

**UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO
EM ENGENHARIA DE PRODUÇÃO**

Angelita Pezzi Pasqualon Bridi

**ALGORITMOS HEURÍSTICOS PARA SOLUÇÃO DE
PROBLEMAS DE SEQUENCIAMENTO EM MÁQUINAS
PARALELAS NÃO-RELACIONADAS EM MULTI-FÁBRICAS**

Santa Maria, RS
2021

Angelita Pezzi Pasqualon Bridi

**ALGORITMOS HEURÍSTICOS PARA SOLUÇÃO DE
PROBLEMAS DE SEQUENCIAMENTO EM MÁQUINAS
PARALELAS NÃO-RELACIONADAS EM MULTI-FÁBRICAS**

Dissertação apresentada ao Curso de Mestrado do Programa de Pós-Graduação em Engenharia de Produção, Área de Concentração em Métodos Quantitativos para a Tomada de Decisão, da Universidade Federal de Santa Maria (UFSM-RS), como requisito parcial para obtenção do grau de **Mestra em Engenharia de Produção**.

Orientador: Prof. Dr. Felipe Martins Müller

Santa Maria, RS
2021

Ficha catalográfica elaborada através do Programa de Geração Automática da Biblioteca Central da UFSM, com os dados fornecidos pelo(a) autor(a).

Bridi, Angelita Pezzi Pasqualon

Algoritmos Heurísticos para Solução de Problemas de Sequenciamento em Máquinas Paralelas Não-Relacionadas em Multi-Fábricas / Angelita Pezzi Pasqualon Bridi - 2021

139 p.; 30 cm

Orientador: Felipe Martins Müller

Dissertação (mestrado) - Universidade Federal de Santa Maria, Centro de Tecnologia, Programa de Pós-Graduação em Engenharia de Produção, RS, 2021

1. Engenharia de Produção 2. Métodos Quantitativos para a Tomada de Decisão 3. Máquinas Paralelas Não-Relacionadas 4. Multi-Fábricas 5. Heurística de 4 Fases I. Müller, Felipe Martins II. Título.

© 2021

Todos os direitos autorais reservados a Angelita Pezzi Pasqualon Bridi. A reprodução de partes ou do todo deste trabalho só poderá ser feita com autorização por escrito do autor. Endereço: Avenida Roraima, Nº 1000, Bairro Camobi, Santa Maria, RS, Brasil, CEP: 97105-900;

Angelita Pezzi Pasqualon Bridi

ALGORITMOS HEURÍSTICOS PARA SOLUÇÃO DE PROBLEMAS
DE SEQUENCIAMENTO EM MÁQUINAS PARALELAS
NÃO-RELACIONADAS EM MULTI-FÁBRICAS

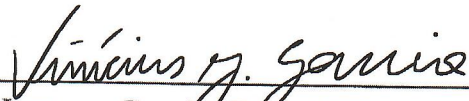
Dissertação apresentada ao Curso de Mestrado do Programa de Pós-Graduação em Engenharia de Produção, Área de Concentração em Métodos Quantitativos para a Tomada de Decisão, da Universidade Federal de Santa Maria (UFSM-RS), como requisito parcial para obtenção do grau de Mestra em Engenharia de Produção.

26 de Outubro de 2021



Documento assinado digitalmente
Felipe Martins Muller
Data: 18/01/2022 10:17:02-0300
Verifique em <https://verificador.itl.br>

Felipe Martins Müller, Dr. (UFSM - Videoconferência)
(Presidente/ Orientador)


Vinicius Jacques Garcia, Dr. (UFSM - Videoconferência)


Hélio Yochihiro Fuchigami, Dr. (UFG - Videoconferência)

Santa Maria, RS
2021

*Ao meu esposo Éder Bridi e meus filhos Rafael e Bernardo pela
paciência, compreensão e amor.*

AGRADECIMENTOS

O término de um mestrado é o resultado de uma trajetória que envolve muita dedicação e obstinação. Nenhum sonho se realiza de maneira fácil e sem esforço. Entretanto, o desenvolvimento deste trabalho não teria sido possível sem a contribuição, estímulo e solicitude de várias pessoas. Por esse motivo, quero agradecer a todos que me auxiliaram nesta caminhada, seja por compartilharem seus saberes e experiências, ou meramente pelo apreço e amizade.

Ao professor Felipe Martins Müller, que acreditou em mim como aluna e tornou-se mais que um orientador: foi um grande amigo e professor. Admirável pelo seu caráter e sua dedicação, és um exemplo de profissional, não esqueço de seus ensinamentos e inestimável confiança.

Aos demais professores do PPGEP, em especial: aos professores que contribuíram significativamente na minha formação e na construção deste trabalho; aos professores que há muito tempo influenciaram positivamente no rumo que resolvi seguir.

Aos professores Hélio Yochihiro Fuchigami e Vinícius Jacques Garcia, pelos importantes apontamentos que engrandeceram esse estudo.

Aos meus pais, Paulo e Terezinha, e irmãos, que mesmo com a distância, sempre se fizeram presentes na minha vida e estarão sempre em meu coração.

Ao meu esposo, Éder Bridi, por ter sido minha base, meu porto seguro, por ter segurado minha mão e ter me feito acreditar que tudo isso seria possível, pela paciência, e principalmente pelo amor nesta caminhada tão árdua e difícil pra mim. Amo você.

Aos meus filhos Rafael e Bernardo, que amo incondicionalmente, e estiveram ao meu lado todos os dias, espero doravante compensá-los das horas de atenção e atenção em que lhes devo.

Aos meus tios Fernando Luiz Bridi e Iara Cristina Bridi agradeço pela ajuda, me auxiliando no meu crescimento pessoal e profissional.

Ao meu grupo de pesquisa, com os quais tive o ensejo de compartilhar momentos de conhecimento mútuo e descontração.

À uma amiga em especial, Greici da Rosa, sua amizade é uma benção que apareceu em meu caminho, e agradeço por todo carinho, apoio e felicidade que me proporcionas desde o começo da nossa amizade, e jamais esquecerei as risadas de bruxa e os shows de horrores compartilhados por mensagens.

À professora Márcia Bitarello, pelos ensinamentos, carinho, e principalmente, por acompanhar minha trajetória acadêmica até os dias de hoje, és muito especial para mim.

Aos amigos que Deus me presenteou nesta caminhada:

Ana Lúcia Londero, Clarice Neumann, Elenice Parizzi, Gabriel Vielmo Gomes, Kaio Vinícius Vilerá, Ronaldo Guisso, Tadeu Vargas, Tailana Militz.

À Universidade Federal de Santa Maria, pela excelência. Também à CAPES e ao CNPq pelo apoio financeiro.

RESUMO

ALGORITMOS HEURÍSTICOS PARA SOLUÇÃO DE PROBLEMAS DE SEQUENCIAMENTO EM MÁQUINAS PARALELAS NÃO-RELACIONADAS EM MULTI-FÁBRICAS

AUTOR: ANGELITA PEZZI PASQUALON BRIDI

ORIENTADOR: FELIPE MARTINS MÜLLER

O problema de sequenciamento de n tarefas independentes em m máquinas paralelas não-relacionadas tem como objetivo de minimizar o tempo total de finalização (*makespan*) das tarefas nas máquinas. Este problema é fundamentado na chamada heurística de quatro fases, estruturada inicialmente para máquinas paralelas idênticas e máquinas paralelas uniformes, cujas fases, por sua vez, são: Fase 1-Alocação Inicial, Fase 2-Fase de Balanceamento, Fase 3-Duplas Trocas e a Fase 4-Trocas Triplas, criada para o problema em questão, de máquinas paralelas não-relacionadas em multi-fábricas. Para o desenvolvimento das 4 Fases, foram propostas 3 heurísticas baseadas em Regras de Despacho para a alocação inicial, denominadas como SPTteste, LPTteste e FCFSteste. As heurísticas em questão foram comparadas com outras duas heurísticas clássicas da literatura, o LPT (*Longest Processing Time*) e o Algoritmo Genético (AG). Com os resultados obtidos, verificou-se que os algoritmos propostos obtiveram melhor desempenho nos experimentos computacionais, discutidos com base no desvio relativo percentual (RPD) em relação a solução inicial e aos limitantes inferiores. Portanto, para ambientes com máquinas paralelas não-relacionadas para multi-fábricas, o método aplicado demonstrou ser vantajoso para programação de produção no setor de manufatura e serviço.

Palavras-chave: Sequenciamento, Máquinas Paralelas, Não-Relacionadas, Multi-Fábricas.

ABSTRACT

APPLICATIONS OF HEURISTICS FOR THE SOLUTION OF SEQUENCING PROBLEMS IN UNRELATED PARALLEL MACHINES IN MULTI-FACTORIES

AUTHOR: ANGELITA PEZZI PASQUALON BRIDI

ADVISOR: FELIPE MARTINS MÜLLER

The sequencing problem of n independent tasks on m unrelated parallel machines aims to minimize the total completion time (makespan) of the tasks on the machines. Based on the four-phase heuristic, and initially structured for identical parallel machines and uniform parallel machines, it has the following phases: Phase 1-Initial Allocation, Phase 2-Balancing Phase, Phase 3-Double Exchanges and Phase 4-Triple Exchanges, created for the problem in question, of unrelated parallel machines in multi-plants. For the development of the 4 Phases, three heuristics based on Dispatch Rules were proposed for the initial allocation. They were named as SPTtest, LPTtest and FCFStest and then compared with two other classic heuristics in the literature, the LPT (Longest Processing Time) and the Genetic Algorithm (GA). With the results obtained, it was verified that the algorithms proposed obtained better performance in the computational experiments, discussed based on the percentage relative deviation (PRD) in relation to the initial solution and the lower bounds. Therefore, for environments with unrelated parallel machines for multi-plants, the method applied proved to be advantageous for production scheduling in the manufacturing and service sector.

Keywords: Scheduling, Parallel Machines, Unrelated, Multi-Factory .

LISTA DE FIGURAS

Figura 2.1	– Primeiro pré-teste com SPTteste.	39
Figura 2.2	– Segundo pré-teste com SPTteste.	40
Figura 2.3	– Terceiro pré-teste com SPTteste.	41
Figura 2.4	– Quarto pré-teste com SPTteste.	42
Figura 2.5	– Quinto pré-teste com SPTteste.	43
Figura 2.6	– Sexto pré-teste com SPTteste.	44
Figura 2.7	– Solução Inicial com critério proposto SPTteste.	45
Figura 2.8	– Primeiro pré-teste com LPTteste.	48
Figura 2.9	– Segundo pré-teste com LPTteste.	49
Figura 2.10	– Terceiro pré-teste com LPTteste.	50
Figura 2.11	– Quarto pré-teste com LPTteste.	51
Figura 2.12	– Quinto pré-teste com LPTteste.	52
Figura 2.13	– Sexto pré-teste com LPTteste.	53
Figura 2.14	– Solução Inicial com critério proposto LPTteste.	54
Figura 2.15	– Primeiro pré-teste com FCFSteste.	57
Figura 2.16	– Segundo pré-teste com FCFSteste.	58
Figura 2.17	– Terceiro pré-teste com FCFSteste.	59
Figura 2.18	– Quarto pré-teste com FCFSteste.	60
Figura 2.19	– Quinto pré-teste com FCFSteste.	61
Figura 2.20	– Solução Inicial com o critério proposto FCFSteste.	62
Figura 3.1	– Representação dos elementos da Fase 2.	64
Figura 3.2	– Exemplo Fase 1 com critério proposto SPTteste.	66
Figura 3.3	– Escolha da primeira tarefa candidata a ser transferida.	67
Figura 3.4	– Solução Final da Fase 2.	68
Figura 4.1	– Solução Inicial da Fase 3 de Duplas Trocas.	72
Figura 4.2	– Máquina mais carregada e menor valor de p_j para sair da máquina. ...	73
Figura 4.3	– Saída de M_1 e entrada das tarefas em M_2 e M_3	74
Figura 4.4	– Candidatas a serem escolhidas para transferência de M_2 para M_1	75
Figura 4.5	– Candidatas a serem escolhidas para transferência de M_3 para M_1	76
Figura 4.6	– Primeira possibilidade de troca entre M_1 e M_3	77
Figura 4.7	– Segunda possibilidade de troca entre M_1 e M_3	78
Figura 4.8	– Terceira possibilidade de troca entre M_1 e M_3	79
Figura 4.9	– Quarta possível candidata para troca entre M_1 e M_3	80
Figura 4.10	– Solução se a escolha fosse a segunda troca entre M_1 e M_3	81
Figura 4.11	– Novos valores de C_i nas três máquinas.	83
Figura 4.12	– Novos valores de C_i nas três máquinas.	85
Figura 4.13	– Comportamento do algoritmo na busca por duplas trocas.	88

Figura 4.14 – Final das buscas por possibilidades de trocas para este caso.	89
Figura 4.15 – Exemplo utilizado para busca de um alvo.	90
Figura 4.16 – Escolha das vizinhanças com um alvo para números grandes de tarefas.	90
Figura 4.17 – Escolha de um alvo.	91
Figura 5.1 – Início da fase de triplas trocas.	94
Figura 5.2 – Identificação da máquina mais carregada e menor valor de p_j	94
Figura 5.3 – Máquina mais carregada e o segundo menor valor de p_j em M_2	97
Figura 5.4 – Valores atualizados de C_i das três máquinas.	102
Figura 6.1 – Limitante inferior para o <i>makespan</i>	104
Figura 6.2 – Desempenho dos cinco algoritmos para 50 tarefas.	109
Figura 6.3 – Desempenho dos cinco algoritmos para 100 tarefas.	113
Figura 6.4 – Desempenho dos cinco algoritmos para 200 tarefas.	115
Figura 6.5 – Gráficos das médias gerais das cinco heurísticas.	117

LISTA DE TABELAS

Tabela 2.1	–	Parâmetros de entrada: tarefas a serem alocadas.	36
Tabela 2.2	–	Relação dos Tempos de processamento das tarefas.	36
Tabela 2.3	–	Tempos de processamento das tarefas nas Máquinas.	36
Tabela 2.4	–	Tempos de processamento das tarefas segundo regra SPT - M1.	37
Tabela 2.5	–	Tempos de processamento das tarefas segundo regra SPT - M2.	37
Tabela 2.6	–	Tempos de processamento das tarefas segundo regra SPT - M3.	37
Tabela 2.7	–	Tempos de processamento das tarefas segundo regra LPT - M1.	46
Tabela 2.8	–	Tempos de processamento das tarefas segundo regra LPT - M2.	46
Tabela 2.9	–	Tempos de processamento das tarefas segundo regra LPT - M3.	47
Tabela 2.10	–	Tempos de processamento das tarefas segundo regra FCFS - M1.	55
Tabela 2.11	–	Tempos de processamento das tarefas segundo regra FCFS - M2.	55
Tabela 2.12	–	Tempos de processamento das tarefas segundo regra FCFS - M3.	55
Tabela 4.1	–	Apresentação das possibilidades de troca.	81
Tabela 4.2	–	Continuidade para encontrar mais possibilidades de troca.	82
Tabela 4.3	–	Exibição das possibilidades de troca entre M_1 e M_2 e entre M_1 e M_3	82
Tabela 4.4	–	Continuidade para encontrar mais possibilidades de troca.	83
Tabela 4.5	–	Valor máximo e valores mínimos para escolha das possíveis candidatas.	84
Tabela 4.6	–	Possíveis candidatas de M_2 para M_1	84
Tabela 4.7	–	Possíveis candidatas de M_3 para M_1	85
Tabela 4.8	–	Valor máximo e valores mínimos para escolha das possíveis candidatas.	86
Tabela 4.9	–	Possíveis candidatas de M_2 para M_1	86
Tabela 4.10	–	Possíveis candidatas de M_3 para M_1	86
Tabela 4.11	–	Apresentação das possibilidades de troca M_1 e M_3	87
Tabela 4.12	–	Vizinhas candidatas para troca.	92
Tabela 5.1	–	Valor máximo e valores mínimos para escolha das possíveis candidatas.	95
Tabela 5.2	–	Possibilidade de tripla troca entre M_1 e M_2	96
Tabela 5.3	–	Continuidade para encontrar mais possibilidades de troca.	97
Tabela 5.4	–	Valor máximo e valores mínimos para escolha das possíveis candidatas.	98
Tabela 5.5	–	Possibilidade de tripla troca entre M_1 e M_2	98
Tabela 5.6	–	Possibilidade de tripla troca entre M_3 e M_2	99

