES-computer modeling in engineering & sciences	1
Complex & intelligent systems	1
Computers & operations research	1
Engineering optimization	1
Euro journal on computational optimization	1
Expert systems with applications	1
IEEE access	1
IEEE transactions on cybernetics	1
IEEE transactions on systems man cybernetics-systems	1
IEEJ transactions on electrical and electronic engineering	1
International journal of advanced manufacturing technology	1
International transactions in operational research	1
Journal of intelligent manufacturing	1
Journal of manufacturing systems	1
Omega-international journal of management science	1
Operational research	1
Peerj computer science	1
Soft computing	1
Total geral	47

Fonte: Elaborado pela autora (2024)

Figura 4 – Ano de publicação para cada classificação de quantidade de objetivos



Fonte: Elaborado pela autora (2024)

*makespan*, em particular, é um dos mais utilizados devido ao seu papel fundamental em aumentar a eficiência na utilização dos recursos.

Quadro 3 – Desempenhos operacionais vs. Classificação quanto a quantidade de funções objetivo

Desempenhos operacionais	Classificação		Total
	Mono-objetivo	Multiobjetivo	
Atraso	4	3	7
Custos de manutenção	1	–	1
<i>Makespan</i>	23	16	39
<i>Makespan</i> e atraso	1	–	1
<i>Makespan</i> ou tempo total de conclusão	1	–	1
Tempo total de conclusão	2	3	5
Custo total de produção	–	1	1
Número total de máquinas utilizadas	–	1	1
Total	32	24	56

Fonte: Elaborado pela autora (2024)

No caso dos estudos mono-objetivo, em menor quantidade, encontramos um trabalho que avalia *makespan* e atraso (Nanthapodej *et al.* (2021b)), outro que analisa os custos totais de manutenção da produção (Hu *et al.* (2023)) e um terceiro que considera tanto *makespan* quanto tempo total de conclusão em casos separados (Li *et al.* (2016a)). Quanto à classificação multiobjetivo, um trabalho aborda os custos totais de produção, englobando custos de produção, ociosidades, custos fixos e penalidades por atraso (Bok *et al.* (2024)), enquanto outro avalia o número total de máquinas efetivamente utilizadas (Zeng *et al.* (2018)), uma vez que o aumento desse número eleva os custos de mão de obra e reduz a utilização dos recursos.

Quanto aos critérios de desempenho ambientais, tanto os estudos classificados como mono-objetivo quanto os multiobjetivo se destacam pelo uso da função objetivo Custo Total Energético (TEC), conforme mostrado no Quadro 4. Nesse contexto, o TEC pode ser interpretado como o consumo ou custo total energético, o último sendo calculado a partir do consumo por meio de uma taxa de conversão de custo. Outro critério frequentemente considerado é a emissão de carbono, geralmente obtida por meio de uma taxa de conversão do consumo energético.

Quadro 4 – Desempenhos ambientais vs. Classificação quanto a quantidade de funções objetivo

Desempenhos ambientais	Classificação		Total
	Mono-objetivo	Multiobjetivo	
TEC	30	18	48
Emissão de carbono	1	4	5
Emissão de carbono e TEC	-	1	1
Desperdícios totais	-	1	1
Ociosidade	1	-	1
Total	32	24	56

Fonte: Elaborado pela autora (2024)

Em Fallahi *et al.* (2023), são calculados custos de emissão de carbono por meio de duas

políticas distintas: uma de taxação de emissão de carbono, onde é aplicado um imposto por unidade de carbono emitido, multiplicado pelo consumo total de energia; e outra de *cap-and-trade*, que estabelece um limite máximo de emissão de carbono dentro de um determinado período, permitindo a compra ou venda de cotas de carbono pelas empresas, dependendo se sua emissão for menor ou maior que o limite. Similarmente, em Gao *et al.* (2023), uma regulamentação de cotas de carbono é considerada, com um limite de emissão estabelecido para cada período, embora neste caso não haja menção à compra e venda de cotas. Por outro lado, em Bok *et al.* (2024), diferentes fontes de geração de eletricidade são contempladas, cada uma com custos e emissões de carbono específicos.

Em Hidri *et al.* (2021), o objetivo é minimizar o *makespan* enquanto se impede a ocorrência de ociosidade nas máquinas, visando eliminar o consumo de energia ociosa. Portanto, espera-se uma redução no consumo total de energia. Já em Salimifard *et al.* (2021), o foco é a redução do desperdício total em um contexto específico de uma indústria de modelagem por injeção. Nesse cenário, o desperdício é influenciado pela sequência das tarefas, onde a troca de tarefas resulta em resíduos nas máquinas, gerando desperdícios que incluem tempo e materiais de limpeza.

Nos estudos mono-objetivo considerados, a modelagem do problema ocorre de duas formas predominantes: (1) um único critério de desempenho é considerado na função objetivo, enquanto o outro é tratado como restrição; (2) uma única função objetivo é formulada, abrangendo todos os critérios de desempenho do problema por meio de uma soma.

Como observado no Quadro 5, a maioria dos trabalhos avaliados se enquadra na primeira opção (66%), com maior ênfase na minimização do TEC com restrição de horizonte de tempo (associado ao *makespan*), representando 43% de todos os casos. Em seguida, temos a otimização do *makespan* com restrição de TEC e a otimização do tempo total de conclusão com limite de TEC.

Nos trabalhos que apresentam uma única função objetivo, os critérios conflitantes são somados ou tratados de forma ponderada. No primeiro caso, 4 trabalhos buscam minimizar a soma do TEC e de uma penalidade ao *makespan*, enquanto 1 otimiza a soma do TEC e atraso penalizado, e outro minimiza a soma do TEC, *makespan* e atraso penalizado. Já para o segundo caso, 2, 1 e 1 trabalhos, respectivamente, ponderam o tempo total de conclusão com TEC, *makespan* e TEC, e atraso e TEC.

Vale ressaltar que Li *et al.* (2016a) tratam de dois modelos, um que minimiza o *makespan* e outro que otimiza o *tempo total de conclusão*, ambos com restrição de TEC. Jiang *et al.* (2023) estuda a otimização do *makespan* com restrição de TEC e vice-versa. E Angel *et al.* (2012) consideram tanto a ponderação do tempo total de conclusão e TEC quanto a minimização do primeiro limitado pelo segundo.

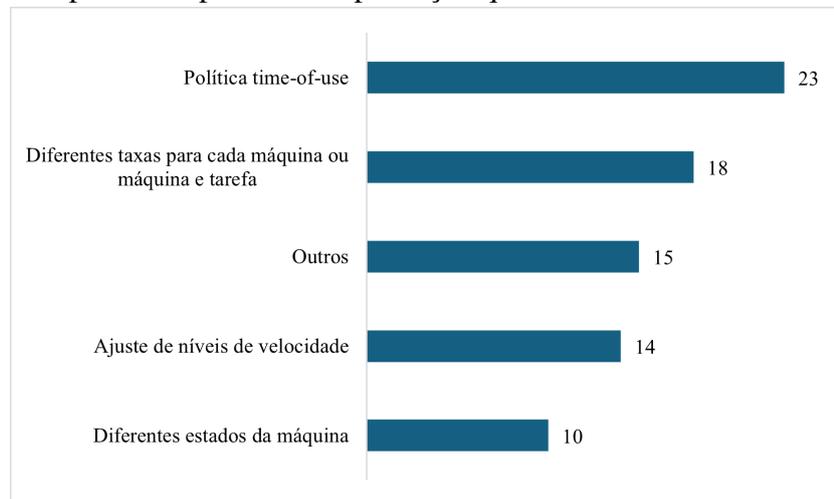
Quadro 5 – Modelagem de trabalhos classificados como mono-objetivo

<b>Função com um único critério de desempenho</b>	
<b>Especificação</b>	<b>Total</b>
Minimizar TEC com restrição do horizonte de tempo	15
Minimizar <i>makespan</i> com limite de TEC	3
Minimizar <i>makespan</i> e com restrição de ociosidade a zero	1
Minimizar tempo total de conclusão com limitação TEC	2
Minimizar TEC com restrição de atendimento no prazo	1
Minimizar <i>makespan</i> com restrição de emissão de carbono	1
Total	23
<b>Função com mais de um critério de desempenho</b>	
<b>Especificação</b>	<b>Total</b>
Minimizar a soma do TEC mais penalização para o <i>makespan</i>	4
Minimizar a ponderação do tempo total de conclusão e TEC	2
Minimizar a ponderação do <i>makespan</i> e TEC	1
Minimizar a soma do custo de manutenção e TEC	1
Minimizar a soma de TEC e atraso penalizado	1
Minimizar a ponderação de atraso e TEC	2
Minimizar a soma do TEC mais <i>makespan</i> e atraso penalizados	1
Total	12

Fonte: Elaborado pela autora (2024)

Observa-se que o TEC é frequentemente empregado devido à capacidade de incorporar diversos aspectos ou políticas da produção que impactam seu desempenho. Além disso, outros fatores são considerados para controlar as emissões de carbono. A Figura 5 destaca alguns desses aspectos que são comumente discutidos pelos autores.

Figura 5 – Aspectos ou políticas da produção que influenciam no controle energético



Fonte: Elaborado pela autora (2024)

Verifica-se que a maior parte dos artigos (41%) adota a política *Time-Of-Use* (TOU), comumente utilizada em diversos países, como técnica para controlar o consumo de energia. Em seguida, 32% dos artigos consideram diferentes taxas de custo ou consumo de energia para cada máquina ou tarefa atribuída a elas. Nesse caso, alguns autores ainda consideram que as máquinas podem diferir, havendo diferentes velocidades fixas para cada uma. Essa abordagem é utilizada com

a premissa de que as máquinas são adquiridas em diferentes períodos de tempo, podem apresentar diferentes tecnologias, entre outros fatores.

Outros aspectos relevantes incluem o ajuste de níveis de velocidade, abordado em 14 artigos (25%), e a consideração dos diferentes estados das máquinas (18%). No primeiro caso, além das decisões de atribuição e sequenciamento das tarefas, deve-se determinar qual nível de velocidade utilizar para processar cada uma delas, uma escolha que impacta tanto o tempo de processamento quanto o consumo energético. No segundo caso, os diferentes estados das máquinas, como em processamento, desligada e ociosa, são considerados, com atribuição de diferentes níveis de consumo energético para cada estado. Além disso, em alguns casos, há também a consideração de um consumo energético fixo associado à mudança de estado.

Outros (25%) artigos, em menor quantidade, consideram diferentes aspectos, tais como:

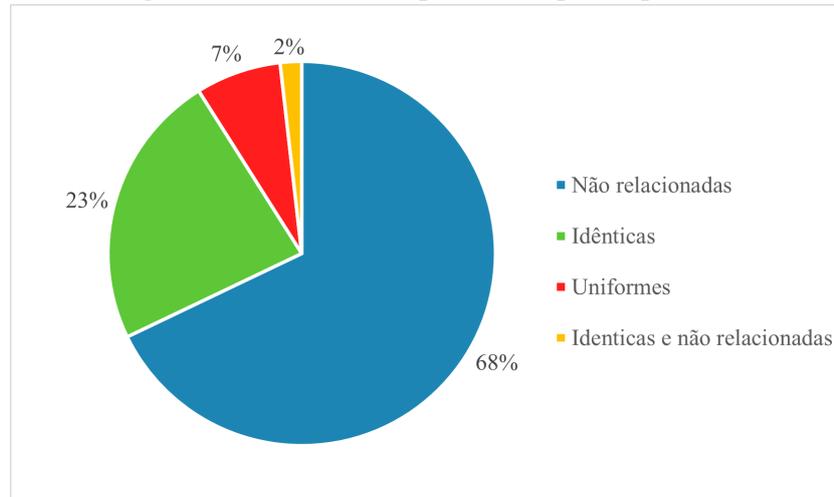
- Opção *not-all-machines*, na qual nem todas as máquinas disponíveis são utilizadas, visto que a inicialização de uma máquina exige um custo fixo (verificado em Feng e Peng (2024)).
- Deterioração da máquina à medida que é utilizada para processar trabalhos, aumentando gradualmente o consumo de energia (observado em Hu *et al.* (2023)).
- Deterioração da tarefa baseada na sua posição de execução, o que aumenta o seu tempo de processamento e, conseqüentemente, os custos de energia (abordado em Wang *et al.* (2023), Qi *et al.* (2020) e Wu *et al.* (2022)).
- Consideração da deterioração e do efeito de aprendizado da máquina, onde o tempo de processamento da tarefa reduz de acordo com a sua posição (observado em Wang e Qi (2023) e Bektur (2022)).
- Desgaste da ferramenta, no qual a troca deve ser realizada quando se atinge um certo limite de desgaste, acarretando em um custo energético (verificado em Zhang *et al.* (2020)).
- Além disso, outros aspectos já mencionados incluem desperdícios dependentes da sequência, eliminação de ociosidades da máquina, consideração de diferentes fontes de energia para o processamento, consumo de energia de aquecimento e imersão em um estudo de caso, taxa de emissão de carbono e controle de emissão de carbono em cada período de tempo.

É importante citar que cerca de 38% dos artigos utilizam mais de uma abordagem. E esses aspectos são discutidos com maior nível de detalhes na Subseção 3.2.2.

Em relação ao contexto específico do ambiente de máquinas avaliado, é perceptível, conforme demonstrado na Figura 6, que a maioria dos estudos se concentra em máquinas paralelas não relacionadas (38 artigos), seguidas por máquinas idênticas (13) e uniformes (4). Além disso, um trabalho se destaca ao considerar dois conjuntos de máquinas paralelas disponíveis para processamento,

um composto por máquinas idênticas e outro por máquinas não relacionadas.

Figura 6 – Diferentes tipos de máquinas paralelas



Fonte: Elaborado pela autora (2024)

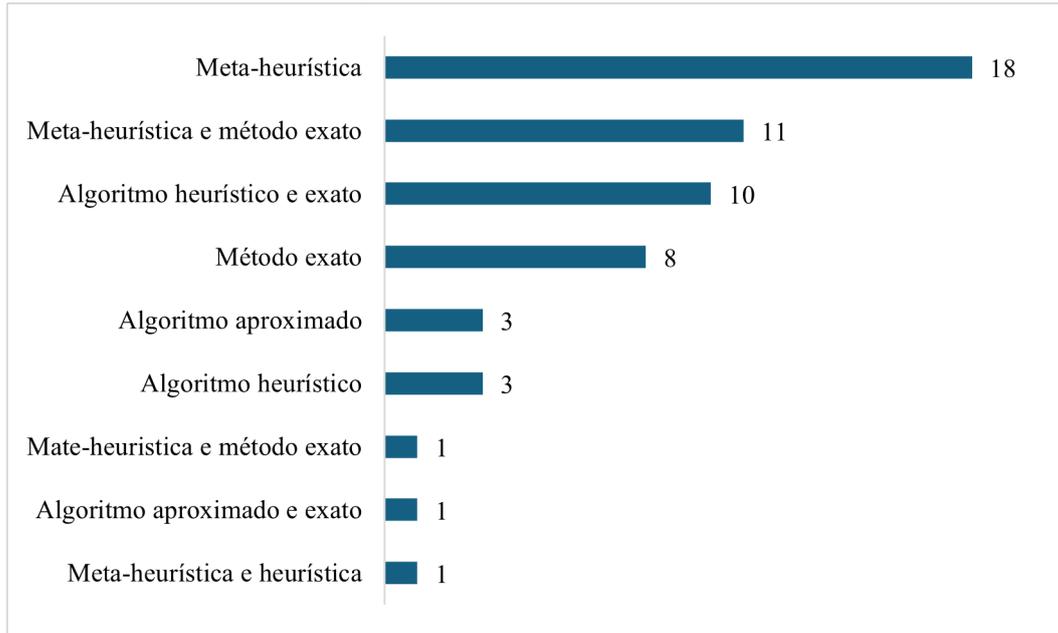
A consideração de máquinas não relacionadas visa adicionar maior realismo aos problemas, uma vez que, em algumas situações, as máquinas podem variar devido a diferenças de idade ou tecnologia empregada. Por outro lado, em muitos casos, as discrepâncias entre as máquinas podem ser irrelevantes. Portanto, considerá-las como idênticas evita despesas e esforços adicionais na coleta de dados e na resolução dos problemas.

Por fim, em relação aos métodos utilizados para resolver os problemas, a Figura 7 destaca o uso predominante de meta-heurísticas, especialmente devido à complexidade dos problemas tratados, que frequentemente são NP-Difíceis. As meta-heurísticas se destacam por sua capacidade de explorar eficientemente o espaço de busca, escapando de ótimos locais e aumentando a probabilidade de encontrar soluções de alta qualidade em tempo razoável.

A maioria das meta-heurísticas se baseia em algoritmos evolutivos de base populacional, que se inspiram na natureza para explorar o espaço de busca de forma eficiente como Zheng e Wang (2016), Qi *et al.* (2020), Pan *et al.* (2018) e outros. Dentre essas abordagens, destacam-se variantes do GA, como os trabalhos de Kurniawan *et al.* (2017), Rego *et al.* (2022), Tan *et al.* (2019), Lu e Qiao (2017), Wang *et al.* (2018) e outros. Além disso, variantes do *Differential Evolution* (DE), como apresentado por Feng e Peng (2024), Zhou *et al.* (2018) e Wu e Che (2019), são amplamente utilizadas. Outras meta-heurísticas significativas são o Algoritmo Memético, como evidenciado nos estudos de Xue *et al.* (2019) e Bektur (2022) e o ACO, apresentado por Liang *et al.* (2015).

É comum também a hibridização de meta-heurísticas, combinando diferentes técnicas para explorar o espaço de busca de maneira mais eficaz. Exemplos incluem a combinação de GA com

Figura 7 – Métodos de resolução



Fonte: Elaborado pela autora (2024)

SA, como demonstrado por Hidri *et al.* (2021), e a combinação de SA e PSO, conforme abordado por Lu e Qiao (2020), hibridização entre o *Non-dominated Sorting Genetic Algorithm-II* (NSGA-II) e DE, conforme estudado por Pan *et al.* (2020), e entre DE e *Adaptive Large Neighborhood Search* (ALNS), como discutido por Nanthapodej *et al.* (2021a). E em relação às heurísticas de solução única, destacam-se a Busca Tabu, conforme apresentada por Zhang *et al.* (2021), Gao *et al.* (2023) e Wu *et al.* (2022), e a VNS, como abordado em trabalhos como Nanthapodej *et al.* (2021b), Wang *et al.* (2023) e Hu *et al.* (2023).

Algoritmos exatos também são bastante aplicados, especialmente em estudos que também exploram meta-heurísticas e algoritmos heurísticos. Eles são frequentemente empregados para validar, comparar e analisar o desempenho dos resultados obtidos por essas abordagens, especialmente em instâncias pequenas. Um dos resolvidores exatos mais utilizados é o Cplex (IBM ILOG CPLEX Optimization Studio), reconhecido por sua eficácia na solução de problemas de otimização.

Nos cenários em que os estudos apresentam uma nova formulação de programação matemática ou aprimoram um modelo já existente na literatura, a aplicação de métodos exatos é fundamental para avaliar a eficácia do modelo proposto. Em particular, Xiao *et al.* (2021) desenvolvem um método de *branch-and-bounds* aprimorado, adaptado às propriedades específicas do Modelo de Programação Linear Inteira Mista (MILP) proposto para sua resolução.

Geralmente, as formulações de modelos se baseiam em programação linear, sendo que apenas um único trabalho explorou um Modelo de Programação Quadrática Inteira Mista (MQIP), empregando um algoritmo de aproximação para sua resolução, apresentado em Pei *et al.* (2020).

Os algoritmos heurísticos desenvolvidos abrangem tanto algoritmos de construção de solução quanto de busca local para aprimoramento. Em geral, eles aplicam o método de restrição- $\epsilon$  para lidar com problemas multiobjetivo. Além disso, 1 artigo específico emprega uma meta-heurística como método de resolução, visualizado em Cota *et al.* (2021).

Em suma, tanto os algoritmos exatos quanto os heurísticos desempenham um papel importante na resolução dos problemas, cada um com suas vantagens e aplicações específicas, dependendo da natureza e complexidade do problema em questão.

Visto isso, o presente estudo visa preencher algumas lacunas identificadas por meio dessa revisão e também incorporar as considerações mais relevantes e tendências observadas. Primeiramente, optou-se por investigar modelos multiobjetivo, uma vez que são mais recentes e ainda pouco explorados na literatura, demandando uma investigação mais aprofundada. Além disso, o ambiente de máquinas paralelas idênticas apresenta aplicações relevantes e carece de uma exploração mais abrangente.

O *makespan* e o TEC foram selecionados como critérios de desempenho a serem analisados, pois são os mais pertinentes para promover um equilíbrio entre a eficiência da produção e as considerações ambientais. Optou-se por abordar a política TOU devido à sua ampla adoção em vários países, proporcionando um controle do consumo energético.

Por fim, será implementada uma meta-heurística devido ao seu potencial para aplicar estratégias que permitam uma busca eficiente de soluções, levando em consideração tanto a diversificação quanto a intensificação. Além disso, o método exato será utilizado como meio de avaliar o desempenho da meta-heurística, especialmente em instâncias de pequena escala.

Além disso, o MILP abordado neste estudo foi inicialmente proposto por Wang *et al.* (2018). Posteriormente, Anghinolfi *et al.* (2021) desenvolveram um artigo onde apresentam uma heurística para resolver este problema. O primeiro artigo foi publicado no *Journal of Cleaner Production* e o segundo no *European Journal of Operational Research*, ambos reconhecidos por disseminarem pesquisas inovadoras e de tendências.

### **3.2.2 Modelagem com questões ambientais**

Esta seção visa explorar como as considerações ambientais são incorporadas nos (PSMP). Essas considerações podem ser observadas tanto nas restrições, que limitam a otimização de medidas de desempenho operacional específicas, quanto na função objetivo, que pode incluir uma medida ambiental única ou ser tratada em conjunto com outras medidas através de soma ponderada em problemas multiobjetivo. O Quadro 6 apresenta uma compilação dos autores discutidos nesta seção,

destacando como os aspectos ambientais são modelados nas restrições dos modelos matemáticos, juntamente com descrições de parâmetros e variáveis relevantes.

Quadro 6 – Restrições com considerações ambientais dos artigos avaliados

Autor	Restrição	Descrição dos parâmetros e variáveis
Li <i>et al.</i> (2016a) e Jiang <i>et al.</i> (2023)	$U = \sum_{i=1}^m \sum_{j=1}^n l_i \times p_j \times x_{ij}$	$n$ : Número de tarefas; $m$ : Número de máquinas; $l_i$ : Custo de processamento por unidade de tempo; $p_j$ : Tempo de processamento da tarefa $j$ ; $x_{ij}$ : Variável que assume 1 se a tarefa $j$ for processada na máquina $i$ ;

Fonte: Elaborado pela autora (2024)

Em Li *et al.* (2016a) as considerações ambientais são abordadas nas restrições do problema, o qual deseja-se realizar um PSMP que diferem no consumo de energia, sem exeder um limite de custo energético ( $\hat{U}$ ), dado que cada máquina  $M_i$  apresenta um custo  $l_i$  por unidade de tempo, logo,  $U \leq \hat{U}$ . Da mesma forma, além de Jiang *et al.* (2023) abordarem esse mesmo problema, também consideram o caso no qual o consumo de energia total é minimizado enquanto existe um prazo comum para o término de todas as tarefas.

Como mencionado anteriormente, as equações matemáticas que incorporam aspectos ambientais também podem ser observadas nas funções de desempenho dos problemas. Portanto, as próximas discussões abordam esse aspecto. O Quadro 7 apresenta uma compilação dos trabalhos discutidos nesta seção, destacando suas equações de desempenho, bem como a descrição de parâmetros e variáveis relevantes.

Quadro 7 – Funções objetivos com considerações ambientais dos artigos avaliados

Autor	Função objetivo	Descrição dos parâmetros e variáveis
Zhang <i>et al.</i> (2020)	<p><math>Min E(C_{max})</math></p> <p><math>Min TEC</math></p> <p>Onde <math>TEC = \sum_{i=1}^n \sum_{j=1}^m \sum_{l=1}^n \sum_{k=1}^r x_{ijkl} p_{ijk} PP_{jk} + \sum_{j=1}^m \sum_{l=1}^n y_{jl} T_c SP_j</math></p>	<p><math>n</math>: Número de tarefas;</p> <p><math>m</math>: Número de máquinas;</p> <p><math>r</math>: Número de níveis de velocidade;</p> <p><math>p_{ijk}</math>: Tempo real de processamento da tarefa <math>i</math> na máquina <math>j</math> na velocidade <math>k</math>;</p> <p><math>PP_{jk}</math>: A potência unitária da máquina <math>j</math> quando sua velocidade de movimento é <math>v_k, k = 1, 2, \dots, r</math>;</p> <p><math>T_c</math>: Tempo de duração da troca de ferramentas;</p> <p><math>SP_j</math>: Potência unitária da máquina <math>j</math> no modo <i>stand-by</i>;</p> <p><math>y_{jl}</math>: Assume 1 se a troca de ferramenta for necessária imediatamente após a conclusão do trabalho na posição <math>l</math> na máquina <math>j</math> e 0 c.c.;</p> <p><math>x_{ijkl}</math>: Assume 1 se a tarefa <math>i</math> for atribuída à posição <math>l</math> na máquina <math>j</math> e sua velocidade de processamento for <math>v_k</math>, e 0 c.c..</p>

Continua na página seguinte

Quadro 7 – Funções objetivos com considerações ambientais dos artigos avaliados

Autor	Função objetivo	Descrição dos parâmetros e variáveis
Wu <i>et al.</i> (2022)	$\text{Min } F = \sum_{j \in N} \alpha_j T_j + \beta \sum_{j \in N} E_j$ <p>Onde <math>E_j \geq e_j + q_j b_j (1 - u_j - z_j) \forall j \in N</math></p>	<p><math>N</math>: Conjunto de Tarefas;</p> <p><math>M</math>: Conjunto de Máquinas;</p> <p><math>E_j</math>: Consumo total de energia extra para concluir o trabalho <math>j</math>;</p> <p><math>T_j</math>: Atraso do trabalho <math>j</math>;</p> <p><math>\alpha_j</math>: Custo de penalidade por atraso no trabalho <math>j</math>;</p> <p><math>\beta</math>: Custo unitário de energia;</p> <p><math>e_j</math>: Energia extra consumida pela tarefa <math>j</math> durante a fase de espera em uma máquina de retenção;</p> <p><math>q_j</math>: Consumo de energia (por unidade de tempo) ao reaquecer o trabalho <math>j</math>;</p> <p><math>b_j</math>: Tempo de processamento de penalidade extra do trabalho <math>j</math>;</p> <p><math>z_j</math>: Assume 1 se o trabalho <math>j</math> for iniciado o mais tardar que seu limite de deterioração, e 0 c.c.;</p> <p><math>u_j</math>: Assume 1 se o <i>status</i> do trabalho <math>j</math> é mantido por um equipamento de retenção e 0 c.c..</p>

Continua na página seguinte

Quadro 7 – Funções objetivos com considerações ambientais dos artigos avaliados

Autor	Função objetivo	Descrição dos parâmetros e variáveis
Wang e Qi (2023)	$\text{Min } E \left( \sum_{i \in M} \sum_{j \in N} \sum_{k \in N} \left( e_{ij}^{\alpha} p'_{ijk} x_{ijk} + e_{ij}^{\alpha} t_{ij} x_{ijk} \right) \right)$ $\text{Min } E(C_{max})$ <p>Onde <math>p'_{ijk} = \left( \frac{p_{ij}}{v_i} \sum_{l \in D} z_{jl} + \alpha_{ij} s_{ijk} \right) k^{\beta_{ij}}</math></p>	<p><math>N</math>: Conjunto de tarefas;</p> <p><math>M</math>: Conjunto de máquinas;</p> <p><math>D</math>: Conjunto de velocidades;</p> <p><math>p_{ij}</math>: Tempo de processamento normal da tarefa <math>j</math> na máquina <math>i</math>;</p> <p><math>\beta_{ij}</math>: Taxa de aprendizado da tarefa <math>j</math> na máquina <math>i</math>;</p> <p><math>\alpha_{ij}</math>: Taxa de deterioração da tarefa <math>j</math> na máquina <math>i</math>;</p> <p><math>v_{il}</math>: Nível de velocidade da máquina <math>i</math>;</p> <p><math>t_{ij}</math>: Tempo de <i>setup</i> da tarefa <math>j</math> na máquina <math>i</math>;</p> <p><math>e_{ij}^{\alpha}</math>: Consumo de energia por unidade de tempo se a tarefa <math>j</math> é processada na máquina <math>i</math>;</p> <p><math>e_{ij}^{\beta}</math>: Consumo de energia por unidade de tempo de <i>setup</i> se a tarefa <math>j</math> é processada na máquina <math>i</math>;</p> <p><math>p'_{ijk}</math>: Tempo de processamento atualizado;</p> <p><math>s_{ijk}</math>: Tempo inicial de processamento da tarefa <math>j</math> na posição <math>k</math> da máquina <math>i</math>;</p> <p><math>x_{ijk}</math>: Assume 1 se a tarefa <math>j</math> for processada na posição <math>k</math> da máquina <math>i</math> e 0 c.c.;</p> <p><math>z_{jl}</math>: Assume 1 se a tarefa <math>j</math> for processada com o nível de velocidade <math>l</math> e 0 c.c..</p>

Continua na página seguinte

Quadro 7 – Funções objetivos com considerações ambientais dos artigos avaliados

Autor	Função objetivo	Descrição dos parâmetros e variáveis
Safarzadeh e Niaki (2019)	$\text{Min } C_{max}$ $\text{Min } \sum_{i=1}^m \sum_{j=1}^n \gamma_i p_j x_{ij}$	<p><math>x_{ij}</math>: Assume 1 se a tarefa <math>j</math> for processada na máquina <math>i</math> e 0 c.c.;</p> <p><math>C_{max}</math>: Corresponde ao <i>makespan</i>;</p> <p><math>p_j</math>: Tempo de processamento da tarefa <math>j</math>;</p> <p><math>\gamma_i</math>: Custo de processamento de uma unidade de tempo na máquina <math>i</math>.</p>

Continua na página seguinte

Quadro 7 – Funções objetivos com considerações ambientais dos artigos avaliados

Autor	Função objetivo	Descrição dos parâmetros e variáveis
Xue <i>et al.</i> (2019)	$\text{Min } C_{max}$ $\text{Min } TEC = \beta \times TEC$ <p>Onde <math>TEC = a \sum_{k=1}^m \sum_{j=1}^n \sum_{i=0}^n t_{jk}^A x_{ijk} + b \sum_{k=1}^m \sum_{j=1}^n \sum_{i=0}^n S_{ijk}^A x_{ijk}</math></p> <p>Onde <math>t_{ik}^A = \frac{t_{ik}}{v_k}</math></p> <p>Onde <math>S_{ijk}^A = \frac{S_{ijk}}{v_k}</math></p>	<p><math>n</math>: Número de tarefas;</p> <p><math>m</math>: Número de máquinas;</p> <p><math>t_{jk}^A</math>: Tempo de processamento atual da tarefa <math>j</math> na máquina <math>k</math>;</p> <p><math>t_{jk}</math>: Tempo de processamento normal da tarefa <math>j</math> na máquina <math>k</math>;</p> <p><math>S_{ijk}^A</math>: Tempo de <i>setup</i> atual da tarefa <math>j</math> ser processada depois da tarefa <math>i</math> na máquina <math>k</math>;</p> <p><math>S_{ijk}</math>: Tempo de <i>setup</i> normal da tarefa <math>j</math> ser processada depois da tarefa <math>i</math> na máquina <math>k</math>;</p> <p><math>v_k</math>: Velocidade de processamento da máquina <math>k</math>;</p> <p><math>a</math>: Coeficiente de consumo energético para o processamento;</p> <p><math>b</math>: Coeficiente de consumo energético para o tempo de <i>setup</i>;</p> <p><math>\beta</math>: Quantidade emitida de carbono por unidade de consumo energético;</p> <p><math>x_{ijk}</math>: Assume 1 se a tarefa <math>i</math> preceder a tarefa <math>j</math> e for processada na máquina <math>k</math>;</p> <p><math>C_{max}</math>: Corresponde ao <i>makespan</i>;</p> <p><math>p_j</math>: Tempo de processamento da tarefa <math>j</math>;</p> <p><math>\gamma_i</math>: Custo de processamento de uma unidade de tempo na máquina <math>i</math>.</p>

Continua na página seguinte

Quadro 7 – Funções objetivos com considerações ambientais dos artigos avaliados

Autor	Função objetivo	Descrição dos parâmetros e variáveis
Nanthapodej <i>et al.</i> (2021b)	$\text{Min } Z = \sum_{m=1}^M \sum_{j=1}^I E_{jm} Y_{jm} + \sum_{j=1}^I a_j T_j + \sum_{m=1}^M B \times \text{Max}_{j=1}^I C_{jm}$	<p><math>i, j</math>: Índice da tarefa;</p> <p><math>m, n</math>: Índice da máquina;</p> <p><math>E_{jm}</math>: Consumo de energia para produzir o trabalho <math>j</math> na máquina <math>m</math>;</p> <p><math>Y_{jm}</math>: Assume 1 se a tarefa <math>j</math> é atribuída a máquina <math>m</math> e 0 c.c.;</p> <p><math>a_j</math>: Custo de penalidade do trabalho <math>j</math> (prioridade do trabalho);</p> <p><math>T_i</math>: Assume 1 se a tarefa <math>i</math> é entregue com atraso e 0 c.c.;</p> <p><math>B</math>: Custo por unidades de tempo de processamento com todas as máquinas;</p> <p><math>C_{jm}</math>: Tempo de conclusão do trabalho <math>j</math> na máquina <math>m</math>.</p>
Nanthapodej <i>et al.</i> (2021a)	$\text{Min } Z = \sum_{m=1}^M \sum_{j=1}^I E_{jm} Y_{jm} + \sum_{m=1}^M B \times \text{Max}_{j=1}^I C_{jm}$	<p><math>i, j</math>: Índice da tarefa;</p> <p><math>m, n</math>: Índice da máquina;</p> <p><math>E_{jm}</math>: Consumo de energia para produzir o trabalho <math>j</math> na máquina <math>m</math>;</p> <p><math>Y_{jm}</math>: Assume 1 se a tarefa <math>j</math> é atribuída a máquina <math>m</math> e 0 c.c.;</p> <p><math>B</math>: Custo por unidades de tempo de processamento com todas as máquinas;</p> <p><math>C_{jm}</math>: Tempo de conclusão do trabalho <math>j</math> na máquina <math>m</math>.</p>

Continua na página seguinte

Quadro 7 – Funções objetivos com considerações ambientais dos artigos avaliados

Autor	Função objetivo	Descrição dos parâmetros e variáveis
Rego <i>et al.</i> (2022)	$\text{Min } C_{max}$ $\text{Min } TEC$ $\text{Onde } TEC \geq \sum_{t=1}^{sizeD} (PEC_t^{off} + PEC_t^{on})$	<p><math>D</math>: Um conjunto de dias no horizonte de planejamento <math>H</math>;</p> <p><math>PEC_t^{on}</math>: Calcula o custo parcial de energia em dólares durante o pico do dia <math>t \in D</math>;</p> <p><math>PEC_t^{off}</math>: Calcula o custo parcial de energia em dólares durante o período fora do pico do dia <math>t \in D</math>;</p> <p><math>PEC_t^{on}</math>: Custo total de energia;</p> <p><math>C_{max}</math>: O tempo máximo de conclusão dos trabalhos.</p>
Anghinolfi <i>et al.</i> (2021)	$\text{Min } C_{max}$ $\text{Min } TEC$ $\text{Onde } TEC = \sum_{h \in H} e_h \sum_{j \in I} \sum_{t=1}^{K-p_j+1} X_{jht} \left( \sum_{i=t}^{t+p_j-1} c_i \right)$	<p><math>J</math>: Conjunto com <math>N</math> tarefas;</p> <p><math>H</math>: Conjunto com <math>M</math> máquinas;</p> <p><math>T</math>: Conjunto do horizonte de tempo de tamanho <math>K</math>;</p> <p><math>e_h</math>: Taxa fixa de consumo energético para cada máquina <math>h</math>;</p> <p><math>c_k</math>: Preço da energia para cada período de tempo <math>k</math>;</p> <p><math>C_{max}</math>: Corresponde ao <i>makespan</i>;</p> <p><math>X_{jht}</math>: Assume 1 se se a tarefa <math>j</math> for processada na máquina <math>h</math> iniciando no período <math>t</math>, e 0 c.c.</p>

Fonte: Elaborado pela autora (2024)

Zhang et al. (2020) apresentam um novo PSMP que considera trocas de ferramentas, onde o consumo energético está diretamente ligado a essas trocas e à velocidade das máquinas. O objetivo do estudo é minimizar simultaneamente o consumo total de energia e o *makespan*. Os autores destacam que no processo de usinagem, por exemplo, o desgaste das ferramentas de corte devido ao atrito afeta a qualidade final do produto. Portanto, as trocas frequentes de ferramentas são necessárias para manter as especificações do produto, embora isso resulte em perda de produtividade e aumento do consumo de energia. Na pesquisa, os autores consideraram dois fatores principais para o desgaste das ferramentas: o tempo de processamento e a velocidade da máquina, onde um aumento na velocidade está associado a uma maior taxa de desgaste.

Nesse problema, as máquinas são não relacionadas e operam com um conjunto de velocidades opcionais. Para uma tarefa  $j$  em uma máquina  $i$  com um tempo de processamento pré-definido  $p_{ij}$ , uma velocidade específica  $v_k$  (onde  $k$  representa os níveis de velocidade) é selecionada. O tempo de processamento real dessa tarefa é então calculado como  $p_{ij}/v_k$ . Além disso, é atribuída uma potência unitária de operação  $PP_{jk}$  para o processamento na máquina  $j$  com o nível de velocidade  $k$ . Para cada máquina, a potência unitária aumenta com o aumento do nível de velocidade. Portanto, a utilização de uma velocidade mais alta na máquina reduz o tempo de processamento, porém resulta em um maior consumo energético (Zhang *et al.*, 2020).

Na mesma pesquisa, há um tempo ocioso fixo na máquina dedicado à troca de ferramentas, acionado quando o acumulado de desgaste ultrapassa um limite superior definido, resultando em um custo de  $SP_j$  por unidade de tempo ocioso. Além disso, cada velocidade  $v_k$  está associada a um tempo unitário de desgaste  $w_k$ , onde o desgaste unitário aumenta com a velocidade, seguindo uma função não decrescente.

O estudo de Wu *et al.* (2022) aborda o PSMP com máquinas idênticas e tarefas sujeitas a deterioração gradual, visando minimizar a soma ponderada do custo de atraso e custo adicional de consumo de energia. Nesse contexto, as tarefas não possuem tempos de processamento fixos e conhecidos previamente; ao contrário, o tempo de processamento pode aumentar à medida que há atrasos ou esperas. Quando todas as tarefas estão disponíveis no período zero, aquelas que iniciam após um limite de deterioração requerem um tempo extra de pré-processamento, seguindo uma função degrau, resultando em um custo adicional de consumo de energia para seu processamento. Além disso, os autores discutem uma alternativa que envolve o uso de um equipamento auxiliar que consome energia continuamente para manter as tarefas prontas para serem processadas no seu tempo “normal”.

A abordagem de deterioração é também explorada por Wang e Qi (2023), onde o tempo de processamento de uma tarefa aumenta conforme seu início é postergado. Além disso, o estudo

considera o efeito do aprendizado, indicando que à medida que uma tarefa é realizada mais tarde, há uma taxa de aprendizado, seja por máquinas ou mão de obra, que pode reduzir esse tempo adicional de processamento.

Nesse caso, os autores consideraram que o tempo de processamento depende do nível de velocidade utilizado na máquina, sendo reduzido à medida que o nível aumenta. Além disso, é adicionado um tempo extra com base numa taxa de deterioração que depende do tempo de início da tarefa. Também é incorporada uma taxa de aprendizado associada a cada posição no agendamento das tarefas, que visa reduzir o tempo de processamento. Assim, o consumo energético total é calculado somando o consumo energético relacionado ao tempo de processamento real das tarefas e o consumo energético do tempo de preparação da máquina necessário para a produção de cada tarefa.

Também, Safarzadeh e Niaki (2019) consideram, em casos práticos, que as máquinas podem ser adicionadas nas estações de produção gradualmente ao longo do tempo, por conta disso, as mesmas podem diferir devido às tecnologias empregadas e/ou os efeitos da degradação. Assim, os autores analisam o problema supondo diferentes custos de processamento em cada máquina. Nesse caso, atribuem uma velocidade especificada  $v_i$  para cada máquina  $i$ , portanto, o processamento da tarefa  $j$  ( $p_j$ ) na máquina  $i$  é definido por  $p_{ij} = \frac{p_j}{v_i}$ , além disso, assumiram que existe um custo  $\beta_i$  associado a cada máquina referente ao consumo energético ou a quantidade de carbono emitida, assim, chamaram de fator de custo  $\gamma_i = \frac{\beta_i}{v_i}$  relacionado ao custo por unidade de tempo de processamento em uma máquina. Nesse estudo, a função objetivo pretende minimizar *makespan* e custo total verde.

De forma parecida, Xue *et al.* (2019) definem que o consumo total de energia é a soma dos consumos de processamento das tarefas em cada máquina e dos tempos de preparação de máquina, que dependem da sequência e de um fator de velocidade. Além disso, existe um coeficiente de consumo associado a cada nível de velocidade de processamento da máquina e ao *setup*. Dessa forma, tanto o tempo de processamento de uma tarefa quanto o *setup* dependente da sequência são atualizados conforme o nível de velocidade selecionado. Por fim, o consumo total de energia é convertido em emissões de carbono, buscando uma minimização simultânea da emissão total de carbono e do *makespan*.

Em Nanthapodej *et al.* (2021b) os autores consideram um consumo  $e_{jm}$  pré-definido para cada tarefa  $j$  ser processada na máquina  $m$ , devido às características de cada tarefa, além de um custo por unidade de tempo de processamento fixo para todas as máquinas. Portanto, a função objetivo neste artigo busca minimizar o consumo energético total, que incorpora o custo de produção de cada tarefa em uma máquina, somado ao custo unitário do tempo total de processamento e uma penalidade pelo atraso na conclusão da tarefa, conforme seu nível de prioridade. Já em Nanthapodej *et al.* (2021a),

apenas o consumo energético é avaliado na função objetivo, semelhante ao descrito anteriormente, com uma restrição adicional de controle de balanceamento de carga entre as máquinas.

Em Rego *et al.* (2022), o custo energético é avaliado por meio de uma política de tarifação de energia que varia ao longo do tempo, sendo discretizado em um horizonte de planejamento de vários dias. Cada dia possui períodos de início e fim de horário de pico, associados a tarifas específicas para esses períodos e outra tarifa para horários fora do pico. Além disso, existem modos de velocidades para o processamento das tarefas relacionadas com um fator de multiplicação do custo. Assim, a função objetivo relacionada ao TEC busca minimizar o custo total de processamento das tarefas, considerando a velocidade de processamento e as tarifas associadas. Da mesma forma, em Anghinolfi *et al.* (2021), é considerado um preço de energia  $c_t$  para cada unidade de tempo  $t$ , e cada máquina  $h$  é associada a um consumo  $e_h$ . O problema abordado visa minimizar tanto o *makespan* quanto o consumo total de energia.

### 3.2.3 Sequenciamento da produção em máquinas paralelas com considerações ambientais

Nesta seção, serão discutidos alguns trabalhos sobre máquinas paralelas com considerações ambientais, abordando diversas características, medidas de desempenho e métodos de resolução desse problema.

Para alguns casos de trabalhos mono-objetivo, Cheng *et al.* (2017) aprimoram um modelo matemático de programação linear inteira mista, o qual visa minimizar o consumo de energia medido pela variação das tarifas energética ao longo do dia (TOU) sujeito a um limite de tempo disponível para o agendamento. Em Liang *et al.* (2015) utiliza-se o ACO para minimizar uma soma ponderada de consumo de energia e atraso total. E Wu *et al.* (2022) estudam a otimização da soma ponderada dos dois desempenhos citados anteriormente para um PSMP idênticas com tarefas em deterioração gradual, utilizando a meta-heurística Busca Tabu integrada com outras técnicas para obter soluções em tempo computacional razoável.

Além disso, Nanthapodej *et al.* (2021b) propõem novo método chamado Busca Adaptativa de Estratégia de Vizinhança Variável para solucionar um PSMP mono-objetivo minimizando o consumo total de energia enquanto considera a prioridade do trabalho e o controle do *makespan*. Já Nanthapodej *et al.* (2021a) propuseram métodos que combinam algoritmos modificados das meta-heurísticas DE e ALNS para resolver o problema que estuda apenas a minimização do consumo total de energia, havendo uma restrição que realiza um balanceamento de carga entre as máquinas. Os autores encontraram resultados satisfatórios quando compararam com os resultados obtidos usando o programa Lingo (método exato) para instâncias pequenas e com o limite inferior gerado pelo Lingo v.11 em 72 horas

de tempo computacional para instâncias médias e grandes.

Em Antoniadis *et al.* (2020) é proposto um algoritmo de aproximação constante que apresenta o objetivo de produzir um cronograma viável com um custo energético mínimo, sujeito a restrição de atendimento das tarefas no prazo estabelecido. Por outro lado, Li *et al.* (2016a) desenvolvem duas heurísticas para um modelo de objetivo único de minimizar o *makespan* ou o somatório dos tempos de conclusão das tarefas sujeito a uma restrição de custo máximo de energia, mostrando, nos resultados, que as heurísticas funcionam muito bem na prática quando comparadas com as soluções ótimas obtidas pelo *Cplex*.

De forma parecida, Jiang *et al.* (2023) propõem um algoritmo de aproximação para um PSMP no qual a função objetivo representa a medida de desempenho de eficiência operacional *makespan* limitado a um consumo energético fixo, e melhoram o algoritmo de Li *et al.* (2016a) para o problema que visa minimizar o consumo energético limitado a um certo tempo máximo disponível para processamento, onde cada máquina apresenta um custo de processamento por unidade de tempo.

Os autores seguem a ideia que, na realidade, normalmente as máquinas são compradas em momentos distintos, por consequência, as organizações dispõem de máquinas mais antigas e mais novas, de modo que essas últimas consomem menos energia, conseqüentemente descarregam uma quantidade menor de poluentes em relação as mais antigas. Assim, os mesmos explanam que os resultados da pesquisa acarretam em importantes questões práticas, uma vez que os algoritmos propostos podem orientar os fabricantes na seleção das máquinas para o processamento das atividades de forma produtiva, atendendo aos clientes no prazo e energeticamente eficiente.

Para os casos mais voltados para a otimização multiobjetivo, Rego *et al.* (2022) propõem uma formulação de programação linear inteira mista para o PSMP não relacionadas com tempos de preparação dependentes da sequência buscando otimizar a TOU e o *makespan*. Além disso, os autores desenvolveram um método adaptado ao problema do algoritmo NSGA-II para resolver instâncias de larga escala. Os resultados foram comparados ao do método exato para instâncias pequenas e a dois algoritmos multiobjetivo para as grandes. No primeiro caso, os autores validaram o NSGA-II, mostrando que o método proposto gera soluções de qualidade e com menor tempo computacional, e no segundo, constataram que o NSGA-II supera os outros dois comparados.

Zheng e Wang (2016) propõem um algoritmo colaborativo multiobjetivo baseado na meta-heurística populacional evolutiva *Fruit Fly Optimization Algorithm* (FOA) para resolver o PSMP não relacionadas com restrições de recursos, objetivando minimizar o *makespan* e a emissão total de carbono.

Shichiyakh *et al.* (2021) trabalharam com a otimização do consumo energético devido

as variações de velocidade nas máquinas, bem como o atraso de conclusão das tarefas, utilizando algoritmos evolucionários (PSO e GA). De forma semelhante, Pan *et al.* (2022) propõem um algoritmo que combina técnicas de DE e de uma meta-heurística baseada no GA para resolver um PSMP não relacionadas com eficiência energética distribuída (i.e., configura-se em uma extensão do PSMP no qual tem-se um conjunto de fábricas disponíveis, cada uma com uma respectiva quantidade de máquinas com diferentes níveis de velocidade) com os mesmos critérios de desempenho mencionados anteriormente.

Em Zhang *et al.* (2020) um novo algoritmo evolutivo é proposto para resolver o PSMP objetivando otimizar em conjunto o consumo total energético, atrelado a troca de ferramentas e a velocidade das máquinas e o *makespan*. Os mesmos objetivos são minimizados em Wang e Qi (2023) por meio de um algoritmo multiobjetivo multi-verso incorporando uma abordagem de simulação estocástica, para solucionar um PSMP com tempos de processamento estocásticos, sendo que a eficiência do algoritmo é mostrada através de comparações com resultados obtidos de outras pesquisas.

Ainda no contexto do PSMPCA, Cota *et al.* (2019) apresentam um modelo de programação inteira mista e um algoritmo *math*-heurístico, Safarzadeh e Niaki (2019) um modelo novo e uma heurística aprimorada e Cota *et al.* (2021) propõem extensões multiobjetivo da metaheurística ALNS com *Learning Automata* (LA), todos para encontrar soluções próximas à fronteira de Pareto para um PSMP bi-objetivo com velocidades variáveis, objetivando minimizar o *makespan* e o consumo total de energia gasto pelas máquinas. Safarzadeh e Niaki (2019) mostraram que a heurística proposta supera a original e comparando as soluções de Pareto da heurística com as soluções exatas para pequenas instâncias, observaram que a heurística obteve aproximações com precisão e rapidamente.

Ademais, Asadpour *et al.* (2022) consideram um modelo bi-objetivo voltado para prazos de entrega como medição do nível de serviço, implementado a meta-heurística SA com o intuito de otimizar o número total de trabalhos atrasados e o consumo total de energia.

Da mesma forma, Anghinolfi *et al.* (2021) apresentam uma heurística construtiva corrigida e melhorada e propõem uma de busca local e Gaggero *et al.* (2023) combinam o MR para otimização multiobjetivo com programação linear inteira mista, ambos para o PSMP idênticas com a finalidade de minimizar o *makespan* e o custo total energia definido por uma taxa de consumo associado a cada máquina e uma tarifa de energia correspondente para cada intervalo de tempo, intencionando processar as atividades em períodos no qual as tarifas de eletricidade são mais baixas. Os primeiros verificaram, por meio de teste computacionais, uma eficácia na abordagem em reduzir o consumo de energia enquanto gera soluções melhores, quando comparado com outros algoritmos.

Em Xue *et al.* (2019), os autores propõem um algoritmo chamado de *Estimation of Distri-*

*bution Evolution Memetic Algorithm* (EDEMA) para resolver o PSMP não relacionadas com critérios de minimização do *makespan* e da emissão total de carbono. Esse é baseado em um algoritmo evolutivo que utiliza um modelo de probabilidade para descrever a distribuição das soluções promissoras no espaço de busca, de forma a desenvolver uma estratégia no cruzamento que obtêm melhores soluções, além de utilizar estruturas de busca em vizinhança para obter descendentes mais eficientes. Os autores mostraram que o algoritmo desenvolvido apresenta um desempenho melhor quando comparado a outros dois da literatura, evidenciando que a utilização de um modelo de probabilidade que controla a direção da evolução é útil para encontrar bons cronogramas.

## 4 METODOLOGIA

A presente pesquisa visa promover conhecimento prático aplicável em sistemas de produção e outras áreas, focando na resolução de problemas de atribuição, ordenação e cronogramação de tarefas a recursos, considerando limitações operacionais e em um tempo computacional razoável. Nesse contexto, classifica-se como pesquisa aplicada, conforme descrito por Silva e Menezes (2005), cujo objetivo é gerar conhecimento com aplicabilidade direta na solução de problemas específicos.

Ademais, quanto aos objetivos, esta pesquisa é exploratória, visto que, segundo Prodanov e Freitas (2013), tem a finalidade de promover maior conhecimento e exploração do assunto investigado. Além disso, quanto à abordagem, é quantitativa, pois, de acordo com o mesmo autor, o problema pode ser quantificado, e as informações são traduzidas em números. Neste estudo, um problema específico de sequenciamento, representado por um modelo matemático, é explorado, apresentando decisões (soluções) que podem ser avaliadas conforme os valores de suas medidas de desempenho.

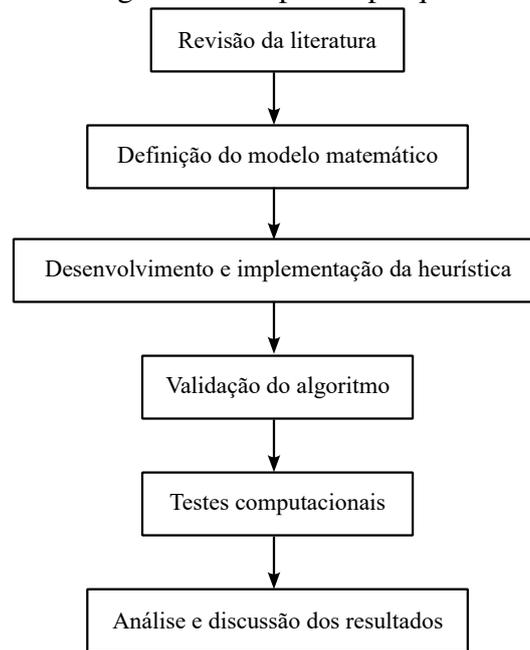
Por fim, o método de pesquisa é caracterizado como simulação, uma vez que, segundo Cauchick-Miguel (2018) a simulação utiliza de técnicas computacionais para realizar simulações que representam determinadas realidades a partir de modelos matemáticos. No caso desta pesquisa, a partir de um modelo matemático que representou um sistema operacional para a tomada de decisão de sequenciamento de tarefas em máquinas paralelas, foram utilizadas técnicas heurísticas para a resolução do problema. Além disso, o algoritmo implementado foi validado e dados foram coletados por meio de experimentos em instâncias de teste, presentes na literatura ou não, buscando identificar o desempenho obtido pela heurística dependendo das variações nos parâmetros. Adicionalmente, os resultados foram comparados com outro método da literatura e com o método de resolução exata.

Este estudo foi conduzido por meio de etapas, conforme ilustrado na Figura 8. Inicialmente, realizou-se uma revisão sistemática da literatura para selecionar o modelo matemático de estudo. Com o modelo definido, procedeu-se à implementação de uma heurística proposta, desenvolvida em linguagem de programação. Posteriormente, foi realizada a validação do algoritmo para verificar se ele descrevia adequadamente o comportamento do sistema real. Nessa etapa da pesquisa, avaliou-se se o algoritmo retornava uma solução viável, atribuindo tarefas às máquinas, definindo uma sequência e cronograma adequados para cada tarefa, com seus respectivos tempos de início e término de processamento.

Em seguida, foram conduzidos experimentos computacionais com diversas instâncias de teste, variando os parâmetros do problema, para identificar o desempenho do algoritmo em termos de soluções e tempo computacional. Os dados foram coletados e analisados para realizar comparações com outros métodos da literatura e com o método de resolução exata, apresentando os resultados por

meio de tabelas e gráficos.

Figura 8 – Etapas da pesquisa



Fonte: Elaborado pela autora (2024)

## 5 DESCRIÇÃO DO PROBLEMA

Neste capítulo detalhamos inicialmente, na Seção 5.1, uma explicação do sequenciamento no ambiente de máquinas selecionado. A Seção 5.2 descreve e justifica as considerações e suposições definidas para o problema. E a Seção 5.3 expõe a notação utilizada para descrever o problema e introduz a formulação de Programação Linear Inteira Mista para resolvê-lo.

### 5.1 O PROBLEMA DE MÁQUINAS PARALELAS

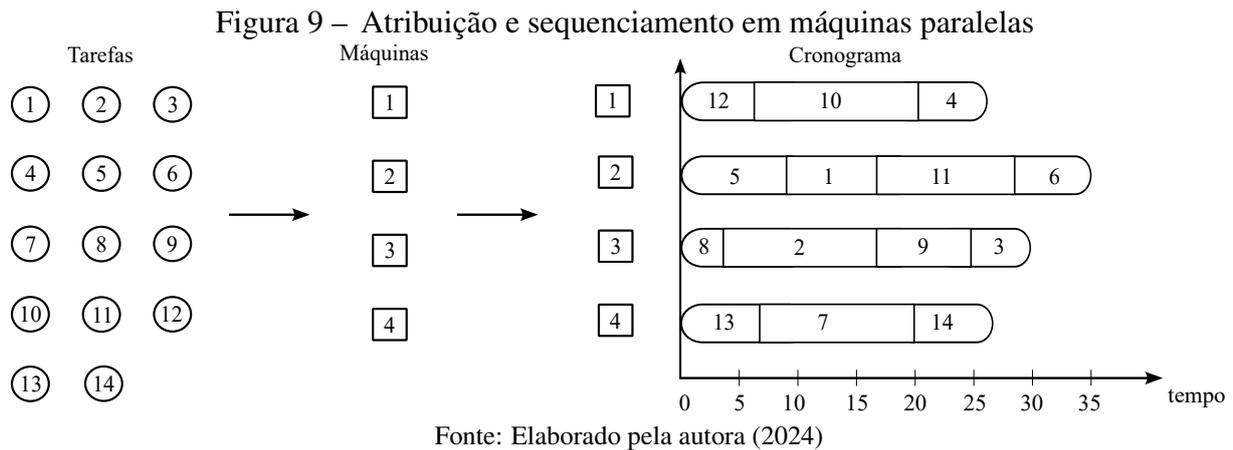
O problema em estudo envolve a sequenciação de um conjunto de tarefas que necessitam ser processadas em um ambiente com múltiplas máquinas simultâneas, sob certas restrições, com o objetivo de otimizar um ou mais critérios. Aspectos ambientais podem ser incorporados tanto nas restrições quanto nos objetivos do problema.

Nesse ambiente, um conjunto  $N = \{1, \dots, n\}$  de tarefas independentes devem ser designadas de maneira eficiente para o conjunto  $H = \{1, \dots, m\}$  de máquinas processarem as atividades em paralelo, no qual existem tempos  $p_{ij}$  de processamento de cada tarefa  $i \in N$  em cada máquina  $j \in H$ , onde cada trabalho deve ser manufaturado exatamente por uma máquina sem preempção (i.e., não admite-se interrupção do processamento de qualquer tarefa com a sua retomada posteriormente), com o objetivo de, normalmente, minimizar uma função objetivo associada a uma medida de desempenho operacional (Fanjul-Peyro *et al.*, 2019).

Por exemplo, na Figura 9, ilustra-se um problema com um total de 14 tarefas a serem processadas em 4 máquinas disponíveis operando em paralelo, cada uma com um tempo de processamento associado. No gráfico de Gantt, observa-se um possível sequenciamento das tarefas nas máquinas: as tarefas 12, 10 e 4 são atribuídas à máquina 1; as tarefas 5, 1, 11 e 6 são atribuídas à máquina 2; as tarefas 8, 2, 9 e 3 são atribuídas à máquina 3; e as tarefas 13, 7 e 14 são atribuídas à máquina 4, seguindo essa mesma ordem. Assim, no gráfico de Gantt, pode-se visualizar o tempo de início e término associado a cada tarefa. Para cada solução, é possível avaliar uma medida de desempenho, como por exemplo, o maior tempo de término (*makespan*), que reflete a disponibilidade das máquinas. Na Figura 9, o *makespan* é de 35 unidades de tempo.

### 5.2 CONSIDERAÇÕES E SUPOSIÇÕES DO PROBLEMA EM ESTUDO

As concessionárias apresentam a dificuldade de fornecer energia de alta qualidade para um conjunto de consumidores de eletricidade em um cenário de alta variabilidade da demanda. Um desafio das empresas do setor público, em particular, é atender de maneira satisfatória a demanda



de energia no período de pico (alta procura energética), no qual dispõem um custo alto pela utilização de mais unidades de geração (Abikarram *et al.*, 2019). Dado que a eletricidade não pode ser armazenada/estocada, isto é, a energia é gerada e fornecida aos clientes instantaneamente, para que as empresas públicas atendam de forma satisfatória seus clientes, uma capacidade reativa sobressalente e dispendiosa deve ser utilizada. Logo, altos custos são necessários para manter as unidades geradoras funcionando constantemente em modo de espera em caso de instabilidade do sistema (Ding *et al.*, 2015).

Visto isso, uma alternativa é convencer os consumidores a utilizarem menos energia em períodos de pico ou deslocarem o consumo para períodos fora do pico. A política de preço por tempo de utilização conhecida como TOU, define diferentes preços de eletricidade ao longo do dia. Assim, os serviços públicos podem definir um preço de eletricidade mais elevado em períodos de alto consumo energético, incentivando os consumidores a mudarem a utilização de energia para períodos onde a tarifa é menor, de modo a evitar uma instabilidade da rede de energia elétrica (Cheng *et al.*, 2017).

Portanto, diante das atuais preocupações ambientais, vários países têm implementado uma política de preços dependentes do horário de utilização (Wang e Li, 2013). Essa estratégia tem se mostrado eficaz na redução do custo total de energia e na mitigação da poluição ambiental.

O esquema TOU é uma das estratégias de precificação da eletricidade mais populares e bem sucedidas utilizadas pelo mundo. Em geral, os períodos são definidos como períodos de pico, médio e fora do pico, no qual a tarifa aumenta do último para o primeiro. Portanto, oferece a oportunidade para as empresas reduzirem o consumo de energia em períodos de alta tarifa e alocarem o consumo para períodos de preço menor. Logo, proporciona benefício tanto para a instalação, com economia de custos, quanto para a concessionária (Zhou *et al.*, 2018).

Além disso, em situações do mundo real, empresas mantêm máquinas novas e mais avançadas funcionando com máquinas mais antigas. Essas máquinas podem diferir em vários fatores,

inclusive no consumo energético (Zhou *et al.*, 2018). Portanto, diferentes taxas de custo/consumo para cada máquina podem ser considerados no momento de realizar a melhor designação de tarefas às máquinas.

Considerando esses aspectos, neste estudo, incorporamos à análise ambiental uma política tarifária conhecida como TOU, juntamente com diferentes taxas de custo energético para cada máquina. As máquinas são consideradas idênticas devido ao tempo de processamento de cada tarefa ser independente da máquina alocada. Essas duas estratégias têm impacto direto no TEC e, conseqüentemente, no *makespan*, representando objetivos conflitantes. Assim, busca-se a minimização simultânea das duas medidas, de modo a obter um conjunto de soluções que conciliem eficiência na produção e controle de custos energéticos.

A otimização da primeira visa reproduzir o contexto prático do mercado tarifário, onde se procura minimizar os custos totais de eletricidade utilizada na fabricação. Isso é relevante, uma vez que os gastos com energia constituem uma parcela significativa dos custos totais de uma empresa. Já a otimização da segunda implica em uma utilização mais eficiente das máquinas, buscando maximizar sua capacidade de produção.

Além disso, para facilitar a análise, focar nas características principais e esclarecer o problema investigado neste artigo, diversas suposições são feitas, a seguir:

- (1) Todas as tarefas chegam simultaneamente uma única vez no instante inicial e estão disponíveis para serem processadas;
- (2) Cada tarefa tem a flexibilidade de ser executada em qualquer máquina disponível no sistema;
- (3) Cada tarefa não pode ser processada por mais de uma máquina;
- (4) Cada máquina é capaz de processar apenas uma tarefa por vez;
- (5) As máquinas operam continuamente, sem interrupções decorrentes de falhas ou avarias, e o consumo de energia durante períodos de ociosidade é insignificante;
- (6) A preempção entre tarefas durante o processo de execução não é permitida, garantindo a integridade das operações;
- (7) Considerações como tempo de configuração, restrições de precedência ou prioridade entre tarefas são desconsideradas;
- (8) Todos os parâmetros são determinísticos.

### 5.3 FORMULAÇÃO MATEMÁTICA

O modelo selecionado para estudo foi proposto inicialmente por Wang *et al.* (2018) e estudado em Anghinolfi *et al.* (2021), no qual envolve o sequenciamento verde à medida que busca a

minimização do custo total energético, chamado de *TEC*, e do *makespan*. Esse modelo também foi trabalhado em Oliveira *et al.* (2023a), onde implementou-se dois métodos de otimização multi-objetivo, conhecidos como restrição- $\epsilon$  e soma ponderada, utilizando da técnica de escalarização para remodelar o problema como mono-objetivo, comparando-os por meio de métricas de performance multi-objetivo.

O problema consiste em sequenciar  $N$  tarefas independentes em  $M$  máquinas paralelas idênticas. Todas as tarefas estão disponíveis para processamento no tempo 0 (i.e., no início do horizonte de tempo), e cada máquina pode processar apenas uma tarefa por vez. Seja  $J = \{1, \dots, N\}$  o conjunto de tarefas a serem processadas, e  $H = \{1, \dots, M\}$  um conjunto de máquinas. Para cada tarefa  $j \in J$ , o tempo de processamento  $p_j$  é idêntico para todas as máquinas. As tarefas devem ser sequenciadas em um horizonte de tempo que consiste em um conjunto  $T = \{1, \dots, K\}$  de intervalos de tempo. As tarefas não podem ser interrompidas, ou seja, cada tarefa  $j$  deve ser processada em um conjunto  $S_j$  de  $p_j$  intervalos de tempo consecutivos em uma única máquina  $h \in H$ .

Devido aos custos variáveis de energia durante o TOU, os intervalos de tempo em  $T$  são organizados em grupos, onde cada grupo engloba uma série de intervalos de tempo que compartilham o mesmo preço de energia. Ou seja, um determinado preço da energia  $c_t$  é atribuído a cada intervalo de tempo  $t \in T$ . Cada máquina  $h \in H$  está associada a uma taxa fixa de consumo de energia  $e_h$ , e o custo do consumo de energia associado ao processamento da tarefa  $j$  na máquina  $h$  é  $e_h \sum_{t \in S_j} c_t$ .

O tempo de conclusão  $C_j$  de uma tarefa  $j$  é o tempo final do último intervalo de tempo designado para a tarefa. Além disso, o  $C_{max}$  de um sequenciamento é o tempo de término do último  $j \in J$ , ou seja,  $C_{max} = \max\{C_j : j \in J\}$ . O valor do *TEC* está associado ao sequenciamento definido como  $\sum_{h \in H} e_h \sum_{j \in J_h} \sum_{k \in S_j} c_k$ , onde  $J_h$  é o conjunto de tarefas sequenciadas na máquina  $h$ . Dessa forma, as variáveis desse problema são:  $X_{jht} \in \{0, 1\}$ ,  $j \in J, h \in H, t \in T$ , que assume valor 1 se a tarefa  $j$  for processada na máquina  $h$  começando no início do intervalo de tempo  $t$ , e 0 caso contrário;  $C_j \geq 0$ ,  $j \in J$ , referente ao tempo de conclusão da tarefa  $j$ ;  $C_{max} \geq 0$ ; e  $TEC \geq 0$ . As notações dos conjuntos, índices, parâmetros e variáveis de decisão para a definição e formulação do problema são sumarizadas no Quadro 8.

Diante disso, este problema pode ser formulado da seguinte maneira:

$$\text{Min } C_{max} \quad (5.1)$$

$$\text{Min } TEC \quad (5.2)$$

$$\text{sujeito a } TEC = \sum_{h \in H} e_h \sum_{j \in J} \sum_{t=1}^{k-p_j+1} X_{jht} \left( \sum_{i=t}^{t+p_j-1} c_i \right), \quad (5.3)$$

$$\sum_{h \in H} \sum_{t=1}^{K-p_j+1} X_{jht} = 1, \quad \forall j \in J, \quad (5.4)$$

Quadro 8 – Notação

Conjuntos	Índices
$J$ : Conjunto de tarefas	$j$ : Índice da tarefa, $j \in J$
$H$ : Conjunto de máquinas	$h$ : Índice da máquina, $h \in H$
$T$ : Conjunto de intervalos de tempo	$t, i$ : Índice da unidade de tempo, $t \in T$
Parâmetros	Variáveis
$N$ : Número de tarefas	$C_j$ : Tempo de conclusão da tarefa $j$
$M$ : Número de máquinas	$C_{max}$ : Tempo máximo de conclusão
$K$ : Horizonte de tempo	$TEC$ : Custo total energético
$p_j$ : Tempo de processamento da tarefa $j$	$X_{jht}$ : Varável binária igual a 1 se a tarefa $j$ for processada na máquina $h$ iniciando no tempo $t$ e 0 c.c.
$e_h$ : Custo da máquina $h$	
$c_t$ : Custo da tarifa no tempo $t$	

Fonte: Elaborado pela autora (2024)

$$\sum_{j \in J} \sum_{i=\max\{0, t-p_j+1\}}^t X_{jhi} \leq 1, \quad \forall t \in T, h \in H, \quad (5.5)$$

$$C_j = \sum_{h \in H} \sum_{t=1}^{K-p_j+1} (t + p_j - 1) X_{jht}, \quad \forall j \in J, \quad (5.6)$$

$$C_{max} \geq C_j, \quad \forall j \in J, \quad (5.7)$$

$$C_{max} \leq K, \quad (5.8)$$

$$C_{max} \geq 0, TEC \geq 0, C_j \geq 0, \quad \forall j \in J, \quad (5.9)$$

$$X_{jht} \in \{0, 1\}, \quad \forall j \in J, h \in H, t \in T. \quad (5.10)$$

As Expressões (5.1) e (5.2) representam as funções objetivos a serem minimizadas, sendo elas  $C_{max}$  e o  $TEC$ , respectivamente. A Equação (5.3) define o cálculo do  $TEC$ . As Restrições (5.4) impõem que cada tarefa seja atribuída a uma única máquina, garantindo que o processamento de qualquer tarefa comece em um único intervalo de tempo em uma única máquina. O conjunto de Restrições (5.5) garantem que no máximo uma única tarefa seja processada em cada intervalo de tempo em cada máquina. As Restrições (5.6) fornecem os horários de conclusão das tarefas, impondo implicitamente a não preempção. As Inequações (5.7) definem o  $C_{max}$ , enquanto que a Inequação (5.8), impõe que o  $C_{max}$  não deva exceder o horizonte temporal disponível (observe que, devido à Equação (5.8), o problema pode não ser solucionável se  $K$  não for suficientemente grande). Finalmente, as Restrições (5.9) e (5.10) definem os domínios das variáveis.

## 6 ALGORITMOS IMPLEMENTADOS

Neste capítulo, são apresentados os algoritmos desenvolvidos para abordar o problema desse estudo. Para instâncias pequenas, é viável obter uma solução exata em tempo computacional razoável. Para isso, emprega-se uma técnica denominada MR exato, discutida detalhadamente na Seção 6.1, que permite alcançar a fronteira de Pareto ótima.

Além disso, dado que o problema abordado é uma generalização do problema de sequenciamento em máquina paralela idêntica, conforme Johnson e Garey (1979), que é NP-difícil, concluímos que ele também pertence a essa classe de problemas. Portanto, serão empregadas heurísticas e meta-heurísticas multi-objetivo para abordá-lo, conforme detalhado na Seção 6.2.

### 6.1 MÉTODO DA RESTRIÇÃO- $\epsilon$ EXATO

O Método da Restrição- $\epsilon$  exato, apresentado no Algoritmo 1, calcula soluções pertencentes à fronteira de Pareto exata, otimizando um único objetivo enquanto mantém o outro como uma restrição, resolvendo através de um algoritmo exato.

Para utilizar este método, são necessárias as seguintes entradas: o conjunto de tarefas  $J = \{1, \dots, N\}$  com os respectivos tempos de processamento  $p_j, j \in J$ ; o conjunto de máquinas disponíveis  $H = \{1, \dots, M\}$ , com as respectivas taxas de consumo  $e_h, h \in H$ ; e o conjunto de unidades de tempo  $T = \{1, \dots, K^{max}\}$ , com os respectivos custos de energia  $c_k, k \in T$ , onde  $K^{max}$  é o maior período do horizonte de tempo.

O método começa com duas listas vazias: a primeira, chamada de *Resultados*, armazena os pares  $TEC$  e  $C_{max}$ , juntamente com a solução de sequenciamento; e a segunda, chamada de *FronteiraMR*, contém as soluções não dominadas e sem repetição dos pares  $TEC$  e  $C_{max}$ .

No Algoritmo 1, a busca por soluções ocorre nas Linhas 3-12, sendo que o objetivo do modelo matemático passa a ser apenas o da Expressão 5.2 do Capítulo 5, e o objetivo da Expressão 5.1 é tratado como restrição do modelo (i.e.,  $C_{max} \leq k$ ). Logo, a cada iteração, o modelo é resolvido na Linha 4 e sua viabilidade é verificada na Linha 5. Caso seja possível encontrar uma solução viável, o número de intervalos de tempo disponíveis para a realização do sequenciamento é reduzido para o valor  $k = C_{max} - 1$ , na Linha 10, e os valores de  $TEC$ ,  $C_{max}$ , bem como a solução de sequenciamento são adicionadas a lista *Resultados* na linha 9. Caso contrário, a *FronteiraMR* é retornada na Linha 14. Nesse algoritmo, o horizonte máximo é gradualmente reduzido a cada iteração até um ponto em que o *solver* do modelo exato não consegue mais encontrar uma solução viável.