Tabela 5.7 – Apresentação das possibilidades de troca entre M_1 e M_2 e entre M_3 e M_2	99
Tabela 5.8 – Valores de C_i com teste.	100
Tabela 5.9 – Valor máximo e valores mínimos para escolha das possíveis candidatas.	100
Tabela 6.1 – Parâmetros de entrada de dados.	103
Tabela 6.2 – Distribuição dos parâmetros de entrada dos dados.	104
Tabela 6.3 – Conjunto dos valores de LB, do problema 1, com 50 tarefas do SPT-teste.	105
Tabela 6.4 – Valores das soluções propostas obtidas e os resultados apresentado por Behnamian e Ghomi (2013), com 50 tarefas.	108
Tabela 6.5 – Valores dos modelos propostos e os resultados apresentado por Behnamian e Ghomi (2013), com 100 tarefas.	110
Tabela 6.6 – Valores dos modelos propostos e os resultados apresentado por Behnamian e Ghomi (2013), com 200 tarefas.	114
Tabela 6.7 – RPD% total alcançados por cada heurística, para 50, 100 e 200 tarefas.	116
Tabela A.1 – Limitante Inferior (LB) para os 18 problemas avaliados.	127
Tabela A.2 – Conjunto de 18 problemas, com 50 tarefas, para 4Fases-SPTteste.	128
Tabela A.3 – Conjunto de 18 problemas, com 100 tarefas, para 4Fases-SPTteste.	129
Tabela A.4 – Conjunto de 18 problemas, com 200 tarefas, para 4Fases-SPTteste.	130
Tabela A.5 – Conjunto de 18 problemas, com 50 tarefas, para 4Fases-LPTteste.	131
Tabela A.6 – Conjunto de 18 problemas, com 100 tarefas, para 4Fases-LPTteste.	132
Tabela A.7 – Conjunto de 18 problemas, com 200 tarefas, para 4Fases-LPTteste.	133
Tabela A.8 – Conjunto de 18 problemas, com 50 tarefas, para 4Fases-FCFSteste.	134
Tabela A.9 – Conjunto de 18 problemas, com 100 tarefas, para 4Fases-FCFSteste.	135
Tabela A.10– Conjunto de 18 problemas, com 200 tarefas, para 4Fases-FCFSteste.	136
Tabela A.11– Conjunto de 18 problemas com 50 tarefas.	137
Tabela A.12– Conjunto de 18 problemas com 100 tarefas.	138
Tabela A.13– Conjunto de 18 problemas com 200 tarefas.	139

LISTA DE SÍMBOLOS

C_i	tempo de processamento das tarefas alocadas na máquina i
F	número de fábricas
j	número das tarefas
M_f	número de máquinas nas fábricas
m	número de máquinas
Min	tempo de processamento mínimo
n	número de tarefas
p_{ij}	tempo de processamento da tarefa j na máquina i
p_j	tempo de processamento da tarefa J_j
T	tempo em segundos
SM_i	relação dos tempos de processamento em cada máquina i

SUMÁRIO

1 INTRODUÇÃO	21
1.1 CLASSIFICAÇÃO DOS 3-CAMPOS	22
1.1.1 Primeiro campo: α	22
1.1.2 Segundo campo: β	24
1.1.3 Terceiro campo: γ	27
1.2 OBJETIVOS	30
1.2.1 Objetivo Geral	30
1.2.2 Objetivos Específicos	30
1.3 CONSIDERAÇÕES SOBRE TEORIA DA COMPLEXIDADE	30
1.4 FORMULAÇÃO MATEMÁTICA	31
2 REVISÃO BIBLIOGRÁFICA	33
2.1 PROBLEMAS DE MÁQUINAS PARALELAS EM MULTI-FÁBRICAS	33
2.2 HEURÍSTICAS CONSTRUTIVAS BASEADAS EM REGRAS DE PRIORIDADE	34
2.2.1 SPT - (<i>Shortest Processing Time</i>)	35
2.2.1.1 Introdução sobre a regra proposta SPTteste	38
2.2.1.2 FASE 1 - Alocação Inicial - SPTteste	38
2.2.2 LPT - (<i>Longest Processing Time</i>)	45
2.2.2.1 Introdução sobre a regra proposta LPTteste	47
2.2.2.2 FASE 1 - Alocação Inicial - LPTteste	48
2.2.3 FCFS - (<i>First Come, First Served</i>)	54
2.2.3.1 Introdução sobre a regra proposta FCFSteste	56
2.2.3.2 FASE 1 - Alocação Inicial - FCFSteste	56
3 FASE DE BALANCEAMENTO PARA $R/multifact/Cmax$	63
4 FASE DE DUPLAS TROCAS PARA $R/multifact/Cmax$	71
5 FASE DE TRIPLAS TROCAS PARA $R/multifact/Cmax$	93
6 RESULTADOS COMPUTACIONAIS	103
6.1 GERAÇÃO DE DADOS	103
6.1.1 Limite Inferior (Lower Bound)	104
6.1.2 Análise dos Resultados	106
6.1.3 Análise dos Resultados entre as Cinco Heurísticas com 50 tarefas. .	106
6.1.4 Análise dos Resultados entre as Cinco Heurísticas com 100 tarefas.	110
6.1.5 Análise dos Resultados entre as Cinco Heurísticas com 200 tarefas.	114
7 CONCLUSÃO	119
Referências	121
ANEXOS	125

1 INTRODUÇÃO

As constantes mudanças e instabilidades nos ambientes de produção, tem levado as organizações dos ramos de serviço e manufatura à optarem por sistemas de produção em multi-fábricas. Esse tipo de escolha, permite que a competitividade e capacidade de resposta dos mercados globais aumentem. Desse modo, tais empresas necessitam trabalhar de forma organizada para atestar um fluxo confiável de bens, serviços e informações (KARIMI; DAVOUDPOUR, 2017).

Em face do cenário atual, para as empresas tornarem-se competitivas na área dos negócios em rápida mudança, precisaram-se adaptar de um sistema centralizado para um sistema descentralizado em diversos campos de tomadas de decisões. O sequenciamento distribuído é um procedimento que possibilita os tomadores de decisões locais criarem sequenciamentos que considerem objetivos e restrições locais dentro dos limites das finalidades gerais do sistema, a fim de aproveitar a tendência da globalização (BEHNAMIAN; GHOMI, 2013).

Para Kanyalkar e Adil (2005) os sistemas de produção de uma fábrica e de multi-fábricas possuem algumas diferenças essenciais. Quando em uma empresa, os produtos são fabricados em um único local e em seguida entregues ao mercado. Logo, na produção em multi-fábricas, os produtos são desenvolvidos em diversas fábricas e em locais geograficamente diferentes. Isso faz com que algumas fábricas fiquem mais próximas dos clientes e outras mais distantes, fatores que influenciam no tempo e no custo de produção.

No ambiente prático, a pesquisa proposta se fundamenta pelas inúmeras situações desenvolvidas por máquinas em paralelo não-relacionadas, frequentemente provenientes da necessidade de melhoria do maquinário existente ou do crescimento da capacidade posta, dado que máquinas de produções tecnológicas diferentes apresentam velocidades de execuções distintas.

Diante do exposto, essa dissertação trata da aplicação de algoritmos heurísticos para o problema de sequenciamento em máquinas paralelas não-relacionadas em multi-fábricas ($R/multifact/Cmax$). O presente estudo tem como objetivo minimizar o tempo total de execução (*makespan*) das tarefas nas máquinas, permitindo assim, enquadrar o problema aqui abordado na teoria de sequenciamento.

O desenvolvimento deste estudo está organizado da seguinte forma:

- Capítulo 1, apresenta-se a introdução, classificação de 3-Campos, e uma breve explicação sobre a Teoria da Complexidade em problemas de sequenciamento.
- Capítulo 2, é feita uma revisão da literatura, apresentando problemas de máquinas paralelas em multi-fábricas, após, heurísticas construtivas baseadas em regras de prioridade. Descrições sobre a aplicação de algumas regras de prioridade são

demonstradas para uma melhor compreensão, etapa essa denominada como Fase 1-Alocação Inicial.

- Capítulo 3, trata da Fase de Balanceamento, denominada também como Fase 2, vista como um processo de melhoramento, baseado na solução encontrada na Fase 1.
- Capítulo 4, apresenta-se a Fase de Duplas Trocas, conceituada de Fase 3, que auxilia o caminho a ser seguido até atingir a melhor troca possível.
- Capítulo 5, trata da Fase de Triplas Trocas, quarta e última fase de melhoramento para o problema $R|multifact|Cmax$.
- Capítulo 6, são apresentados os resultados computacionais obtidos pelos métodos propostos.

O estudo é finalizado com a apresentação da conclusão e referências bibliográficas utilizadas, que deram embasamento para o desenvolvimento dos métodos aplicados.

1.1 CLASSIFICAÇÃO DOS 3-CAMPOS

Suponha que m máquinas M_i ($i=1, \dots, m$) devem executar n tarefas J_j ($j=1, \dots, n$). Um sequenciamento (*schedule*) é a alocação destas tarefas as máquinas. Um sequenciamento é factível se as condições impostas pela natureza do problema (características próprias das tarefas e dos processadores) são completamente satisfeitas. Um sequenciamento é ótimo se ele minimiza um dado critério de otimalidade.

Para Lawler et al. (1989) o conjunto de características dos processadores, das tarefas e o critério de otimalidade definem um tipo de problema que pode ser representado por uma classificação de três campos $\alpha | \beta | \gamma$. De acordo com Müller (1993) as Classificações de 3-Campos são apresentadas, de maneira mais detalhada a seguir:

1.1.1 Primeiro campo: α

Este campo, descreve o tipo de problema e o número de máquinas nele existentes. Normalmente, ele é apresentado através de um único símbolo.

Se $\alpha_1 \in \{\circ, P, Q, R\}$, cada tarefa J_j consiste de uma operação simples que pode ser realizada em qualquer processador M_i ; o tempo de processamento de J_j em M_i será representado por p_{ij} . Estes quatro valores de α_1 podem ser descritos como:

(a) Máquina única (1) - ambiente de produção que possui apenas uma máquina. Pode ser considerado como um caso particular de todos os outros ambientes mais complexos. Descrito como:

- $\alpha_1 = \circ$: processador único; $p_{1j} = p_j$;

(b) Máquinas paralelas idênticas (P) - ambiente de produção onde existem m máquinas idênticas. Cada tarefa pode ser processada em qualquer uma das máquinas com a mesma velocidade:

- $\alpha_1 = P$: processadores paralelos idênticos; $p_{ij} = p_j$ para todo M_i ;

(c) Máquinas paralelas uniformes (Q) - várias máquinas podem executar as mesmas tarefas com velocidades de processamento diferentes, mas que seguem uma relação conhecida entre si:

- $\alpha_1 = Q$: processadores paralelos uniformes; $p_{ij} = p_j / s_i$ para uma dada velocidade de processamento s_i de M_i ;

(d) Máquinas paralelas não-relacionadas (R) - generalização do ambiente anterior, no qual as velocidades de processamento são diferentes em cada máquina, mas não seguem uma relação conhecida entre elas. Dessa forma, a velocidade de processamento depende da tarefa a ser executada:

- $\alpha_1 = R$: processadores paralelos não-relacionados; o tempo p_{ij} que a tarefa j gasta na máquina i é igual a $p_{ij} = p_j / s(i,j)$, se as velocidades das máquinas são independentes da tarefa.

(e) *Flow shop* (F) - ambiente de produção que possui m máquinas especializadas em série, onde cada uma das tarefas deve ser processada em todas as máquinas, seguindo um mesmo roteiro, isto é, deve ser executada primeiro na máquina 1, depois na máquina 2, e assim sucessivamente. Em geral, depois de concluir a operação em uma máquina, as tarefas entram numa fila antes de iniciarem seu processamento na máquina seguinte:

- Se $\alpha_1 = F$, tem-se um *Flow Shop*, no qual cada J_j consiste numa cadeia (O_{1j}, \dots, O_{mj}) . O_{ij} deve ser processado em M_i durante p_{ij} unidades de tempo.

(f) *Flexible flow shop* (FF)- este ambiente é uma generalização do *flow shop* e do ambiente com máquinas paralelas, em que existem s estágios de processamento em série, com um número de máquinas em paralelo em cada um. Todas as tarefas seguem o mesmo critério e devem passar por uma das máquinas de todos os estágios, primeiramente no estágio 1, a seguir no estágio 2, e assim sucessivamente.

- (g) *Job Shop* (J)- neste ambiente, as operações de cada tarefa são executadas numa sequência específica de máquinas especializadas, havendo uma rota própria através das máquinas para cada tarefa. Desse modo, o *job shop* é caracterizado por permitir diferentes fluxos de tarefas entre as máquinas, e diferentes números de operações por tarefa.
- Se $\alpha_1 = J$, tem-se um *Job Shop*, no qual cada J_j consiste numa cadeia (O_{1j}, \dots, O_{mj}) . O_{ij} deve ser processado em um dado processador u_{ij} durante p_{ij} unidades de tempo, com $u_{ij} \neq u_{i+1j}$, para $i = 1, \dots, m_j - 1$.
- (h) *Flexible job shop* (FJ)- este ambiente de produção é uma extensão do *job shop*, sendo constituído por vários centros de trabalho. Em cada centro de trabalho existem várias máquinas capazes de executar a mesma operação. Dessa forma, há uma rota própria para cada tarefa ao longo dos centros de trabalho, mas as operações são realizadas em apenas uma das máquinas de cada centro.
- (i) *Open shop* (O)- neste ambiente, cada tarefa deve ser processada em várias máquinas, mas não necessariamente em todas. Além disso, não existem roteiros de processamento pré-estabelecidos para as tarefas, isto é, não existe nenhuma sequência obrigatória de processamento das tarefas ao longo das máquinas. Assim:
- Se $\alpha_1 = O$, tem-se um *Open Shop*, no qual J_j consiste de um conjunto de operações (O_{1j}, \dots, O_{mj}) . O_{ij} deve ser processado em M_i durante p_{ij} unidades de tempo, mas a ordem em que estas operações são executadas é irrelevante.