---

**Algoritmo 1:** MR- $\epsilon$ 

---

**Entrada:** Conjunto  $J = \{1, \dots, N\}$ , com  $p_j, j \in J$ .

Conjunto  $H = \{1, \dots, M\}$  de máquinas, com a taxa de consumo unitária  $e_h, h \in H$ .

Conjunto  $T = \{1, \dots, K^{max}\}$  de unidades de tempo, com os custos de energia  $c_k, k \in T$ , onde  $K^{max}$  é o maior período do horizonte de tempo.

**Resultado:** Uma fronteira de Pareto ótima.

```

1 Resultados  $\leftarrow \emptyset$ ;
2 FronteiraMR  $\leftarrow \emptyset$ ;
3 while verdadeiro do
4   | TEC, Cmax, S  $\leftarrow$  MetodoExato(J, H, T);
5   | if o resolvidor não encontrar uma solução viável para o modelo then
6   |   | Sair da repetição;
7   | end
8   | else
9   |   | Resultados  $\leftarrow$  Adicionar(TEC, Cmax, S);
10  |   | Kmax = Cmax - 1;
11  |   | end
12 end
13 FronteiraMR  $\leftarrow$  Pares TEC e Cmax de Resultados excluindo soluções dominadas e que
    | apresentam pares iguais;
14 return FronteiraMR

```

---

## 6.2 ALGORITMOS HEURÍSTICOS

Estes algoritmos oferecem uma alternativa ao método exato mencionado na seção anterior, permitindo uma aproximação da fronteira de Pareto-ótima para grandes instâncias em um tempo computacional viável para a tomada de decisões rápidas e eficientes.

A heurística de sequenciamento é apresentada na Subseção 6.2.1 e foi desenvolvida por Anghinolfi *et al.* (2021). Este método corrige e aprimora uma outra proposta recente. Testes computacionais revelaram que essa abordagem supera significativamente outras alternativas comparadas, como as abordagens evolutivas multi-objetivo *Non-dominated Sorting Genetic Algorithm-III* (NSGA-III) e *Multiobjective Evolutionary Algorithm Based on Decomposition* (MOEA/D). Ela é composta por uma heurística construtiva, descrita na Seção 6.2.1.1, e por uma heurística de refinamento, abordada na Seção 6.2.1.2. A combinação dessas duas heurísticas por meio de um algoritmo de restrição- $\epsilon$  para encontrar a fronteira de Pareto aproximada é detalhada na Seção 6.2.1.3. Adicionalmente, uma meta-heurística multiobjetivo é empregada para resolver o problema, como apresentado na Seção 6.2.2. Por sua vez, a Subseção 6.2.3 descreve as abordagens implementadas, que combinam diferentes estratégias e adaptam as heurísticas apresentadas, visando aprimorar as soluções encontradas.

### 6.2.1 Sequenciamento Guloso Dividido

O *Split-Greedy Scheduler* (SGS) é um algoritmo de sequenciamento que combina uma heurística construtiva gulosa, que realiza a alocação preemptiva de tarefas aproveitando os períodos de menor custo energético e posteriormente converte um sequenciamento com preempção de tarefas para um sem preempção, sem alterar os valores de  $TEC$  e  $C_{max}$ , com uma heurística de refinamento que busca melhorar o  $TEC$  sem comprometer o  $C_{max}$  por meio de trocas de blocos, conforme será detalhado posteriormente. Por fim, obtém-se uma fronteira de Pareto aproximada, considerando os *trade-offs* entre o custo total de energia e o *makespan*.

#### 6.2.1.1 Heurística Gananciosa Dividida

A Heurística gananciosa dividida, do inglês *Split-Greedy Heuristic* (SGH), constrói uma solução inicial considerando a minimização do  $TEC$ , com um horizonte de tempo fixo. Para facilitar a compreensão, algumas definições de Anghinolfi *et al.* (2021) são necessárias.

**Definição 6.2.1 (Unidades de tempo adjacentes e consecutivas)** *Duas unidades de tempo  $k$  e  $k + 1$  na mesma máquina são consideradas adjacentes, e duas unidades de tempo  $k$  e  $k + t$ , onde  $t > 1$ , na mesma máquina são consideradas como unidades de tempo consecutivas, se todas as unidades de tempo de  $k + 1$  até  $k + t - 1$  forem utilizadas para processar tarefas, ou seja, não estiverem livres.*

As unidades de tempo são consideradas livres se não estiverem sendo processadas tarefas nelas. Uma localização  $l$  que contém as unidades de tempo  $F \subseteq T$  em uma máquina  $h \in H$  é denotada como  $l = (h, F)$ . O custo energético dessa localização,  $EC_l$ , é dado por  $e_h \sum_{t \in F} c_t$ .

Figura 10 – Exemplo para unidades de tempo adjacentes e consecutivas



Fonte: Elaborado pela autora (2024)

A Figura 10 apresenta um exemplo com 8 unidades de tempo  $t$  na máquina  $h$ , onde estão programadas duas tarefas,  $j$  e  $j'$ . As unidades de tempo 1, 2, 7 e 8 são consideradas livres, pois não estão sendo utilizadas para processar nenhuma tarefa, ao contrário dos períodos de tempo 3, 4, 5 e 6. Os períodos 1 e 2 são ditos livres e adjacentes, assim como os períodos 7 e 8. No entanto, os períodos 2 e 7 são livres e consecutivos, pois todas as unidades de tempo entre eles estão ocupadas.

**Definição 6.2.2 (Localização livre)** *Uma localização livre para uma tarefa  $j$  é uma localização  $(h, F)$  que agrupa apenas unidades de tempo adjacentes e/ou consecutivas livres da máquina  $h$ , de modo que  $|F| = p_j$ .*

**Definição 6.2.3 (Localização dividida)** *Uma localização dividida é uma localização que inclui pelo menos duas unidades de tempo livres consecutivas.*

Uma localização atribuída para uma tarefa  $j$  na máquina  $h$  é representada como  $(h, A_j)$ , onde  $A_j$  é o conjunto de unidades de tempo adjacentes atribuídas à tarefa  $j$  e  $|A_j| = p_j$ . Observe que o horário de início de uma tarefa  $j$ ,  $s_j$ , é igual a  $\min_{t \in A_j} \{t\}$ .

Retomando o exemplo da Figura 10, a tarefa  $j$  é atribuída a uma localização com um par de unidades de tempo adjacentes, enquanto  $j'$  é atribuída a uma localização com um par de unidades de tempo consecutivas, chamada de localização dividida.

Se uma tarefa  $j$  é atribuída a uma localização dividida, diz-se que  $j$  é agendada dividida em  $l$ . Se pelo menos uma tarefa é agendada dividida em  $l$ , temos um sequenciamento dividido, que é o equivalente a um sequenciamento com preempção.

**Definição 6.2.4 (Sequenciamento Dividido)** *Um sequenciamento dividido  $S = \{(h, I_j), j \in J : h \in H, I_j \subseteq T\}$  é um conjunto de localizações atribuídas e localizações divididas atribuídas de tal forma que, para toda localização  $(h, I_i)$  e  $(h, I_j)$  atribuídas às tarefas  $i, j \in J$ ,  $I_i \cap I_j = \emptyset$ , se  $i \neq j$ .*

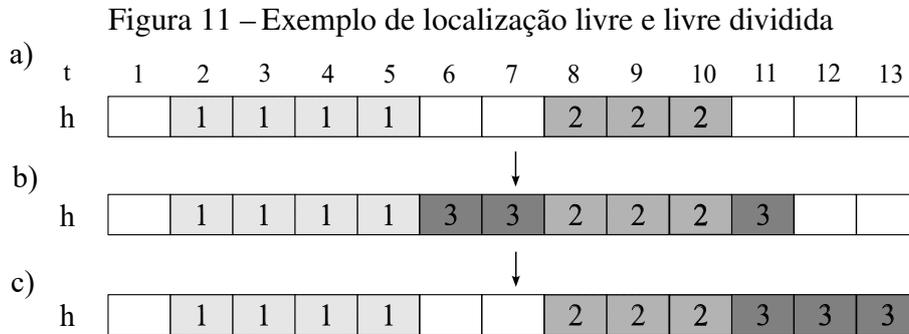
**Definição 6.2.5 (Sequenciamento Viável)** *Um sequenciamento viável  $S = \{(h, A_j), j \in J : h \in H, A_j \subseteq T\}$  é um conjunto de atribuições de tarefas a localizações não divididas, de modo que  $(h, A_j)$  e  $(h, A_{j'}), A_j \cap A_{j'} = \emptyset$ .*

Um sequenciamento viável apresenta apenas locais atribuídos às tarefas com unidades de tempo adjacentes na mesma máquina e satisfaz a condição de não sobreposição para tarefas atribuídas à mesma máquina, ou seja, não pode haver intervalos de tempo comuns.

Visto isso, um sequenciamento dividido  $S'$  pode ser convertido em um sequenciamento viável  $S$  (com tarefas atribuídas a localizações com unidades de tempo adjacentes apenas) sem alterar os valores de  $TEC$  e  $makespan$ . Portanto, pode-se afirmar que um sequenciamento dividido  $S'$  é equivalente a  $S$ , se  $C_{\max}(S') = C_{\max}(S)$  e  $TEC(S') = TEC(S)$ .

Como exemplo, a Figura 11a) mostra duas tarefas, 1 e 2, com tempos de processamento  $p_1 = 4$  e  $p_2 = 3$ , respectivamente. Elas têm suas localizações atribuídas como  $(h, A_1)$ , onde  $A_1 = \{2, 3, 4, 5\}$ , e  $(h, A_2)$ , onde  $A_2 = \{8, 9, 10\}$ , na máquina  $h$ , i.e., seus horários de início são 2 e 8, nessa

ordem. Possíveis localizações livres para a atribuição de uma terceira tarefa ( $j = 3$ ) com tempo de processamento  $p_3 = 3$  incluem  $(h, \{1, 6, 7\})$ ,  $(h, \{6, 7, 11\})$ ,  $(h, \{7, 11, 12\})$  e  $(h, \{11, 12, 13\})$ . As três primeiras são caracterizadas como localizações divididas, enquanto a última é uma localização não dividida, contendo apenas unidades de tempo adjacentes.

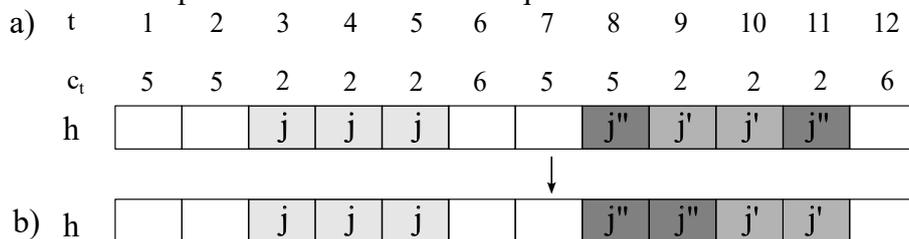


Fonte: Elaborado pela autora (2024)

Uma atribuição da tarefa 3 a qualquer uma das três primeiras localizações geraria um sequenciamento dividido, como evidenciado em 11b) com uma atribuição à localização  $(h, \{6, 7, 11\})$ . No entanto, a atribuição da tarefa 3 à localização  $(h, \{11, 12, 13\})$  gera um sequenciamento viável, ou seja, não preemptado, conforme ilustrado em 11c).

O sequenciamento dividido pode ser convertido em um viável sem alterar os valores de  $TEC$  e  $C_{max}$ . Na Figura 12a), apresenta-se um sequenciamento com a atribuição das tarefas  $j$  e  $j'$  à máquina  $h$  em localizações contendo apenas unidades de tempo adjacentes  $(h, \{3, 4, 5\})$  e  $(h, \{9, 10\})$ , com custo energético  $EC_{I_j} = 6$  e  $EC_{I_{j'}} = 4$ , assumindo  $e_h = 1$ , respectivamente. A tarefa  $j''$  é atribuída à mesma máquina na localização dividida  $(h, \{8, 11\})$ , com  $EC_{I_{j''}} = 7$ .

Figura 12 – Exemplo de conversão de um sequenciamento dividido em um viável



Fonte: Elaborado pela autora (2024)

Portanto, temos um sequenciamento dividido com  $TEC = 17$  e  $C_{max} = 11$ , o qual pode ser convertido em um sequenciamento viável equivalente, conforme mostrado na Figura 12b), com a atribuição das tarefas  $j$ ,  $j'$  e  $j''$  aos respectivos locais  $(h, \{3, 4, 5\})$ ,  $(h, \{8, 9\})$  e  $(h, \{10, 11\})$ , que apresentam apenas unidades de tempo adjacentes.

O pseudocódigo 2 demonstra como converter qualquer sequenciamento dividido  $S$ , que

contém uma coleção de conjuntos de locais atribuídos às tarefas  $j \in J$ , em um sequenciamento viável correspondente.

---

**Algoritmo 2:** Converter\_sequenciamento

---

**Entrada:** Conjunto  $J = \{1, \dots, N\}$  com  $p_j, j \in J$ .

Conjunto  $H = \{1, \dots, M\}$  de máquinas.

Conjunto  $T = \{1, \dots, K^{max}\}$  de unidades de tempo.

Sequenciamento dividido  $S = \{S_h, h \in H\}$ .

**Resultado:** Sequenciamento viável  $S'$  equivalente a  $S$ .

```

1  $S' \leftarrow \{S'_h, h \in H\}$ , no qual  $S'_h = \emptyset, h \in H$ ;
2 for  $h \in H$  do
3    $s_{aux1} \leftarrow 0, s_{aux2} \leftarrow 0$  e  $p \leftarrow 0$ ;
4   while existir uma tarefa  $j$  tal que  $\hat{s}_j = \min\{s_i : i \in J, (h, I_i) \in S_h, s_i > s_{aux2}\}$  do
5      $s'_j \leftarrow \max(s_{aux1} + p, \hat{s}_j)$ ;
6     Atribuir  $l' = (h, \{s'_j, s'_j + 1, \dots, s'_j + p_j - 1\})$  a  $j$  e adicionar  $l'$  a  $S'_h$ ;
7      $s_{aux1} \leftarrow s'_j, s_{aux2} \leftarrow \hat{s}_j$  e  $p \leftarrow p_j$ ;
8   end
9 end
10 return  $S'$ 

```

---

No Algoritmo 2, observe que para cada máquina  $h$  (Linhas 2 a 9), um sequenciamento viável  $S'_h$  é construído utilizando as variáveis  $s_{aux1}$ ,  $s_{aux2}$  e  $p$  como auxiliares (Linha 3). A partir disso, busca-se uma tarefa  $j \in J$  com o menor valor de tempo de início na máquina  $h$  que seja maior que  $s_{aux2}$ , atribuindo esse valor a  $\hat{s}_j$  (Linha 4). Se essa tarefa existir, seu tempo de início é atualizado para o maior valor entre  $s_{aux1} + p$  e  $\hat{s}_j$  (Linha 5). Em seguida, um novo local  $l'$  na máquina  $h$ , contendo apenas unidades de tempo adjacentes ao início em  $s'_j$  e término em  $s'_j + p_j - 1$ , é atribuído à tarefa  $j$ , atualizando o sequenciamento equivalente de  $S'_h$  (Linha 6). Depois disso, os auxiliares  $s_{aux1}$ ,  $s_{aux2}$  e  $p$  são atualizados para  $s'_j$ ,  $\hat{s}_j$  e  $p_j$  (Linha 7).

Portanto, este algoritmo ajusta a alocação de tarefas preemptadas de  $S$  para localizações não divididas, em  $S'$ , sem alterar as unidades de tempo utilizadas para processar tarefas, garantindo que os valores objetivos de  $TEC$  e  $C_{max}$  permaneçam inalterados.

Neste algoritmo, uma correção foi feita em relação ao apresentado por Anghinolfi *et al.* (2021), pois eles consideram apenas as variáveis auxiliares  $s$  e  $p$  para ajustar os tempos de início. Aqui,  $s$  é inicializada como zero na linha 3 e é utilizada nas linhas 4, 5 e 7. Na primeira, é usada para encontrar a tarefa  $j$  com o menor tempo de início que seja maior que  $s$ . Na segunda, o tempo de início da tarefa  $j$  selecionada é ajustado para ser o maior entre  $s + p$  e  $\hat{s}_j$ , e na última,  $s$  é atualizado para  $\hat{s}_j$ .

No entanto, em alguns casos de sequenciamento dividido, a atualização de  $s'_j$  pode ser feita de forma incorreta. Isso ocorre porque o valor de  $s$  atualizado na Linha 7, para ser o menor tempo de

início de uma tarefa com a condição de ser maior que o  $s$  anterior, não deve ser o mesmo utilizado na Linha 4 para ajustar  $s'_j$  na próxima iteração. Esse ajuste deve ser feito para o maior valor entre o tempo de início ajustado da iteração anterior  $s'_j$  mais  $p$  ou  $\hat{s}_j$ .

A heurística construtiva, denominada SGH, busca construir um sequenciamento de maneira iterativa, seguindo a regra do tempo de processamento mais longo e atribuindo as tarefas aos locais com os menores valores de custo energético dentro de um horizonte de tempo limitado. O pseudocódigo desta pode ser verificado no Algoritmo 3.

---

**Algoritmo 3:** Heurística\_gananciosa\_dividida

---

**Entrada:** Conjunto  $J = \{1, \dots, N\}$ , com  $p_j, j \in J$ .

Conjunto  $H = \{1, \dots, M\}$  de máquinas, com a taxa de consumo unitária  $e_h, h \in H$ .

Conjunto  $T = \{1, \dots, K^{max}\}$  de unidades de tempo, com os custos de energia  $c_k, k \in T$ , onde  $K^{max}$  é o maior período do horizonte de tempo.

**Resultado:** Um sequenciamento viável de  $N$  tarefas em um horizonte de tempo  $K^{max}$  em  $M$  máquinas, se existir, c.c., um sequenciamento vazio.

```

1  $S \leftarrow \{S_h, h \in H\}, S_h = \emptyset, h \in H;$ 
2 Construir uma lista  $\hat{P}$  dos elementos de  $P_j$  e os ordenar de forma não-crescente;
3 Para todo  $p \in P_j$ , construir uma lista  $F_p \leftarrow \{j \in J : p_j = p\};$ 
4 for  $p \in \hat{P}$  do
5   for  $h \in H$  do
6     Construir uma lista  $L_{ph}$  com as localizações alternativas na máquina  $h$  para alguma
       tarefa  $j$ , no qual,  $p_j = p;$ 
7   end
8   for  $j \in F_p$  do
9     if  $L_{ph} = \emptyset \forall h \in H$  then
10      return “Não existe solução viável”
11    end
12    else
13      Selecionar aleatoriamente a localização  $\hat{l} = (\hat{h}, \hat{I})$  dentre as localizações de menor
        custo energético provenientes de  $\cup_{h \in H} L_{ph};$ 
14      Designar a localização  $\hat{l}$  para a tarefa  $j$  e adicionar  $\hat{l}$  em  $S_{\hat{h}};$ 
15      Remover de  $L_{p\hat{h}}$  qualquer localização  $(\hat{h}, I)$  no qual  $I \cap \hat{I} \neq \emptyset;$ 
16      Construir as novas localizações divididas  $(\hat{h}, I')$ , com  $|I'| = p$  e inserir em  $L_{p\hat{h}};$ 
17    end
18  end
19 end
20 if O sequenciamento  $S$  construído é dividido then
21   Converter o sequenciamento dividido  $S$  em um viável utilizando o Algoritmo
     Converter_sequenciamento;
22 end
23 return  $S$ 

```

---

Com base no Algoritmo 3, a construção de uma solução inicial começa com a criação de

uma lista  $\hat{P}$  dos tempos de processamento distintos  $p \in P_j$  das tarefas  $j \in J$ , ordenados de forma não-crescente (Linha 2). Em seguida, é construída uma lista de listas  $F$  de tamanho  $|P_j|$ , onde  $F_p$  armazena as tarefas  $j \in J$  com o tempo de processamento  $p$  (Linha 3). Para cada tempo de processamento distinto  $p \in \hat{P}$ , ordenado pela regra do maior tempo de processamento (LPT), as tarefas correspondentes são alocadas em localizações livres (possivelmente divididas) visando obter o menor custo energético (Linhas 4 a 19).

Esse procedimento começa com a construção de uma lista  $L_{ph}$  das localizações livres e livres divididas disponíveis para alocar uma tarefa com tempo de processamento  $p$  para cada máquina  $h \in H$  (Linhas 5 a 7). As localizações livres (e divididas) são construídas ao longo dos intervalos de tempo, iterando uma unidade até alcançar  $K^{max}$ . Uma fila é definida e, se a unidade de tempo  $t$  é considerada livre, ela é enfileirada até que se alcance uma capacidade de  $p$  unidades, indicando que uma nova localização (possivelmente dividida) foi encontrada. O custo energético desse local pode ser definido pelo custo do local anterior retirando o custo da unidade de tempo mais antiga da fila e acrescentando a unidade de tempo mais atual da fila à nova localização.

Portanto, a identificação de um novo local ocorre sempre que um novo período de tempo livre é identificado, removendo o período de tempo mais antigo da fila da localização anterior e acrescentando o novo período de tempo. Assim, o tempo de início desse local é igual ao elemento mais novo da fila.

Para cada tarefa  $j$  com tempo de processamento  $p$  (ou seja,  $j \in F_p$ ), uma localização  $\hat{l}$  dentre as de menor custo energético, considerando todas as máquinas, é selecionada aleatoriamente (Linha 13). Essa seleção uniforme aleatória foi definida em Anghinolfi *et al.* (2021), pois observaram que a escolha aleatória produziu melhores resultados de *TEC* em comparação com uma escolha determinística que prioriza a alocação das tarefas nos primeiros períodos de tempo.

Em seguida, a localização selecionada  $\hat{l} = (\hat{h}, \hat{I})$  é designada para a tarefa  $j$  e adicionada ao sequenciamento da máquina  $\hat{h}$  (Linha 14). Posteriormente, nas Linhas 15 e 16, a lista  $L_{p\hat{h}}$  é atualizada removendo as localizações  $l'$  de  $L_{p\hat{h}}$  onde  $l' \cap \hat{I} \neq \emptyset$  e adicionando as novas localizações divididas que surgem da designação de  $j$  para  $\hat{l}$ . Se não houver localização livre em qualquer uma das máquinas para alocar uma tarefa, isso indica um problema sem solução viável, uma vez que  $K^{max}$  é muito restrito para sequenciar todas as tarefas (Linhas 9 a 11).

É importante salientar que os autores realizam a atualização das Linhas 13 e 16 criando novamente a lista  $L_{p\hat{h}}$ , conforme o explanado sobre a Linha 6. No entanto, para este trabalho, desenvolvemos uma atualização mais eficiente. O Algoritmo 4 mostra um pseudocódigo de atualização de  $L_{p\hat{h}}$  retirando as localizações  $l'$  onde  $l' \cap \hat{I} \neq \emptyset$ . Nesse algoritmo, as variáveis *periodo\_inicial* e

$periodo\_final$  serão utilizadas na próxima atualização, portanto, explicadas posteriormente.

---

**Algoritmo 4:** Atualizar\_Lista\_Locais1

---

**Entrada:** Lista  $L_{p\hat{h}}$  das localizações livres de tamanho  $p$  na máquina  $\hat{h}$ .

O local  $\hat{l}$  selecionado para alocar a tarefa  $j$ .

O tempo de processamento  $p_j$  da tarefa  $j$ .

A posição de  $\hat{l}$  na lista  $L_{p\hat{h}}$ , denotada de  $pos\_local$ .

**Resultado:** Atualização de  $L_{p\hat{h}}$  retirando as localizações  $l'$  onde  $I' \cap \hat{I} \neq \emptyset$ .

```

1 if  $pos\_local > p_j$  then
2    $pos\_del \leftarrow pos\_local - p_j + 1$ ; % Identificar a posição da lista a partir do qual deletar de
    $L_{p\hat{h}}$ 
3    $periodo\_inicial \leftarrow$  primeiro elemento de  $I'$  da posição  $pos\_del$  de  $L_{p\hat{h}}$ ;
4    $periodo\_final \leftarrow$  penúltimo elemento de  $I'$  da posição  $pos\_del$  de  $L_{p\hat{h}}$ ;
5   if  $pos\_local + p_j - 1 \leq |L_{p\hat{h}}|$  then
6     | Deletar as localizações a partir da posição  $pos\_del$  um total de  $2 \times p_j - 1$  vezes;
7   end
8   else
9     | Deletar as localizações de  $L_{p\hat{h}}$  a partir da posição  $pos\_del$  até a última;
10  end
11 end
12 else
13   if  $pos\_local \geq 2$  then
14     |  $periodo\_inicial \leftarrow$  primeiro elemento de  $I'$  da posição 1 de  $L_{p\hat{h}}$ ;
15     |  $periodo\_final \leftarrow$  primeiro elemento de  $I'$  da posição  $pos\_local - 1$  de  $L_{p\hat{h}}$ ;
16   end
17   Deletar as localizações da posição 1 até  $pos\_local$  de  $L_{p\hat{h}}$ ;
18   if  $|L_{p\hat{h}}| \leq p_j$  then
19     | Deletar todas as localizações de  $L_{p\hat{h}}$ ;
20   end
21   else
22     | Deletar as localizações a partir da posição 1 um total de  $p_j - 1$  vezes;
23   end
24 end
25 return  $L_{p\hat{h}}$ 

```

---

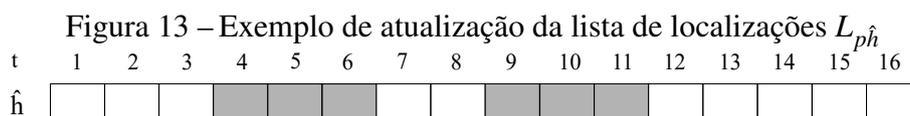
Cada localização em  $L_{p\hat{h}}$  possui uma posição na lista, e como sabemos que as localizações são criadas sequencialmente, elas estão ordenadas de acordo com o primeiro período de tempo de cada local. Portanto, podemos recuperar a posição de  $\hat{l}$  na lista  $L_{p\hat{h}}$  e denotá-la como  $pos\_local$ .

Assim, se  $pos\_local > p_j$  (Linha 1), podemos definir a posição a partir da qual devemos deletar as localizações de  $L_{p\hat{h}}$  que apresentam interseção de períodos de tempo com  $\hat{I}$ , denotada como  $pos\_del$ , como sendo  $pos\_local - p_j + 1$  (Linha 2). Em seguida, verificamos se o tamanho da lista  $L_{p\hat{h}}$  é grande o suficiente (Linha 5) para deletar  $2 \times p_j - 1$  locais  $l'$  de  $L_{p\hat{h}}$  a partir da posição  $pos\_del$

(Linha 6). Caso contrário, deletamos as localizações  $l'$  de  $L_{p\hat{h}}$  da posição  $pos\_del$  até a última (Linha 9).

Por outro lado, se  $pos\_local \leq p_j$  (Linha 12), devemos deletar as localizações  $l'$  de  $L_{p\hat{h}}$  da posição 1 até  $pos\_local$  (Linha 17). Além disso, se o novo tamanho de  $L_{p\hat{h}}$  for menor ou igual a  $p_j$  (Linha 18), deletamos toda a lista (Linha 19); caso contrário, deletamos as localizações  $l'$  de  $L_{p\hat{h}}$  das posições 1 até  $p_j - 1$  (Linha 22). Ao final desse procedimento, obtemos uma lista  $L_{p\hat{h}}$  que não contém localizações  $l'$  onde ocorre interseção entre  $l'$  e  $\hat{I}$  da localização  $\hat{l}$  selecionada para atribuir a tarefa  $j$  (Linha 25).

Para ilustrar o procedimento descrito, considere o exemplo na Figura 13, que mostra um sequenciamento na máquina  $\hat{h}$  com um horizonte de tempo total de 16 unidades. Os quadrados pintados representam alocações de tarefas nos respectivos períodos de tempo, enquanto os quadrados em branco indicam períodos de tempo ociosos.



Fonte: Elaborado pela autora (2024)

Suponha que desejamos alocar uma tarefa  $j$  com tempo de processamento  $p_j = 3$ . A lista  $L_{p\hat{h}}$  de possíveis localizações  $l'$  para a máquina  $\hat{h}$  incluiria os seguintes subconjuntos de períodos de tempo  $l'$ :

- (1)  $\{1, 2, 3\}$ ;
- (2)  $\{2, 3, 7\}$ ;
- (3)  $\{3, 7, 8\}$ ;
- (4)  $\{7, 8, 12\}$ ;
- (5)  $\{8, 12, 13\}$ ;
- (6)  $\{12, 13, 14\}$ ;
- (7)  $\{13, 14, 15\}$ ;
- (8)  $\{14, 15, 16\}$ .

Suponha que a localização selecionada  $\hat{l}$  seja  $\hat{I} = \{7, 8, 12\}$ , com posição em  $L_{p\hat{h}}$  igual a  $pos\_local = 4$ . Verificamos então se  $pos\_local > p_j$ , o que neste caso é verdadeiro, já que  $4 > 3$ . Assim, calculamos  $pos\_del = 4 - 3 + 1 = 2$ , indicando que devemos deletar um conjunto de locais a partir da posição 2 em  $L_{p\hat{h}}$ . Em seguida, verificamos se  $pos\_local + p_j - 1 \leq |L_{p\hat{h}}|$ , ou seja,  $4 + 3 - 1 \leq 8$ . Como esta condição é verdadeira, deletamos  $2 \times 3 - 1 = 5$  locais consecutivos de  $L_{p\hat{h}}$  a partir da posição 2, ou seja, da 2ª à 6ª posição. Estes locais incluem os períodos de interseção 7, 7 e 8,

7, 8 e 12, 8 e 12, e 12, respectivamente. Isso resulta na nova lista  $L_{p\hat{h}}$ :

- (1) {1,2,3};
- (2) {13,14,15};
- (3) {14,15,16}.

Agora, considerando a lista inicial  $L_{p\hat{h}}$ , se a localização selecionada  $\hat{l}$  for aquela com  $\hat{I} = \{13, 14, 15\}$ , então  $pos\_local = 7$ . Dessa forma, a condição da linha 1 do Algoritmo 4 é satisfeita, e  $pos\_del$  é definido como  $7 - 3 + 1 = 5$ . Em seguida, a condição da linha 5 não é satisfeita, resultando na eliminação das localizações das posições 5 até a última. Isso resulta na nova lista  $L_{p\hat{h}}$ :

- (1) {1,2,3};
- (2) {2,3,7};
- (3) {3,7,8};
- (4) {7,8,12}.

Novamente, considerando a lista inicial  $L_{p\hat{h}}$ , uma nova suposição pode ser realizada. Digamos que a localização selecionada  $\hat{l}$  seja aquela com  $\hat{I} = \{1, 2, 3\}$ , então  $pos\_local = 1$ . Nesse caso, a condição da linha 1 não é satisfeita, resultando na eliminação das localizações da posição 1 à 1, ou seja, apenas a 1ª posição é eliminada. Em seguida, a condição da linha 18 não é satisfeita, e o procedimento continua com a eliminação dos locais que agora estão na posição 1 até a  $p_j - 1 = 2$ . Isso resulta na nova lista  $L_{p\hat{h}}$ :

- (1) {7,8,12};
- (2) {8,12,13};
- (3) {12,13,14};
- (4) {13,14,15};
- (5) {14,15,16}.

Após a remoção dos locais de  $L_{p\hat{h}}$  que têm interseção com o subconjunto de períodos de tempo de  $\hat{I}$ , o processo de atualização continua com a construção das novas localizações divididas que surgem da atribuição da tarefa  $j$  ao local  $\hat{l}$ . Este procedimento é apresentado no pseudocódigo do Algoritmo 5.

De acordo com o Algoritmo 5, quando o tempo de processamento  $p_j$  da tarefa  $j$  é igual a 1, e/ou quando a localização  $\hat{l}$  é a primeira da lista  $L_{p\hat{h}}$ , não há necessidade de criar novas localizações divididas. Em vez disso, apenas locais são excluídos conforme o Algoritmo 4.

No entanto, quando esses dois casos não ocorrem, é necessário identificar o intervalo de períodos de tempo para iniciar a construção das novas localizações divididas. Ao contrário do Algoritmo 3, onde a construção enfileira os períodos de tempo livres até alcançar uma capacidade

---

**Algoritmo 5:** Atualizar\_Lista\_Locais2
 

---

**Entrada:** Lista  $L_{p\hat{h}}$  das localizações livres de tamanho  $p$  na máquina  $\hat{h}$ .

O tempo de processamento  $p_j$  da tarefa  $j$  que se deseja alocar.

A posição de  $\hat{l}$  na lista  $L_{p\hat{h}}$ , denotada de  $pos\_local$ .

A posição  $pos\_del$ .

As unidades de tempo  $periodo\_inicial$  e  $periodo\_final$ .

**Resultado:** Atualização de  $L_{p\hat{h}}$  adicionando os locais divididos que surgem a partir da atribuição da tarefa  $j$  ao local  $\hat{l}$  de  $L_{p\hat{h}}$ .

```

1 if  $p_j > 1$  e  $pos\_local \neq 1$  then
2   if  $pos\_del = 0$  then
3      $nova\_pos \leftarrow 1$ ;
4   end
5   else
6      $nova\_pos \leftarrow pos\_del$ ;
7   end
8   for  $t \in periodo\_inicial$  até  $periodo\_final$  do
9     Construir a localização dividida  $l'$  enfileirando os períodos de tempo  $t$  livres até atingir
      o tamanho  $p_j$ ;
10    if  $nova\_pos = 1$  then
11      Adicionar a localização dividida  $l'$  na posição  $nova\_pos$ ;
12      Calcular o custo energético somando cada tarifa de cada período e multiplicando
      pela taxa de consumo da máquina;
13       $nova\_pos \leftarrow nova\_pos + 1$ ;
14    end
15    else
16      Adicionar a localização dividida  $l'$  na posição  $nova\_pos$ ;
17      Calcular o custo energético com base no custo da localização da posição
       $nova\_pos - 1$ ;
18       $nova\_pos \leftarrow nova\_pos + 1$ ;
19    end
20  end
21 end
22 return  $L_{p\hat{h}}$ 

```

---

$p_j$ , começando de 1 até  $K^{max}$ , nessa nova atualização, esse intervalo é determinado pelas variáveis  $periodo\_inicial$  e  $periodo\_final$ , definidas nas linhas 3 e 4 e 14 e 15 do Algoritmo 4. Se  $pos\_local \geq p_j$ ,  $periodo\_inicial$  é o primeiro período de tempo da localização na posição  $pos\_del$  de  $L_{p\hat{h}}$ , e  $periodo\_final$  é o penúltimo período de tempo da localização de mesma posição. Caso contrário, se  $pos\_local \geq 2$ ,  $periodo\_inicial$  e  $periodo\_final$  são, respectivamente, o primeiro período de tempo da localização na posição 1 e  $pos\_local - 1$  de  $L_{p\hat{h}}$ .

Após a identificação dos valores de  $periodo\_inicial$  e  $periodo\_final$ , a construção das localizações é realizada (Linha 9). Se  $nova\_pos = 1$ , a localização  $l'$  criada é adicionada nessa posição em  $L_{p\hat{h}}$  (Linha 11), e seu custo energético é calculado somando a tarifa de energia em cada período

de tempo de  $I'$  multiplicado pela taxa de consumo da máquina (Linha 12). Em seguida,  $nova\_pos$  é atualizada acrescentando uma unidade a ela mesma (Linha 13).

Agora, se  $nova\_pos \geq 2$ , a lista criada  $I'$  é adicionada nessa posição (Linha 16). No entanto, o custo energético dessa localização pode ser calculado com base no custo da localização anterior, i.e., da posição  $nova\_pos - 1$  (Linha 17), seguido da atualização de  $nova\_pos$ .

Assim, a adição das novas localizações  $I'$  é realizada estrategicamente para manter a mesma característica da criação inicial de  $L_{p\hat{h}}$ , na qual os locais são dispostos em  $L_{p\hat{h}}$  de forma sequenciada em  $I'$  para todos  $I' = (\hat{h}, I') \in L_{p\hat{h}}$ . Este processo garante uma estrutura consistente e ordenada que facilita a manipulação subsequente das localizações.

Usando o exemplo inicial para ilustrar o funcionamento do algoritmo, tínhamos a lista  $L_{p\hat{h}}$  com os conjuntos  $I'$  para todas as localizações  $I' \in L_{p\hat{h}}$  da seguinte forma:  $\{1, 2, 3\}$ ,  $\{2, 3, 7\}$ ,  $\{3, 7, 8\}$ ,  $\{7, 8, 12\}$ ,  $\{8, 12, 13\}$ ,  $\{12, 13, 14\}$ ,  $\{13, 14, 15\}$  e  $\{14, 15, 16\}$ .

Ao selecionar a localização  $\hat{I} \in L_{p\hat{h}}$  com  $\hat{I} = \{7, 8, 12\}$  para programar a tarefa  $j$ , sabíamos que neste caso  $pos\_local = 4$ ,  $pos\_del = 2$ ,  $p_j = 3$ , e os períodos de tempo 7, 8 e 12 na máquina  $\hat{h}$  seriam ocupados. Além disso, definimos  $periodo\_inicial = 2$ , o primeiro elemento de  $I'$  na posição  $pos\_del$ , e  $periodo\_final = 3$ , o penúltimo elemento da mesma posição.

Após a remoção das localizações  $I' \in L_{p\hat{h}}$  que apresentavam interseção de  $I'$  com  $\hat{I}$ , restaram:  $\{1, 2, 3\}$ ,  $\{13, 14, 15\}$  e  $\{14, 15, 16\}$ .

Portanto, como  $p_j > 1$  e  $pos\_local \neq 1$ , a criação de novas localizações começa enfileirando os períodos de tempo livres a partir do período 2. Esse processo continua até alcançar o período 3 como o primeiro período de um novo  $I'$ . Assim, o enfileiramento começa com o período 2, seguido pelo período 3 e, em seguida, pelo período 13, formando o  $I' = \{2, 3, 13\}$ . O próximo período livre é o 14, formando o  $I' = \{3, 13, 14\}$ . O primeiro  $I'$  é colocado na posição  $nova\_pos = pos\_del = 2$ , e o segundo na posição  $nova\_pos = nova\_pos + 1 = 3$ , na lista  $L_{p\hat{h}}$ . Isso resulta em uma nova lista  $L_{p\hat{h}}$  com localizações  $I'$  contendo  $I'$  da seguinte forma:

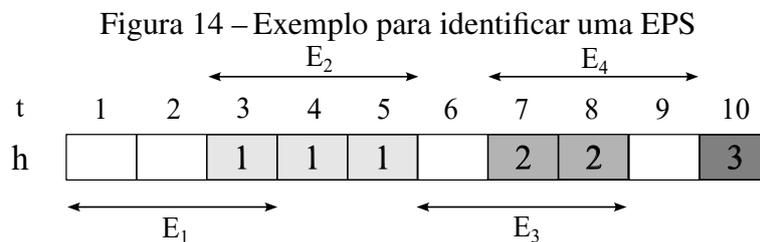
- (1)  $\{1, 2, 3\}$ ;
- (2)  $\{2, 3, 13\}$ ;
- (3)  $\{3, 13, 14\}$ ;
- (4)  $\{13, 14, 15\}$ ;
- (5)  $\{14, 15, 16\}$ .

### 6.2.1.2 Busca Local

A solução obtida pela heurística construtiva SGH pode ser aprimorada por meio de uma busca baseada em trocas, conhecida como *Exchange-search* (ES). Para uma compreensão mais profunda dessa abordagem, são necessárias as seguintes definições.

**Definição 6.2.6 (Sequência de Períodos Trocáveis - EPS)** Um bloco EPS é um subconjunto  $E$  de intervalos de tempo adjacentes em uma máquina específica  $h$  dentro de um cronograma viável  $S$ . Em um EPS, se uma unidade de tempo é utilizada para processar uma tarefa  $j$ , então todas as unidades de tempo atribuídas à tarefa  $j$  devem estar contidas em  $E$ .

Na Figura 14 o subconjunto  $E_1 = \{1, 2, 3\}$  na máquina  $h$  não pode ser considerado um EPS, uma vez que a unidade de tempo 3 é utilizada para executar parte da tarefa 1, porém  $E_1$  não contém todos os períodos utilizados para processar a tarefa 1, ou seja, os períodos 4 e 5 não estão contidos em  $E_1$ . Nesse caso, possíveis EPS's de tamanho 3 seriam  $E_2$ , que abrange todos os períodos utilizados para processar a tarefa 1, e  $E_3$  e  $E_4$ , que englobam todos os períodos usados para processar a tarefa 2.

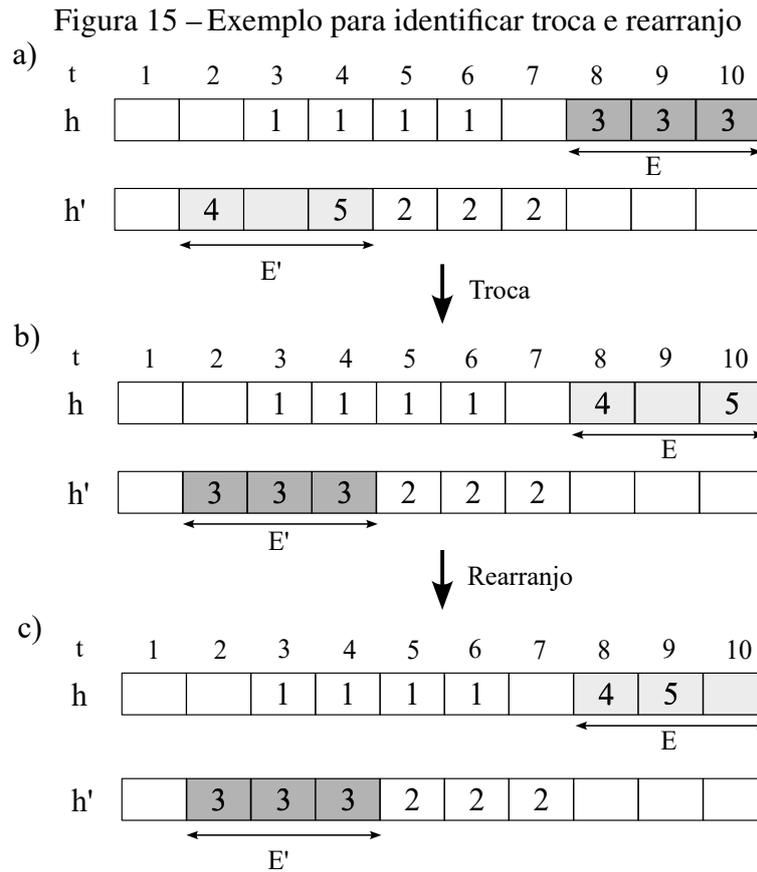


Fonte: Elaborado pela autora (2024)

**Definição 6.2.7 (Troca de EPS)** Uma troca entre dois EPS's  $E$  e  $E'$ , pertencentes às máquinas  $h$  e  $h'$ , respectivamente, tal que  $|E| = |E'|$  e  $E \cap E' = \emptyset$  se  $h = h'$ , é um procedimento que permite o sequenciamento das tarefas atribuídas às unidades de tempo em  $E$ , na máquina  $h$ , para as unidades de tempo em  $E'$ , na máquina  $h'$ , e vice-versa, sem alterar a ordem de atribuição das tarefas dentro das unidades de tempo de  $E$  e  $E'$ .

**Definição 6.2.8 (Rearranjo de EPS)** Um rearranjo de um EPS  $E$  em um sequenciamento viável  $S$  é um procedimento que reatribui o conjunto de tarefas sequenciadas em  $E$ , denotado por  $J_E(S)$ , em localizações com períodos de tempo adjacentes.

A Figura 15 exemplifica uma troca e rearranjo de EPS. Na Figura 15a), temos um sequenciamento viável  $S$  de 5 tarefas.



Fonte: Elaborado pela autora (2024)

Nesta ilustração, o EPS  $E = \{8, 9, 10\}$  na máquina  $h$  é trocado com o EPS  $E' = \{2, 3, 4\}$  na máquina  $h'$ , resultando no sequenciamento em 15b). Em seguida, o rearranjo em  $E$  é realizado, resultando no novo sequenciamento em 15c). Esse rearranjo tem como objetivo atribuir as tarefas envolvidas na troca nos períodos de tempo de menor custo energético.

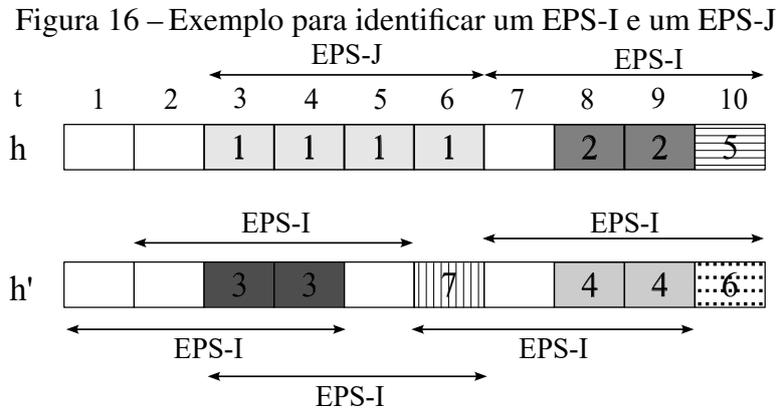
Ainda, conforme ilustrado nas Figuras 14 e 15, é possível notar que nem todas as unidades de tempo em um EPS são atribuídas às tarefas. Nesse caso, podemos distinguir um EPS-J de um EPS-I.

**Definição 6.2.9 (Sequência de Períodos Trocáveis - J (EPS-J))** Um EPS-J é um EPS que contém uma única tarefa, sem nenhuma unidade de tempo ociosa.

**Definição 6.2.10 (Sequência de Períodos Trocáveis - I (EPS-I))** Um EPS-I é um EPS que contém no mínimo uma unidade de tempo ociosa.

Diante disso, a troca de blocos só pode ocorrer entre um EPS-J e um EPS-I que possuem a mesma cardinalidade de  $E$ . Durante esse procedimento, todas as tarefas contidas no EPS-J são atribuídas ao EPS-I, enquanto as tarefas contidas no EPS-I são rearranjadas no EPS-J, caso existam. Esse procedimento combinado de troca e rearranjo é denominado “movimento”.

Para ilustrar um EPS-J e um EPS-I, considere a Figura 16, que mostra um possível sequenciamento  $S$  para 7 tarefas, utilizando as máquinas  $h$  e  $h'$ .



Neste exemplo, um possível EPS-J de tamanho 4 é o subconjunto de  $T$  formado por  $\{3, 4, 5, 6\}$ , que inclui todos os períodos atribuídos à tarefa 1 na máquina  $h$  e não apresenta períodos ociosos. Assim, possíveis EPS-I's trocáveis com esse EPS-J são  $\{7, 8, 9, 10\}$  na máquina  $h$  e  $\{1, 2, 3, 4\}$ ,  $\{2, 3, 4, 5\}$ ,  $\{3, 4, 5, 6\}$ ,  $\{6, 7, 8, 9\}$  e  $\{7, 8, 9, 10\}$  na máquina  $h'$ .

O pseudocódigo da busca local pode ser visto no Algoritmo 6. Este procedimento busca refinar a solução utilizando uma política de primeira melhoria, reduzindo o  $TEC$  sem aumentar o  $makespan$ .

Inicialmente todos blocos denominados de EPS-J e EPS-I para cada tempo de processamento  $p \in \hat{P}$  são criados e armazenados em  $blocos\_epsj$  e  $blocos\_epsi$ , respectivamente.

O EPS-J de cada tarefa  $i \in F_p$  com tempo de processamento  $p$  é formado pelo subconjunto de  $T$ ,  $E_J = \{s_j, s_j + 1, \dots, s_j + p_j - 1\}$ , na máquina  $h_J$ , onde  $s_j$  é o tempo de início da tarefa  $j$ . A identificação de um conjunto EPS-I, formado por  $p$  unidades de tempo adjacentes,  $E_I = \{s, s + 1, \dots, s + p - 1\}$  na máquina  $h_I$  é feita de tal modo que o primeiro período do subconjunto  $E_I$ ,  $s$ , seja um tempo de início de uma tarefa  $j \in J$  ou um período ocioso, e o último período de  $E_I$ ,  $s + p - 1$ , deve ser um tempo de conclusão de uma tarefa  $j \in J$  ou um período ocioso. No entanto, caso  $s$  seja um tempo de início e  $s + p - 1$  seja um tempo de conclusão, deve existir pelo menos uma unidade de tempo livre entre eles. Os subconjuntos de  $T$  formados por períodos adjacentes são testados de 1 até  $K^{max}$ , iterando de uma em uma unidade até completar um subconjunto de períodos de tempo de tamanho  $p$ , para cada máquina.

Para cada tempo de processamento distinto (Linha 4) e para cada EPS-J de  $blocos\_epsj$  (Linha 5) e EPS-I de  $blocos\_epsi$  (Linha 6) de cardinalidade igual a  $p$ , i.e.,  $|EPS_{J_p}| = |EPS_{I_p}| = p$ , a função Viabilidade (Linha 7) verifica se, no melhor caso de movimentação, o  $TEC$  proveniente

**Algoritmo 6:** Busca\_local

**Entrada:** Conjunto  $J = \{1, \dots, N\}$ , com  $p_j, j \in J$ .

Conjunto  $H = \{1, \dots, M\}$  de máquinas, com a taxa de consumo unitária  $e_h, h \in H$ .

Conjunto  $T = \{1, \dots, K^{max}\}$  de unidades de tempo, com os custos de energia  $c_k, k \in T$ , onde  $K^{max}$  é o maior período do horizonte de tempo.

Sequenciamento viável  $S = \{S_h, h \in H\}$ .

Conjunto ordenado de forma não-crescente de tempos de processamento distintos  $\hat{P}$ .

**Resultado:** Um novo sequenciamento viável  $S$ .

```

1 Criar os blocos EPS-I e EPS-J  $\forall p \in \hat{P}$  e armazenar em blocos_epsi e blocos_epsj,
  respectivamente;
2 while melhoria = verdade do
3   melhoria  $\leftarrow$  falso;
4   for  $p \in \hat{P}$  do
5     for  $EPS_{jp} \in \textit{blocos\_epsj}$  do
6       for  $EPS_{ip} \in \textit{blocos\_epsi}$  do
7         viabilidade, novo_tec, menor_custo  $\leftarrow$  Viabilidade( $EPS_{jp}, EPS_{ip}$ );
8         if viabilidade = verdade then
9            $\theta \leftarrow$  Movimento(novo_tec, menor_custo);
10          if  $\theta < TEC(S)$  then
11            Atualizar o Sequenciamento  $S$ ;
12            Atualizar os blocos EPS-I e EPS-J  $\forall p \in \hat{P}$  e armazenar em blocos_epsi
              e blocos_epsj, respectivamente;
13            melhoria  $\leftarrow$  verdade;
14            Retornar para a linha 5;
15          end
16        end
17      end
18    end
19  end
20 end
21 return  $S$ 

```

desta (*novo\_tec*) é melhor que o *TEC* original do sequenciamento  $S$ , se sim, temos que uma atribuição de *verdade* para a variável *viabilidade*, c.c., falso.

O cálculo do *novo\_tec* é realizado retirando do *TEC* de  $S$  o custo do  $EPS_{Jp}$  e do  $EPS_{Ip}$  e somando o custo da ocupação de todas as unidades de tempo pertencentes ao  $EPS_{Ip}$  e do custo da quantidade de unidades de tempo ocupadas no  $EPS_{Ip}$  das tarifas de menor custo do  $EPS_{Jp}$  (*menor\_custo*) nas suas respectivas máquinas.

Se o *novo\_tec* for menor que o *TEC* de  $S$ , i.e., *viabilidade* = *verdade* então realizamos o movimento entre ambos. A função Movimento (Linha 9) realiza a troca e o rearranjo das tarefas de  $EPS_{Ip}$  nos períodos e máquina de  $EPS_{Jp}$  e o cálculo do novo *TEC* proveniente dessa movimentação,  $\theta$ , é definido pela redução de *menor\_custo* de *novo\_tec* e pela adição do real custo de ocupação das tarefas de  $EPS_{Ip}$  nos períodos de  $EPS_{Jp}$ .

Em seguida, se  $\theta$  for menor que o  $TEC$  de  $S$  (Linha 10), o novo sequenciamento e seus respectivos valores de  $TEC$  e  $makespan$  são atualizados (Linha 11), e  $blocos\_epsi$  e  $blocos\_epsj$  são atualizados para todo  $p \in \hat{P}$  (Linha 12). Após isso, o procedimento retorna para a Linha 5. Assim, a heurística de refinamento adota uma política de “primeira melhoria”, repetindo esse processo até que não seja mais possível realizar nenhuma troca de EPS que melhore a solução.

### 6.2.1.3 Heurística Gananciosa Dividida com Busca Local

No algoritmo Heurística Gananciosa Dividida com Busca Local, do inglês, *Split-Greedy Scheduler with Exchange Search* (SGS-ES), o método Restrição- $\varepsilon$  aproximado é empregado para gerar a fronteira aproximada de Pareto, como delineado no Algoritmo 7. Neste algoritmo, a heurística construtiva SGH é combinada com o algoritmo de refinamento ES para minimizar o  $TEC$  dentro de um horizonte de tempo  $\hat{K}$ . A cada iteração, este horizonte é reduzido em uma unidade, e a solução é adicionada ao conjunto de soluções, das quais as dominadas são posteriormente removidas.

---

#### Algoritmo 7: SGS\_ES

---

**Entrada:** Conjunto  $J = \{1, \dots, N\}$ , com  $p_j, j \in J$ .  
 Conjunto  $H = \{1, \dots, M\}$  de máquinas, com a taxa de consumo unitária  $e_h, h \in H$ .  
 Conjunto  $T = \{1, \dots, K^{max}\}$  de unidades de tempo, com os custos de energia  $c_k, k \in T$ , onde  $K^{max}$  é o maior período do horizonte de tempo.

**Resultado:** Uma fronteira aproximada de Pareto, caracterizada pelo par  $(C_{max}, TEC)$ .

- 1  $K^{min} \leftarrow \max\{\lfloor \sum_{j \in J} p_j / M \rfloor, \max_{j \in J} \{p_j\}\};$
- 2  $F' \leftarrow \emptyset;$
- 3  $\hat{K} \leftarrow K^{max};$
- 4 **while**  $\hat{K} \geq K^{min}$  **do**
- 5      $\hat{T} \leftarrow \{1, \dots, \hat{K}\};$
- 6      $S \leftarrow \text{Heuristica\_gananciosa\_dividida}(J, H, \hat{T});$
- 7     **if**  $S$  é um sequenciamento viável **then**
- 8          $S \leftarrow \text{Busca\_local}(J, H, \hat{T}, S);$
- 9     **end**
- 10    **else**
- 11         Sair do *While*;
- 12    **end**
- 13    Reportar  $TEC$  e  $C_{max}$  de  $S$  e adicionar em  $F'$ ;
- 14     $\hat{K} \leftarrow \hat{K} - 1$
- 15 **end**
- 16 **return** Fronteira aproximada  $F'$  com as soluções não dominadas.

---

No algoritmo mencionado, o valor de  $K^{min}$  é determinado como o maior valor entre duas condições:

- (1) O somatório dos tempos de processamento dividido pelo número de máquinas, arredondado para baixo.
- (2) O maior valor dos tempos de processamento  $p_j$  para todas as tarefas  $j$  no conjunto  $J$ .

O conjunto  $F'$  armazena o par  $C_{max}$  e  $TEC$  de cada solução gerada para um horizonte de tempo máximo  $\hat{K}$ , inicializado como  $K^{max}$ . Este horizonte de tempo é reduzido em uma unidade a cada iteração até que  $\hat{K}$  seja menor que  $K^{min}$ , ou até que o horizonte de tempo definido em  $\hat{T} = \{1, \dots, \hat{K}\}$  seja tão pequeno que não seja possível gerar um sequenciamento viável.

Ao final, o conjunto de soluções em  $F$  é analisado e as soluções dominadas são eliminadas. Isso permite manter apenas as soluções que representam o melhor compromisso entre  $TEC$  e  $C_{max}$  para o problema dado.

### 6.2.2 Two-Phase Pareto Local Search

A meta-heurística, denominada de Busca Local de Pareto de Duas fases, do inglês *Two-Phase Pareto Local Search* (2PPLS), é proposta por Lust e Teghem (2010) e utiliza algoritmos aproximativos nas duas fases do método. Estas fases são: (1) Buscar um conjunto aproximado de boas soluções eficientes suportadas, empregando um algoritmo aproximativo com um único objetivo (mono-objetivo), resultante da agregação linear dos múltiplos objetivos do POM inicial; (2) Buscar um conjunto de soluções eficientes não-suportadas aproximadas, obtidas a partir da aplicação de um procedimento de busca em vizinhança.

Ao empregar métodos heurísticos no procedimento, não necessariamente serão obtidas soluções eficientes ou suportadas, mas estas tendem a compor um conjunto próximo das soluções eficientes suportadas.

Na primeira fase, busca-se soluções eficientes suportadas aproximadas, alcançadas por uma heurística construtiva que visa minimizar a soma ponderada dos dois objetivos do problema ( $TEC$  e  $C_{max}$ ). Nesse método, testa-se um conjunto de pesos  $\lambda = \lambda_1, \lambda_2, \dots, \lambda_w$  para a minimização da função mono-objetiva ponderada  $\sum_{i=1:w} f_i \times \lambda_i$ . Este procedimento está detalhado no Algoritmo 8.

No contexto deste trabalho, essa ponderação foi adaptada e incorporada ao algoritmo construtivo *Heuristica\_gananciosa\_dividida*, onde uma localização  $\hat{l}$  é selecionada aleatoriamente dentre aquelas com a menor ponderação entre  $TEC$  e  $C_{max}$ , resultante da atribuição de  $j$  a  $\hat{l}$ , para um conjunto de pesos  $\lambda = \{\lambda_1, \lambda_2\}$ . Ao invés da abordagem original que seleciona dentre aquelas com menor  $TEC$  para um horizonte de tempo  $\hat{K}$ .

Inicialmente, obtêm-se as soluções extremas (Linhas 1-4) resolvendo o problema considerando os conjuntos de pesos  $\lambda = (1, 0)$  e  $\lambda = (0, 1)$ . Ou seja, o primeiro visa otimizar exclusivamente

---

**Algoritmo 8:** Fase\_1
 

---

**Entrada:** Conjunto  $J = \{1, \dots, N\}$ , com  $p_j, j \in J$ .  
 Conjunto  $H = \{1, \dots, M\}$  de máquinas, com a taxa de consumo unitária  $e_h, h \in H$ .  
 Conjunto  $T = \{1, \dots, K^{max}\}$  de unidades de tempo, com os custos de energia  $c_k, k \in T$ , onde  $K^{max}$  é o maior período do horizonte de tempo.

**Resultado:** Uma fronteira  $\hat{F}$  aproximada de Pareto das soluções eficientes suportadas.

- 1  $\lambda \leftarrow (1, 0)$ ;
- 2  $x_1 \leftarrow \text{Heuristica\_gananciosa\_dividida}(J, H, \hat{T}, \lambda)$ ;
- 3  $\lambda \leftarrow (0, 1)$ ;
- 4  $x_2 \leftarrow \text{Heuristica\_gananciosa\_dividida}(J, H, \hat{T}, \lambda)$ ;
- 5  $\hat{F} \leftarrow \{x_1, x_2\}$ ;
- 6  $\hat{F} \leftarrow \text{Heuristica\_recursiva}(x_1, x_2)$ ;
- 7 **return** Fronteira  $\hat{F}$ .

---

o  $TEC$ , obtendo a solução  $x_1$ , e o segundo minimiza somente o  $C_{max}$ , obtendo  $x_2$ . As duas soluções são adicionadas à fronteira  $\hat{F}$  (Linha 5) e um algoritmo recursivo que recebe ambas as soluções é chamado (Linha 6).

A função  $\text{Heuristica\_recursiva}$ , Algoritmo 9, utiliza os valores de  $TEC$  e  $C_{max}$  das duas soluções suportadas ( $x_l$  e  $x_s$ ) que recebe como argumento para calcular um novo conjunto de pesos normalizados (Linha 1) para obter uma nova solução suportada. Uma ponderação do tipo  $TEC \times \lambda_1 + C_{max} \times \lambda_2$  é realizada para resolver o problema mono-objetivo por meio da  $\text{Heuristica\_gananciosa\_dividida}$ , que retorna uma solução  $x_m$  (Linha 3). Essa solução é comparada com todas as outras de  $\hat{F}$ , se essa for dominada por alguma, não é adicionada no conjunto de soluções não-dominadas  $\hat{F}$  (Linha 4).  $\text{Add\_solucao}$  é apresentado no Algoritmo 10.

---

**Algoritmo 9:** Heuristica\_recursiva
 

---

**Entrada:** Duas soluções suportadas  $x_l$  e  $x_r$ .  
**Resultado:** Uma nova solução  $x_m$ .

- 1  $\lambda_1 = \frac{C_{max}(x_l) - C_{max}(x_r)}{C_{max}(x_l) - C_{max}(x_r) + TEC(x_r) - TEC(x_l)}$  e  $\lambda_2 = 1 - \lambda_1$ ;
- 2  $\lambda \leftarrow (\lambda_1, \lambda_2)$ ;
- 3  $x_m \leftarrow \text{Heuristica\_gananciosa\_dividida}(J, H, \hat{T}, \lambda)$ ;
- 4  $\hat{F} \leftarrow \text{Add\_solucao}(x_m, \hat{F})$ ;
- 5 **if**  $z(x_m)$  estiver contida no triângulo formado por  $z(x_l)$ ,  $z(x_r)$  e o ponto ideal formado por  $(z(x_l), z(x_r))$  **then**
- 6      $\hat{F} \leftarrow \text{Heuristica\_recursiva}(x_l, x_m)$ ;
- 7      $\hat{F} \leftarrow \text{Heuristica\_recursiva}(x_m, x_r)$ ;
- 8 **end**
- 9 **return** Fronteira  $\hat{F}$ .

---

Após isso, a heurística recursiva será chamada novamente se a imagem da solução  $x_m$  no

espaço dos objetivos,  $z(x_m)$ , estiver localizada no interior do triângulo formado pelos pontos  $z(x_l)$ ,  $z(x_r)$  e o ponto ideal formado por ambos. Tendo os pares de soluções  $(z(x_l), z(x_m))$  e  $(z(x_m), z(x_r))$  como argumentos (Linhas 6 e 7).

Pontos encontrados localizados sobre a reta que conecta dois vértices do triângulo são desconsiderados para uma nova recursão, visando evitar redundâncias nas ponderações, uma vez que geram combinações idênticas às previamente consideradas (Costa *et al.*, 2015).

---

**Algoritmo 10:** Add\_solução

---

**Entrada:** Uma solução  $x_m$  e a fronteira  $\hat{F}$ .

**Resultado:** Uma nova fronteira  $\hat{F}$ .

```

1 Adicionar  $\leftarrow true$ ;
2 for  $x \in \hat{F}$  do
3   if  $x$  domina  $x_m$  then
4     Adicionar  $\leftarrow false$ ;
5     Sair do for;
6   end
7   if  $x_m$  domina  $x$  then
8     Deletar  $x$  de  $\hat{F}$ ;
9   end
10 end
11 if Adicionar = true then
12   Adicionar  $x_m$  em  $\hat{F}$ ;
13 end
14 return Fronteira  $\hat{F}$ .
```

---

Em Add\_solucão, o procedimento de adição da solução encontrada  $x_m$  à fronteira  $\hat{F}$  ocorre com a comparação da imagem desta solução no espaço de objetivos com cada solução em  $\hat{F}$ . Se houver uma solução em  $\hat{F}$  com objetivos menores ou iguais aos de  $x_m$ , sendo pelo menos um deles melhor, então a solução  $x_m$  não é adicionada a  $\hat{F}$ , uma vez que é dominada, e essa fronteira é retornada. Caso contrário, a solução em  $\hat{F}$  dominada por  $x_m$  é removida de  $\hat{F}$ , e a busca continua até que todas as soluções dominadas sejam removidas, resultando na nova fronteira  $\hat{F}$  ao final do processo.

A segunda fase do método busca encontrar soluções eficientes não-suportadas a partir das soluções suportadas encontradas anteriormente, utilizando uma busca em vizinhança. Esse procedimento é denominado de *Pareto Local Search* (PLS) e foi proposto por Paquete *et al.* (2004). O algoritmo correspondente está detalhado no Algoritmo 11.

Em PLS, duas fronteiras são criadas:  $F_e$ , que armazena as soluções eficientes aproximadas encontradas na primeira fase do 2PPLS e pertencentes a  $\hat{F}$ , e  $F_{aux}$ , uma fronteira vazia usada para auxiliar no desenvolvimento do método. Enquanto a fronteira  $F'$  não estiver vazia (Linhas 4 a 17),

**Algoritmo 11:** PLS

**Entrada:** Conjunto  $J = \{1, \dots, N\}$ , com  $p_j, j \in J$ .

Conjunto  $H = \{1, \dots, M\}$  de máquinas, com a taxa de consumo unitária  $e_h, h \in H$ .

Conjunto  $T = \{1, \dots, K^{max}\}$  de unidades de tempo, com os custos de energia  $c_k, k \in T$ , onde  $K^{max}$  é o maior período do horizonte de tempo.

Uma fronteira  $\hat{F}$  de soluções suportadas aproximadas.

**Resultado:** Uma fronteira aproximada  $F_e$  das soluções eficientes.

```

1  $F_e \leftarrow \hat{F}$ ;
2  $F' \leftarrow \hat{F}$ ;
3  $F_{aux} \leftarrow \emptyset$ ;
4 while  $|F'| \neq 0$  do
5   for  $x \in F'$  do
6     for vizinho  $x'$  de  $x$  do
7       if  $x'$  não é dominado por  $x$  then
8          $F_e \leftarrow \text{Add\_solução}(x', F_e)$ ;
9         if Se  $x'$  foi adicionado em  $F_e$  then
10           $F_{aux} \leftarrow \text{Add\_solução}(x', F_{aux})$ ;
11        end
12      end
13    end
14  end
15   $F' \leftarrow F_{aux}$ ;
16   $F_{aux} \leftarrow \emptyset$ ;
17 end
18 return Fronteira eficiente aproximada  $F_e$ .
```

para cada solução  $x$  em  $F'$  (Linhas 5 a 14), procuramos por um vizinho  $x'$  de  $x$ . Se  $x'$  não for dominado por  $x$  (Linhas 7 a 12), verificamos em  $\text{Add\_solução}$  se  $x'$  não é dominado por nenhuma outra solução ou se ele domina alguma em  $F_e$ , fazendo os ajustes necessários em  $F_e$  (Linha 8). Em seguida, se  $x'$  for adicionado a  $F_e$ , aplicamos novamente PLS para adicionar  $x'$  a  $F_{aux}$  e eliminamos as soluções de  $F_{aux}$  que são dominadas por  $x'$ . Após examinar todos os vizinhos de todas as soluções em  $F'$ , as novas soluções encontradas e armazenadas em  $F_{aux}$  são transferidas para  $F'$ , e  $F_{aux}$  é reinicializado como vazio (Linhas 15 e 16). O algoritmo termina quando  $F'$  se torna vazio.

### 6.2.3 Abordagem proposta

Neste trabalho, foram testadas cinco abordagens distintas, e o entendimento de uma delas requer a apresentação de uma definição.

**Definição 6.2.11 (Sequência de Períodos Trocáveis - K (EPS-K))** *Um EPS-K é um EPS que contém uma ou mais tarefas, sem nenhuma unidade de tempo ociosa.*

As cinco abordagens são:

- (1) A proposta de Anghinolfi *et al.* (2021), explanada na Subseção 6.2.1, que consiste na aplicação de um algoritmo Restrição- $\epsilon$  para minimizar o  $TEC$ , reduzindo gradualmente o horizonte de tempo. Além disso, a solução obtida pela heurística construtiva é refinada por um algoritmo de busca local que realiza movimentos entre blocos para minimizar o  $TEC$  sem piorar o  $C_{max}$ , utilizando uma política de “primeira melhoria”;
- (2) A mesma proposta de Anghinolfi *et al.* (2021), com algumas modificações: substituições da política de “primeira melhoria” por “melhor melhoria” no algoritmo de busca local e consideração de movimentações entre EPS-K e EPS-I;
- (3) A aplicação da meta-heurística 2PPLS, que utiliza uma busca em vizinhança na segunda fase para encontrar outras soluções não dominadas, aumentando gradualmente o  $C_{max}$  para melhorar o valor do  $TEC$ . A busca em vizinhança é realizada por meio de movimentos entre EPS-J’s e EPS-I’s, seguindo uma política de “primeira melhoria”;
- (4) A mesma abordagem do número 3, mas com a mudança da política de “primeira melhoria” para “melhor melhoria” na busca em vizinhança, e considerando que a busca em vizinhança é realizada por meio de movimentos entre EPS-K’s e EPS-I’s.
- (5) Meta-heurística 2PPLS com a primeira fase definida pela Abordagem 2 e segunda fase caracterizada pela busca em vizinhança do 2PPLS com as estratégias de política de “melhor melhoria” e movimentações entre EPS-I e EPS-K.

Essas cinco abordagens foram testadas para analisar sua eficácia na resolução do problema em questão.

Na Abordagem 3, a busca por vizinhos  $x'$  de uma solução  $x$  é conduzida por meio do refinamento que envolve movimentos entre EPS’s no sequenciamento de  $x$ . No entanto, adaptamos essa busca para permitir trocas que resultam em um aumento no valor de  $C_{max}(x)$  em  $\theta$  unidades, desde que haja uma melhoria no valor de  $TEC$ . Esse processo segue uma política de “primeira melhoria”, onde os movimentos são realizados até que não seja possível encontrar um  $TEC$  menor sem piorar o  $C_{max}$  de  $x'$ , indicando a convergência para um ótimo local dentro dessa estrutura de vizinhança.

A busca na vizinhança considera incrementos graduais de piora (de 1 unidade) no valor de  $C_{max}(x)$ , começando de 0 (para refinar a solução corrente) e avançando até a diferença entre o menor  $C_{max}$  de uma solução  $q$  da fronteira aproximada  $F_e$  que seja maior que o  $C_{max}$  da solução corrente  $x$ , e o próprio  $C_{max}$  da solução corrente  $x$ , i.e.,  $max\_piora = \min\{C_{max}(q) : q \in F_e, C_{max}(q) > C_{max}(x)\} - C_{max}(x)$ . Portanto, novas soluções vizinhas  $x'$  podem ser exploradas por meio de movimentos de blocos no sequenciamento da solução  $x$  que permitem uma piora do  $C_{max}(x)$  em  $\theta \in \{0, 1, \dots, max\_piora - 1, max\_piora\}$  desde que  $TEC(x') < TEC(x)$ . Ainda, caso uma solução vizinha  $x'$  de ótimo local

encontrada for dominada por alguma outra solução da fronteira aproximada de Pareto  $F_e$  a busca dos vizinhos da solução corrente  $x$  para e a próxima solução de  $F'$  passa a ser a corrente. Esse processo explora de forma abrangente as possíveis melhorias no sequenciamento, garantindo uma busca completa pelo espaço de soluções vizinhas.

A política de “melhor melhoria” mapeia todos os possíveis blocos de EPS-J e EPS-I com cardinalidade  $p \in P_j$  e testa cada um deles com o objetivo de selecionar o movimento que resulta na maior melhoria no valor do  $TEC$ , como demonstrado no Pseudocódigo 12.

Inicialmente, na Linha 1, todos blocos definidos como EPS-K e EPS-I para cada tempo de processamento  $p \in \hat{P}$  são criados e armazenados em *blocos\_epsi* e *blocos\_epsk*, respectivamente.

Em seguida, uma solução  $x'$  é inicializada com a solução  $x$ , e será avaliada por meio de movimentos de blocos com o objetivo de encontrar um vizinho  $x'$  com um melhor valor de  $TEC$ . Portanto, enquanto for possível melhorar a solução  $x'$  (Linhas 3 a 26), para cada tempo de processamento distinto  $p \in P_j$  (Linhas 5 a 25), e para cada bloco de EPS-K e EPS-I de cardinalidade  $p$  (Linhas 9 e 10, respectivamente) todas as possibilidades de trocas são testadas. Esses EPS's são formados por subconjuntos de tempo  $T$  que permitem uma busca em um horizonte de tempo  $K^{max}$  definido por  $C_{max}(x) + piora$ , onde *piora* pode variar conforme explicado anteriormente.

A cada iteração a função Viabilidade (Linha 11) identifica se o  $TEC$ , no melhor caso de movimentação, (*novo\_tec*) é melhor que o  $TEC(x')$ , se sim, temos uma atribuição de *verdade* para a variável *viabilidade*, c.c., *falso*. Ademais, como já explanado, o cálculo do *novo\_tec* é realizado de forma eficiente. Se o *novo\_tec* for menor que o  $TEC$  de  $x'$ , i.e., *viabilidade* = *verdade* então realizamos o movimento entre o EPS-K e EPS-I identificados, obtendo o  $TEC$  proveniente do movimento,  $\theta$  (Linha 13).