1.1.2 Segundo campo: β

Nestes campos são descritas as características de processamento, e as restrições das tarefas e dos recursos. Este termo pode ser apresentado através de um único símbolo, múltiplos símbolos, e até mesmo sem nenhum símbolo. Os possíveis símbolos de β são:

- (a) Data de liberação (*release date* - r_j) - este termo indica que as tarefas possuem uma data mínima a partir da qual sua execução pode ser iniciada. Assim, nenhuma tarefa pode entrar em processo antes de sua data de liberação:
- $\beta_3 \in \{r_j, \circ\}$
 - $\beta_3 = r_j$: tempos de disponibilidade são especificados e podem diferir para cada tarefa;
 - $\beta_3 = \circ$: todo $r_j = 0$

- (b) Data de entrega de uma tarefa (*due date* - d_j) - é o momento em que uma tarefa deveria ser concluída. Geralmente, existem penalidades quando este prazo não é cumprido.
- (c) Data de limite de entrega (*deadline* - \bar{d}_j) - este campo indica que existe um limite máximo de tempo em que uma tarefa deve ser impreterivelmente concluída.
- (d) Tempos de preparação (*set-up*) dependentes das máquinas e tarefas (s) - nestes problemas, existe um tempo de preparação entre duas tarefas, o qual depende da máquina onde as tarefas serão processadas, e da própria tarefa. Esse tempo inclui o tempo para obtenção das ferramentas, posicionamento dos materiais a serem usados no trabalho, processos de limpeza, preparação e ajuste das ferramentas, inspeção de materiais, etc. Durante esse tempo, as máquinas não podem executar operações específicas de nenhuma tarefa.
- (e) Tempo de preparação (*set-up*) dependente da sequência ($s_{i,j}$) - este símbolo indica que o tempo de preparação das máquinas depende tanto da tarefa a ser processada, quanto daquela que foi processada imediatamente antes numa mesma máquina.
- (f) Tempo de preparação dependente das famílias de produtos (*batch set-up problem*): neste tipo de problema, as tarefas são agrupadas em famílias de produtos antes de serem processadas. A transição de uma família para outra requer um grande tempo de preparação, enquanto que a transição entre tarefas de uma mesma família necessita de um tempo de preparação muito menor.
- (g) Interrupção (*preemptions* - $pmtm$) - neste tipo de problema, existe a possibilidade de uma tarefa ter sua execução interrompida antes de sua conclusão. O tempo já processado não é perdido; e, após o seu retorno, a tarefa deve permanecer na máquina apenas o tempo restante necessário para concluir sua operação. Se existirem máquinas paralelas, a tarefa não é obrigada a terminar sua execução na mesma máquina em que começou. Durante o período de interrupção, pode-se executar a operação de outra tarefa na máquina em que estava sendo processada. Podendo ser representado:
- $\beta_1 = pmtm$: Preempção (divisão da tarefa) é permitida; o processamento de cada operação pode ser interrompido e retomado tempos depois;
 - $\beta_1 = \circ$: Preempção não é permitida.
- (h) Recirculação (*recrc*) - em sistemas com esta característica, uma tarefa pode visitar uma mesma máquina mais de uma vez.
- (i) Restrições de precedência (*prec*) - este parâmetro indica que existe uma ordem de execução entre as tarefas. Dessa forma, uma determinada tarefa só pode iniciar seu processamento após o término de outra específica. Descrito:

- $\beta_2 = \text{prec}$: uma relação de precedência \rightarrow é especificada entre as tarefas. Isto é derivado de um grafo direto acíclico G com um conjunto de vértices $\{1, \dots, n\}$. Se G contém um caminho direto de j para k , escreve-se $J_j \rightarrow J_k$ e é necessário que J_j seja finalizado antes de J_k se iniciar.
- (j) Restrições de dependência entre operações - em alguns ambientes, depois de se executar uma determinada operação de uma dada tarefa, deve-se obrigatoriamente executar a próxima operação dentro de uma janela de tempo. Um caso particular deste tipo de restrição (*no-wait* - *nwt*) ocorre em ambientes onde não existem filas intermediárias entre duas máquinas. Neste caso, o tempo de espera entre as operações deve ser nulo, e o início da primeira operação deve ser postergado o bastante para garantir que a segunda máquina esteja disponível no instante em que a primeira finalize seu trabalho.
- (k) Bloqueio (*block*) - esta condição pode ocorrer em ambientes *flow shop* quando existe um local de armazenamento de tamanho limitado entre duas máquinas consecutivas. Dessa forma, quando este local estiver lotado, a máquina imediatamente anterior a ele não pode liberar a tarefa que acabou de ser executada, o que impede temporariamente que uma nova tarefa seja processada.
- (l) Permutação (*prmu*) - em alguns ambientes *flow shop*, as tarefas devem ser executadas obrigatoriamente de acordo com a regra FIFO (*First In First Out*). Isto implica que a ordem de execução das tarefas na primeira máquina é mantida a mesma durante todo o sistema.
- (m) Tempo de comunicação/transporte entre máquinas - neste tipo de problema, uma tarefa deve aguardar um tempo mínimo entre duas operações consecutivas, mesmo que a máquina da frente esteja disponível no instante de término da operação da máquina anterior.
- (n) Restrições de escolha de máquinas - em alguns problemas que possuem máquinas paralelas, algumas tarefas não podem ser executadas em todas as máquinas de um estágio.
- (o) Restrição do número de tarefas (*nbr*) - neste ambiente, existe um número máximo de tarefas que podem ser processadas, dentre todas as tarefas disponíveis.

A seguir será descrito o sistema de multi-fábricas abordado neste estudo e como será tratado:

- Considere um conjunto $J = \{J_1, J_2, \dots, J_n\}$ de n tarefas que devem ser indicadas a um conjunto $I = \{M_1, M_2, \dots, M_m\}$ de m máquinas, distribuídas geograficamente.

Considere também $\{p_{ij}\}$ uma matriz $m \times n$ de tempos de processamento, onde $\{p_{ij}\}$ é o tempo que a i -ésima máquina leva para realizar a j -ésima tarefa e, $X = X_{ij}$, um conjunto de variáveis de decisão, onde $X_{ij} = 1$ caso a j -ésima tarefa seja designada para a máquina i , e $X_{ij} = 0$ caso contrário. Este problema pode ser visto como $R||Cmax$, classificado por Gary e Johnson (1979) como sendo um problema NP-Difícil, devido as velocidades de processamento serem diferentes em cada máquina, mas não seguem uma relação entre elas. Desse modo, não se conhece métodos exatos na literatura que resolvam o problema em um tempo computacional aceitável, a não ser para instâncias muito pequenas e assim, faz-se necessário escolher algoritmos heurísticos para resolver o problema proposto.

- Todas as tarefas estão disponíveis para todas as fábricas.
- As máquinas são idênticas em cada fábrica, e dentro de cada fábrica as máquinas que são paralelas, estas são idênticas, porém, as fábricas não possuem nenhuma relação, e utilizam tarefas com tempos de processamento diferentes para cada máquina e sua velocidade depende das tarefas executadas.
- Distribuí-se as tarefas em todas as fábricas, considerando um parque de máquinas total, replicando o conjunto de tarefas em cada fábrica.
- O objetivo é minimizar o tempo de finalização máximo das tarefas, de forma não preemptiva e o problema será tratado como $R|multifact|Cmax$.

1.1.3 Terceiro campo: γ

Este campo refere-se à função objetivo e define os critérios de otimização do problema. Antes de apresentar os possíveis símbolos deste campo, convém apresentar algumas definições:

- (a) Tempo de término de uma tarefa (*completion time* - C_j) - corresponde ao instante de término de processamento da tarefa j .
 - o tempo de finalização: C_j ;
- (b) Tempo de fluxo de uma tarefa (*flowtime* - F_j) - é a soma dos tempos de espera e de processamento de uma tarefa.
- (c) Desvio da data de entrega (*lateness*) de uma tarefa (L_j) - é a diferença entre a data de término de uma tarefa e a sua data de entrega, podendo assumir um positivo quando a tarefa estiver atrasada, ou negativo, quando está adiantada.

- a demora: $L_j = C_j - d_j$, (onde d_j é o tempo limite de finalização de uma tarefa J_j);
- (d) *Tardiness* (T_j) - é semelhante ao *lateness*, mas não considera as tarefas que estiverem adiantadas. Pode ser interpretado como o tempo de atraso das tarefas.
- o atraso: $T_j = \max \{0, C_j - d_j\}$;
- (e) *Earliness* (E_j) - é semelhante ao *lateness*, mas não considera as tarefas que estiverem atrasadas. Pode ser interpretado como o tempo de antecipação das tarefas.
- a antecipação: $E_j = \max \{-L_j, 0\}$.

Em geral, o termo γ está relacionado com os prazos de entrega das tarefas ou com os tempos de término de processamento.

Os critérios de otimização baseados no tempo de conclusão das tarefas conduzem a um menor tempo médio de permanência destas no sistema, o que resulta num menor tempo médio de resposta aos clientes, e menor custo médio de estoques em processo. Os principais critérios de otimização deste tipo são:

- (a) *Makespan* (C_{max}) - é o instante de término da última tarefa a deixar o sistema. A sua minimização normalmente conduz a níveis elevados de utilização das máquinas.
- (b) Tempo total de término (*total completion time*) - é a soma dos instantes de conclusão de cada tarefa. Este critério vem ganhando cada vez mais importância nas indústrias com a busca por uma produção mais enxuta, uma vez que, por meio dele, consegue-se uma redução do estoque em processo (*work in process* - WIP), e uma utilização mais estável dos equipamentos.
- (c) Tempo total de término ponderado (*total weighted completion time*) - é a soma ponderada dos instantes de término de cada tarefa, de acordo com um peso atribuído a cada uma delas.

Os principais critérios de otimização baseados no tempo de fluxo são:

- (d) Tempo de fluxo total (*total flowtime*)
- (e) Soma ponderada dos tempos de fluxo (*total weighted flowtime*)

Os critérios de otimização baseados nas datas de entrega das tarefas são de grande importância em ambientes de produção *make-to-order*. Sua minimização resulta num melhor serviço prestado aos clientes. Dessa forma, estes critérios geralmente têm um forte impacto na preferência dos clientes, e podem constituir uma vantagem competitiva. Os principais critérios de otimização deste tipo são:

- (f) Desvio máximo com relação à data de entrega (*maximum lateness* - L_{max}) - mede a maior violação de tempo de entrega entre todas as tarefas.
- (g) Soma dos atrasos (*total tardiness*).
- (h) Soma ponderada dos atrasos (*total weighted tardiness*) - é definido como a soma ponderada da função *tardiness* de cada tarefa.
- (i) Número total de tarefas em atraso (*number of tardy jobs*).
- (j) Soma dos atrasos e antecipações (*total earliness-tardiness penalty*).
- (k) Soma ponderada dos atrasos e antecipações (*total weighted earliness-tardiness penalty*).
- Além disso, há outros objetivos não tão frequentes:
- (l) Minimização do tempo de preparação das máquinas.
- (m) Minimização do custo de preparação das máquinas.

O critério de otimalidade geralmente escolhido envolve a minimização de $f_{max} \in \{C_{max}, L_{max}\}$, conforme 1.1.

$$f_{max} = \max_{1 \leq j \leq n} f_j(C_j) \quad (1.1)$$

onde: $f_j(C_j) = C_j, L_j$, respectivamente.

Ou de: $\sum f_j \in \{\sum C_j, \sum T_j, \sum U_j, \sum w_j C_j, \sum w_j T_j, \sum w_j U_j\}$ conforme 1.2.

$$\sum f_j = \sum_{j=1}^n f_j(C_j) \quad (1.2)$$

Onde: $f_j(C_j) = C_j, T_j, U_j, w_j C_j, w_j T_j, w_j U_j$ respectivamente e w_j é um peso que indica a importância relativa da tarefa J_j e $f_j(t)$ é uma função real não decrescente que mede o custo que ocorre se J_j é finalizado no tempo t .

Deve-se notar que $\sum w_j C_j$ e $\sum w_j L_j$ diferem por uma constante $\sum w_j d_j$ e, portanto, são equivalentes. Além disso, qualquer sequenciamento minimizando L_{max} também minimiza T_{max} e U_{max} , mas não vice-versa.

Conforme a classificação apresentada, o problema a ser abordado neste estudo é do tipo $R|multifact|C_{max}$. O problema $R|multifact|C_{max}$ é definido como alocar n tarefas independentes a m máquinas não-relacionadas (*unrelated*), sendo que cada tarefa tem um tempo de processamento diferente para cada máquina, e tem como objetivo minimizar o tempo máximo de processamento da máquina mais carregada (*makespan*) (BAKER, 1974), além disso, considera-se que todas as tarefas estão disponíveis para iniciar sua execução ao mesmo tempo, e uma máquina com mais de uma tarefa alocada a ela, deve

executá-las uma após a outra, em alguma sequência, sem interrupção, ou seja, a tarefa que iniciar sua execução em uma máquina deve permanecer nela até o seu final (MÜLLER; DIAS, 2002).

1.2 OBJETIVOS

1.2.1 Objetivo Geral

Desenvolver uma metodologia de sequenciamento para aplicação em máquinas paralelas não-relacionadas através de métodos heurísticos afim de encontrar resultados que podem ser compreendidos como ótimos em reduzidos tempos de processamento e por conseguinte, compará-los com outras metodologias encontradas na literatura.

1.2.2 Objetivos Específicos

- Desenvolver um algoritmo que utilize uma estratégia eficiente de busca aproximada aplicando técnicas de resolução;
- Implementar os algoritmos de otimização para o problema proposto;
- Realizar os testes computacionais e comparar os resultados obtidos com heurísticas já utilizados na literatura.

1.3 CONSIDERAÇÕES SOBRE TEORIA DA COMPLEXIDADE

O conhecimento prático e de simples compreensão é que alguns problemas são mais fáceis de resolver que outros. Em alguns problemas de sequenciamento, apresentam-se algoritmos que são capazes de resolver problemas com milhares de tarefas, em compensação para outros problemas, os melhores algoritmos existentes encontram dificuldades para resolver situações com somente algumas poucas tarefas (MÜLLER, 1993).

A teoria da complexidade fornece um conjunto de análises matemáticas onde os problemas são tidos como de fácil ou difícil resolução. Os problemas de fácil resolução tem complexidade de tempo polinomial (**Classe P**), caso contrário, o algoritmo tem complexidade de tempo não determinístico polinomial (**Classe NP**).

Na teoria da complexidade, nos problemas combinatórios, o número de combinações aumenta exponencialmente com o crescimento do número de itens, tornando infactível a determinação, por meio da busca exaustiva da solução ótima do problema. Desse modo,

objetivando ordenar diferentes problemas sobre os aspectos de complexidade temporal foram definidas várias classes de complexidade que agrupam problemas distintos que podem ser solucionados em tempos similares.

De acordo com Limberger (1997), dado como exemplo, um problema matemático genérico, formado por uma quantidade de n itens e considerando um sistema computacional que permita um milhão de combinações em um segundo, para a situação de 20 itens ($n = 20$), encontra-se a solução ótima em 1 segundo ($2^n = 20^{20} = 1.048.580$) combinações. Se aplicar para o mesmo problema o dobro de itens, em progressão geométrica, ($n = 40$), será preciso um tempo estimado de duas semanas de processamento, levando em consideração a mesma velocidade da máquina acima, para a determinação da solução ótima.

Desse modo, para problemas considerados P , determinados algoritmos precisam de um tempo polinomial para o alcance da solução, e considera-se que são eficientemente resolvidos. Para problemas de sequenciamento tratados em gestão industrial, por exemplo, não se tem conhecimento de um algoritmo polinomial para o alcance da solução e estes são classificados na classe de problemas NP (MÜLLER, 1993).

As soluções proporcionadas pela Teoria da Complexidade no que se refere os Problemas de Otimização Combinatória mostram a existência de uma classe de problemas NP -*Completo*, em que existirá um algoritmo de complexidade polinomial, capaz de solucionar todos os problemas desta classe ou então nenhum deles poderá ser resolvido em tempo polinomial. O fato de certo problema descrito como NP -*Completo* é aceito como forte indicação contra a existência de algoritmos polinomiais e em efeito como justificava para a aplicação de algoritmos heurísticos, determinados a técnicas de enumeração ou a outras formas de otimização, com o objetivo de conseguir soluções aproximadas de boa qualidade (LIMBERGER, 1997).

Apenas os problemas da classe P são resolvidos com eficiência em tempo polinomial e sabemos que $P \subseteq NP$, pois para um determinado grupo de problemas NP caso solucionado em tempo polinomial também podem ser reduzidos e resolvidos em tempo polinomial (BARBOSA, 2020).

1.4 FORMULAÇÃO MATEMÁTICA

Um aspecto significativo dos métodos exatos é a de encontrar soluções ótimas para problemas combinatórios. Por isso, a aplicação desses métodos faz-se essencial para que se corroborem os métodos heurísticos usados. Dessa forma, é apresentado neste subitem um modelo matemático apresentado por Martello, Soumis e Toth (1997), para o problema $R||Cmax$. O modelo utiliza os seguintes parâmetros:

- n conjunto de tarefas que devem ser processadas;

- m número de máquinas;
- p_{ij} tempo de processamento da tarefa j na máquina i ;
- X_{ij} 1 se a tarefa j for processada na máquina i e 0 caso contrário;
- C_{max} tempo máximo de processamento de conclusão das tarefas nas máquinas;
- j número da tarefa.

A formulação matemática do modelo é demonstrada a seguir: o tempo de finalização

$$\text{Minimizar } Z = C_{max} \quad (1.3)$$

Sujeito a:

$$\sum_{i=1}^m x_{ij} = 1, j = 1, \dots, n \quad (1.4)$$

$$\sum_{j=1}^n p_{ij}x_{ij} \leq z, i = 1, \dots, m \quad (1.5)$$

$$x_{ij} \in \{0, 1\}, i = 1, \dots, m; j = 1, \dots, n \quad (1.6)$$

Onde:

$$x_{ij} = \begin{cases} 1 & \text{se } J_j \text{ é processado por } M_i \\ 0 & \text{caso contrário.} \end{cases}$$

A função objetivo 1.3 minimiza o *makespan*, ou seja, o tempo de finalização máximo das tarefas. A restrição 1.4 garante que cada tarefa j seja executada por apenas uma máquina. As restrições 1.5, impõem que o tamanho de z de cada máquina i não é excedida. E por fim, a restrição 1.6 menciona o domínio das variáveis.

O próximo capítulo trata de uma revisão bibliográfica sobre heurísticas construtivas baseadas em regras de prioridade, e algumas regras clássicas são estudadas para a escolha de um algoritmo heurístico para aplicação e implementação do presente estudo.

2 REVISÃO BIBLIOGRÁFICA

Este capítulo apresenta uma pesquisa bibliográfica realizada acerca do problema de sequenciamento em máquinas paralelas em multi-fábricas, seguida da metodologia aplicada na presente pesquisa sobre heurísticas construtivas baseadas em regras de prioridade.

2.1 PROBLEMAS DE MÁQUINAS PARALELAS EM MULTI-FÁBRICAS

Com a multinacionalização econômica e o crescimento na procura por produtos e serviços, as organizações tornaram-se mais competitivas e com isso passaram a trabalhar com ambientes de produção distribuídas, ou multi-fábricas, em diversas localizações geograficamente. Esse sistema de manufatura distribuída é aplicado em várias indústrias, tais como: automotivo, siderúrgico, alimentício, entre outros (FU et al., 2021).

Este método de manufatura distribuída, ou seja, programação de produção em multi-fábricas, permite uma aproximação das fábricas com os clientes, reduz seus custos, atende com mais agilidade as transformações do mercado, porém, cada fábrica possui algumas restrições particulares, como por exemplo, custos de trabalho, políticas governamentais, tributação, instalações de transportes e avanços de máquina (BEHNAMIAN; GHOMI, 2020).

O número de pesquisadores em busca de soluções para problemas de sequenciamento em multi-fábricas vem aumentando nas últimas décadas. Recentemente, Zhang e Xing (2019) abordaram o problema de multi-fábricas homogêneas e utilizaram como critério o *makespan*. Propuseram duas heurísticas construtivas para gerar uma boa solução inicial para metaheurísticas.

Para Bargaoui, Driss e Ghédira (2016) resolver problemas de programação é uma área significativa do campo de pesquisa operacional. Em sua pesquisa, os autores propuseram uma nova generalização do problema de programação de fluxo de permutação com ambiente multi-fábricas, a fim de minimizar o *makespan*.

Meng, Pan e Wang (2019) trataram um problema em que um conjunto de pedidos precisa ser fabricado em diversas fábricas e cada pedido composto de algumas tarefas definidas necessita ser processada na mesma fábrica. Um modelo matemático foi proposto para minimizar o *makespan*, e em seguida desenvolveram três metaheurísticas, uma colônia de abelhas artificial e um método guloso.

No trabalho de Lei et al. (2020), os autores estudaram máquinas paralelas não-relacionadas com a minimização do *makespan* em uma rede de multi-fábricas heterogêneas. Um novo algoritmo é apresentado, no qual, quatro estruturas de vizinhança, um operador de busca local são introduzidos e um limite inferior é fornecido.

Em Wachtel e Elalouf (2020), propuseram um algoritmo baseado em um esquema de aproximação totalmente polinomial para agendar tarefas entre uma fábrica principal e outro conjunto de multi-fábricas. A tomada de decisão necessita equilibrar a carga da tarefa entre dois conjuntos de fábricas, enquanto considera as propriedades específicas de cada tarefa, como complexidade, data de vencimento, lucro obtido se concluído em tempo.

Behnamian e Ghomi (2020) abordaram o problema de sequenciamento em multi-fábricas com fábricas heterogêneas. As fábricas disponíveis para processar as tarefas possuem velocidades diferentes em que cada fábrica têm máquinas paralelas idênticas. Para minimizar a soma da antecipação e atraso, e do tempo total de conclusão, os autores apresentaram um modelo de programa linear inteiro misto e um algoritmo heurístico.

Os pesquisadores Lohmer e Lasch (2021) fizeram uma revisão abrangente e sistemática da literatura sobre problemas de planejamento e programação da produção em multi-fábricas. Eles analisaram diversas contribuições da literatura de acordo com a configuração da loja, tipo de fábrica, demanda, estrutura de rede, funções objetivo e métodos de solução.

Desse modo, os empenhos tecnológicos são fundamentais para que a indústria progrida em direção a redes multi-fábricas, devido a globalização e à transformação digital (LOHMER; LASCH, 2021).

2.2 HEURÍSTICAS CONSTRUTIVAS BASEADAS EM REGRAS DE PRIORIDADE

Os problemas de sequenciamento podem ser usados a uma infinidade de sistemas. Desde as primeiras publicações nessa área, heurísticas construtivas são utilizadas para encontrar soluções mais rápidas em um tempo finito de execução e são baseadas em regras de prioridade, também conhecidas como regras de despacho, grande parte delas são regras de decisão ou regras de sequenciamento. Estas regras são aplicadas através da escolha de algum critério para determinar as funções das tarefas e parâmetros das máquinas (ALHARKAN, 2005).

Para Baker (1974) e Morton e Pentico (1993), as regras de prioridade foram denominadas das seguintes maneiras:

- Regras Locais: a prioridade de um processamento é definida por informações locais disponíveis, ou seja, determina-se as normas das tarefas que estão na fila da máquina que ocorrerá o sequenciamento. Como exemplo de normas, pode-se citar, o tempo de processamento, data de entrega, entre outros;
- Regras Globais: utilizam todas as informações disponíveis e são aplicadas para despachar as tarefas;

- Regras Estáticas: a prioridade para as regras estáticas não alteram com o tempo, ou melhor, o valor da prioridade é independente do instante atual.
- Regras Dinâmicas: ao contrário das regras estáticas, nas regras dinâmicas, o valor da prioridade muda constantemente com o tempo.
- Regras de Previsão: são usadas para dar prioridade as tarefas de acordo com o que a tarefa vai deparar no futuro e de acordo com a condição na máquina local.

Existem diversas regras de prioridade estudadas, desenvolvidas, e implementadas por pesquisadores, porém, nas próximas seções serão apresentadas especificamente as regras mais clássicas aplicadas e investigadas na literatura.

2.2.1 SPT - (*Shortest Processing Time*)

Na regra de prioridade SPT (*Shortest Processing Time*) as tarefas são ordenadas de acordo com os tempos de processamento crescente e então atribuídas uma a uma a primeira máquina que ficar disponível, até que todas as tarefas sejam alocadas (SANLAVILLE; SCHMIDT, 1998).

Não obstante o pequeno número de referências sobre a aplicação da regra de prioridade SPT (*Shortest Processing Time*) em máquinas paralelas não-relacionadas, algumas delas são discutidas a seguir.

Ramezani e Saidi-Mehrabad (2012) abordaram um problema de sequenciamento de máquinas paralelas não-relacionadas de múltiplos produtos com processos de retrabalho, ou seja, uma série de tarefas consideradas imperfeitas, probabilidade de retrabalho (todas atividades necessárias para sequenciar itens que não foram produzidos de acordo com as normas predefinidas) e tempo de liberação menor que 0. Usaram o *makespan* como função objetivo. Desenvolveram alguns métodos heurísticos com base em regras de prioridade e aplicaram a heurística SPT (*Shortest Processing Time*) juntamente com outras quatro regras para resolver o modelo proposto.

Shao, Shao e Pi (2021) estudaram um problema de máquinas não-relacionadas em várias fábricas, tendo como critério o *makespan*, onde cada fábrica inclui um flow shop flexível idêntico com restrição de não espera. Quatorze regras de prioridade envolvendo classificação simples e método de decomposição foram desenvolvidas para construir uma sequência de critérios de tarefas, entre tais regras, utilizaram o SPT (*Shortest Processing Time*) para classificar as tarefas de acordo com o tempo de processamento total.

Ao abordar as regras de prioridade e buscando na literatura maneiras diferentes de aplicações da regra SPT (*Shortest Processing Time*), apresenta-se agora um exemplo da aplicação da regra SPT em máquinas paralelas não-relacionadas. Considere um

problema com 3 máquinas e 20 tarefas para serem executadas. Os dados serão apresentados na Tabela 2.1, onde j representa o número das tarefas, p_j corresponde os tempos de processamento das tarefas nas máquinas.

Tabela 2.1 – Parâmetros de entrada: tarefas a serem alocadas.

j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
p_j	1	15	12	13	10	19	4	7	11	16	2	17	18	8	1	13	6	3	20	14

Fonte: Autora, 2021.

Na Tabela 2.2, SM_i representa a relação dos tempos de processamento de cada tarefa em cada máquina.

Tabela 2.2 – Relação dos Tempos de processamento das tarefas.

j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
SM_1	0.90	1.10	2.30	1.20	1.00	0.70	1.30	1.40	1.50	1.20	0.50	0.80	2.10	1.90	0.30	0.60	2.60	1.32	1.06	1.69
SM_2	1.20	0.60	0.80	0.90	1.20	3.10	2.70	2.40	4.30	1.60	1.30	2.10	3.10	3.40	1.50	6.80	1.10	4.20	0.70	1.70
SM_3	1.70	1.20	0.70	0.90	2.50	1.00	2.30	4.20	1.90	2.90	2.60	1.10	3.50	2.80	2.70	1.90	0.80	0.50	2.20	3.70

Fonte: Autora, 2021.

Tabela 2.3 – Tempos de processamento das tarefas nas Máquinas.

j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
p_{jM_1}	1.11	13.64	5.22	10.83	10.00	27.14	3.08	5.07	7.33	13.33	4.00	21.25	8.57	4.21	3.33	21.67	2.31	2.27	18.87	8.28
p_{jM_2}	0.83	25.00	15.00	14.44	8.33	6.13	1.48	2.92	2.56	10.00	1.54	8.10	5.81	2.35	0.67	1.91	5.45	0.71	28.57	8.24
p_{jM_3}	0.59	12.50	17.14	14.44	4.00	19.00	1.74	1.67	5.79	5.52	0.77	15.45	5.14	2.86	0.37	6.84	7.50	6.00	9.09	3.78

Fonte: Autora, 2021.

A fase inicial de um método heurístico tem como característica a escolha de uma determinada regra de prioridade, que faz-se necessário no desenvolvimento da maioria dos algoritmos construtivos para os problemas de sequenciamento.

Atribuindo a regra SPT (*Shortest Processing Time*) os novos valores dos tempos de processamento de cada tarefa em cada máquina, que constam nas Tabelas 2.4, 2.5 e 2.6, foram obtidos através de:

$$p_{ij} = \frac{p_j}{s_{(i,j)}} \quad (2.1)$$

Tabela 2.4 – Tempos de processamento das tarefas segundo regra SPT - M1.

j	1	18	17	7	15	11	14	8	3	9	20	13	5	4	10	2	19	12	16	6
p_{jM_1}	1.11	2.27	2.31	3.08	3.33	4.00	4.21	5.07	5.22	7.33	8.28	8.57	10.00	10.83	13.33	13.64	18.87	21.25	24.67	27.14

Fonte: Autora, 2021.

Tabela 2.5 – Tempos de processamento das tarefas segundo regra SPT - M2.

j	15	18	1	7	11	16	14	9	8	17	13	6	12	20	5	10	4	3	2	19
p_{jM_2}	0.67	0.71	0.83	1.48	1.54	1.91	2.35	2.56	2.92	5.45	5.81	6.13	8.10	8.24	8.33	10.00	14.44	15.00	25.00	28.57

Fonte: Autora, 2021.

Tabela 2.6 – Tempos de processamento das tarefas segundo regra SPT - M3.

j	15	1	11	8	7	14	20	5	15	10	9	18	16	17	19	2	4	12	3	6
p_{jM_3}	0.37	0.59	0.77	1.67	1.74	2.86	3.78	4.00	5.14	5.52	5.79	6.00	6.84	7.50	9.09	12.50	14.44	15.45	17.14	19.00

Fonte: Autora, 2021.