Se  $\theta$  for menor que  $TEC(x')$ , a “melhor melhoria” é então identificada por meio das Linhas 15 a 17 e a atualização de  $x'$  é realizada na Linha 21. Na Linha 22, atualiza-se os *blocos\_epsi* e *blocos\_epsk* de todas as cardinalidades de  $p \in \hat{P}$  que surgem e que deixam de ser EPS-I ou EPS-K a partir da movimentação de EPS's realizada. Esse processo é repetido até que não haja nenhuma troca entre os EPS-I's e EPS-K's de cardinalidade  $p$  que melhorem o  $TEC$  de  $x'$ .

Um subconjunto  $E$  é identificado como um EPS-K que pode conter uma ou mais tarefas se o primeiro elemento do subconjunto  $E$  é o tempo de início de uma tarefa  $j \in J$ , se o último elemento de  $E$  é o tempo de conclusão de uma tarefa  $j' \in J$ , e se todas as unidades de tempo do subconjunto  $E$  são ocupadas, ou seja, são utilizadas para processar tarefas.

Na Abordagem 5, busca-se explorar a construção das soluções por meio da Abordagem 1 para definir uma fronteira inicial mais eficiente na primeira fase do algoritmo 2PPLS. E a busca em

**Algoritmo 12:** Melhor\_melhoria

**Entrada:** Conjunto  $J = \{1, \dots, N\}$ , com  $p_j, j \in J$ .

Conjunto  $H = \{1, \dots, M\}$  de máquinas, com a taxa de consumo unitária  $e_h, h \in H$ .

Conjunto  $T = \{1, \dots, K^{max}\}$  de unidades de tempo, com os custos de energia  $c_k, k \in T$ , onde  $K^{max}$  é o maior período do horizonte de tempo.

Uma solução corrente  $x$ .

Conjunto ordenado de forma não-crescente de tempos de processamento distintos  $\hat{P}$ .

Horizonte de tempo  $K^{max}$

**Resultado:** Uma solução vizinha  $x'$  de  $x$ .

```

1 Criar os blocos EPS-I e EPS-K  $\forall p \in \hat{P}$  e armazenar em blocos_epsi e blocos_epsk,
  respectivamente;
2  $x' \leftarrow x$ ;
3 while melhorar do
4   melhorar  $\leftarrow$  falso;
5   for  $p \in \hat{P}$  do
6     melhorar_aux  $\leftarrow$  verdade;
7     while melhorar_aux = verdade do
8       melhorar_aux  $\leftarrow$  falso, melhor_movimento  $\leftarrow$  0 e sol  $\leftarrow$   $\emptyset$ ;
9       for  $EPS_{k_p} \in \text{blocos\_epsi}$  do
10        for  $EPS_{i_p} \in \text{blocos\_epsj}$  do
11          viabilidade, novo_tec, menor_custo  $\leftarrow$  Viabilidade( $EPS_{k_p}, EPS_{i_p}$ );
12          if viabilidade = verdade then
13             $\theta \leftarrow$  Movimento( $novo\_tec, menor\_custo$ );
14          end
15          if  $\theta < TEC(x')$  e  $\theta < melhor\_movimento$  then
16            melhor_movimento  $\leftarrow$   $\theta$ , sol  $\leftarrow$  resposta do movimento e
17            melhorar_aux  $\leftarrow$  verdade;
18          end
19        end
20      end
21      if melhorar_aux = verdade then
22         $x' \leftarrow$  sol;
23        Atualizar os blocos EPS-I e EPS-K  $\forall p \in \hat{P}$  e armazenar em blocos_epsi e
24        blocos_epsk, respectivamente, e melhorar  $\leftarrow$  verdade;
25      end
26    end
27  end
28 return  $x'$ 

```

vizinhança permite uma exploração ainda maior do espaço de soluções.

## 7 RESULTADOS E DISCUSSÃO

Este capítulo apresenta os resultados obtidos por meio da abordagem proposta. Os resultados são apresentados e discutidos para avaliar a eficácia e o desempenho do método proposto em resolver o problema em questão.

Os experimentos são realizados em instâncias de teste da literatura, classificadas como pequenas e grandes, e os resultados para os algoritmos implementados são reportados para cada uma. Os algoritmos foram implementados em linguagem de programação Julia e executados em um computador com processador Intel(R) Core (TM) i5-1135G7, equipado com CPU de @ 2.40GHz, 8 GB de memória RAM e sistema operacional Windows 11.

Para resolver as pequenas instâncias de forma exata, foi utilizado o pacote JuMP (*Julia for Mathematical Programming*) de modelagem matemática, com o Cbc (*COIN-OR Branch and Cut solver*) como solucionador do problema de programação linear inteira mista. Por fim, os testes estatísticos empregados para comparar e analisar os resultados foram desenvolvidos com o auxílio do pacote *HypothesisTests* de Julia.

### 7.1 INSTÂNCIAS UTILIZADAS

As instâncias de testes foram propostas por Wang *et al.* (2018) e são classificadas em Anghinolfi *et al.* (2021) como instâncias de pequena escala (numeradas de 1 a 30) e de grande escala (numeradas de 31 a 60). Além disso, são disponibilizadas pelos mesmos no site <https://github.com/ORresearcher/A-bi-objective-heuristic-approach-for-green-identical-parallel-machine-scheduling/tree/master/Data>.

As pequenas contam com o número de tarefas  $N \in \{6, 10, 15, 20, 25\}$  e seus respectivos tempos de processamento  $p_j, j \in J$ , o número de máquinas paralelas disponíveis  $M \in \{3, 5, 7\}$  com seus valores de custo de consumo energético associado a cada uma  $e_h, h \in H$ , e o total de intervalos de tempo  $K \in \{50, 80\}$  disponíveis em cada máquina com suas respectivas tarifas  $c_t, t \in T$ . Por outro lado, as grandes contam com o número de tarefas  $N \in \{30, 60, 100, 150, 200\}$  e seus respectivos tempos de processamento  $p_j, j \in J$ , o número de máquinas paralelas disponíveis  $M \in \{8, 16, 20, 25\}$  com seus valores de custo de consumo energético associado a cada uma  $e_h, h \in H$ , e o total de intervalos de tempo  $K \in \{100, 300\}$  disponíveis em cada máquina com suas respectivas tarifas  $c_t, t \in T$ .

## 7.2 RESULTADOS COMPUTACIONAIS

Aqui apresentamos os resultados dos indicadores de performance avaliados para os grupos de instâncias pequenas e grandes, além de realizarmos algumas discussões.

A *Pureza* e o  $D_r$  medem a qualidade das soluções. A *Pureza* verifica o número de soluções de uma fronteira que apresentam os mesmos valores de  $TEC$  e  $C_{max}$  quando comparadas a uma fronteira de referência. Os valores de *Pureza* variam entre 0 e 1, sendo que valores maiores indicam uma fronteira de melhor qualidade. O  $D_r$  mede a distância euclidiana entre os pontos do método avaliado e uma fronteira de referência, onde se desejam valores menores de  $D_r$ . A fronteira de referência é obtida pela aplicação do método exato ou realizando a união das soluções encontradas por todos os métodos, retirando as soluções dominadas.

Por outro lado, o *Hipervolume* avalia a aproximação e a dispersão dos pares obtidos em relação a um ponto de referência. Um maior número de soluções menos dispersas e mais próximas da fronteira de referência resulta em um maior *Hipervolume*. O ponto de referência utilizado é o Ponto de Nadir da união de todas as fronteiras obtidas pelos métodos analisados.

Testamos várias abordagens, mudando o conjunto de estratégias utilizadas. As estratégias adotadas nas diferentes abordagens incluem variações na política de melhoria da busca local e nas considerações de movimentações entre EPS's. A política de “melhor melhoria”, por exemplo, explora todas as possibilidades de movimentações entre EPS's, selecionando a que proporciona a maior melhoria no  $TEC$ . A estratégia de permitir que um EPS-K contenha uma ou mais tarefas aumenta as possibilidades de movimentações, potencialmente levando a melhores soluções.

### 7.2.1 Instâncias Pequenas

Para as instâncias pequenas, seis abordagens são analisadas:

- (1) MIP: Fronteira exata obtida pelo algoritmo de restrição- $\epsilon$  exato;
- (2) Abordagem 1: Proposta de Anghinolfi *et al.* (2021);
- (3) Abordagem 2: Proposta de Anghinolfi *et al.* (2021) com as estratégias de política de “melhor melhoria” na busca local e consideração de movimentações entre EPS-I e EPS-K;
- (4) Abordagem 3: Meta-heurística 2PPLS com política de “primeira melhoria”;
- (5) Abordagem 4: Meta-heurística 2PPLS com as estratégias de política de “melhor melhoria” na busca local e consideração de movimentações entre EPS-I e EPS-K;
- (6) Abordagem 5: Meta-heurística 2PPLS com a primeira fase definida pela Abordagem 2 e segunda fase caracterizada pela busca em vizinhança com as estratégias de política de “melhor melhoria”

e movimentações entre EPS-I e EPS-K.

A Tabela 1 reporta o *Hipervolume* e a média geral e mediana deste para cada abordagem. Além disso, uma piora percentual (linha “GAP”) do *Hipervolume* médio de cada abordagem em relação ao da fronteira exata (coluna “MIP”) é apresentada.

O “GAP” entre o resultado médio de uma fronteira de referência e outra, para um certo indicador de performance, é calculado conforme a Equação (7.1).

(2.4).

$$GAP = \frac{M_{p,r} - M_{p,i}}{M_{p,r}} \quad (7.1)$$

Onde  $M$  é o valor médio de comparação,  $p$  é o indicador de performance avaliado,  $r$  é a fronteira de referência, e  $i$  é a fronteira comparada.

Tabela 1 – Resultados para *Hipervolume* para instâncias pequenas.

Instâncias	Hipervolume					
	MIP	Abordagem 1	Abordagem 2	Abordagem 3	Abordagem 4	Abordagem 5
1	0,7460	0,7460	0,7460	0,7455	0,7460	0,7460
2	0,7756	0,7744	0,7743	0,7715	0,7725	0,7747
3	0,7775	0,7605	0,7605	0,7632	0,7622	0,7667
4	0,7617	0,7617	0,7617	0,7612	0,7594	0,7612
5	0,8019	0,8008	0,8004	0,7977	0,7999	0,8017
6	0,8247	0,8237	0,8234	0,8210	0,8174	0,8237
7	0,7295	0,7295	0,7295	0,7281	0,7274	0,7295
8	0,8153	0,8138	0,8141	0,8137	0,8138	0,8142
9	0,7687	0,7566	0,7604	0,7570	0,7596	0,7615
10	0,7732	0,7719	0,7731	0,7716	0,7710	0,7726
11	0,8128	0,8128	0,8128	0,8112	0,8126	0,8128
12	0,8544	0,8521	0,8522	0,8501	0,8507	0,8525
13	0,6900	0,6900	0,6900	0,6845	0,6876	0,6900
14	0,7548	0,7531	0,7533	0,7532	0,7537	0,7539
15	0,7520	0,7505	0,7511	0,7492	0,7495	0,7507
16	0,7993	0,7990	0,7988	0,7983	0,7982	0,7991
17	0,7145	0,7140	0,7142	0,7127	0,7123	0,7142
18	0,8221	0,8174	0,8180	0,8168	0,8207	0,8188
19	0,5794	0,5766	0,5789	0,5764	0,5784	0,5790
20	0,7485	0,7485	0,7485	0,7461	0,7471	0,7485
21	0,7458	0,7457	0,7453	0,7449	0,7454	0,7457
22	0,7970	0,7969	0,7968	0,7966	0,7958	0,7969
23	0,7743	0,7706	0,7717	0,7719	0,7719	0,7711
24	0,8413	0,8403	0,8405	0,8401	0,8402	0,8406
25	0,7351	0,7351	0,7351	0,7186	0,7330	0,7351
26	0,7683	0,7683	0,7683	0,7649	0,7683	0,7683
27	0,7425	0,7425	0,7425	0,7282	0,7288	0,7425
28	0,7438	0,7438	0,7438	0,7380	0,7380	0,7438
29	0,4531	0,4531	0,4531	0,4531	0,4531	0,4531
30	0,7532	0,7532	0,7532	0,7474	0,7429	0,7532
Médias	0,75521	0,75342	0,75371	0,75108	0,75191	0,75406
GAP	0,0000%	0,2371%	0,1977%	0,5460%	0,4367%	0,1523%
Mediana	0,76503	0,75854	0,76045	0,75908	0,75947	0,76135

Fonte: Elaborado pela autora (2024)

Conforme a Tabela 1, dentre as abordagens de 1 a 5, a Abordagem 5 apresenta o maior *Hipervolume* médio (0,75406), seguida da Abordagem 1 (0,75371) e da Abordagem 2 (0,75342).

A princípio, pode-se afirmar que a política de “melhor melhoria” e a utilização de EPS-K são eficazes em melhorar o *Hipervolume*, visto que a média da Abordagem 2 é maior que a média da Abordagem 1, e a média da Abordagem 4 é maior que a da Abordagem 3. Além disso, a Abordagem 5 supera todas as outras com a maior média, pois, além de utilizar essas estratégias, inclui uma segunda fase que realiza buscas em vizinhança, permitindo encontrar novas soluções a partir das soluções da primeira fase. Além disso, as reduções das médias obtidas pelas Abordagens 1 a 5 em relação ao “MIP” são, respectivamente, 0,2371%, 0,1977%, 0,5460%, 0,4367% e 0,1523%. A menor redução é observada na Abordagem 5, o que indica que esta abordagem é a que mais se aproxima do resultado “MIP”, seguida pela Abordagem 2.

Para analisar se as diferenças entre os métodos são significativas, a Tabela 2 reporta o *valor-p* dos testes estatísticos de Wilcoxon realizados par a par.

Tabela 2 – Estatística de teste *p* para os resultados de *Hipervolume* para as instâncias pequenas.

Abordagens	Estatística de teste <i>p</i>					
	MIP	Abordagem 1	Abordagem 2	Abordagem 3	Abordagem 4	Abordagem 5
MIP	-	7,63E-06	7,63E-06	3,73E-09	1,49E-08	3,81E-06
Abordagem 1	7,63E-06	-	0,098373	0,000053	0,018655	0,000763
Abordagem 2	7,63E-06	0,098373	-	2,84E-06	0,000339	0,066536
Abordagem 3	3,73E-09	0,000053	2,84E-06	-	0,114516	1,86E-07
Abordagem 4	1,49E-08	0,018655	0,000339	0,114516	-	0,000013
Abordagem 5	3,81E-06	0,000763	0,066536	1,86E-07	0,000013	-

Fonte: Elaborado pela autora (2024)

Considerando um nível de significância de 5%, definimos que um valor da estatística *valor-p* menor que 0,05 implica que não há evidência estatística suficiente para considerarmos que as medianas provenientes das amostras entre os métodos avaliados são iguais. Diante disso, verifica-se que o “MIP” apresenta diferenças significativas com todas as abordagens. Logo, podemos dizer que o “MIP” reporta resultados de *Hipervolume* superiores. Entretanto, ao comparar as abordagens heurísticas entre si, observa-se que a proposta de Anghinolfi *et al.* (2021) (Abordagem 1) supera as Abordagens 3 e 4. No entanto, não há diferença significativa entre as Abordagens 1 e 2, em *Hipervolume*. Por fim, a Abordagem 5 apresenta diferenças significativas em relação a todas as outras.

Com base nisso, podemos inferir que a Abordagem 5 supera a Abordagem 1, proposta por Anghinolfi *et al.* (2021), em *Hipervolume*, visto que não há evidências estatísticas suficientes para afirmar que não há diferença significativa entre as medianas de ambas (*valor-p* = 0,000763). Além disso, a Abordagem 2 pode ser considerada equivalente à Abordagem 1, visto que *valor-p* = 0,098373.

Então, considerando que a Abordagem 5 é uma implementação da Abordagem 1 com

as estratégias de “melhor melhoria”, permitindo movimentações entre EPS-I e EPS-K, além da implementação de uma segunda fase com busca em vizinhança, conclui-se que a combinação dessas estratégias é eficaz para encontrar soluções que apresentam uma fronteira de melhor *Hipervolume*. A política de “melhor melhoria” toma as melhores decisões em cada momento, enquanto permitir que os EPS-K’s contenham uma ou mais tarefas aumenta as possibilidades de movimentações, possibilitando encontrar soluções que não seriam identificadas sem essa estratégia. A busca em vizinhança, por sua vez, permite descobrir novas soluções a partir de outras já existentes.

Quanto ao indicador de desempenho *Pureza*, a Tabela 3 apresenta os resultados das médias e medianas de cada abordagem e o “GAP” entre a abordagem “MIP” e todas as outras. Da mesma forma, a Tabela 4 mostra a estatística de teste *valor-p* da comparação par a par entre os métodos.

Tabela 3 – Resultados para *Pureza* para instâncias pequenas.

Instâncias	Pureza					
	MIP	Abordagem 1	Abordagem 2	Abordagem 3	Abordagem 4	Abordagem 5
1	1,0000	1,0000	1,0000	0,9677	1,0000	1,0000
2	1,0000	0,8919	0,8649	0,7368	0,8158	0,9189
3	1,0000	0,2143	0,2857	0,6923	0,6923	0,6154
4	1,0000	1,0000	1,0000	0,8889	0,7778	0,8889
5	1,0000	0,8667	0,8000	0,8000	0,8667	0,9333
6	1,0000	0,8235	0,7647	0,5882	0,4118	0,8235
7	1,0000	1,0000	1,0000	0,8519	0,7037	1,0000
8	1,0000	0,8919	0,9189	0,8611	0,9412	0,9459
9	1,0000	0,3043	0,2917	0,3182	0,4091	0,5417
10	1,0000	0,8148	0,9630	0,7407	0,6923	0,9259
11	1,0000	1,0000	1,0000	0,9286	0,9286	1,0000
12	1,0000	0,6333	0,6129	0,3636	0,4063	0,6667
13	1,0000	1,0000	1,0000	0,8000	0,9000	1,0000
14	1,0000	0,7949	0,7436	0,7750	0,8000	0,8750
15	1,0000	0,7778	0,8649	0,6667	0,7027	0,8378
16	1,0000	0,9149	0,9111	0,7083	0,7708	0,9167
17	1,0000	0,8095	0,9048	0,6667	0,7143	0,9048
18	1,0000	0,4048	0,4286	0,5581	0,5476	0,5952
19	1,0000	0,7000	0,8571	0,8000	0,9500	0,9048
20	1,0000	1,0000	1,0000	0,7895	0,7500	1,0000
21	1,0000	0,9583	0,8750	0,6800	0,8000	0,9583
22	1,0000	0,9388	0,8980	0,8750	0,7708	0,9388
23	1,0000	0,6571	0,6286	0,6765	0,6061	0,6857
24	1,0000	0,7561	0,8500	0,7805	0,7857	0,8293
25	1,0000	1,0000	1,0000	0,6154	0,9231	1,0000
26	1,0000	1,0000	1,0000	0,8571	1,0000	1,0000
27	1,0000	1,0000	1,0000	0,6667	0,6667	1,0000
28	1,0000	1,0000	1,0000	0,7500	0,7500	1,0000
29	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000
30	1,0000	1,0000	1,0000	0,7778	0,6667	1,0000
Médias	1,0000	0,8384	0,8488	0,7394	0,7583	0,8902
GAP	0,0000%	16,1569%	15,1223%	26,0623%	24,1671%	10,9779%
Mediana	1,0000	0,9034	0,9079	0,7625	0,7708	0,9296

Fonte: Elaborado pela autora (2024)

Conforme a Tabela 3, das abordagens heurísticas, identificou-se que a Abordagem 5

apresenta uma média superior e uma menor piora em relação à fronteira “MIP”, de 0,8902 e 10,9779%, respectivamente. Portanto, a fronteira obtida por esta é a mais próxima ao resultado ótimo (“MIP”) para *Pureza*, seguida da Abordagem 2, com média geral e “GAP” de 0,8488 e 15,1223%, respectivamente.

Avaliando se há diferenças estatísticas entre as medianas dessas abordagens, não temos evidência estatística suficiente para afirmar que a mediana da Abordagem 5 é igual a todas as outras. Logo, conforme o *valor-p* na Tabela 4, podemos inferir que a Abordagem 5 é significativamente superior a todas as outras. Já a Abordagem 2, de acordo com os valores de teste da estatística *valor-p* apresentados na Tabela 4, supera apenas as Abordagens 3 e 4. Para as Abordagens 1 e 5, a diferença não é significativa. Portanto, em termos de *Pureza*, podemos afirmar que a Abordagem 2 é equivalente à Abordagem 1.

Tabela 4 – Estatística de teste *p* para os resultados de *Pureza* para as instâncias pequenas.

Abordagens	Estatística de teste <i>p</i>					
	MIP	Abordagem 1	Abordagem 2	Abordagem 3	Abordagem 4	Abordagem 5
MIP	-	7,63E-06	7,63E-06	3,73E-09	1,49E-08	3,81E-06
Abordagem 1	7,63E-06	-	0,495079	0,000487	0,010069	0,001709
Abordagem 2	7,63E-06	0,495079	-	0,000216	0,004155	0,005600
Abordagem 3	3,73E-09	0,000487	0,000216	-	0,380984	1,42E-07
Abordagem 4	1,49E-08	0,010069	0,004155	0,380984	-	8,49E-07
Abordagem 5	3,81E-06	0,001709	0,005600	1,42E-07	8,49E-07	-

Fonte: Elaborado pela autora (2024)

Neste caso, as estratégias de “melhor melhoria” e movimentações entre EPS-I e EPS-K não resultaram em melhorias significativas. Com base na estatística, infere-se que as Abordagens 1 e 2 e as 3 e 4 são equivalentes entre si. No entanto, novamente, a combinação das duas com a segunda fase do 2PPLS exploram melhor o espaço de soluções, permitindo uma melhoria significativa de 5,8176% da Abordagem 5 em relação à 1.

Para o indicador de performance  $D_r$ , a Tabela 5 apresenta os resultados das médias e medianas de cada abordagem e o “GAP” entre a Abordagem 1 e todas as outras. Da mesma forma, a Tabela 6 mostra a estatística de teste *valor-p* da comparação par a par entre os métodos.

Menores valores de  $D_r$  indicam um melhor desempenho. Visto isso, a Abordagem 5 se sai melhor dentre as cinco heurísticas, com um  $D_r$  médio de 0,0015, seguido pela Abordagem 2, com 0,0019. Em relação ao “GAP”, é possível verificar que a Abordagem 5 melhora a média da Abordagem 1 em 28,852% e a 2 melhora a 1 em 8,317%.

Com base nos testes estatísticos apresentados na Tabela 6, novamente, a Abordagem 5 é superior a todas as outras, com um valor de *valor-p* igual a 3,81E-06, 0,004181, 0,048279, 4,10E-07 e 0,000052, respectivamente. No entanto, assim como no indicador de *Pureza*, as diferenças entre a Abordagem 1 com a 2 e a 3 com a 4 não são significativas, o que sugere que, com base no  $D_r$ , essas

Tabela 5 – Resultados para  $D_r$  para instâncias pequenas.

Instâncias	$D_r$					
	MIP	Abordagem 1	Abordagem 2	Abordagem 3	Abordagem 4	Abordagem 5
1	0,0000	0,0000	0,0000	0,0003	0,0000	0,0000
2	0,0000	0,0018	0,0021	0,0065	0,0050	0,0014
3	0,0000	0,0205	0,0196	0,0164	0,0186	0,0111
4	0,0000	0,0000	0,0000	0,0032	0,0058	0,0032
5	0,0000	0,0016	0,0023	0,0062	0,0029	0,0003
6	0,0000	0,0019	0,0025	0,0071	0,0150	0,0019
7	0,0000	0,0000	0,0000	0,0010	0,0026	0,0000
8	0,0000	0,0012	0,0008	0,0010	0,0006	0,0006
9	0,0000	0,0114	0,0137	0,0106	0,0092	0,0112
10	0,0000	0,0018	0,0001	0,0022	0,0027	0,0008
11	0,0000	0,0000	0,0000	0,0020	0,0002	0,0000
12	0,0000	0,0033	0,0029	0,0077	0,0056	0,0022
13	0,0000	0,0000	0,0000	0,0053	0,0023	0,0000
14	0,0000	0,0018	0,0015	0,0024	0,0017	0,0008
15	0,0000	0,0019	0,0009	0,0026	0,0025	0,0012
16	0,0000	0,0003	0,0003	0,0012	0,0013	0,0003
17	0,0000	0,0007	0,0004	0,0025	0,0031	0,0004
18	0,0000	0,0056	0,0048	0,0065	0,0024	0,0038
19	0,0000	0,0029	0,0006	0,0032	0,0006	0,0004
20	0,0000	0,0000	0,0000	0,0025	0,0018	0,0000
21	0,0000	0,0002	0,0008	0,0025	0,0018	0,0002
22	0,0000	0,0007	0,0008	0,0005	0,0014	0,0007
23	0,0000	0,0041	0,0028	0,0026	0,0023	0,0034
24	0,0000	0,0014	0,0010	0,0018	0,0017	0,0010
25	0,0000	0,0000	0,0000	0,0254	0,0032	0,0000
26	0,0000	0,0000	0,0000	0,0056	0,0000	0,0000
27	0,0000	0,0000	0,0000	0,0207	0,0198	0,0000
28	0,0000	0,0000	0,0000	0,0073	0,0073	0,0000
29	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
30	0,0000	0,0000	0,0000	0,0091	0,0162	0,0000
Médias	0,0000	0,0021	0,0019	0,0055	0,0046	0,0015
GAP	-	0,000%	8,317%	-162,614%	-117,803%	28,852%
Mediana	0,0000	0,0007	0,0005	0,0029	0,0024	0,0004

Fonte: Elaborado pela autora (2024)

abordagens são equivalentes.

Tabela 6 – Estatística de teste  $p$  para os resultados de  $D_r$  para as instâncias pequenas.

Abordagens	Estatística de teste $p$					
	MIP	Abordagem 1	Abordagem 2	Abordagem 3	Abordagem 4	Abordagem 5
MIP	-	7,63E-06	7,63E-06	3,73E-09	1,49E-08	3,81E-06
Abordagem 1	7,63E-06	-	0,154045	0,000053	0,008215	0,004181
Abordagem 2	7,63E-06	0,154045	-	0,000041	0,000828	0,048279
Abordagem 3	3,73E-09	0,000053	0,000041	-	0,156910	4,10E-07
Abordagem 4	1,49E-08	0,008215	0,000828	0,156910	-	0,000052
Abordagem 5	3,81E-06	0,004181	0,048279	4,10E-07	0,000052	-

Fonte: Elaborado pela autora (2024)

Em resumo, para as instâncias de menor porte, fica evidente que, dentre os métodos heurísticos empregados, a Abordagem 5 se destaca como superior em relação a todas as outras. Adicionalmente, constatamos que a Abordagem 2 é tão eficaz quanto a proposta de Anghinolfi *et al.*

(2021).

Por outro lado, as Abordagens 3 e 4 apresentaram os piores desempenhos. Acreditamos que isso se deve à própria estrutura da meta-heurística 2PPLS, a qual, na primeira fase, encontra apenas um número limitado e espaçado de soluções utilizando a heurística construtiva. Esperava-se que a busca em vizinhança, na segunda fase, fosse capaz de encontrar as demais soluções restantes da fronteira, porém isso não ocorre com eficiência. Portanto, possíveis melhorias nesse algoritmo podem ser aplicadas no procedimento da segunda fase, com uma busca em vizinhança mais eficiente.

No entanto, visando superar essa dificuldade, a Abordagem 5 implementa o método da restrição- $\epsilon$  aproximado na primeira fase, encontrando um número bem maior de soluções e de qualidade. A partir disso, a busca em vizinhança na segunda fase do 2PPLS consegue encontrar outras soluções.

Além disso, para todas as medidas de desempenho, as estratégias adicionais de “melhor melhoria” e EPS-K nas Abordagens não foram capazes de encontrar soluções melhores que aumentassem o desempenho da fronteira de forma significativa. No entanto, a utilização dessas duas estratégias em conjunto com a busca em vizinhança do 2PPLS proporciona decisões tomadas com maior eficiência, um maior número de movimentações e exploração do espaço de solução. Isso permite encontrar melhores soluções e fronteiras aproximadas com melhores desempenhos.

É importante salientar que o valor 1 de Pureza indica que todas as soluções encontradas pela fronteira analisada estão contidas na de referência. Portanto, como a fronteira “MIP” é a de referência, espera-se o valor 1 para todas as instâncias. Da mesma forma, para o  $D_r$ , o “MIP” apresenta o valor 0,00 em todas as instâncias, o que é esperado, pois não há erro ou distância entre os pontos, uma vez que a fronteira de referência é a própria exata analisada.

Quanto ao tempo computacional, a Tabela 7 mostra o tempo computacional em segundos para executar o procedimento de cada um dos métodos de otimização. E a Tabela 8 mostra a estatística de teste *valor-p* da comparação par a par entre os métodos. A estatística de teste *valor-p* mostra que todos os tempos das comparações par a par apresentam diferenças significativas.

Para as instâncias pequenas, em média o “MIP” retorna uma fronteira em 25,0351 segundos. Todas as abordagens heurísticas retornam uma fronteira em poucos segundos, melhorando o tempo computacional do “MIP” em 98,4405%, 97,6847%, 87,8162%, 93,4453% e 96,1419%. A Abordagem 1 é a mais rápida, com média de 0,3904 segundos, seguida da 2, com 0,5796 segundos, e da 5, com 0,9659 segundos. A melhoria do desempenho medida pelos indicadores de performance da fronteira proveniente da Abordagem 5 em relação à 1 custou um tempo maior em 147,40%. Já a melhoria da Abordagem 2, mesmo que não significativa, resultou em um aumento do tempo em 48,47%. Apesar

Tabela 7 – Resultados para tempo computacional para instâncias pequenas.

Instâncias	Tempo Computacional					
	MIP	Abordagem 1	Abordagem 2	Abordagem 3	Abordagem 4	Abordagem 5
1	39,2927	0,1085	0,1579	0,6005	0,3689	0,2445
2	22,9879	0,2619	0,3506	1,4123	0,8065	0,5552
3	25,6184	0,1544	0,1841	0,7431	0,4762	0,4544
4	20,5789	0,4766	0,6511	1,4979	0,7718	0,9626
5	11,4556	0,2636	0,3569	1,0511	0,5827	0,5718
6	25,9600	0,5379	0,7568	1,7540	1,0114	1,1492
7	11,9017	0,1136	0,1639	1,1757	0,5317	0,2500
8	18,8225	0,3395	0,5412	2,2502	1,3389	0,8650
9	29,5240	0,1965	0,2825	2,3603	1,0423	0,7292
10	30,4499	0,5838	0,8406	2,2602	1,4709	1,3173
11	15,6835	0,3012	0,4766	1,9387	1,3142	0,7769
12	41,7939	0,8194	1,1992	6,4400	2,4826	1,7734
13	12,6579	0,1253	0,1634	1,2397	0,6518	0,2806
14	21,9195	0,3442	0,5165	3,8571	1,9793	0,9702
15	18,1996	0,2810	0,4030	3,8602	2,1134	0,7185
16	40,6696	0,9183	1,7306	8,4356	4,8887	2,3477
17	20,2926	0,4044	0,6476	3,1823	1,7344	0,9368
18	59,5337	0,9582	1,5260	12,3374	5,3698	3,3161
19	10,6209	0,1551	0,2310	1,7917	0,7314	0,4054
20	36,6411	0,3626	0,5134	2,9494	1,7253	0,9258
21	18,8451	0,3234	0,5096	3,4058	1,9376	0,8764
22	57,3490	0,7607	1,1689	7,7037	4,0454	2,0979
23	34,7733	0,4391	0,6474	7,0731	3,8738	1,2078
24	10,3473	1,2415	1,7892	9,3563	6,3092	3,0073
25	14,5295	0,0681	0,0810	0,1906	0,3878	0,1181
26	16,4244	0,1952	0,2491	0,4273	0,2155	0,3393
27	9,5270	0,1244	0,1439	0,3416	0,1462	0,1890
28	15,4109	0,2663	0,3361	0,5815	0,2971	0,5438
29	9,0265	0,1889	0,2725	0,4297	0,2061	0,3424
30	50,2174	0,3985	0,4985	0,8602	0,4186	0,7040
Médias	25,0351	0,3904	0,5796	3,0502	1,6410	0,9659
GAP	0,0000%	98,4405%	97,6847%	87,8162%	93,4453%	96,1419%

Fonte: Elaborado pela autora (2024)

Tabela 8 – Estatística de teste  $p$  para os resultados de tempo computacional para as instâncias pequenas.

Abordagens	Estatística de teste $p$					
	MIP	Abordagem 1	Abordagem 2	Abordagem 3	Abordagem 4	Abordagem 5
MIP	-	1,86E-09	1,86E-09	1,86E-09	1,86E-09	1,86E-09
Abordagem 1	1,86E-09	-	1,86E-09	1,86E-09	1,86E-09	1,86E-09
Abordagem 2	1,86E-09	1,86E-09	-	1,86E-09	2,05E-07	1,86E-09
Abordagem 3	1,86E-09	1,86E-09	1,86E-09	-	5,59E-09	1,86E-09
Abordagem 4	1,86E-09	1,86E-09	2,05E-07	5,59E-09	-	7,91E-05
Abordagem 5	1,86E-09	1,86E-09	1,86E-09	1,86E-09	7,91E-05	-

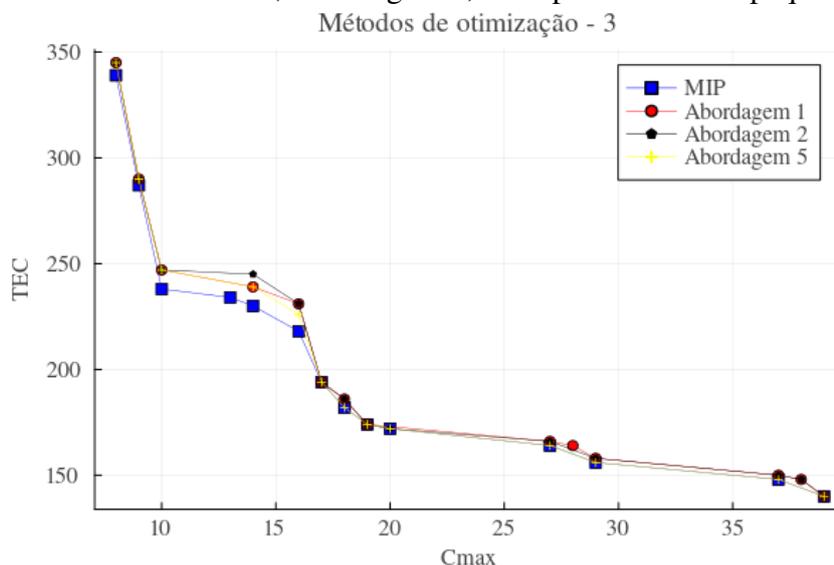
Fonte: Elaborado pela autora (2024)

disso, essas heurísticas ainda apresentam um tempo computacional relativamente baixo.

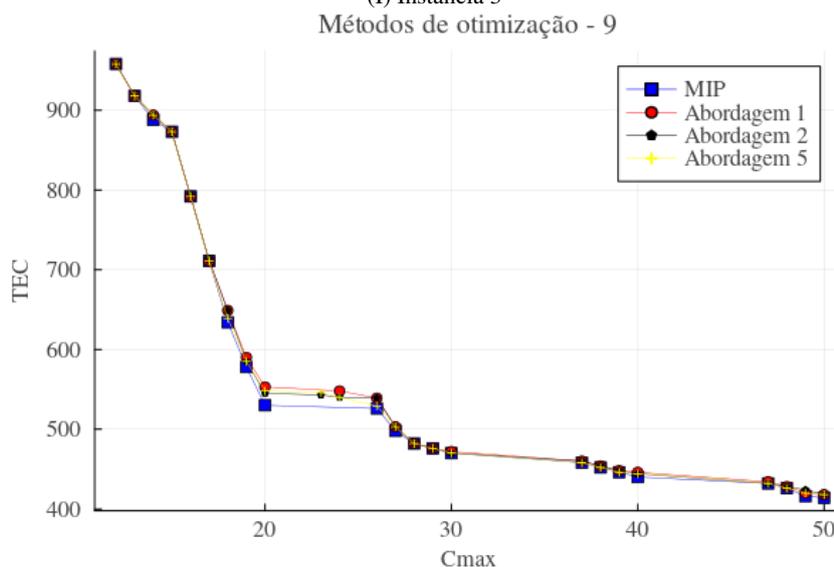
A Figura 17, mostra as fronteiras obtidas pelas Abordagens “MIP”, 1, 2 e 5 para as instâncias 3 e 9. Nelas podemos notar que a fronteira da Abordagem 5 (em amarelo) é a mais próxima da fronteira ótima (em azul), encontrando mais pontos em comum. E a fronteira obtida pela Abordagem 2 (linha preta) consegue encontrar pontos mais próximos da ótima em relação à Abordagem 1 (linha

vermelha). No Apêndice A, da Figura 19 até a 48 é possível visualizar as fronteiras obtidas por esses métodos para todas as instâncias pequenas.