Utilizando as informações contidas nas Tabelas 2.4, 2.5 e 2.6 , uma implementação para o problema de sequenciamento em máquinas paralelas não-relacionadas é sugerida no próximo subitem. A implementação proposta começa com uma alocação inicial imposta pelo método SPT e busca um melhoramento nos resultados por meio da aplicação de uma nova forma aqui denominada como SPTteste. Apresenta-se agora o Algoritmo 1 e os passos para implementação da heurística SPTteste, adaptada para este estudo.

Algoritmo 1: Alocação Inicial com pré-teste para SPTteste .

Entrada de Dados;

Solução Inicial - Algoritmo Construtivo

repita

 Posicionar as tarefas (p_j) nas M máquinas

 Aplicar pré-teste nas máquinas:

 Escolher a tarefa (p_j) que proporcione o menor valor de C_i

 Alocar a tarefa escolhida na máquina com menor valor de C_i

 Verificar número da tarefa alocada (j)

 Eliminar do sequenciamento as demais tarefas (p_j) com número (j) igual
 nas demais máquinas

até que todas tarefas estejam alocadas;

retorna SoluçãoInicial

2.2.1.1 Introdução sobre a regra proposta SPTteste

A heurística proposta SPTteste é fundamentada na regra de prioridade SPT (*Shortest Processing Time*) classificada por Baker (1974), Morton e Pentico (1993). O algoritmo SPTteste, adaptado para o problema de sequenciamento em máquinas paralelas não-relacionadas em multi-fábricas ($R/multifact/Cmax$) pode ser resumido em uma primeira fase de alocação inicial, onde antes da alocação de cada tarefa em uma das máquinas, ocorre um pré-teste e a escolha para alocação ocorre através da máquina que possuir o menor C_i , ou seja, a máquina que tiver a menor soma dos tempos de processamento das tarefas na máquina i . Esta fase é tida como a fase construtiva do algoritmo, na qual busca-se uma organização balanceada das tarefas nas máquinas e capaz de cumprir em pouco tempo este processo.

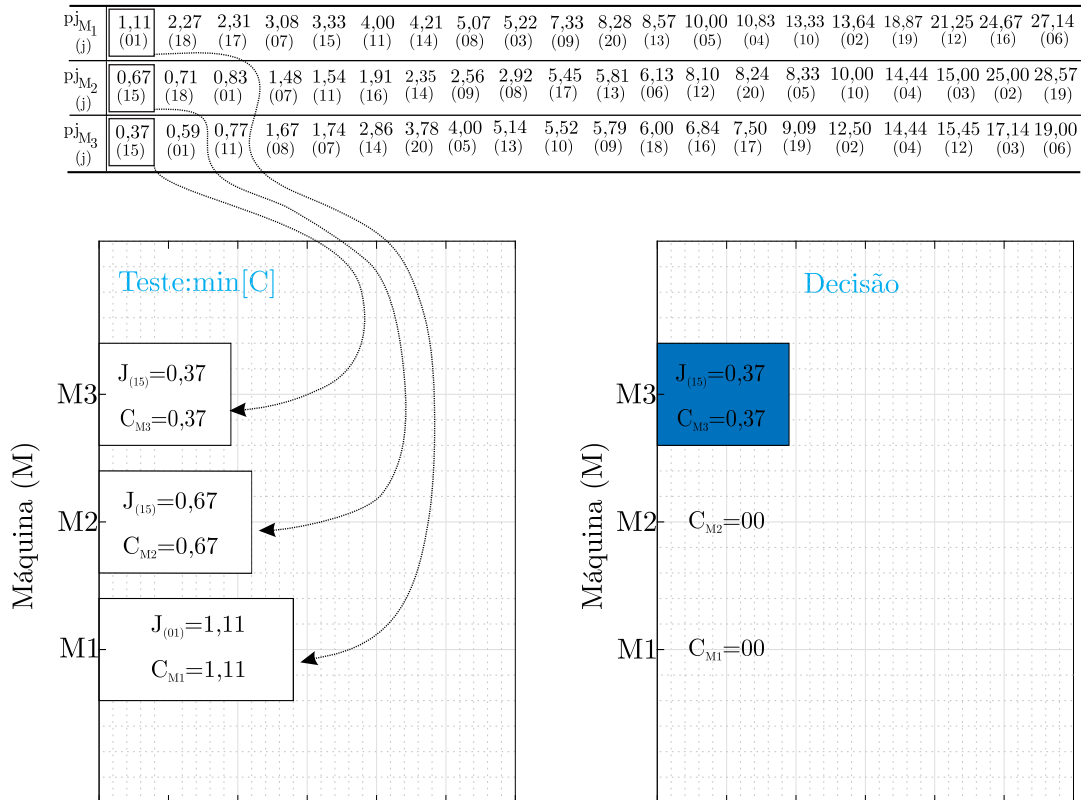
2.2.1.2 FASE 1 - Alocação Inicial - SPTteste

A alocação inicial pode ser considerada a FASE 1 de um procedimento construtivo, nesse caso a condição imposta para o início do algoritmo é a ordenação das tarefas conforme regra de prioridade SPT (*Shortest Processing Time*), as tarefas são ordenadas de forma não decrescente de seu tempo de processamento e a tarefa com menor tempo de processamento é agendada inicialmente.

Aplicando a regra SPTteste em máquinas paralelas não-relacionadas, teremos a sequência inicial $\{J_1, J_{15}, J_{15}\}$ com tempos de processamento $p_{jM1} = 1, 11$, $p_{jM2} = 0, 67$ e $p_{jM3} = 0, 37$. Primeiramente, posiciona-se as tarefas mencionadas nas suas respectivas

máquinas, método esse denominado como pré-teste, ou melhor, SPTteste. Em seguida, certifica-se qual tarefa será alocada e qual máquina que ficará com menor C_i , conforme Figura 2.1.

Figura 2.1 – Primeiro pré-teste com SPTteste.

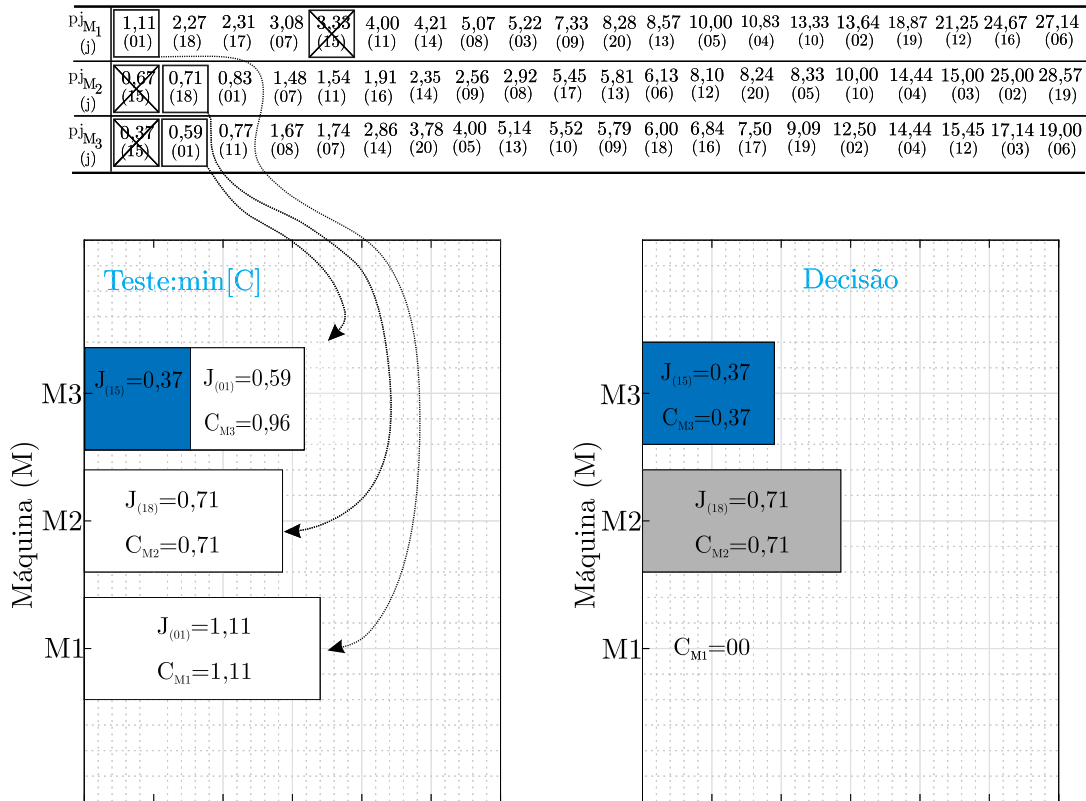


Fonte: Autora, 2021.

Neste primeiro pré-teste a tarefa escolhida $J_{15} \{p_{jM_3} = 0,37\}$, será alocada na máquina M_3 por ser a tarefa de menor valor de tempo de processamento, e seguidamente todas as tarefas J_{15} em M_1 e J_{15} em M_2 serão eliminadas do sequenciamento.

O segundo passo é ordenar para o pré-teste as tarefas $\{J_1, J_{18}, J_1\}$ com tempos de processamento $p_{jM_1} = 1,11$, $p_{jM_2} = 0,71$ e $p_{jM_3} = 0,59$. A Figura 2.2 apresenta o procedimento para o segundo pré-teste, a escolha da tarefa e para qual máquina será alocada.

Figura 2.2 – Segundo pré-teste com SPTteste.



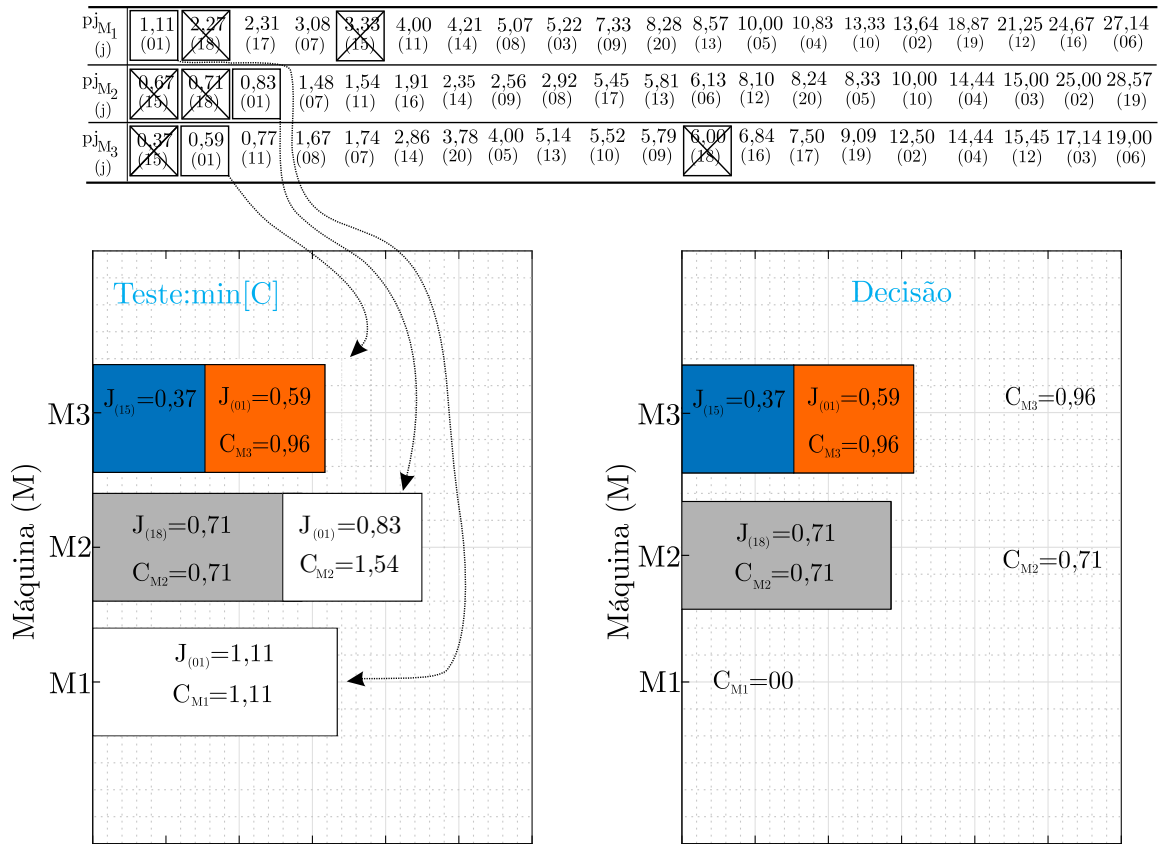
Fonte: Autora, 2021.

No segundo pré-teste a tarefa $J_{18} \{p_{jM_2} = 0,71\}$ é escolhida e alocada na máquina M_2 por ter o menor valor de tempo de processamento de M_1 e M_3 . Todas as tarefas J_{18} em M_1 e J_{18} em M_3 são eliminadas do sequenciamento. Observe que na máquina M_3 houve um pré-teste da tarefa $J_{15} \{p_{jM_3} = 0,37\} + J_1 \{p_{jM_3} = 0,59\} = 0,96$, que resultou em não ser a máquina escolhida para a alocação da tarefa por não ter apresentado o menor valor de C_i . Para cada tarefa alocada são atualizados os valores de C_i , que representa a soma dos tempos de processamentos das tarefas na máquina i .

À vista disso, objetiva-se alcançar uma distribuição razoavelmente balanceada na FASE 1 do algoritmo e assim, uma solução inicial que distribui as tarefas aos três processadores.

No terceiro pré-teste as tarefas $J_1 \{p_{jM_1} = 1,11\}$ e $J_1 \{p_{jM_3} = 0,59\}$ retornam para suas respectivas máquinas e são novamente analisadas. A Figura 2.3 demonstra o pré-teste desta etapa:

Figura 2.3 – Terceiro pré-teste com SPTteste.

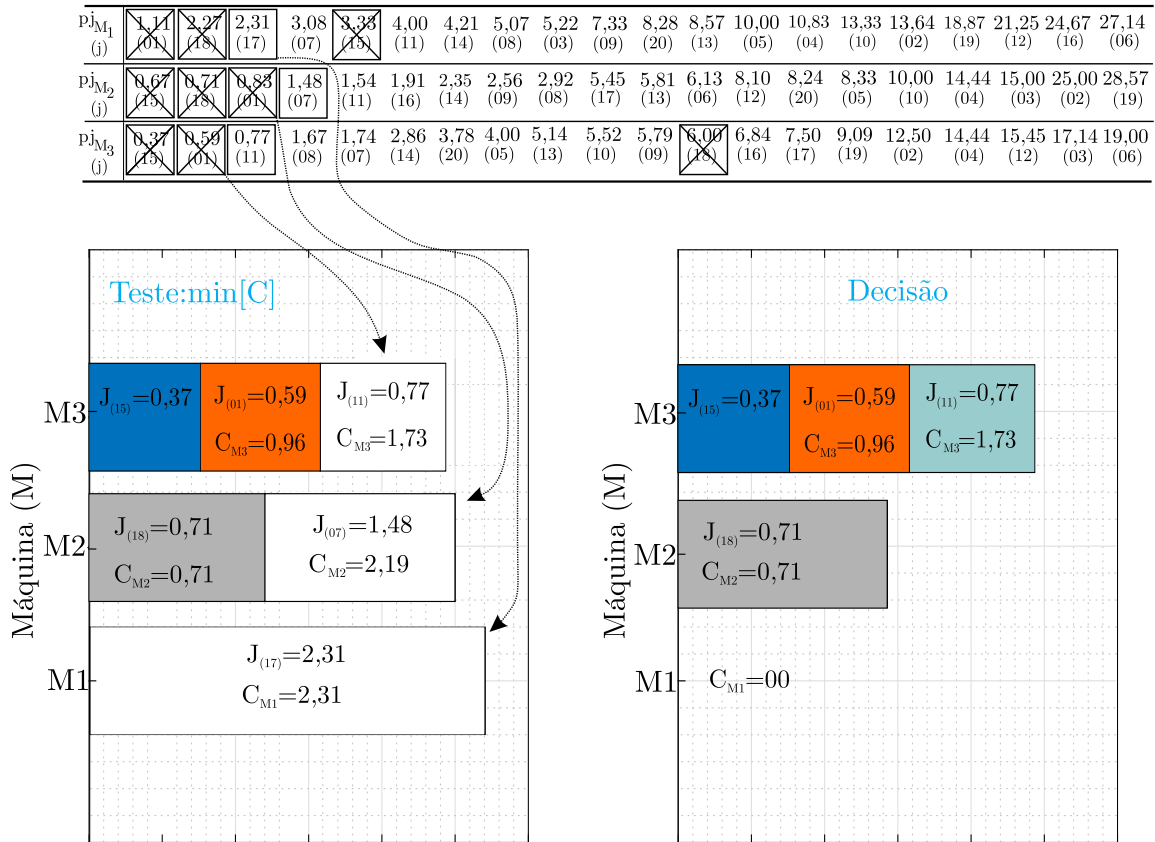


Fonte: Autora, 2021.

Observe que ao ocorrer o pré-teste, a máquina M_3 apresentou o menor valor de C_i das demais máquinas, evidentemente a tarefa $\{p_{jM_3} = 0,59\}$ é alocada em M_3 e as tarefas J_1 das máquinas M_1 e J_1 em M_2 são eliminadas do sequenciamento.

Os pré-testes vão ocorrendo sequencialmente até que todas as tarefas sejam alocadas e um balanceamento entre as máquinas seja possível. O próximo pré-teste acontece do mesmo modo que as etapas anteriores, conforme a Figura 2.4 apresenta:

Figura 2.4 – Quarto pré-teste com SPTteste.

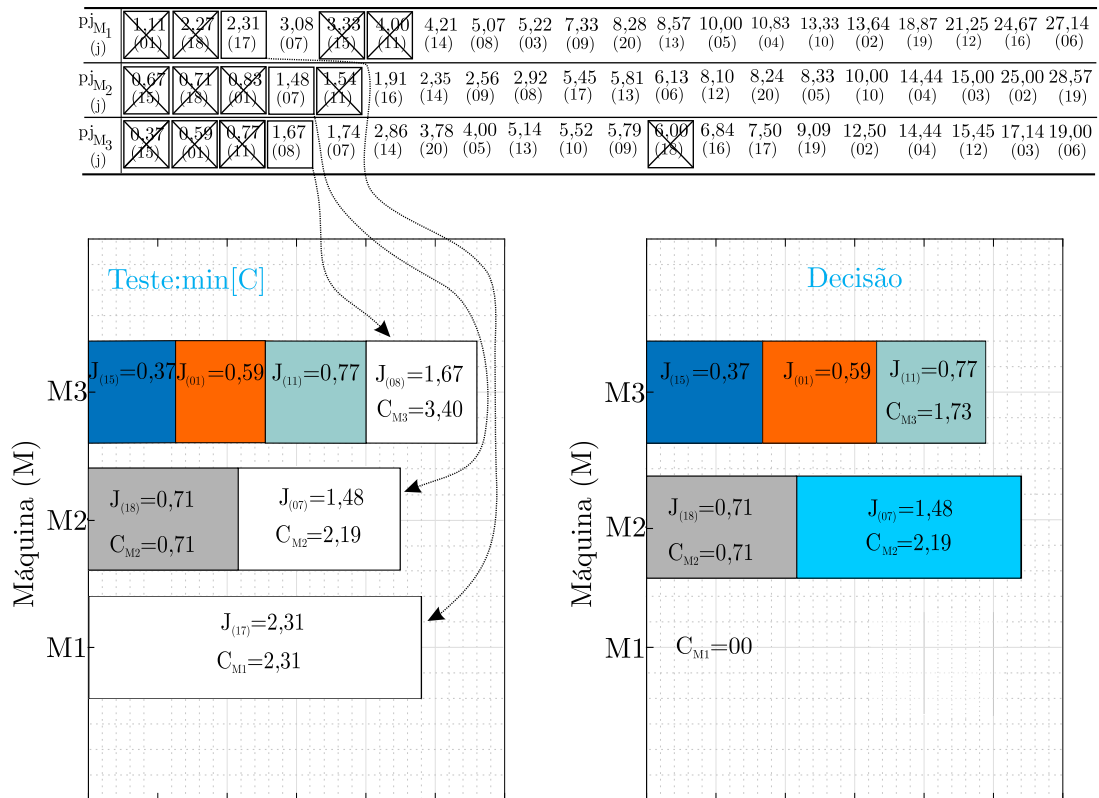


Fonte: Autora, 2021.

Nota-se que neste quarto pré-teste a tarefa $J_{11} \{p_{jM_3} = 0,77\}$ foi escolhida para alocação. Esta escolha obteve-se da seguinte forma: $J_{15} \{p_{jM_3} = 0,37\} + J_{01} \{p_{jM_3} = 0,59\} + J_{11} \{p_{jM_3} = 0,77\} = 1,73$. O resultado é a soma de todas as tarefas alocadas na máquina M_3 e o menor C_i dentre as demais máquinas. Seguidamente, as tarefas J_{11} das máquinas M_2 e J_{11} em M_1 são eliminadas do sequenciamento.

O pré-teste seguinte será das tarefas $J_{17} \{p_{jM_1} = 2,31\}$, $J_{07} \{p_{jM_2} = 1,48\}$ e $J_{08} \{p_{jM_3} = 1,67\}$, conforme Figura 2.5:

Figura 2.5 – Quinto pré-teste com SPTteste.

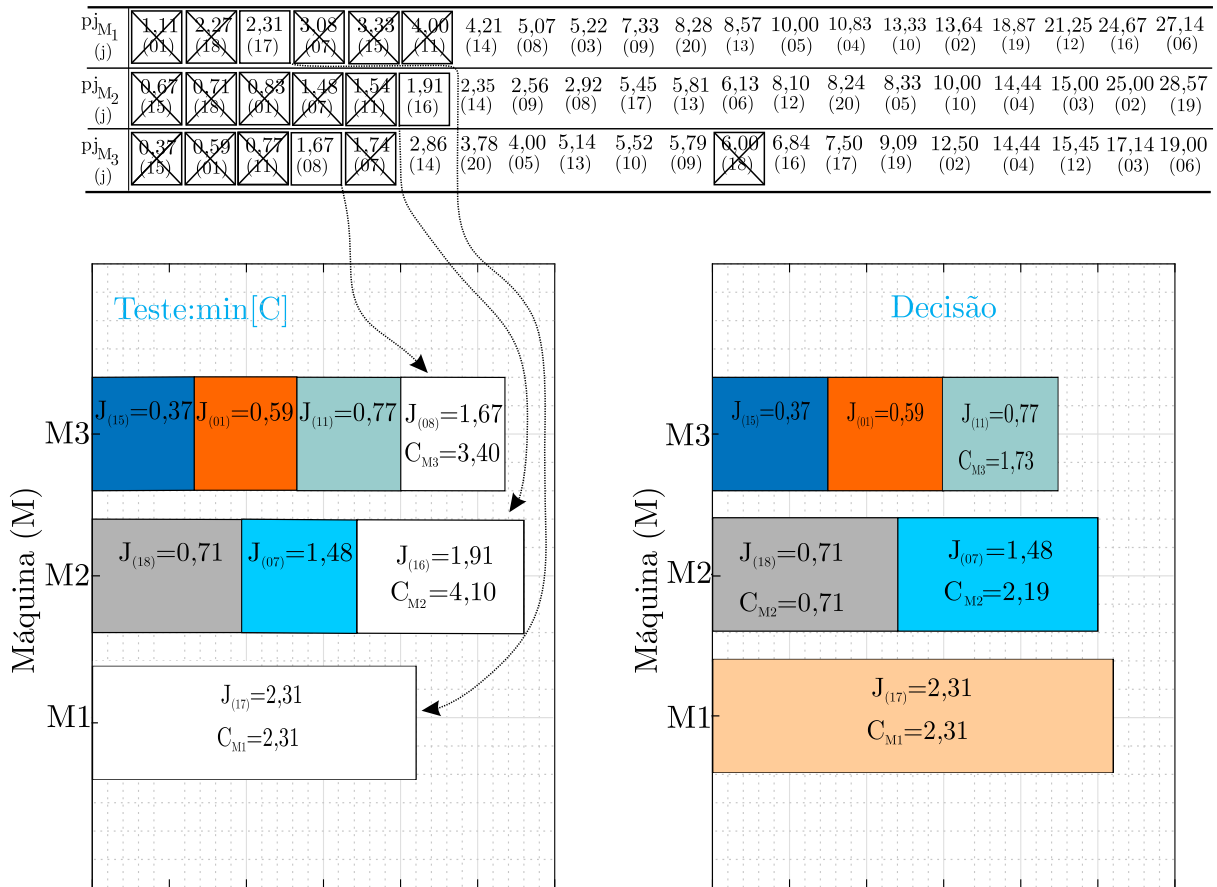


Fonte: Autora, 2021.

A Figura 2.6 apresenta o pré-teste feito nas três máquinas. Desta vez a máquina M_2 apresentou no pré-teste o menor valor de C_i para alocação da tarefa. A tarefa escolhida é $J_{07} \{p_{jM_2} = 1,48\}$. Observa-se que a máquina M_1 , ainda não possui nenhuma tarefa alocada, isso ocorre devido ao pré-teste não identificar até o momento nenhuma tarefa que consiga deixar o valor de C_i menor que M_2 e M_3 . As tarefas J_7 das máquinas M_1 e M_3 são excluídas da fila do sequenciamento.

No sexto pré-teste, de acordo com a Figura 2.6, a tarefa $J_{17} \{p_{jM_1} = 2,31\}$ demonstrou na escolha atual, o menor valor de C_i . Observa-se que os pré-testes realizados até o momento, pelo critério SPTteste, vai deixando o valor de C_i das três máquinas aproximados.

Figura 2.6 – Sexto pré-teste com SPTteste.



Fonte: Autora, 2021.

Dessa maneira, prosseguindo com os pré-testes, obtém-se uma solução inicial para o problema de sequenciamento em máquinas paralelas não-relacionadas, com o valor da solução, de acordo com a Figura 2.7:

Baseado na regra de prioridade mencionada, alguns pesquisadores desenvolveram e implementaram o LPT (*Longest Processing Time*) em máquinas paralelas idênticas e máquinas paralelas uniformes. A seguir, será citado alguns autores que utilizaram este critério em máquinas paralelas não-relacionadas:

Ramezani e Saidi-Mehrabad (2012) utilizaram a regra de prioridade LPT para atribuir as tarefas mais longas nas máquinas, em seguida, a tarefa mais longa, entre aquelas ainda não processadas em uma máquina que possui carga mínima é alocada na máquina. Essa heurística procura neste caso, alocar as tarefas mais curtas no final do sequenciamento, onde podem ser usadas pra balancear as cargas.

Ezugwu, Adeleke e Viriri (2018) aplicaram a regra de tempo de processamento mais longo primeiro, LPT (*Longest Processing Time*) para melhorar ainda mais a velocidade e o desempenho de otimização. Por ser a regra mais apropriada para o problema em questão, é usada para encontrar sequenciamentos de boa qualidade. O ótimo é a melhor solução possível. Além disso, o modelo LPT (*Longest Processing Time*) proposto, garante que as tarefas com tempos de processamento sejam alocadas mais no final do sequenciamento gerado, de modo a manter o equilíbrio de carga em todas as máquinas.

Li et al. (2021b) apresentaram três subproblemas relacionados a multi-fábricas. O primeiro envolve a alocação de uma fábrica para cada tarefa, o segundo subproblema é determinar a sequência de tarefas em cada fábrica e o terceiro é alocar uma máquina para cada tarefa em cada etapa, um modelo matemático é proposto baseado na posição da máquina para otimizar o *makespan* e utilizaram três regras de prioridade, entre elas, o LPT (*Longest Processing Time*) para gerar uma ordenação com base na diminuição do tempo total de conclusão das tarefas em todos os estágios.

Para mostrar as características da regra de prioridade LPT (*Longest Processing Time*), apresenta-se a implementação para máquinas não-relacionadas, em estudo. As Tabelas 2.7, 2.8 e 2.9, demonstram os tempos de processamento em ordem decrescente com base inicial nos dados das Tabelas 2.1, 2.2 e 2.3:

Tabela 2.7 – Tempos de processamento das tarefas segundo regra LPT - M1.

j	6	16	12	19	2	10	4	5	13	20	9	3	8	14	11	15	7	17	18	1
p_{jM1}	27,14	21,67	21,25	18,87	13,63	13,33	10,83	10,00	8,57	8,28	7,33	5,22	5,00	4,21	4,00	3,33	3,08	2,31	2,27	1,11

Tabela 2.8 – Tempos de processamento das tarefas segundo regra LPT - M2.

j	19	2	3	4	10	5	20	12	6	13	17	8	9	14	16	11	7	1	18	15
p_{jM2}	28,57	25,00	15,00	14,44	10,00	8,33	8,24	8,10	6,13	5,81	5,45	2,92	2,56	2,35	1,91	1,54	1,48	0,83	0,71	0,67

Tabela 2.9 – Tempos de processamento das tarefas segundo regra LPT - M3.

j	6	3	12	4	2	19	17	16	18	9	10	13	5	20	14	7	8	11	1	15
p_{jM3}	19,00	17,14	15,45	14,44	12,50	9,09	7,50	6,84	6,00	5,79	5,52	5,14	4,00	3,78	2,86	1,74	1,67	0,77	0,59	0,37

Conforme informações descritas nas Tabelas mencionadas, é possível demonstrar no subitem 2.1.2.1, um método proposto de implementação para problemas em máquinas paralelas não-relacionadas ($R/multifactor/Cmax$) através da regra de prioridade LPT (*Longest Processing Time*), com um processo adaptado e denominado como LPTteste. Apresenta-se agora o Algoritmo 2 e os passos para implementação da heurística LPTteste.

Algoritmo 2: Alocação Inicial com pré-teste para LPTteste.