Figura 17 – Fronteiras MIP, Abordagem 1, 2 e 5 para instâncias pequenas 3 e 9



(I) Instância 3



(II) Instância 9

Fonte: Elaborado pela autora (2024)

### 7.2.2 Instâncias Grandes

Para as instâncias grandes, avaliamos somente as 5 abordagens heurísticas, visto a incapacidade de obter a fronteira exata para essas instâncias. Em cada tabela que avalia os indicadores de performance são reportadas as médias gerais de cada abordagem, as medianas e o percentual e melhora ou piora desse em relação à média geral da abordagem de referência (abordagem 1, proposta de Anghinolfi *et al.* (2021)) (linha “GAP”).

Tabela 9 – Resultados para *Hipervolume* para instâncias grandes.

Instâncias	Hipervolume				
	Abordagem 1	Abordagem 2	Abordagem 3	Abordagem 4	Abordagem 5
31	0,8814	0,8814	0,8766	0,8779	0,8821
32	0,8109	0,8109	0,8063	0,8048	0,8113
33	0,7769	0,7770	0,7764	0,7742	0,7771
34	0,6937	0,6944	0,6912	0,6838	0,6947
35	0,6012	0,6012	0,5929	0,5916	0,6012
36	0,8548	0,8540	0,8556	0,8561	0,8552
37	0,8714	0,8710	0,8699	0,8705	0,8714
38	0,8489	0,8494	0,8475	0,8487	0,8497
39	0,7725	0,7722	0,7701	0,7712	0,7737
40	0,7802	0,7806	0,7761	0,7778	0,7833
41	0,9291	0,9291	0,9286	0,9282	0,9291
42	0,8529	0,8529	0,8502	0,8503	0,8529
43	0,8625	0,8626	0,8596	0,8589	0,8632
44	0,8268	0,8270	0,8271	0,8282	0,8274
45	0,7900	0,7900	0,7858	0,7878	0,7911
46	0,8296	0,8299	0,8243	0,8332	0,8311
47	0,8730	0,8732	0,8712	0,8725	0,8739
48	0,8727	0,8728	0,8685	0,8699	0,8741
49	0,8025	0,8027	0,7988	0,7988	0,8045
50	0,8169	0,8179	0,8228	0,8200	0,8255
51	0,8484	0,8500	0,8406	0,8471	0,8519
52	0,8682	0,8683	0,8593	0,8606	0,8690
53	0,8881	0,8882	0,8871	0,8876	0,8885
54	0,8836	0,8836	0,8798	0,8808	0,8840
55	0,8451	0,8451	0,8445	0,8447	0,8453
56	0,8871	0,8868	0,8848	0,8845	0,8864
57	0,7513	0,7520	0,7362	0,7515	0,7537
58	0,9066	0,9066	0,9062	0,9066	0,9068
59	0,8993	0,8994	0,8987	0,8992	0,8998
60	0,8169	0,8172	0,8154	0,8148	0,8172
Médias	0,83141	0,83158	0,82839	0,82939	0,83251
GAP	0,0000%	-0,0207%	0,3627%	0,2423%	-0,1321%
Mediana	0,8487	0,8497	0,8460	0,8479	0,8508

Fonte: Elaborado pela autora (2024)

Conforme a Tabela 9, dentre as Abordagens de 1 a 5, a Abordagem 5 apresenta o maior *Hipervolume* médio. Além disso, a Abordagem 5 apresenta a maior redução média do *Hipervolume* em relação à 1, de 0,1321%, e para a Abordagem 2, essa redução é de 0,0207%. Para analisar se as diferenças entre os métodos são significativas, a Tabela 10 reporta o *valor-p* dos testes estatísticos de Wilcoxon realizados par a par.

Tabela 10 – Estatística de teste *p* para os resultados de *Hipervolume* para as instâncias grandes.

Abordagens	Estatística de teste <i>p</i>				
	Abordagem 1	Abordagem 2	Abordagem 3	Abordagem 4	Abordagem 5
Abordagem 1	-	0,014722	6,92E-06	0,000608	6,56E-07
Abordagem 2	0,014722	-	7,99E-06	0,000153	2,46E-07
Abordagem 3	6,92E-06	7,99E-06	-	0,091950	5,59E-09
Abordagem 4	0,000608	0,000153	0,091950	-	8,33E-07
Abordagem 5	6,56E-07	2,46E-07	5,59E-09	8,33E-07	-

Fonte: Elaborado pela autora (2024)

O valor da estatística *valor-p* na Tabela 10 mostra que a Abordagem 5 é superior a todas as outras em *Hipervolume* para grandes instâncias. Ainda, o *valor-p* = 0,014722 para a comparação entre as Abordagens 1 e 2 revela que a 2 supera a 1. Nesse caso, as estratégias de “melhor melhoria” e EPS-K foram capazes de melhorar a fronteira da 2. Mas, não melhoraram a fronteira da Abordagem 4 em relação à 3 de forma significativa.

Quanto ao indicador de desempenho *Pureza*, a Tabela 11 apresenta as médias e medianas de cada abordagem e o “GAP” entre a Abordagem 1 e todas as outras. Da mesma forma, a Tabela 12 mostra a estatística de teste *valor-p* da comparação par a par entre os métodos.

Tabela 11 – Resultados para *Pureza* para instâncias grandes.

Instâncias	Pureza				
	Abordagem 1	Abordagem 2	Abordagem 3	Abordagem 4	Abordagem 5
31	0,6170	0,5870	0,1739	0,1458	0,8542
32	0,8246	0,7895	0,4151	0,3208	0,8983
33	0,7742	0,8475	0,6500	0,4035	0,8644
34	0,5738	0,7333	0,1935	0,1290	0,7419
35	1,0000	1,0000	0,4865	0,2973	1,0000
36	0,6818	0,5417	0,8947	0,9474	0,8000
37	0,6176	0,5143	0,3636	0,4545	0,6286
38	0,3438	0,4286	0,0758	0,4375	0,4219
39	0,6909	0,7143	0,3333	0,4528	0,9107
40	0,2549	0,3654	0,2115	0,3922	0,7885
41	0,9375	0,9375	0,7500	0,7500	1,0000
42	0,9286	0,9286	0,8077	0,8846	0,9286
43	0,4561	0,4528	0,1176	0,0638	0,7358
44	0,2500	0,2090	0,2903	0,5469	0,2857
45	0,4068	0,4138	0,1818	0,3091	0,7407
46	0,4643	0,3667	0,1429	0,6071	0,3448
47	0,4179	0,4030	0,3000	0,4839	0,7049
48	0,2289	0,3218	0,2381	0,2738	0,6854
49	0,3782	0,3457	0,0894	0,1098	0,7907
50	0,3333	0,3789	0,2784	0,2556	0,8351
51	0,6154	0,5000	0,2308	0,5333	0,5333
52	0,5000	0,6000	0,2381	0,3889	0,7368
53	0,4500	0,4727	0,2857	0,3273	0,5091
54	0,4483	0,4821	0,3148	0,2830	0,7451
55	0,7652	0,8036	0,5780	0,6182	0,9099
56	0,8095	0,7000	0,5500	0,5000	0,6818
57	0,4091	0,4091	0,2857	0,6190	0,6667
58	0,6875	0,7000	0,4194	0,6129	0,9000
59	0,3736	0,4516	0,4000	0,4675	0,6023
60	0,4688	0,4921	0,4386	0,3966	0,5500
Médias	0,5569	0,5630	0,3578	0,4337	0,7265
GAP	0,0000%	-1,0942%	35,7462%	22,1186%	-30,4511%
Mediana	0,4844	0,4960	0,2952	0,4205	0,7413

Fonte: Elaborado pela autora (2024)

Conforme a Tabela 11, a Abordagem 5 aponta a maior média de *Pureza*, com 0,7265, seguido da 2, com 0,5630 e da 1, com 0,5569. Além disso, a Abordagem 5 melhora a média da 1 em 30,4511% e a 2 melhora a da 1 em 1,0942%.

De acordo com os valores de teste da estatística *valor-p* apresentados na Tabela 12, a Abordagem 5 é superior a todas as outras e a Abordagem 2 é equivalente à 1 (*valor-p* = 0,652813). Ainda, as estratégias da Abordagem 4 mostraram ser eficientes quando se compara 4 e 3 (*valor-p* = 0,020266).

Tabela 12 – Estatística de teste *p* para os resultados de *Pureza* para as instâncias grandes.

Abordagens	Estatística de teste <i>p</i>				
	Abordagem 1	Abordagem 2	Abordagem 3	Abordagem 4	Abordagem 5
Abordagem 1	-	0,652813	9,98E-07	0,012048	0,000018
Abordagem 2	0,652813	-	2,35E-06	0,007612	3,20E-07
Abordagem 3	9,98E-07	2,35E-06	-	0,020266	9,31E-09
Abordagem 4	0,012048	0,007612	0,020266	-	8,59E-06
Abordagem 5	0,000018	3,20E-07	9,31E-09	8,59E-06	-

Fonte: Elaborado pela autora (2024)

Para o indicador de performance  $D_r$ , a Tabela 13 apresenta as médias e medianas de cada abordagem e o “GAP” entre a Abordagem 1 e todas as outras. Da mesma forma, a Tabela 14 mostra a estatística de teste *valor-p* da comparação par a par entre os métodos.

Menores valores de  $D_r$  indicam um melhor desempenho. Conforme a Tabela 13 a Abordagem 5 apresenta a menor média, de 0,0015. Essa melhora a Abordagem 1 em 121,70% e a 2 melhora a 1 em 1,56%.

A análise estatística dos resultados de  $D_r$  para as grandes instâncias, Tabela 14 aponta que a 5 supera todas as outras. A 2 é equivalente à 1 e a 4 é equivalente à 3. Portanto, nesse caso, as estratégias de “melhor melhoria” e utilização de “EPS-K” não retornaram fronteiras significativamente melhores.

Em suma, para as grandes instâncias, podemos inferir que, entre os métodos heurísticos empregados, a Abordagem 5 é superior a todas as outras. Além disso, dado que a Abordagem 2 apresenta uma melhoria significativa da *Pureza* em relação à Abordagem 1, podemos inferir que ela também é melhor que 1. Novamente, as Abordagens 3 e 4 apresentaram os piores desempenhos.

As estratégias de “melhor melhoria” e a utilização de EPS-K ajudam a encontrar melhores fronteiras. Quando essas estratégias são aplicadas ao 2PPLS com uma busca em vizinhança na segunda fase, os resultados melhoram ainda mais.

Quanto ao tempo computacional, a Tabela 15 mostra o tempo computacional em segundos para executar o procedimento de cada uma das heurísticas. Conforme os testes estatísticos realizados, apresentados na Tabela 16, os tempos entre todas as abordagens apresentam diferenças significativas.

Para as instâncias grandes, a Abordagem 1 é, novamente, a mais rápida, executando as instâncias grandes com uma média de tempo de 90,8097 segundos. A melhoria do desempenho medida

Tabela 13 – Resultados para  $D_r$  para instâncias grandes.

Instâncias	$D_r$				
	Abordagem 1	Abordagem 2	Abordagem 3	Abordagem 4	Abordagem 5
31	0,0019	0,0017	0,0059	0,0050	0,0005
32	0,0014	0,0011	0,0041	0,0055	0,0005
33	0,0015	0,0004	0,0011	0,0028	0,0004
34	0,0014	0,0006	0,0038	0,0104	0,0004
35	0,0000	0,0000	0,0079	0,0091	0,0000
36	0,0067	0,0112	0,0011	0,0005	0,0027
37	0,0025	0,0030	0,0032	0,0025	0,0022
38	0,0032	0,0028	0,0053	0,0041	0,0023
39	0,0016	0,0018	0,0043	0,0029	0,0001
40	0,0040	0,0041	0,0080	0,0063	0,0009
41	0,0002	0,0002	0,0023	0,0032	0,0000
42	0,0002	0,0001	0,0009	0,0007	0,0001
43	0,0025	0,0016	0,0045	0,0049	0,0006
44	0,0032	0,0031	0,0024	0,0013	0,0017
45	0,0037	0,0034	0,0066	0,0044	0,0011
46	0,0072	0,0085	0,0146	0,0050	0,0073
47	0,0038	0,0030	0,0045	0,0039	0,0009
48	0,0033	0,0029	0,0052	0,0043	0,0011
49	0,0020	0,0019	0,0053	0,0049	0,0004
50	0,0062	0,0050	0,0024	0,0037	0,0008
51	0,0114	0,0202	0,0211	0,0128	0,0068
52	0,0088	0,0038	0,0187	0,0132	0,0018
53	0,0031	0,0023	0,0045	0,0045	0,0017
54	0,0027	0,0022	0,0037	0,0035	0,0006
55	0,0007	0,0005	0,0013	0,0011	0,0001
56	0,0019	0,0040	0,0087	0,0100	0,0055
57	0,0083	0,0071	0,0072	0,0043	0,0018
58	0,0025	0,0015	0,0028	0,0018	0,0005
59	0,0022	0,0020	0,0018	0,0016	0,0010
60	0,0026	0,0022	0,0027	0,0033	0,0015
Médias	0,0034	0,0034	0,0055	0,0047	0,0015
GAP	0,00%	1,56%	39,41%	29,02%	-121,70%
Mediana	0,0025	0,0023	0,0044	0,0042	0,0009

Fonte: Elaborado pela autora (2024)

Tabela 14 – Estatística de teste  $p$  para os resultados de  $D_r$  para as instâncias grandes.

Abordagens	Estatística de teste $p$				
	Abordagem 1	Abordagem 2	Abordagem 3	Abordagem 4	Abordagem 5
Abordagem 1	-	0,090024	0,000608	0,015460	3,98E-06
Abordagem 2	0,090024	-	9,90E-05	0,027741	1,54E-06
Abordagem 3	0,000608	9,90E-05	-	0,134735	4,66E-08
Abordagem 4	0,015460	0,027741	0,134735	-	8,33E-07
Abordagem 5	3,98E-06	1,54E-06	4,66E-08	8,33E-07	-

Fonte: Elaborado pela autora (2024)

pelos indicadores de performance da fronteira proveniente da Abordagem 5 em relação à 1 custou um tempo maior em 165,63%. Já a melhoria da Abordagem 2 resultou em um aumento do tempo em 55,02%.

A Figura 18, mostra as fronteiras obtidas pelos métodos 1, 2 e 5 para as instâncias 40 e 50. É possível verificar que a fronteira obtida pela Abordagem 5 (linha amarela) é a mais próxima do ponto ideal. As fronteiras das outras instâncias grandes podem ser visualizadas no Apêndice B, nas

Tabela 15 – Resultados para tempo computacional para instâncias grandes.

Instâncias	Tempo Computacional				
	Abordagem 1	Abordagem 2	Abordagem 3	Abordagem 4	Abordagem 5
31	3,3315	6,2822	9,2500	12,9199	6,3040
32	4,1808	5,7752	8,6559	15,8523	10,5057
33	5,8276	11,3798	17,6638	57,6409	17,2537
34	7,6632	16,9870	41,3183	233,4821	28,1459
35	6,2090	7,5146	29,4288	205,7353	15,8398
36	4,5335	7,1433	6,1021	13,2112	11,0388
37	7,5947	13,2138	15,8101	38,3461	22,8594
38	12,6700	19,5095	37,5492	107,2890	48,8593
39	20,5847	41,7082	61,7974	217,6672	103,8467
40	33,6996	90,1789	95,8869	329,8059	142,9761
41	4,8900	6,9418	7,0463	12,8630	11,1638
42	9,3143	14,1348	12,8530	29,7917	23,3861
43	16,0447	26,3160	43,2909	166,4663	52,7811
44	26,1753	45,4034	84,4184	339,0495	95,1415
45	50,9296	106,1523	130,9842	509,4285	217,0600
46	21,0782	30,2481	16,7058	50,7285	46,2685
47	50,2439	77,4082	88,5633	134,4180	127,2480
48	78,9724	139,5035	120,1596	268,9026	206,4317
49	115,8290	192,5082	358,8046	723,1498	406,5366
50	193,7142	258,7668	380,7989	728,1733	1017,9185
51	44,4064	59,7620	59,8750	178,2101	91,5139
52	81,8405	111,2497	70,1709	137,1248	176,7645
53	128,9159	192,7922	259,4928	535,7720	312,9181
54	210,0718	280,1764	285,5600	981,0010	447,7150
55	333,9569	636,7831	874,2025	4024,3304	773,4653
56	65,4278	94,1215	60,5534	124,3402	144,0730
57	137,6406	178,4160	97,7254	163,3871	272,3183
58	207,1026	285,3763	224,5301	482,1509	445,6756
59	386,1527	642,8130	731,6613	1792,6198	1009,6593
60	455,2905	624,6357	642,4035	1531,5845	950,9202
Médias	90,8097	140,7734	162,4421	471,5147	241,2196
GAP	0,00%	-55,02%	-78,88%	-419,23%	-165,63%

Fonte: Elaborado pela autora (2024)

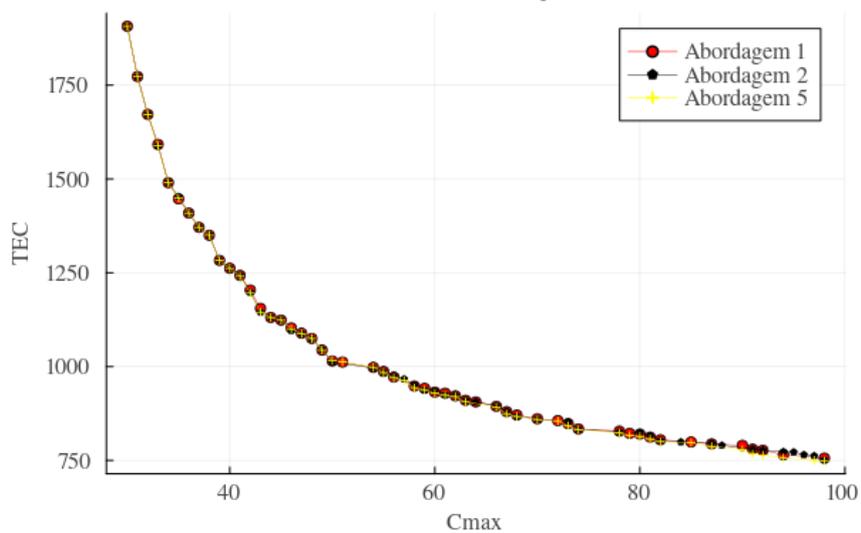
Tabela 16 – Estatística de teste  $p$  para os resultados de tempo computacional para as instâncias grandes.

Abordagens	Estatística de teste $p$				
	Abordagem 1	Abordagem 2	Abordagem 3	Abordagem 4	Abordagem 5
Abordagem 1	-	1,86E-09	1,06E-05	1,86E-09	1,86E-09
Abordagem 2	1,86E-09	-	0,032721	1,86E-08	1,86E-09
Abordagem 3	1,06E-05	0,032721	-	1,86E-09	5,59E-05
Abordagem 4	1,86E-09	1,86E-08	1,86E-09	-	0,000170
Abordagem 5	1,86E-09	1,86E-09	5,59E-05	0,000170	-

Fonte: Elaborado pela autora (2024)

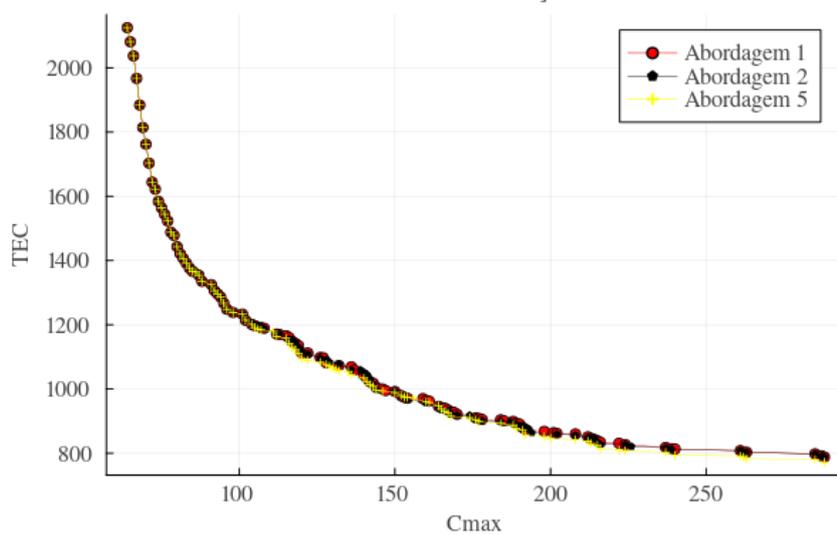
Figuras 49 até 78.

Figura 18 – Fronteiras Abordagem 1, 2 e 5 para instâncias grandes 40 e 50  
Métodos de otimização - 40



(I) Instância 40

Métodos de otimização - 50



(II) Instância 50

Fonte: Elaborado pela autora (2024)

## 8 CONCLUSÕES

Nesta dissertação, foi abordado um problema de sequenciamento de máquinas paralelas com considerações ambientais em um contexto bi-objetivo. A variante estudada visa minimizar simultaneamente duas funções objetivo conflitantes: o *makespan*, que mede o maior tempo de término das tarefas, e o consumo total de energia, considerando tarifas que variam ao longo do dia, política comumente conhecida como *time-of-use*.

Além de apresentar o problema e realizar uma revisão sistemática sobre o tema, este trabalho propôs cinco abordagens heurísticas diferentes. A revisão sistemática identificou lacunas e tendências na literatura, que foram incorporadas neste estudo. O trabalho adaptou o algoritmo proposto por Anghinolfi *et al.* (2021) e a meta-heurística 2PPLS para resolver o PSMPCA. No primeiro caso, foram consideradas duas adaptações: a troca da política de refinamento e a definição de EPS-K. No segundo caso, as adaptações incluíram a troca da política de refinamento, a definição de EPS-K e a busca em vizinhança permitindo pioras no *makespan* até certo ponto, a fim de explorar melhor o espaço de soluções.

Dessas adaptações, surgiram quatro novas propostas, das quais duas se mostraram mais eficientes que a Abordagem 1 re-implementada, originalmente proposta por Anghinolfi *et al.* (2021). Esta abordagem é recente e promissora, conforme testes estatísticos realizados pelos autores que demonstraram sua superioridade em relação a outras alternativas, como os algoritmos evolutivos multi-objetivo NSGA-III e MOEA/D.

As fronteiras obtidas por cada abordagem foram avaliadas por meio de indicadores de performance que medem a qualidade da fronteira em termos de número de soluções, espaçamento entre elas e proximidade do ponto ideal. Os testes computacionais mostraram que a Abordagem 5 supera significativamente a Abordagem 1 em todas as medidas de performance, tanto para instâncias pequenas quanto grandes. Por outro lado, a Abordagem 2 apresentou melhorias significativas em apenas um indicador de performance, apesar de retornar medidas centrais melhores.

A Abordagem 2 aplica um algoritmo de restrição- $\epsilon$ , que resolve o problema a cada iteração, minimizando o *TEC* enquanto mantém o horizonte de tempo como restrição. A heurística construtiva desse método aloca as tarefas às unidades de tempo com menores tarifas nas máquinas de menor custo energético. A heurística de refinamento realiza trocas entre blocos do sequenciamento para reduzir o custo total energético sem piorar o *makespan*, repetindo o processo até alcançar um ótimo local, utilizando a política de “melhor melhoria”.

A Abordagem 5 consiste na meta-heurística 2PPLS, onde a primeira fase é caracterizada

por encontrar soluções a partir da Abordagem 2, e a segunda fase explora o espaço de soluções por meio de uma busca em vizinhança previamente definida. No entanto, essas melhorias aumentaram o tempo computacional, embora ainda resolvam o problema em tempo razoável e competitivo.

Essa pesquisa apresenta os seguintes impactos ambientais, sociais e econômicos:

- **Impactos Ambientais:** A pesquisa aborda diretamente a questão do consumo energético, um dos principais aspectos ambientais em processos industriais. A minimização do consumo de energia, especialmente em horários de pico onde a tarifa é mais alta, contribui para a redução da demanda de energia e, conseqüentemente, diminui a necessidade de geração de energia por fontes poluentes. Esta abordagem promove práticas mais sustentáveis na gestão de recursos energéticos, resultando em menor emissão de gases de efeito estufa e outras substâncias nocivas ao meio ambiente.
- **Impactos Sociais:** A otimização do consumo energético também possui implicações sociais significativas. Reduzindo a carga energética durante os picos, a pesquisa contribui para a estabilidade do fornecimento de energia elétrica, prevenindo apagões e interrupções que podem afetar a qualidade de vida das comunidades. Além disso, ao promover práticas sustentáveis, a pesquisa incentiva uma consciência ambiental maior tanto nas indústrias quanto entre os consumidores finais. A revisão sistemática realizada nesta pesquisa também beneficia a comunidade acadêmica, fornecendo um mapeamento abrangente do estado da arte e identificando lacunas que podem ser exploradas em futuras pesquisas, fomentando assim o avanço do conhecimento e a colaboração entre pesquisadores.
- **Impactos Econômicos:** Economicamente, a pesquisa proporciona vantagens competitivas às empresas que adotam as heurísticas propostas. A redução do consumo energético pode resultar em diminuição significativa dos custos operacionais, aumentando a eficiência e a rentabilidade. Empresas que conseguem otimizar seus processos com foco na sustentabilidade podem também melhorar sua imagem corporativa, atraindo consumidores preocupados com práticas ambientais responsáveis e investidores interessados em empresas sustentáveis.

O agendamento da eficiência energética está se tornando um tópico cada vez mais relevante. Em futuras pesquisas, diversas estratégias de eficiência energética serão exploradas. Além disso, pretende-se aprimorar as propostas deste trabalho, desenvolvendo uma busca em vizinhança mais eficiente para a meta-heurística 2PPLS e reduzindo o tempo computacional dos algoritmos em geral.

Dessa forma, a pesquisa não apenas avança o estado da arte em otimização de sequenciamento de produção com considerações ambientais, mas também fornece uma heurística aplicável e adaptável a diferentes contextos industriais, promovendo práticas mais sustentáveis e economicamente

vantajosas.

## REFERÊNCIAS

- ABDEL-BASSET, M.; ABDEL-FATAH, L.; SANGAIAH, A. K. Metaheuristic algorithms: A comprehensive review. **Computational intelligence for multimedia big data on the cloud with engineering applications**, Elsevier, p. 185–231, 2018.
- ABIKARRAM, J. B.; MCCONKY, K.; PROANO, R. Energy cost minimization for unrelated parallel machine scheduling under real time and demand charge pricing. **Journal of cleaner production**, Elsevier, v. 208, p. 232–242, 2019.
- AFZALIRAD, M.; SHAFIPOUR, M. Design of an efficient genetic algorithm for resource-constrained unrelated parallel machine scheduling problem with machine eligibility restrictions. **Journal of Intelligent Manufacturing**, Springer, v. 29, n. 2, p. 423–437, 2018.
- ANGEL, E.; BAMPIS, E.; KACEM, F. Energy aware scheduling for unrelated parallel machines. In: **IEEE. 2012 IEEE International Conference on Green Computing and Communications**. [S. l.], 2012. p. 533–540.
- ANGHINOLFI, D.; PAOLUCCI, M.; RONCO, R. A bi-objective heuristic approach for green identical parallel machine scheduling. **European journal of operational research**, Elsevier, v. 289, n. 2, p. 416–434, 2021.
- ANTONIADIS, A.; GARG, N.; KUMAR, G.; KUMAR, N. Parallel machine scheduling to minimize energy consumption. In: **SIAM. Proceedings of the fourteenth annual ACM-SIAM symposium on discrete algorithms**. [S. l.], 2020. p. 2758–2769.
- ARENALES, M.; ARMENTANO, V.; MORABITO, R.; YANASSE, H. **Pesquisa Operacional**. Rio de Janeiro: Elsevier, 2015. ISBN 978-85-352-5193-7.
- ASADPOUR, M.; HODAEI, Z.; AZAMI, M.; KEHTARI, E.; VESAL, N. A green model for identical parallel machines scheduling problem considering tardy jobs and job splitting property. **Sustainable Operations and Computers**, Elsevier, v. 3, p. 149–155, 2022.
- BALANÇO ENERGÉTICO NACIONAL 2023. **Empresa de Pesquisa Energética**. Rio de Janeiro: EPE: [S. n.], 2023. Disponível em: <https://www.epe.gov.br/pt/publicacoes-dados-abertos/publicacoes/balanco-energetico-nacional-2023>. Acesso em: 27 jul. 2023.
- BANDYOPADHYAY, S.; PAL, S. K.; ARUNA, B. Multiobjective gas, quantitative indices, and pattern classification. **IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)**, IEEE, v. 34, n. 5, p. 2088–2099, 2004.
- BEKTUR, G. An nsga-ii-based memetic algorithm for an energy-efficient unrelated parallel machine scheduling problem with machine-sequence dependent setup times and learning effect. **Arabian Journal for Science and Engineering**, Springer, v. 47, n. 3, p. 3773–3788, 2022.
- BITTENCOURT, M. L.; OLIVEIRA, C. M. R. de; KRAMER, R.; LINS, I. D. Simheurística para programação de máquina única com tempos de setup dependente da sequência e estocásticos. In: **Anais do LV Simpósio Brasileiro de Pesquisa Operacional**. São José dos Campos - SP: SOBRAPO, 2023.
- BLUM, C.; ROLI, A. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. **ACM computing surveys (CSUR)**, Acm New York, NY, USA, v. 35, n. 3, p. 268–308, 2003.

BOK, Y.; LEE, N. K.; JO, S.; LEE, S.; KWEON, S. J.; NA, H. S. The production scheduling problem employing non-identical parallel machines with due dates considering carbon emissions and multiple types of energy sources. **Expert Systems with Applications**, Elsevier, v. 238, p. 121990, 2024.

BÉRUBÉ, J.-F.; GENDREAU, M.; POTVIN, J.-Y. An exact formula not shown-constraint method for bi-objective combinatorial optimization problems: Application to the traveling salesman problem with profits. **European Journal of Operational Research**, Elsevier Science, v. 194, n. 1, p. 39–50, 2009.

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

CHE, A.; WU, X.; PENG, J.; YAN, P. Energy-efficient bi-objective single-machine scheduling with power-down mechanism. **Computers & Operations Research**, Elsevier, v. 85, p. 172–183, 2017.

CHE, A.; ZHANG, S.; WU, X. Energy-conscious unrelated parallel machine scheduling under time-of-use electricity tariffs. **Journal of Cleaner Production**, v. 156, p. 688–697, 2017. ISSN 0959-6526. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0959652617307175>.

CHE, A.; ZHANG, S.; WU, X. Energy-conscious unrelated parallel machine scheduling under time-of-use electricity tariffs. **Journal of cleaner production**, Elsevier, v. 156, p. 688–697, 2017.

CHENG, J.; CHU, F.; ZHOU, M. An improved model for parallel machine scheduling under time-of-use electricity price. **IEEE Transactions on Automation Science and Engineering**, IEEE, v. 15, n. 2, p. 896–899, 2017.

CHENG, J.; WU, P.; CHU, F. Mixed-integer programming for unrelated parallel machines scheduling problem considering electricity cost and makespan penalty cost. In: IEEE. **2019 international conference on industrial engineering and systems management (IESM)**. [S. l.], 2019. p. 1–5.

COSTA, L. C. A. **Uma heurística baseada em busca local de pareto para o pollution-routing problem bi-objetivo**. Dissertação (Mestrado em Engenharia de Produção) – Universidade Federal da Paraíba, João Pessoa, 2015.

COSTA, L. C. A. d. *et al.* Uma huerística baseada em busca local de pareto para o pollution-routing problem bi-objetivo. In: **Anais do LV Simpósio Brasileiro de Pesquisa Operacional**. [S. l.]: SOBRAPO, 2015.

COTA, L. P.; COELHO, V. N.; GUIMARÃES, F. G.; SOUZA, M. J. Bi-criteria formulation for green scheduling with unrelated parallel machines with sequence-dependent setup times. **International Transactions in Operational Research**, Wiley Online Library, v. 28, n. 2, p. 996–1017, 2021.

COTA, L. P.; GUIMARÃES, F. G.; RIBEIRO, R. G.; MENEGHINI, I. R.; OLIVEIRA, F. B. de; SOUZA, M. J.; SIARRY, P. An adaptive multi-objective algorithm based on decomposition and large neighborhood search for a green machine scheduling problem. **Swarm and Evolutionary Computation**, Elsevier, v. 51, p. 100601, 2019.

DING, J.-Y.; SONG, S.; ZHANG, R.; CHIONG, R.; WU, C. Parallel machine scheduling under time-of-use electricity prices: New models and optimization approaches. **IEEE Transactions on Automation Science and Engineering**, IEEE, v. 13, n. 2, p. 1138–1154, 2015.

DONTHU, N.; KUMAR, S.; MUKHERJEE, D.; PANDEY, N.; LIM, W. M. How to conduct a bibliometric analysis: An overview and guidelines. **Journal of business research**, Elsevier, v. 133, p. 285–296, 2021.

- FALLAHI, A.; SHAHIDI-ZADEH, B.; NIAKI, S. T. A. Unrelated parallel batch processing machine scheduling for production systems under carbon reduction policies: Nsga-ii and mogwo metaheuristics. **Soft Computing**, Springer, v. 27, n. 22, p. 17063–17091, 2023.
- FANG, K.; UHAN, N. A.; ZHAO, F.; SUTHERLAND, J. W. Scheduling on a single machine under time-of-use electricity tariffs. **Annals of Operations Research**, Springer, v. 238, n. 1, p. 199–227, 2016.
- FANJUL-PEYRO, L.; RUIZ, R.; PEREA, F. Reformulations and an exact algorithm for unrelated parallel machine scheduling problems with setup times. **Computers & Operations Research**, Elsevier, v. 101, p. 173–182, 2019.
- FENG, X.; PENG, H. Robust identical parallel machine scheduling with two-stage time-of-use tariff and not-all-machine option. **International Journal of Production Research**, Taylor & Francis, v. 62, n. 1-2, p. 380–403, 2024.
- FUCHIGAMI, H. Y. Material didático, versão 8.0, **Introdução ao sequenciamento da produção**. 2021.
- GAGGERO, M.; PAOLUCCI, M.; RONCO, R. Exact and heuristic solution approaches for energy-efficient identical parallel machine scheduling with time-of-use costs. **European Journal of Operational Research**, Elsevier, 2023.
- GALVÃO, M. C. B.; RICARTE, I. L. M. Revisão sistemática da literatura: conceituação, produção e publicação. **Logeion: Filosofia da informação**, v. 6, n. 1, p. 57–73, 2019.
- GANDIBLEUX, X. Multiple criteria optimization: state of the art annotated bibliographic surveys. Springer Science & Business Media, 2006.
- GAO, M.; LIU, C.; LIU, Z.; CHEN, X. A tabu search algorithm for the unrelated parallel machine scheduling problem with varied carbon emission constraints in different time intervals. **Journal of the Operational Research Society**, Taylor & Francis, p. 1–15, 2023.
- GENDREAU, M.; POTVIN, J.-Y. **Handbook of Metaheuristics**. 3. ed. [S. l.]: Springer, 2010. ISBN 978-3-319-91086-4.
- GEOFFRION, A. M. Proper efficiency and the theory of vector maximization. **Journal of mathematical analysis and applications**, Elsevier, v. 22, n. 3, p. 618–630, 1968.
- GIRET, A.; TRENTESAUX, D.; PRABHU, V. Sustainability in manufacturing operations scheduling: A state of the art review. **Journal of Manufacturing Systems**, Elsevier, v. 37, p. 126–140, 2015.
- GOLDBARG, M. C.; LUNA, H. P. L. **Otimização combinatória e programação linear: modelos e algoritmos**. [S. l.]: Elsevier, 2005.
- GRAHAM, R.; LAWLER, E.; LENSTRA, J.; KAN, A. Optimization and approximation in deterministic sequencing and scheduling: a survey. In: HAMMER, P.; JOHNSON, E.; KORTE, B. (Ed.). **Discrete Optimization II**. Elsevier, 1979, (Annals of Discrete Mathematics, v. 5). p. 287–326. Disponível em: <https://www.sciencedirect.com/science/article/pii/S016750600870356X>.
- HEYDAR, M.; MARDANEH, E.; LOXTON, R. Approximate dynamic programming for an energy-efficient parallel machine scheduling problem. **European Journal of Operational Research**, Elsevier, v. 302, n. 1, p. 363–380, 2022.

HIDRI, L.; ALQAHTANI, A.; GAZDAR, A.; YOUSSEF, B. B. Green scheduling of identical parallel machines with release date, delivery time and no-idle machine constraints. **Sustainability**, MDPI, v. 13, n. 16, p. 9277, 2021.

HU, C.; LU, S.; KONG, M.; LIU, X.; PARDALOS, P. M. Parallel machine scheduling with linearly increasing energy consumption cost. **Annals of Mathematics and Artificial Intelligence**, Springer, v. 91, n. 2, p. 239–258, 2023.

ISHIBUCHI, H.; YOSHIDA, T.; MURATA, T. Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling. **IEEE transactions on evolutionary computation**, IEEE, v. 7, n. 2, p. 204–223, 2003.

JASZKIEWICZ, A. On the performance of multiple-objective genetic local search on the 0/1 knapsack problem—a comparative experiment. **IEEE Transactions on Evolutionary Computation**, IEEE, v. 6, n. 4, p. 402–412, 2002.

JIANG, Y.; TANG, X.; LI, K.; CHENG, T.; JI, M. Approximation algorithms for bi-objective parallel-machine scheduling in green manufacturing. **Computers & Industrial Engineering**, Elsevier, v. 176, p. 108949, 2023.

JOHNSON, D.; GAREY, M. **A guide to the theory of np-completeness. computers and intractability**. [S. l.]: WH Freeman and Company New York, NY, USA, 1979.

KRAMER, R. H. F. R. **Uma abordagem heurística para o pollution-routing problem**. Dissertação (Mestrado em Engenharia de Produção) – Universidade Federal da Paraíba, João Pessoa, 2014.

KURNIAWAN, B.; CHANDRAMITASARI, W.; GOZALI, A. A.; WENG, W.; FUJIMURA, S. Triple-chromosome genetic algorithm for unrelated parallel machine scheduling under time-of-use tariffs. **IEEJ Transactions on Electrical and Electronic Engineering**, Wiley Online Library, v. 15, n. 2, p. 208–217, 2020.

KURNIAWAN, B.; GOZALI, A.; WENG, W.; FUJIMURA, S. A genetic algorithm for unrelated parallel machine scheduling minimizing makespan cost and electricity cost under time-of-use (tou) tariffs with job delay mechanism. In: IEEE. **2017 IEEE international conference on industrial engineering and engineering management (IEEM)**. [S. l.], 2017. p. 583–587.

LI, K.; ZHANG, X.; LEUNG, J. Y.-T.; YANG, S.-L. Parallel machine scheduling problems in green manufacturing industry. **Journal of Manufacturing Systems**, Elsevier, v. 38, p. 98–106, 2016.

LI, Z.; YANG, H.; ZHANG, S.; LIU, G. Unrelated parallel machine scheduling problem with energy and tardiness cost. **The International Journal of Advanced Manufacturing Technology**, Springer, v. 84, p. 213–226, 2016.

LIANG, P.; YANG, H.-d.; LIU, G.-s.; GUO, J.-h. *et al.* An ant optimization model for unrelated parallel machine scheduling with energy consumption and total tardiness. **Mathematical Problems in Engineering**, Hindawi, v. 2015, 2015.

LU, H.; QIAO, F. An improved genetic algorithm for a parallel machine scheduling problem with energy consideration. In: IEEE. **2017 13th IEEE conference on automation science and engineering (CASE)**. [S. l.], 2017. p. 1487–1492.

LU, H.; QIAO, F. A sustainable parallel-machine scheduling problem with time constraint based on hybrid metaheuristic algorithm. In: IEEE. **2020 Chinese Automation Congress (CAC)**. [S. l.], 2020. p. 1506–1510.

LUST, T.; TEGHEM, J. Two-phase pareto local search for the biobjective traveling salesman problem. **Journal of Heuristics**, Springer, v. 16, n. 3, p. 475–510, 2010.

MANSOURI, S. A.; AKTAS, E.; BESIKCI, U. Green scheduling of a two-machine flowshop: Trade-off between makespan and energy consumption. **European journal of operational research**, Elsevier, v. 248, n. 3, p. 772–788, 2016.

MAVROTAS, G. Effective implementation of the  $\epsilon$ -constraint method in multi-objective mathematical programming problems. **Applied mathematics and computation**, Elsevier, v. 213, n. 2, p. 455–465, 2009.

MÓDOS, I.; ŠUCHA, P.; HANZÁLEK, Z. On parallel dedicated machines scheduling under energy consumption limit. **Computers & Industrial Engineering**, Elsevier, v. 159, p. 107209, 2021.

NANTHAPODEJ, R.; LIU, C.-H.; NITISIRI, K.; PATTANAPAIROJ, S. Hybrid differential evolution algorithm and adaptive large neighborhood search to solve parallel machine scheduling to minimize energy consumption in consideration of machine-load balance problems. **Sustainability**, MDPI, v. 13, n. 10, p. 5470, 2021.

NANTHAPODEJ, R.; LIU, C.-H.; NITISIRI, K.; PATTANAPAIROJ, S. Variable neighborhood strategy adaptive search to solve parallel-machine scheduling to minimize energy consumption while considering job priority and control makespan. **Applied Sciences**, MDPI, v. 11, n. 11, p. 5311, 2021.

NATTAF, M.; ARTIGUES, C.; LOPEZ, P.; MEDINA, R.; PARADA, V.; PRADENAS, L. A batch sizing and scheduling problem on parallel machines with different speeds, maintenance operations, setup times and energy costs. In: IEEE. **2015 International Conference on Industrial Engineering and Systems Management (IESM)**. [S. l.], 2015. p. 883–891.

OLIVEIRA, C. M. R. de; BITTENCOURT, M. L.; KRAMER, R. Análise de métodos de otimização multi-objetivo para um problema de sequenciamento verde de máquinas paralelas. In: **Anais do XIII Congresso Brasileiro de Engenharia de Produção**. Ponta Grossa: APREPRO, 2023.

OLIVEIRA, C. M. R. de; BITTENCOURT, M. L.; MOTA, M. R. R.; SOUZA, B. L. de; MACEDO, R. J. S. de; SOUZA, H. R. O. de; MOURA, M. das C. Implementação de um algoritmo genético para a estimação de parâmetros de um processo de renovação generalizado. In: **Anais do LV Simpósio Brasileiro de Pesquisa Operacional**. São José dos Campos - SP: SOBRAPO, 2023.

PAN, Z.; LEI, D.; WANG, L. A knowledge-based two-population optimization algorithm for distributed energy-efficient parallel machines scheduling. **IEEE transactions on cybernetics**, IEEE, v. 52, n. 6, p. 5051–5063, 2020.

PAN, Z.; LEI, D.; WANG, L. A knowledge-based two-population optimization algorithm for distributed energy-efficient parallel machines scheduling. **IEEE TRANSACTIONS ON CYBERNETICS**, v. 52, n. 6, p. 5051–5063, JUN 2022. ISSN 2168-2267.

PAN, Z.; LEI, D.; ZHANG, Q. *et al.* A new imperialist competitive algorithm for multiobjective low carbon parallel machines scheduling. **Mathematical problems in engineering**, Hindawi, v. 2018, 2018.

PAQUETE, L.; CHIARANDINI, M.; STÜTZLE, T. Pareto local optimum sets in the biobjective traveling salesman problem: An experimental study. In: **Metaheuristics for multiobjective optimisation**. [S. l.]: Springer, 2004. p. 177–199.

PEI, Z.; WAN, M.; JIANG, Z.-Z.; WANG, Z.; DAI, X. An approximation algorithm for unrelated parallel machine scheduling under tou electricity tariffs. **IEEE Transactions on Automation Science and Engineering**, IEEE, v. 18, n. 2, p. 743–756, 2020.

PINEDO, M. L. **Scheduling: Theory, Algorithms, and Systems**. 5. ed. [S. l.]: Springer Publishing Company, Incorporated, 2016.

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.

QI, Y.; FU, Y.; SUN, C.; GUO, X. Multi-objective parallel-machine scheduling problem considering energy consumption and deteriorating effect. In: IEEE. **2020 Chinese Automation Congress (CAC)**. [S. l.], 2020. p. 6917–6922.

REGO, M. F.; PINTO, J. C. E.; COTA, L. P.; SOUZA, M. J. A mathematical formulation and an nsga-ii algorithm for minimizing the makespan and energy cost under time-of-use electricity price in an unrelated parallel machine scheduling. **PeerJ Computer Science**, PeerJ Inc., v. 8, p. e844, 2022.

ROCHOLL, J.; MÖNCH, L.; FOWLER, J. W. Electricity power cost-aware scheduling of jobs on parallel batch processing machines. In: IEEE. **2018 Winter Simulation Conference (WSC)**. [S. l.], 2018. p. 3420–3431.

SABERI-ALIABAD, H.; REISI-NAFCHI, M.; MOSLEHI, G. Energy-efficient scheduling in an unrelated parallel-machine environment under time-of-use electricity tariffs. **Journal of Cleaner Production**, Elsevier, v. 249, p. 119393, 2020.

SAFARZADEH, H.; NIAKI, S. T. A. Bi-objective green scheduling in uniform parallel machine environments. **Journal of cleaner production**, Elsevier, v. 217, p. 559–572, 2019.

SALIMIFARD, K.; LI, J.; MOHAMMADI, D.; MOGHDANI, R. A multi objective volleyball premier league algorithm for green scheduling identical parallel machines with splitting jobs. **Applied Intelligence**, Springer, v. 51, n. 7, p. 4143–4161, 2021.

SANATI, H.; MOSLEHI, G.; REISI-NAFCHI, M. Unrelated parallel machine energy-efficient scheduling considering sequence-dependent setup times and time-of-use electricity tariffs. **EURO Journal on Computational Optimization**, Elsevier, v. 11, p. 100052, 2023.

SEWARD, L. E.; DOANE, D. P. **Estatística Aplicada à Administração e Economia-4**. [S. l.]: AMGH editora, 2014.

SHARMA, A.; ZHAO, F.; SUTHERLAND, J. W. Econological scheduling of a manufacturing enterprise operating under a time-of-use electricity tariff. **Journal of Cleaner Production**, v. 108, p. 256–270, 2015. ISSN 0959-6526. Disponível em: <https://www.sciencedirect.com/science/article/pii/S095965261500712X>.

SHICHIYAKH, R. A.; VORONKOVA, O. Y.; SHAKHBANOVA, I. K.; SHAFRANSKAYA, C. Y.; TITOVA, S. V.; POLTARYKHIN, A. L. Parallel-machine scheduling to minimize tardiness penalty and energy cost by using evolutionary algorithms. **Industrial Engineering & Management Systems**, v. 20, n. 4, p. 637–644, 2021.

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

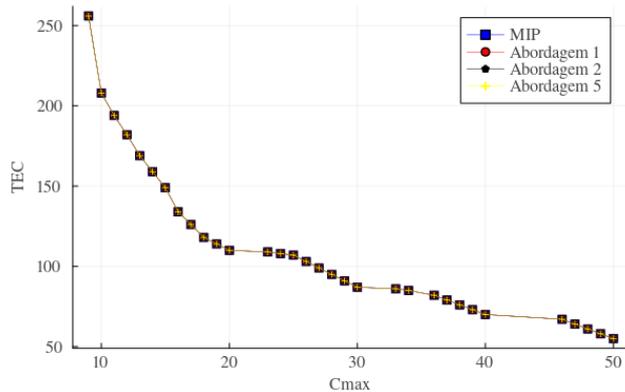
- SOUZA, M. J. F. **Inteligência Computacional para Otimização: metaheurísticas**. Universidade Federal de Ouro Preto, Ouro Preto, Minas Gerais: [S. n.], 2022. Disponível em: <http://www.decom.ufop.br/prof/marcone/Disciplinas/InteligenciaComputacional/InteligenciaComputacional.pdf>.
- TAN, M.; DUAN, B.; SU, Y. Economic batch sizing and scheduling on parallel machines under time-of-use electricity pricing. **Operational Research**, Springer, v. 18, p. 105–122, 2018.
- TAN, M.; YANG, H.-L.; SU, Y.-X. Genetic algorithms with greedy strategy for green batch scheduling on non-identical parallel machines. **Memetic Computing**, Springer, v. 11, n. 4, p. 439–452, 2019.
- WANG, L.; QI, Y. Scheduling an energy-aware parallel machine system with deteriorating and learning effects considering multiple optimization objectives and stochastic processing time. **CMES-Computer Modeling in Engineering & Sciences**, TECH SCIENCE PRESS 871 CORONADO CENTER DR, SUTE 200, HENDERSON, NV 89052 USA, v. 135, n. 1, p. 325–339, 2023.
- WANG, S.; WANG, X.; YU, J.; MA, S.; LIU, M. Bi-objective identical parallel machine scheduling to minimize total energy consumption and makespan. **Journal of cleaner production**, Elsevier, v. 193, p. 424–440, 2018.
- WANG, Y.; CHE, A.; FENG, J. Energy-efficient unrelated parallel machine scheduling with general position-based deterioration. **International Journal of Production Research**, Taylor & Francis, v. 61, n. 17, p. 5886–5900, 2023.
- WANG, Y.; LI, L. Time-of-use based electricity demand response for sustainable manufacturing systems. **Energy**, Elsevier, v. 63, p. 233–244, 2013.
- WOOLSON, R. F. Wilcoxon signed-rank test. **Wiley encyclopedia of clinical trials**, Wiley Online Library, p. 1–3, 2007.
- WU, X.; CHE, A. A memetic differential evolution algorithm for energy-efficient parallel machine scheduling. **Omega**, Elsevier, v. 82, p. 155–165, 2019.
- WU, X.; GUO, P.; WANG, Y.; WANG, Y. Decomposition approaches for parallel machine scheduling of step-deteriorating jobs to minimize total tardiness and energy consumption. **Complex & Intelligent Systems**, Springer, p. 1–16, 2022.
- XIAO, Y.; ZHENG, Y.; YU, Y.; ZHANG, L.; LIN, X.; LI, B. A branch and bound algorithm for a parallel machine scheduling problem in green manufacturing industry considering time cost and power consumption. **Journal of Cleaner Production**, Elsevier, v. 320, p. 128867, 2021.
- XUE, Y.; RUI, Z.; YU, X.; SANG, X.; LIU, W. Estimation of distribution evolution memetic algorithm for the unrelated parallel-machine green scheduling problem. **Memetic Computing**, Springer, v. 11, p. 423–437, 2019.
- YANG, A.; QIAN, B.; HU, R.; WANG, L.; LI, S.-H. Single-machine green scheduling problem of multi-speed machine. In: SPRINGER. **Intelligent Computing Theories and Application: 15th International Conference, ICIC 2019, Nanchang, China, August 3–6, 2019, Proceedings, Part II 15**. [S. l.], 2019. p. 669–677.
- YIN, Y.; WANG, Y.; CHENG, T.; LIU, W.; LI, J. Parallel-machine scheduling of deteriorating jobs with potential machine disruptions. **Omega**, Elsevier, v. 69, p. 17–28, 2017.
- ZANDI, A.; RAMEZANIAN, R.; MONPLAISIR, L. Green parallel machines scheduling problem: A bi-objective model and a heuristic algorithm to obtain pareto frontier. **Journal of the Operational Research Society**, Taylor & Francis, v. 71, n. 6, p. 967–978, 2020.

- ZENG, Y.; CHE, A.; WU, X. Bi-objective scheduling on uniform parallel machines considering electricity cost. **Engineering Optimization**, Taylor & Francis, v. 50, n. 1, p. 19–36, 2018.
- ZHANG, H.; WU, Y.; PAN, R.; XU, G. Two-stage parallel speed-scaling machine scheduling under time-of-use tariffs. **Journal of Intelligent Manufacturing**, Springer, v. 32, p. 91–112, 2021.
- ZHANG, L.; DENG, Q.; GONG, G.; HAN, W. A new unrelated parallel machine scheduling problem with tool changes to minimise the total energy consumption. **International Journal of Production Research**, Taylor & Francis, v. 58, n. 22, p. 6826–6845, 2020.
- ZHENG, X.-L.; WANG, L. A collaborative multiobjective fruit fly optimization algorithm for the resource constrained unrelated parallel machine green scheduling problem. **IEEE Transactions on Systems, Man, and Cybernetics: Systems**, IEEE, v. 48, n. 5, p. 790–800, 2016.
- ZHOU, S.; LI, X.; DU, N.; PANG, Y.; CHEN, H. A multi-objective differential evolution algorithm for parallel batch processing machine scheduling considering electricity consumption cost. **Computers & Operations Research**, v. 96, p. 55–68, 2018. ISSN 0305-0548. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0305054818300923>.
- ZHU, W.; TIANYU, L. A novel multi-objective scheduling method for energy based unrelated parallel machines with auxiliary resource constraints. **IEEE Access**, IEEE, v. 7, p. 168688–168699, 2019.
- ZITZLER, E. **Evolutionary algorithms for multiobjective optimization: Methods and applications**. [S. l.]: Shaker Ithaca, 1999. v. 63.
- ZITZLER, E.; DEB, K.; THIELE, L. Comparison of multiobjective evolutionary algorithms: Empirical results. **Evolutionary computation**, MIT Press, v. 8, n. 2, p. 173–195, 2000.

## APÊNDICE A – FRONTEIRAS DE PARETO PARA INSTÂNCIAS PEQUENAS

Figura 19 – Fronteiras da instância 1

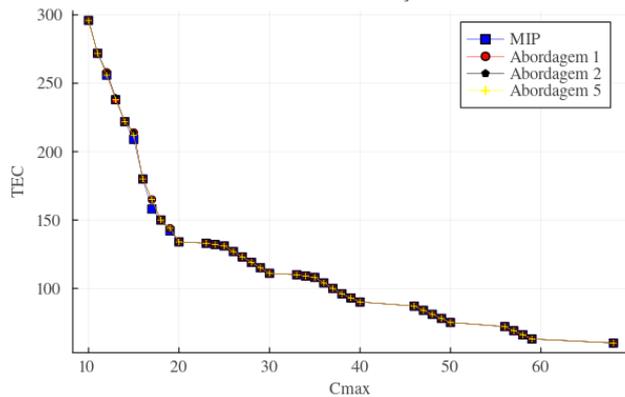
Métodos de otimização - 1



Fonte: Elaborado pela autora (2024)

Figura 20 – Fronteiras da instância 2

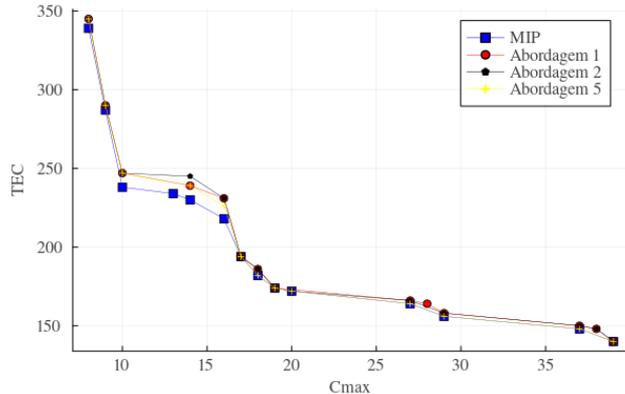
Métodos de otimização - 2



Fonte: Elaborado pela autora (2024)

Figura 21 – Fronteiras da instância 3

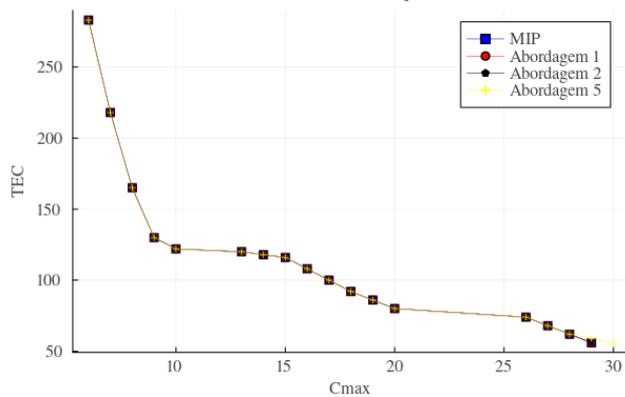
Métodos de otimização - 3



Fonte: Elaborado pela autora (2024)

Figura 22 – Fronteiras da instância 4

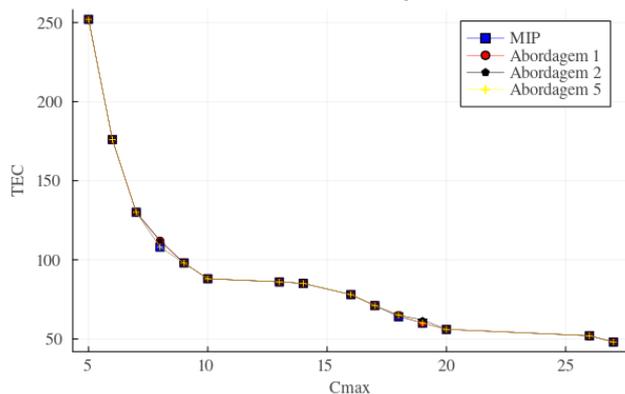
Métodos de otimização - 4



Fonte: Elaborado pela autora (2024)

Figura 23 – Fronteiras da instância 5

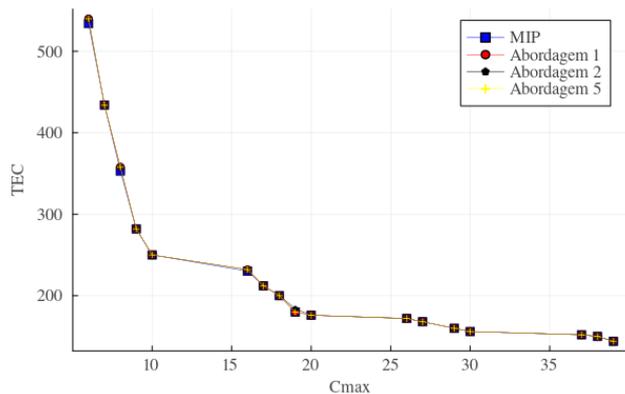
Métodos de otimização - 5



Fonte: Elaborado pela autora (2024)

Figura 24 – Fronteiras da instância 6

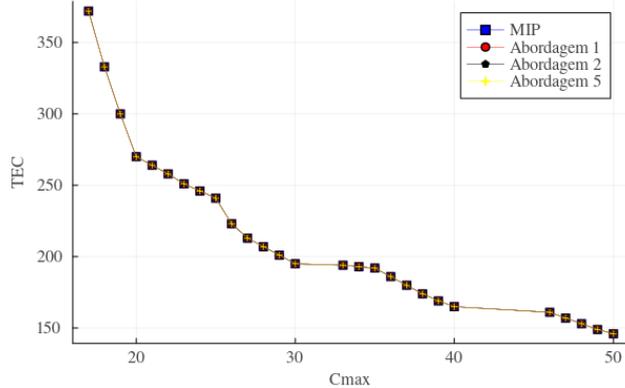
Métodos de otimização - 6



Fonte: Elaborado pela autora (2024)

Figura 25 – Fronteiras da instância 7

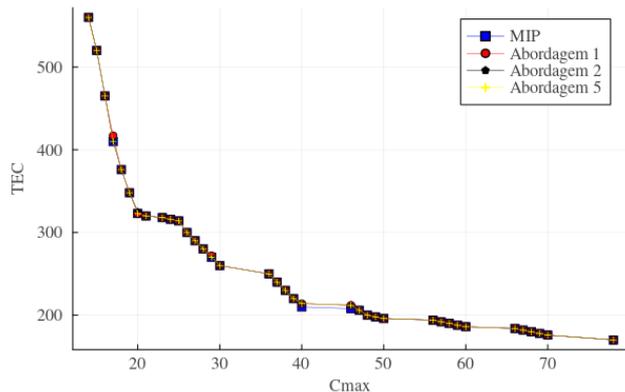
Métodos de otimização - 7



Fonte: Elaborado pela autora (2024)

Figura 26 – Fronteiras da instância 8

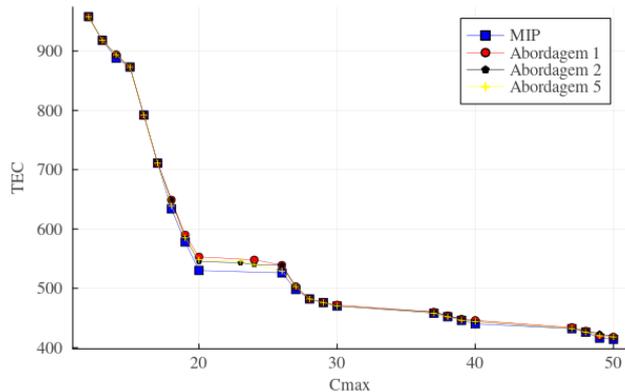
Métodos de otimização - 8



Fonte: Elaborado pela autora (2024)

Figura 27 – Fronteiras da instância 9

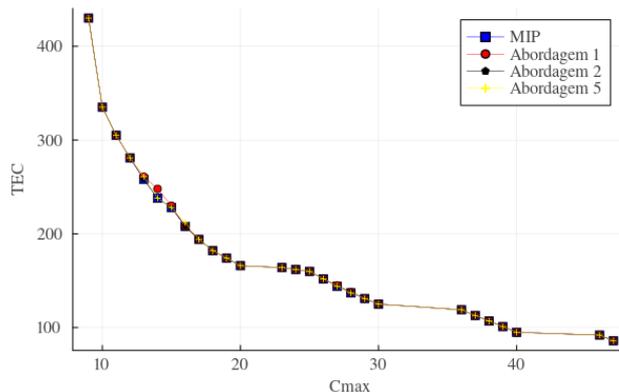
Métodos de otimização - 9



Fonte: Elaborado pela autora (2024)

Figura 28 – Fronteiras da instância 10

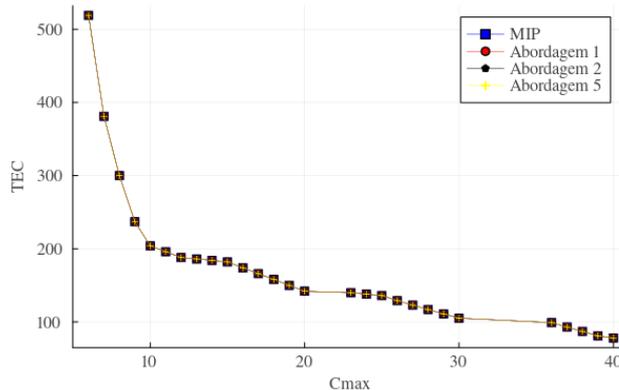
Métodos de otimização - 10



Fonte: Elaborado pela autora (2024)

Figura 29 – Fronteiras da instância 11

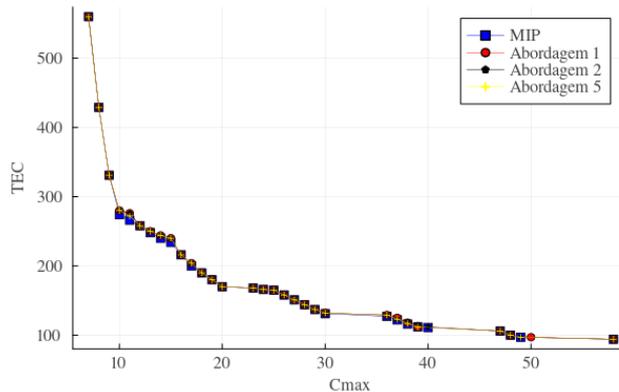
Métodos de otimização - 11



Fonte: Elaborado pela autora (2024)

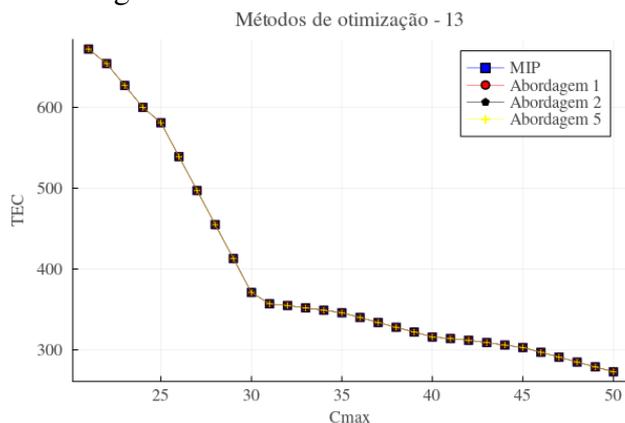
Figura 30 – Fronteiras da instância 12

Métodos de otimização - 12



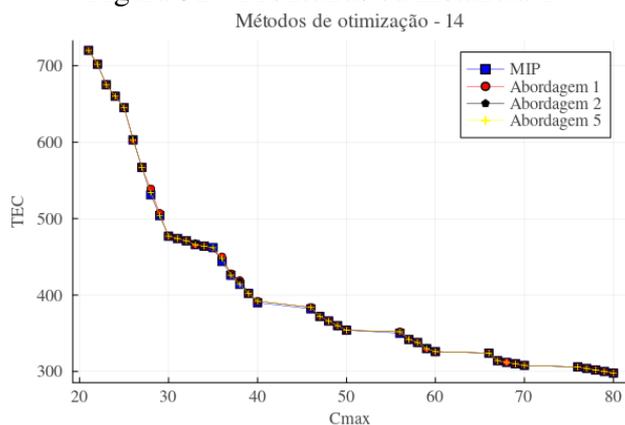
Fonte: Elaborado pela autora (2024)

Figura 31 – Fronteiras da instância 13



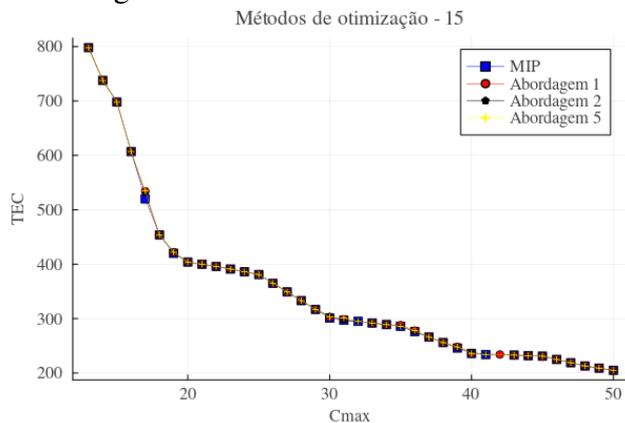
Fonte: Elaborado pela autora (2024)

Figura 32 – Fronteiras da instância 14



Fonte: Elaborado pela autora (2024)

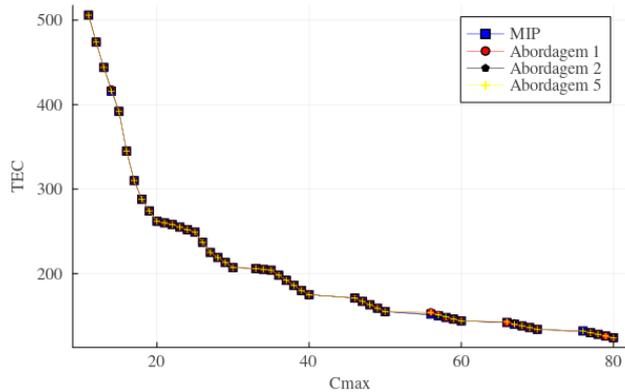
Figura 33 – Fronteiras da instância 15



Fonte: Elaborado pela autora (2024)

Figura 34 – Fronteiras da instância 16

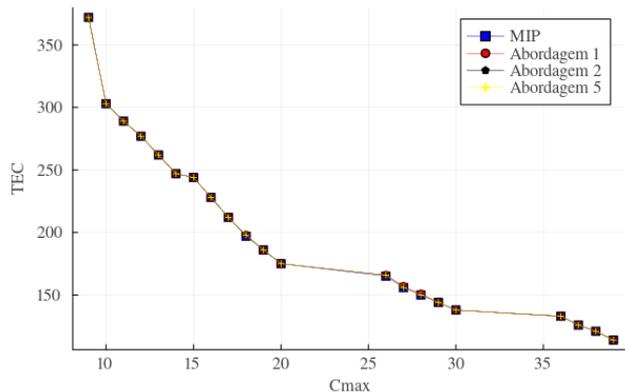
Métodos de otimização - 16



Fonte: Elaborado pela autora (2024)

Figura 35 – Fronteiras da instância 17

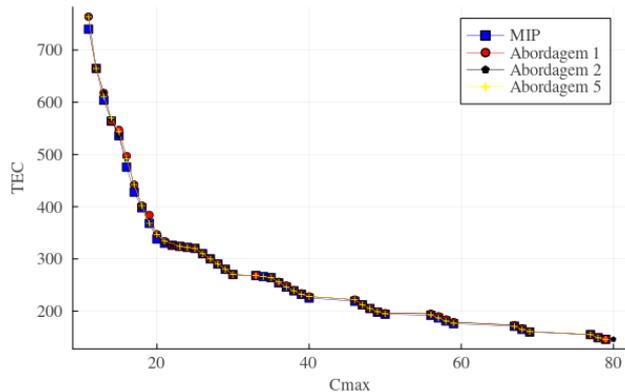
Métodos de otimização - 17



Fonte: Elaborado pela autora (2024)

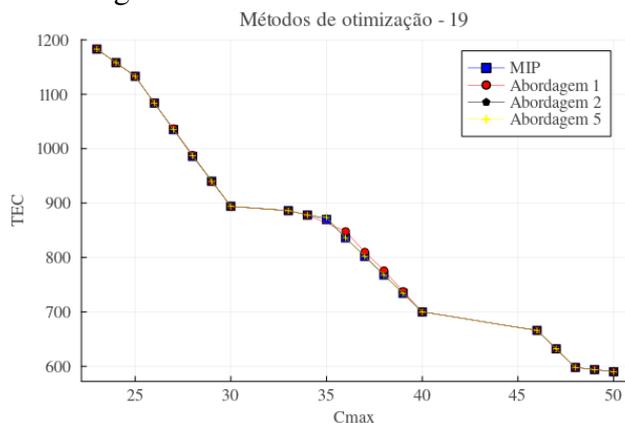
Figura 36 – Fronteiras da instância 18

Métodos de otimização - 18



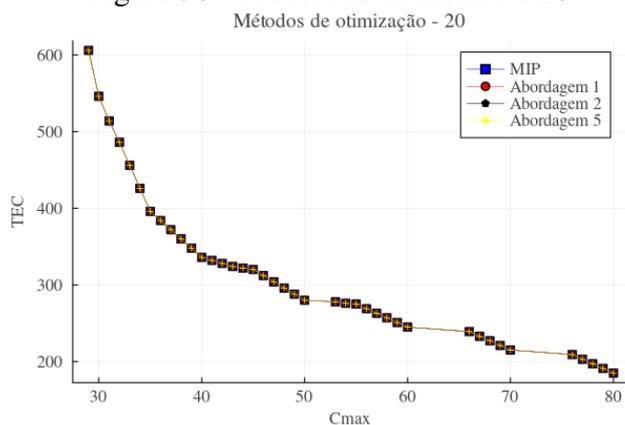
Fonte: Elaborado pela autora (2024)

Figura 37 – Fronteiras da instância 19



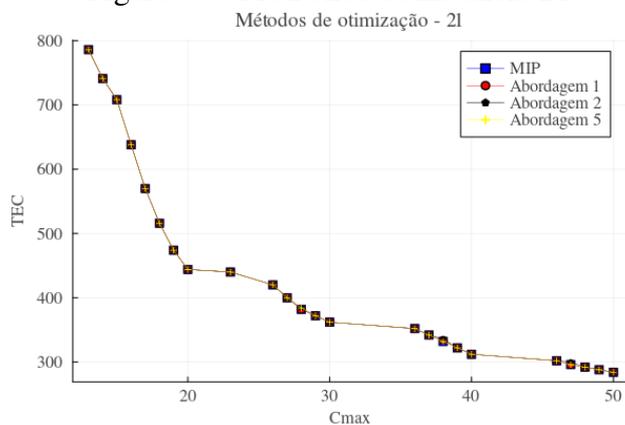
Fonte: Elaborado pela autora (2024)

Figura 38 – Fronteiras da instância 20



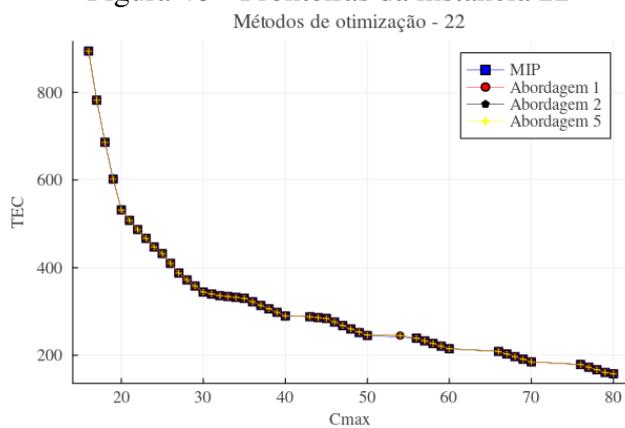
Fonte: Elaborado pela autora (2024)