Entrada de Dados;

Solução Inicial - Algoritmo Construtivo

repita

 Posicionar as tarefas (p_j) nas M máquinas

 Aplicar pré-teste nas máquinas:

 Escolher a tarefa (p_j) que proporcione o menor valor de C_i

 Alocar a tarefa escolhida na máquina com menor valor de C_i

 Verificar número da tarefa alocada (j)

 Eliminar do sequenciamento as demais tarefas (p_j) com número (j) igual nas demais máquinas

 Atualizar o valor de C_i das máquinas

até que todas tarefas estejam alocadas;

retorna SoluçãoInicial

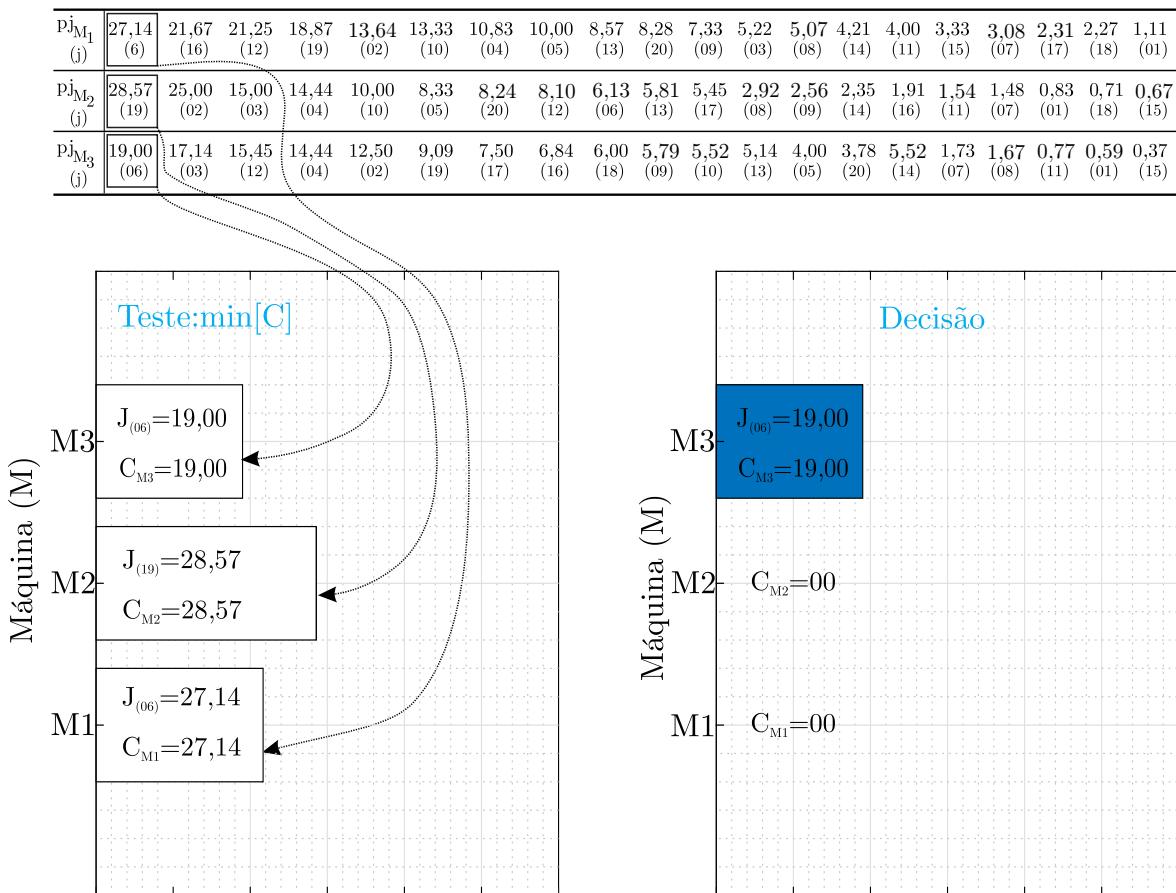
2.2.2.1 Introdução sobre a regra proposta LPTteste

Baseada na regra de prioridade LPT (*Longest Processing Time*) sugerida por Graham et al. (1979), a heurística LPTteste é proposta para a fase de alocação inicial das tarefas nas máquinas, ou seja, a fase construtiva do algoritmo. Este critério utiliza o mesmo modo de pré-teste aplicada na regra proposta SPTteste, realiza-se um pré-teste antes da escolha da alocação da tarefa j na máquina i e a escolha da tarefa acontece na máquina que tiver o menor valor de C_i .

2.2.2.2 FASE 1 - Alocação Inicial - LPTteste

Aplicando a regra de prioridade adaptada LPTteste, a Figura 2.8 demonstra a sequência inicial das tarefas $\{J_6, J_{19}, J_6\}$ com os tempos de processamento $p_{jM_1} = 27,14$, $p_{jM_2} = 28,57$ e $p_{jM_3} = 19,00$ para o pré-teste. O primeiro passo é posicionar as tarefas citadas, cada tarefa relacionada a sua máquina. Após, dá-se início ao pré-teste LPTteste, e verifica-se qual máquina que ficará com menor valor de C_i .

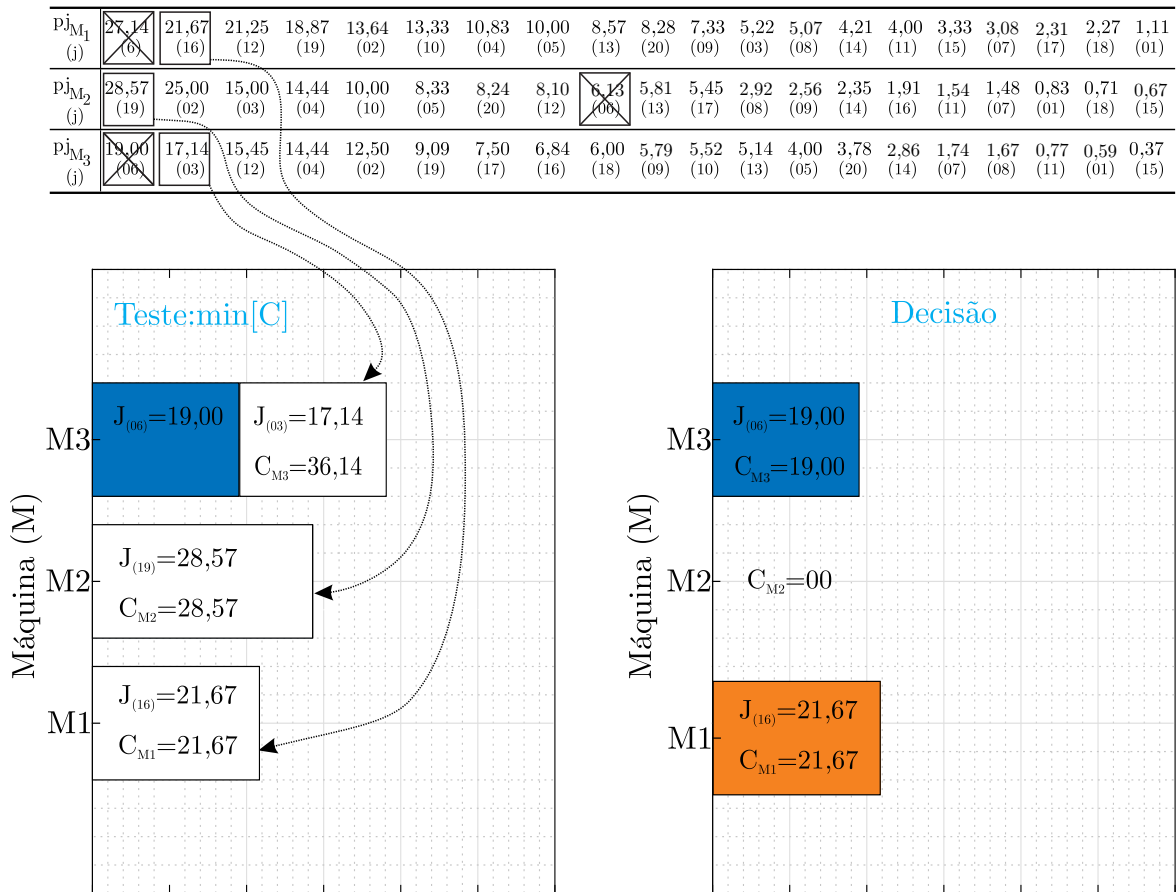
Figura 2.8 – Primeiro pré-teste com LPTteste.



Fonte: Autora, 2021.

Na Figura 2.8, aplicando o pré-teste, a tarefa escolhida $J_6 \{p_{jM_3} = 19,00\}$ será alocada na máquina M_3 por ter apresentado menor valor de C_i . De acordo com a Figura 2.9, e continuando o pré-teste, as próximas tarefas a serem distribuídas nas máquinas são $\{J_{16}, J_{19}, J_3\}$. Observe que as tarefas $J_6 \{p_{jM_3} = 19,00\}$ e $J_{19} \{p_{jM_2} = 28,57\}$ retornam para o pré-teste, isso ocorreu em virtude de não terem sido escolhidas ou eliminadas.

Figura 2.9 – Segundo pré-teste com LPTteste.

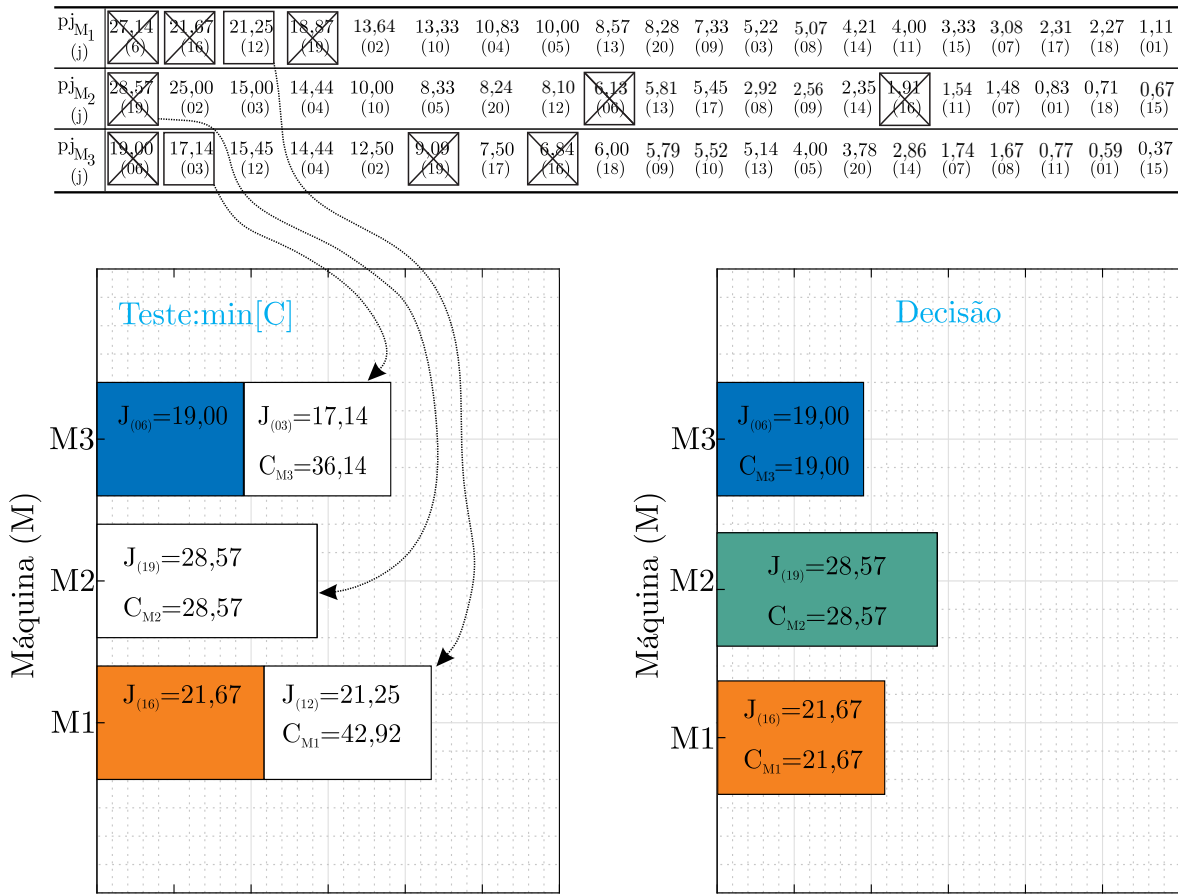


Fonte: Autora, 2021.

Nota-se na Figura 2.9, que a tarefa escolhida para alocação é J_{16} $\{p_{jM_1} = 21,67\}$ na máquina M_1 e conforme cada tarefa vai sendo alocada, os valores de C_i são atualizados.

O pré-teste seguinte será das tarefas $\{J_{12}, J_{19}, J_3\}$ com os tempos de processamento $p_{jM_1} = 21,25$, $p_{jM_2} = 28,57$ e $p_{jM_3} = 17,14$, como mostra a Figura 2.10:

Figura 2.10 – Terceiro pré-teste com LPTteste.

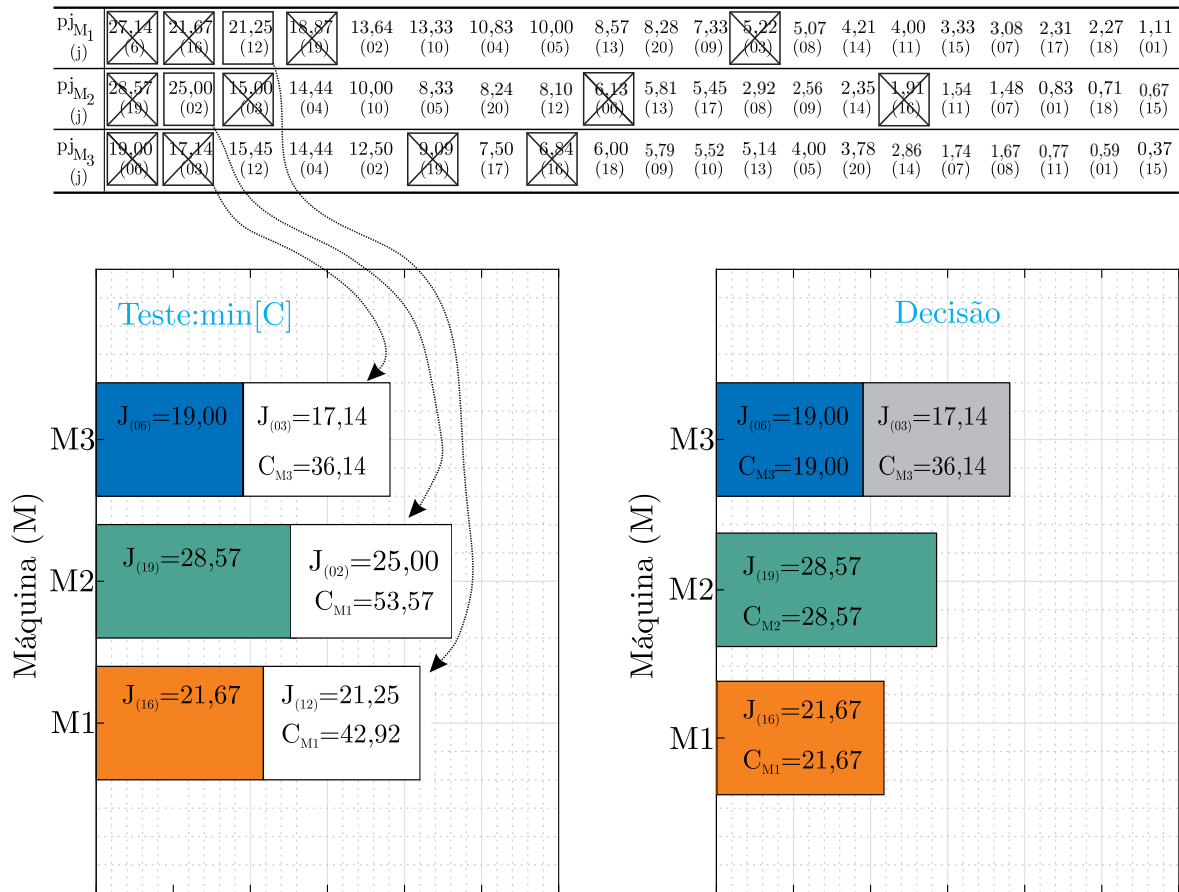


Fonte: Autora, 2021.