Figura 39 – Fronteiras da instância 21



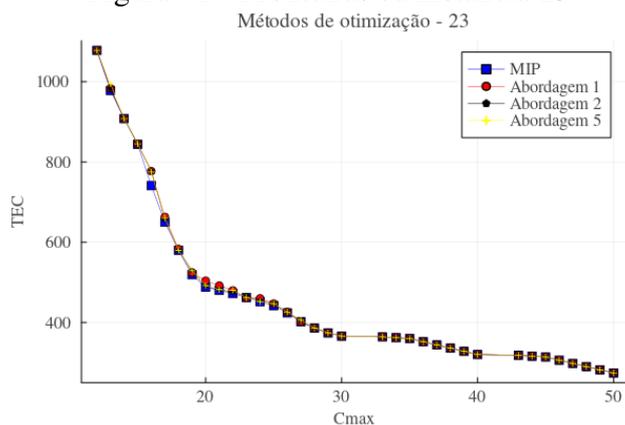
Fonte: Elaborado pela autora (2024)

Figura 40 – Fronteiras da instância 22



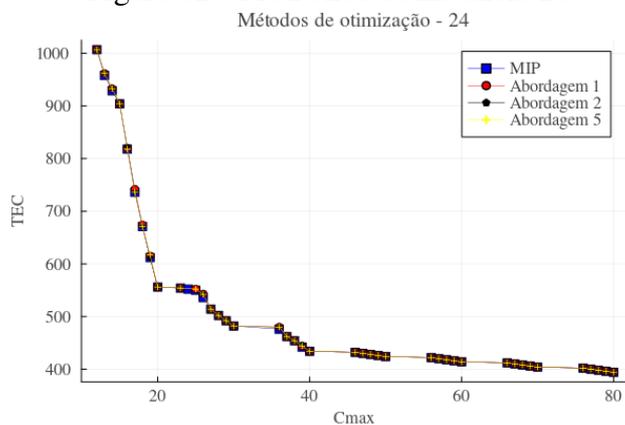
Fonte: Elaborado pela autora (2024)

Figura 41 – Fronteiras da instância 23



Fonte: Elaborado pela autora (2024)

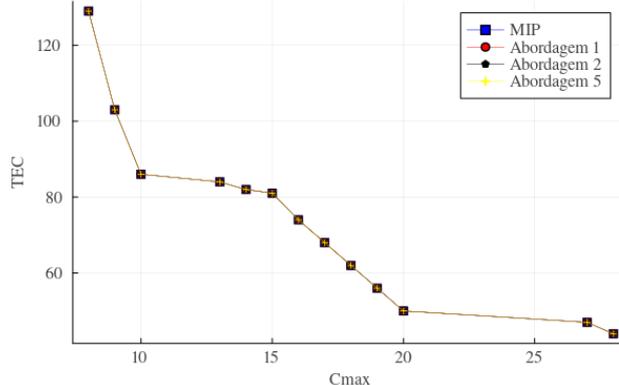
Figura 42 – Fronteiras da instância 24



Fonte: Elaborado pela autora (2024)

Figura 43 – Fronteiras da instância 25

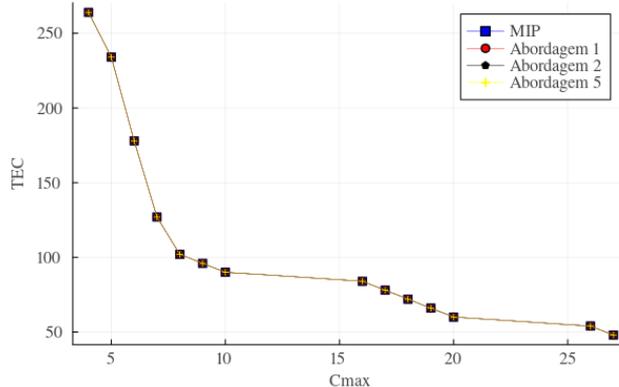
Métodos de otimização - 25



Fonte: Elaborado pela autora (2024)

Figura 44 – Fronteiras da instância 26

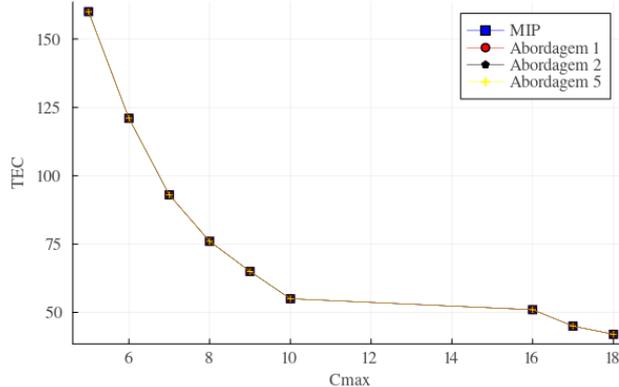
Métodos de otimização - 26



Fonte: Elaborado pela autora (2024)

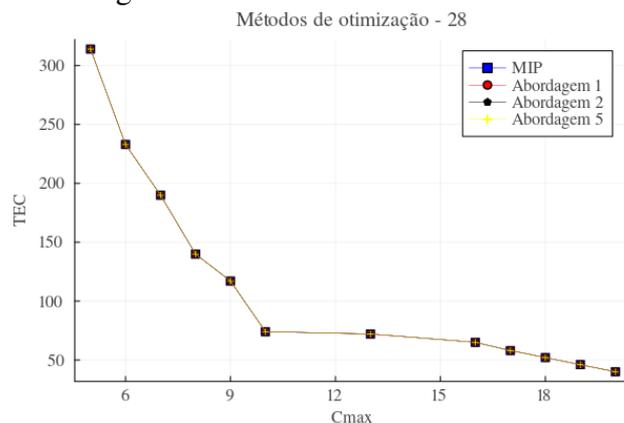
Figura 45 – Fronteiras da instância 27

Métodos de otimização - 27



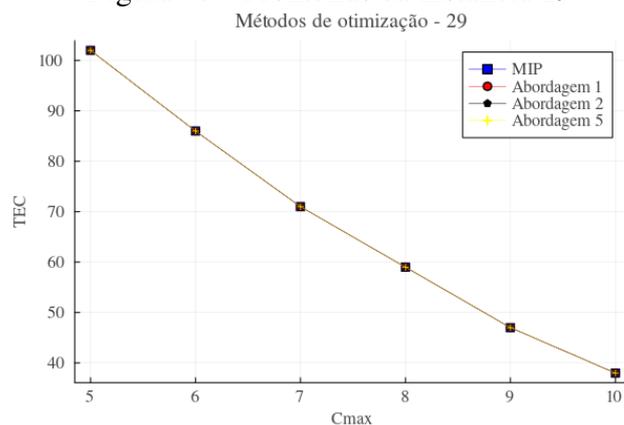
Fonte: Elaborado pela autora (2024)

Figura 46 – Fronteiras da instância 28



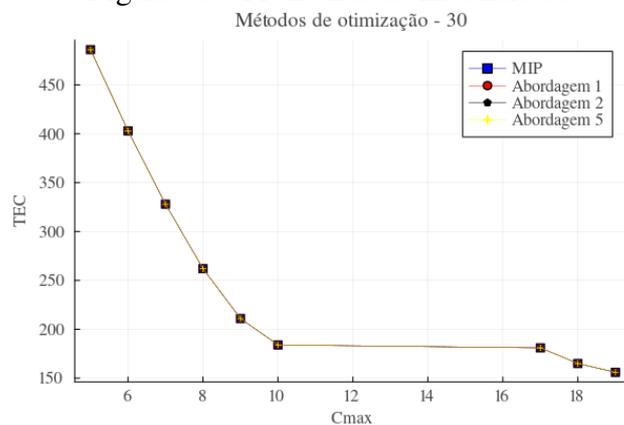
Fonte: Elaborado pela autora (2024)

Figura 47 – Fronteiras da instância 29



Fonte: Elaborado pela autora (2024)

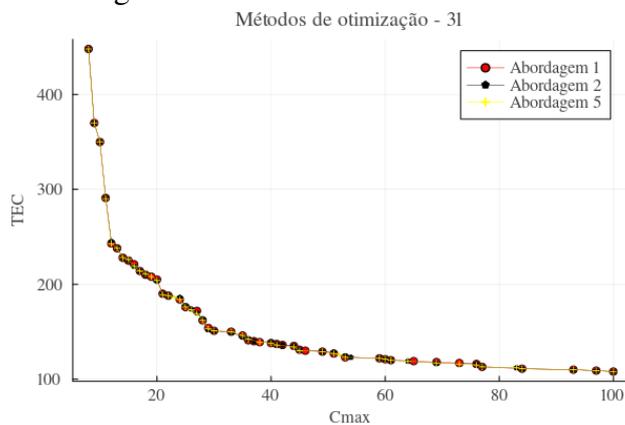
Figura 48 – Fronteiras da instância 30



Fonte: Elaborado pela autora (2024)

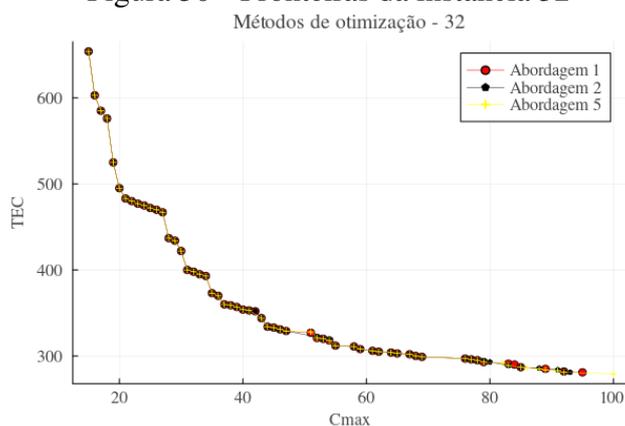
## APÊNDICE B – FRONTEIRAS DE PARETO PARA INSTÂNCIAS GRANDES

Figura 49 – Fronteiras da instância 31



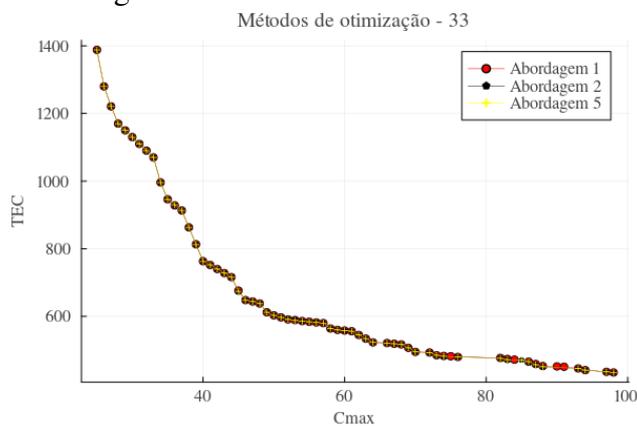
Fonte: Elaborado pela autora (2024)

Figura 50 – Fronteiras da instância 32



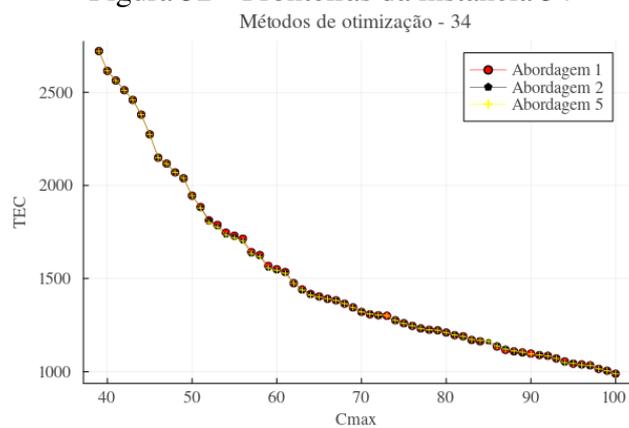
Fonte: Elaborado pela autora (2024)

Figura 51 – Fronteiras da instância 33



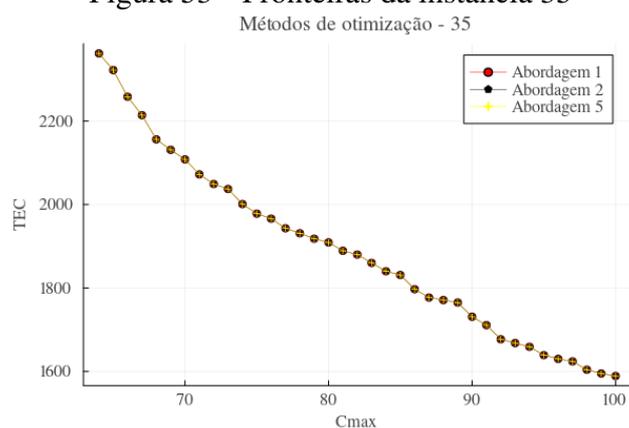
Fonte: Elaborado pela autora (2024)

Figura 52 – Fronteiras da instância 34



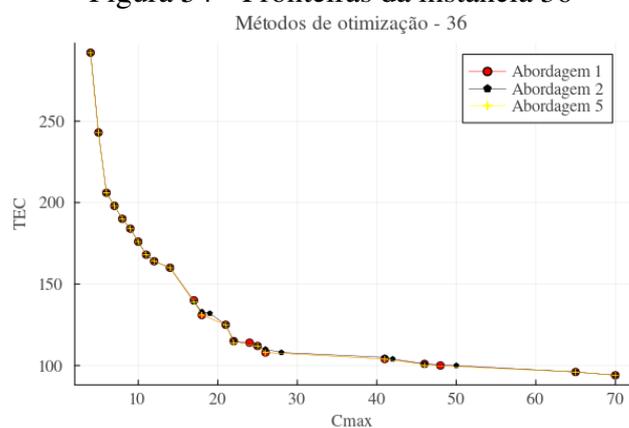
Fonte: Elaborado pela autora (2024)

Figura 53 – Fronteiras da instância 35



Fonte: Elaborado pela autora (2024)

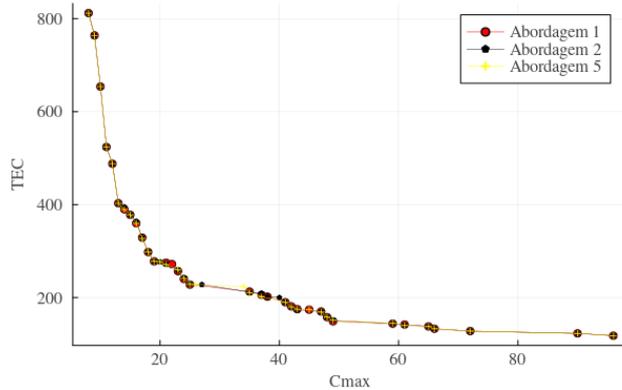
Figura 54 – Fronteiras da instância 36



Fonte: Elaborado pela autora (2024)

Figura 55 – Fronteiras da instância 37

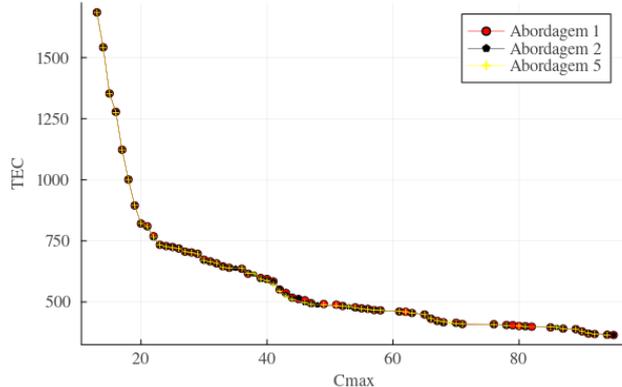
Métodos de otimização - 37



Fonte: Elaborado pela autora (2024)

Figura 56 – Fronteiras da instância 38

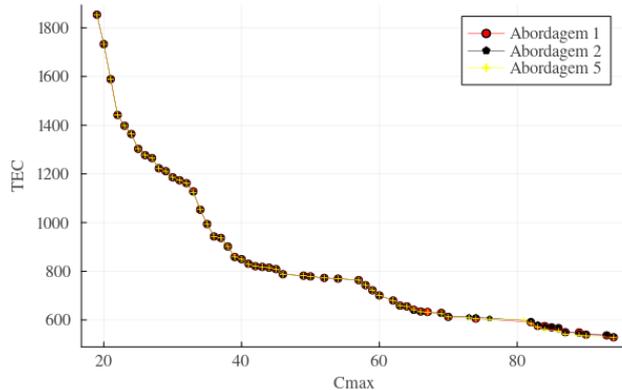
Métodos de otimização - 38



Fonte: Elaborado pela autora (2024)

Figura 57 – Fronteiras da instância 39

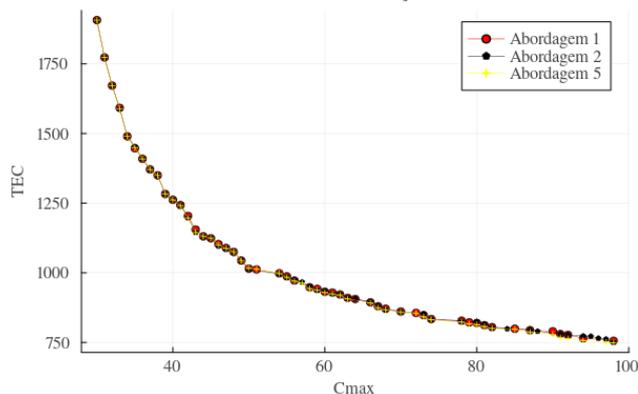
Métodos de otimização - 39



Fonte: Elaborado pela autora (2024)

Figura 58 – Fronteiras da instância 40

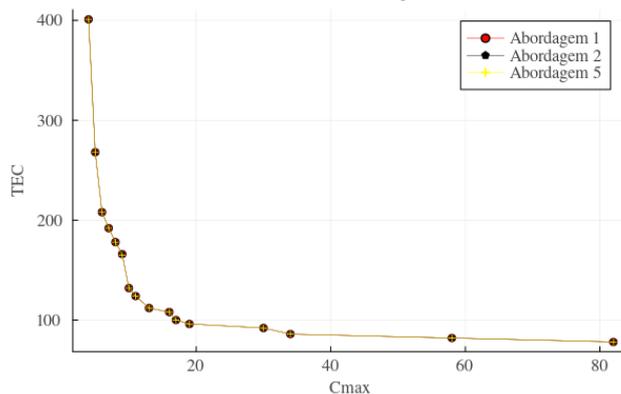
Métodos de otimização - 40



Fonte: Elaborado pela autora (2024)

Figura 59 – Fronteiras da instância 41

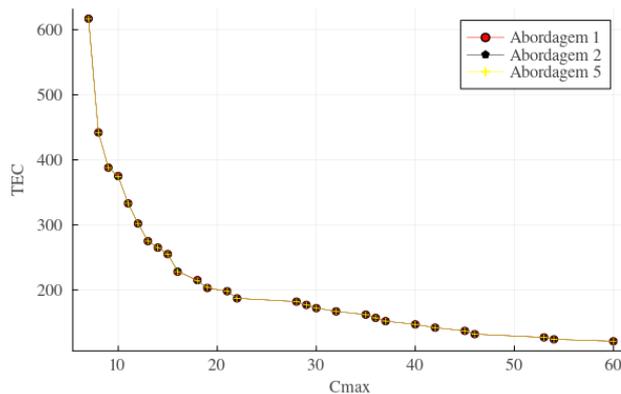
Métodos de otimização - 41



Fonte: Elaborado pela autora (2024)

Figura 60 – Fronteiras da instância 42

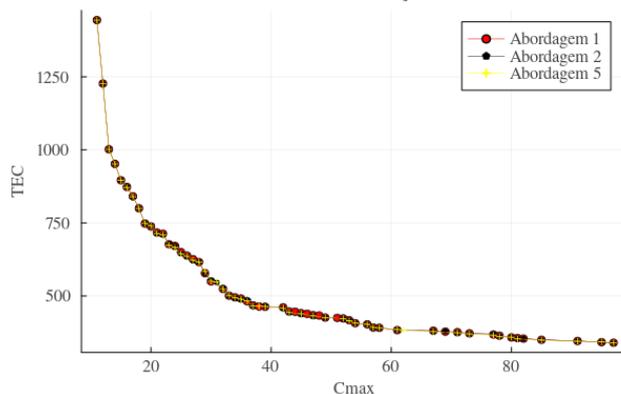
Métodos de otimização - 42



Fonte: Elaborado pela autora (2024)

Figura 61 – Fronteiras da instância 43

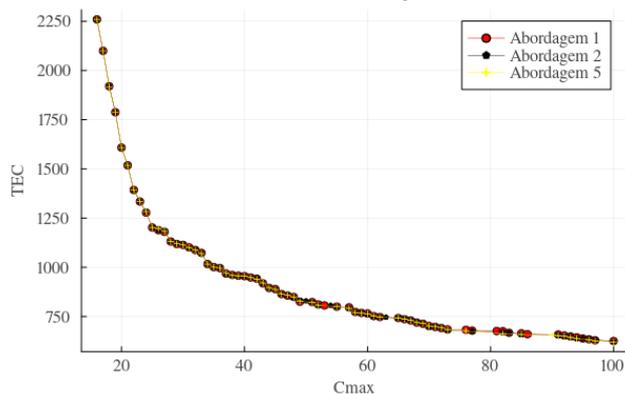
Métodos de otimização - 43



Fonte: Elaborado pela autora (2024)

Figura 62 – Fronteiras da instância 44

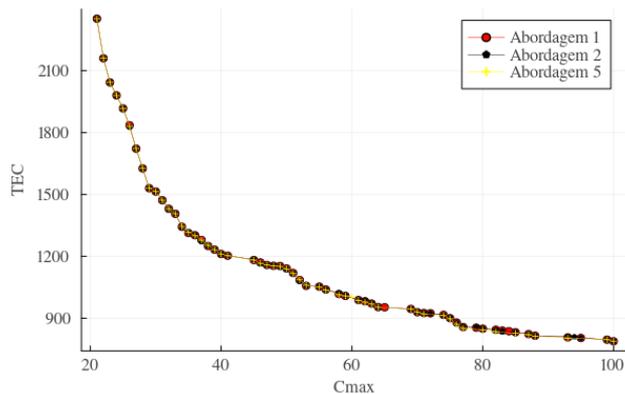
Métodos de otimização - 44



Fonte: Elaborado pela autora (2024)

Figura 63 – Fronteiras da instância 45

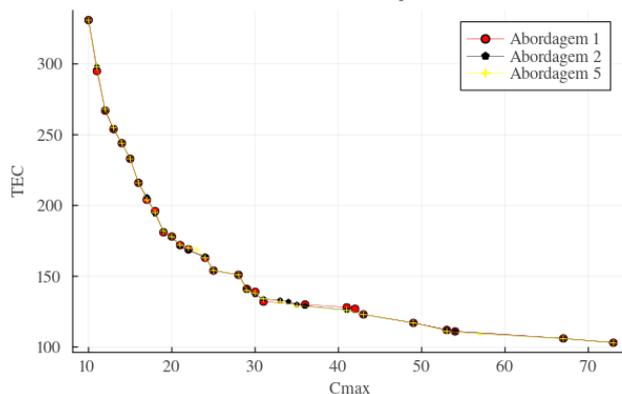
Métodos de otimização - 45



Fonte: Elaborado pela autora (2024)

Figura 64 – Fronteiras da instância 46

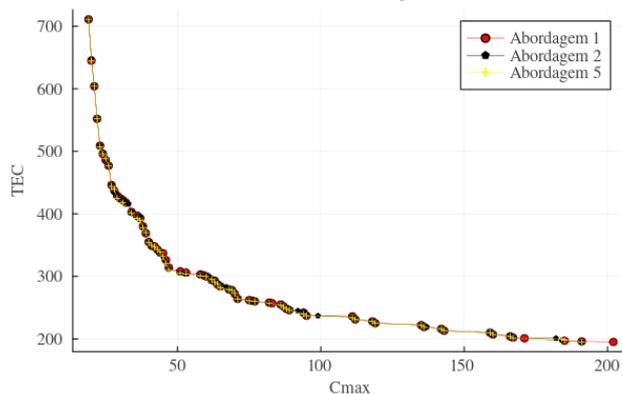
Métodos de otimização - 46



Fonte: Elaborado pela autora (2024)

Figura 65 – Fronteiras da instância 47

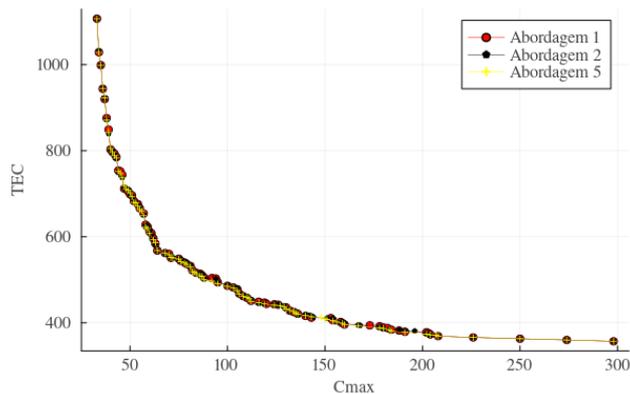
Métodos de otimização - 47



Fonte: Elaborado pela autora (2024)

Figura 66 – Fronteiras da instância 48

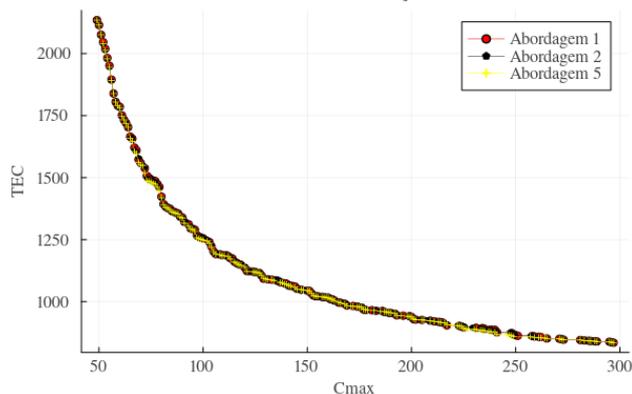
Métodos de otimização - 48



Fonte: Elaborado pela autora (2024)

Figura 67 – Fronteiras da instância 49

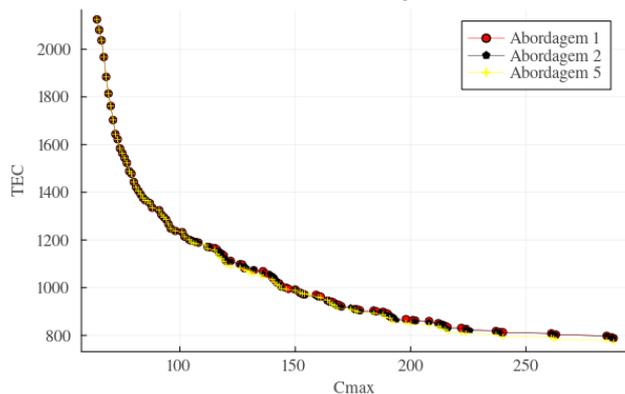
Métodos de otimização - 49



Fonte: Elaborado pela autora (2024)

Figura 68 – Fronteiras da instância 50

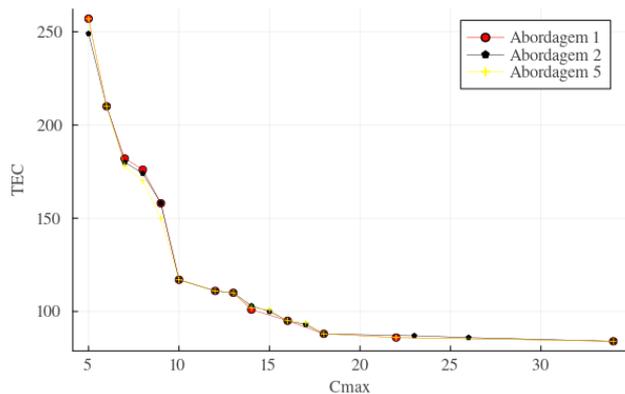
Métodos de otimização - 50



Fonte: Elaborado pela autora (2024)

Figura 69 – Fronteiras da instância 51

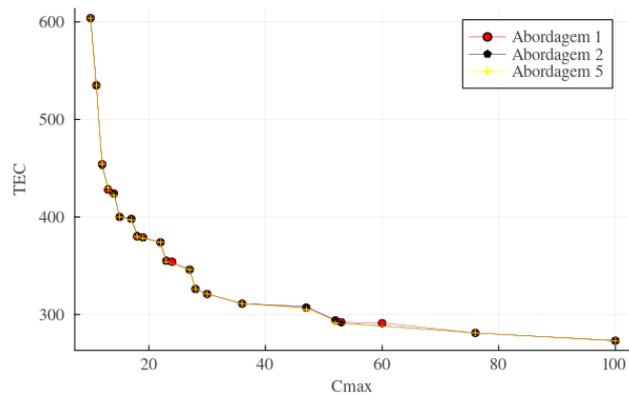
Métodos de otimização - 51



Fonte: Elaborado pela autora (2024)

Figura 70 – Fronteiras da instância 52

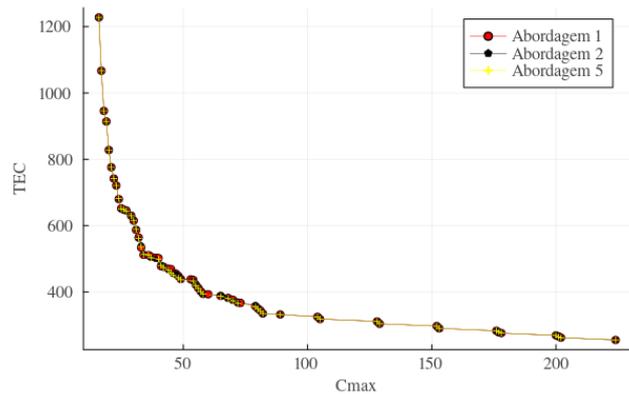
Métodos de otimização - 52



Fonte: Elaborado pela autora (2024)

Figura 71 – Fronteiras da instância 53

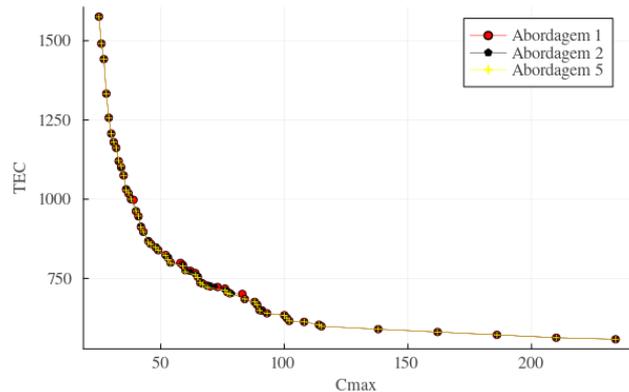
Métodos de otimização - 53



Fonte: Elaborado pela autora (2024)

Figura 72 – Fronteiras da instância 54

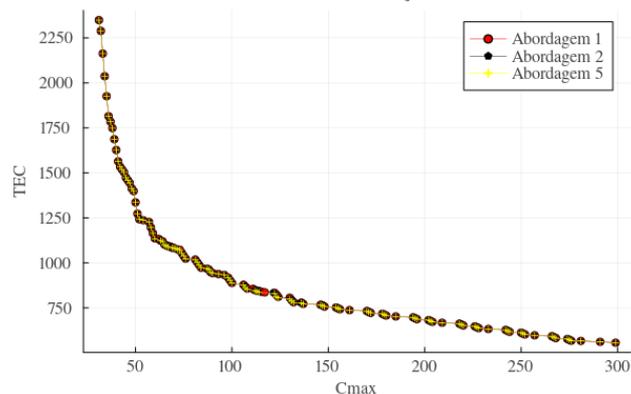
Métodos de otimização - 54



Fonte: Elaborado pela autora (2024)

Figura 73 – Fronteiras da instância 55

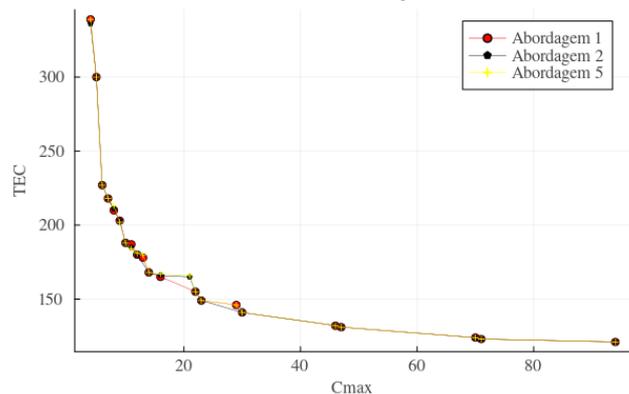
Métodos de otimização - 55



Fonte: Elaborado pela autora (2024)

Figura 74 – Fronteiras da instância 56

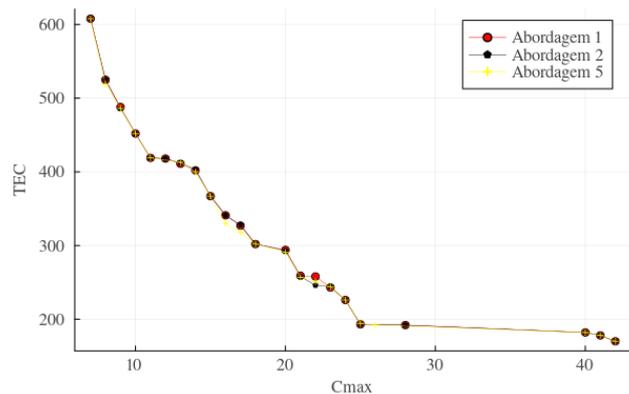
Métodos de otimização - 56



Fonte: Elaborado pela autora (2024)

Figura 75 – Fronteiras da instância 57

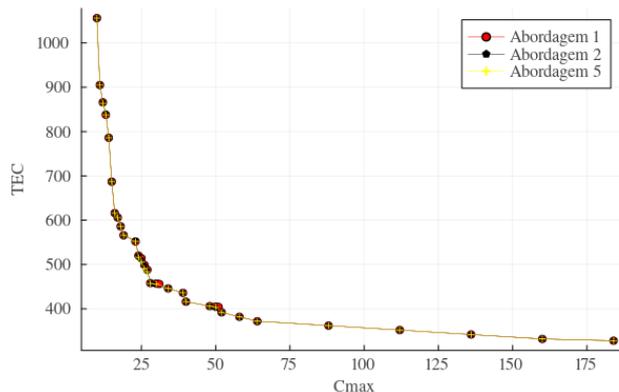
Métodos de otimização - 57



Fonte: Elaborado pela autora (2024)

Figura 76 – Fronteiras da instância 58

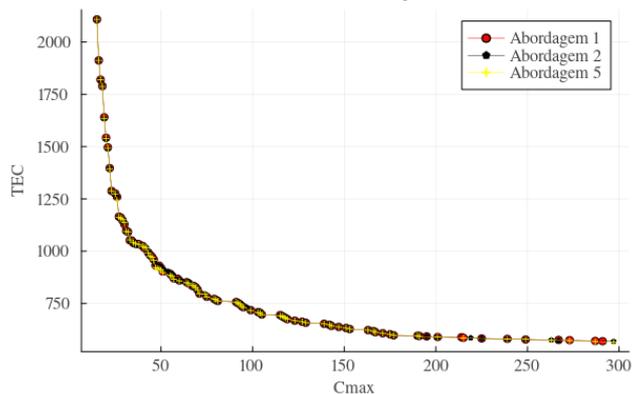
Métodos de otimização - 58



Fonte: Elaborado pela autora (2024)

Figura 77 – Fronteiras da instância 59

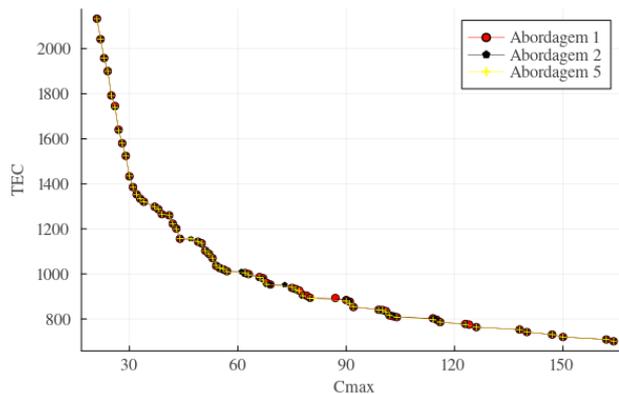
Métodos de otimização - 59



Fonte: Elaborado pela autora (2024)

Figura 78 – Fronteiras da instância 60

Métodos de otimização - 60



Fonte: Elaborado pela autora (2024)