Na Figura 2.10, a tarefa $J_{16} \{p_{jM_2} = 28,57\}$ é escolhida para ser alocada na máquina M_1 . Observa-se que agora as três máquinas encontram-se com tarefas alocadas e o pré-teste vai mantendo um balanceamento entre as tarefas e os valores de C_i .

A escolha de uma regra de prioridade nesta primeira fase de alocação inicial é fundamental. Os pré-testes auxiliam mais rapidamente na escolha da tarefa e da máquina em que acontecerá o sequenciamento, neste caso, em máquinas paralelas não-relacionadas. No quarto pré-teste, na Figura 2.11, as tarefas selecionadas são $\{J_{12}, J_2, J_3\}$ e seus tempos de processamento $p_{jM_1} = 21,25$, $p_{jM_2} = 25,00$ e $p_{jM_3} = 17,14$.

Figura 2.11 – Quarto pré-teste com LPTteste.

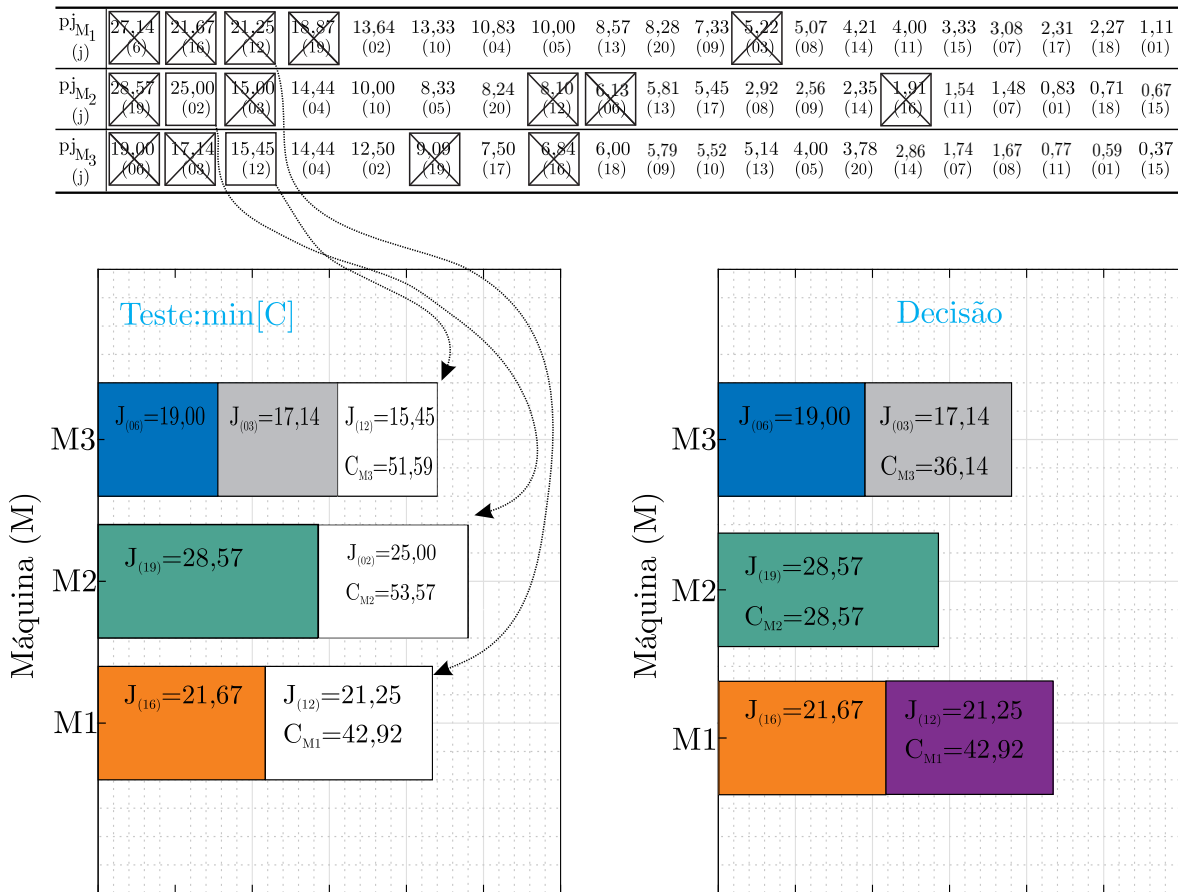


Fonte: Autora, 2021.

Constata-se que na máquina M_3 aconteceu um pré-teste da tarefa J_6 $\{p_{jM_3} = 19,00\}$ + a tarefa J_3 $\{p_{jM_3} = 17,14\} = 36,14$. No pré-teste este valor resultante é o que apresentou menor valor em C_i .

Dando sequencia nos pré-testes, as próximas tarefas $\{J_{12}, J_2, J_{12}\}$ com tempos de processamento $p_{jM_1} = 21,25$, $p_{jM_2} = 25,00$ e $p_{jM_3} = 15,45$ são inseridas conforme sua respectiva máquina. Observe que a máquina M_1 apresentou o menor valor de C_i neste pré-teste, assim, a tarefa J_{12} $\{p_{jM_1} = 21,25\}$ é alocada em M_1 , e respectivamente todas as tarefas J_{12} das máquinas M_2 e M_3 são eliminadas do sequenciamento, conforme apresentado na Figura 2.12:

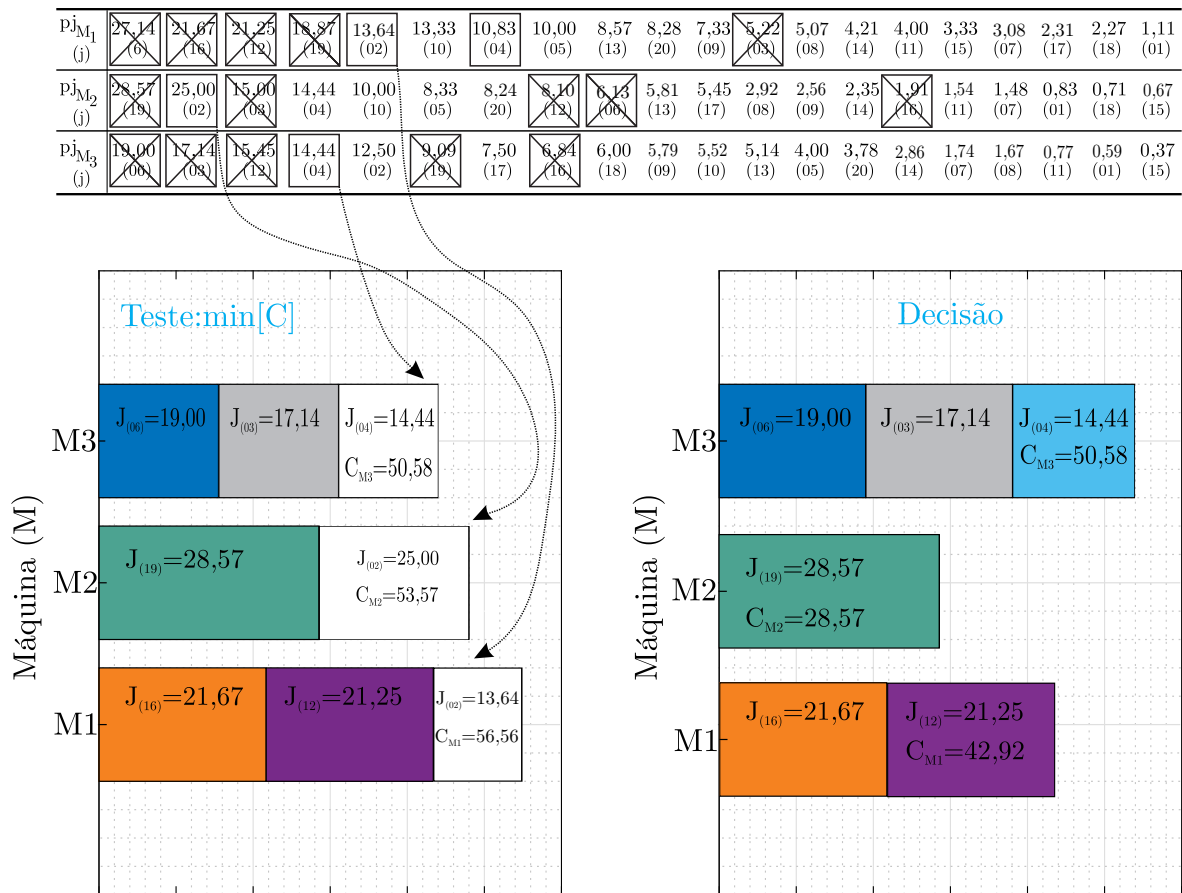
Figura 2.12 – Quinto pré-teste com LPTteste.



Fonte: Autora, 2021.

Dando continuidade ao pré-teste, a Figura 2.13 demonstra qual a tarefa que será escolhida e em qual máquina será alocada:

Figura 2.13 – Sexto pré-teste com LPTteste.



Fonte: Autora, 2021.

Através dos cálculos apresentado, é possível identificar a máquina que possui o menor valor de C_i para alocação da tarefa pré-testada. :

$$M_1 = J_{16} \{p_{jM_1} = 21,67\} + J_{12} \{p_{jM_1} = 21,25\} + J_2 \{p_{jM_1} = 13,64\} = 56,56.$$

$$M_2 = J_{19} \{p_{jM_2} = 28,57\} + J_2 \{p_{jM_2} = 25,00\} = 53,57.$$

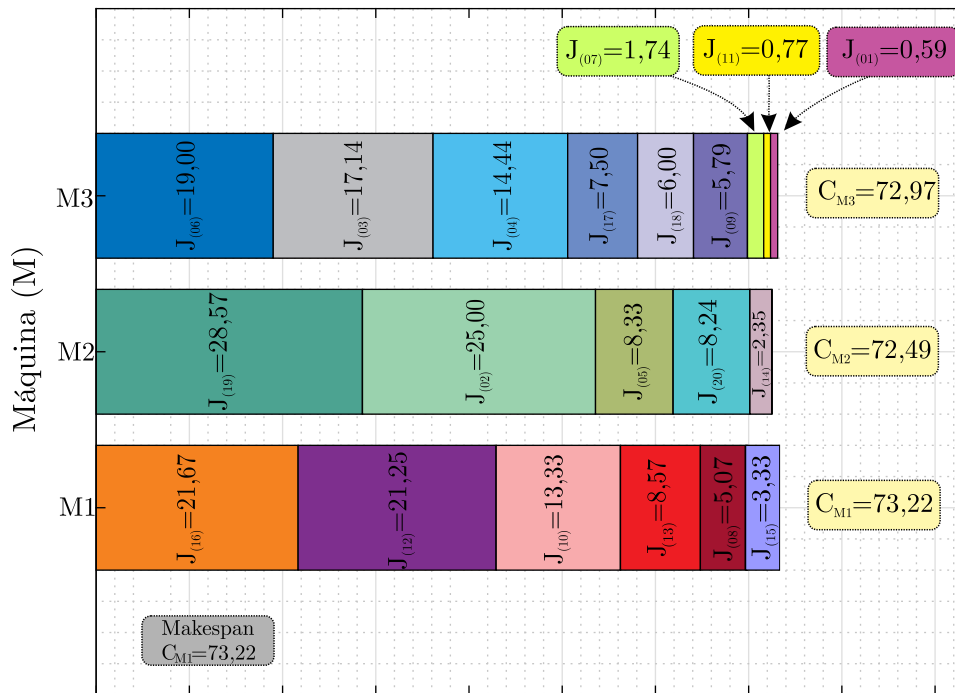
$$M_3 = J_6 \{p_{jM_3} = 19,00\} + J_3 \{p_{jM_3} = 17,14\} + J_4 \{p_{jM_3} = 14,44\} = 50,58.$$

A Figura 2.13 apontou a tarefa $J_4 \{p_{jM_3} = 14,44\}$ para ser alocada na máquina M_3 . Posteriormente, todas as tarefas J_4 serão excluídas das máquinas M_1 e M_2 .

Prosseguindo com os pré-testes e aplicando o critério adaptado LPTteste, é possível obter uma solução inicial para o problema de sequenciamento em máquinas paralelas não-relacionadas. A Figura 2.14 demonstra, através do diagrama de *Gantt*, como as tarefas ficaram alocadas e os valores de C_i nas três máquinas:

Figura 2.14 – Solução Inicial com critério proposto LPTteste.

$P_{M_1}^j$	$\frac{27,14}{(6)}$	$\frac{21,67}{(16)}$	$\frac{21,25}{(13)}$	$\frac{18,87}{(19)}$	$\frac{18,64}{(02)}$	$\frac{13,33}{(16)}$	$\frac{10,83}{(04)}$	$\frac{10,00}{(05)}$	$\frac{8,57}{(13)}$	$\frac{8,28}{(20)}$	$\frac{7,33}{(09)}$	$\frac{5,22}{(03)}$	$\frac{5,07}{(08)}$	$\frac{4,21}{(13)}$	$\frac{4,00}{(11)}$	$\frac{3,33}{(13)}$	$\frac{3,06}{(07)}$	$\frac{2,31}{(17)}$	$\frac{2,27}{(18)}$	$\frac{1,11}{(01)}$
$P_{M_2}^j$	$\frac{28,57}{(19)}$	$\frac{25,00}{(02)}$	$\frac{15,00}{(08)}$	$\frac{14,44}{(04)}$	$\frac{10,00}{(16)}$	$\frac{8,33}{(05)}$	$\frac{8,24}{(20)}$	$\frac{8,10}{(12)}$	$\frac{6,13}{(06)}$	$\frac{5,81}{(13)}$	$\frac{5,45}{(17)}$	$\frac{2,92}{(08)}$	$\frac{2,56}{(09)}$	$\frac{2,35}{(13)}$	$\frac{1,91}{(16)}$	$\frac{1,54}{(11)}$	$\frac{1,48}{(07)}$	$\frac{0,83}{(08)}$	$\frac{0,71}{(18)}$	$\frac{0,67}{(15)}$
$P_{M_3}^j$	$\frac{19,00}{(06)}$	$\frac{17,14}{(08)}$	$\frac{15,45}{(12)}$	$\frac{14,44}{(04)}$	$\frac{12,50}{(02)}$	$\frac{9,09}{(19)}$	$\frac{7,50}{(17)}$	$\frac{6,84}{(16)}$	$\frac{6,00}{(18)}$	$\frac{5,79}{(09)}$	$\frac{5,52}{(10)}$	$\frac{5,14}{(13)}$	$\frac{4,00}{(05)}$	$\frac{3,78}{(20)}$	$\frac{2,86}{(14)}$	$\frac{1,74}{(07)}$	$\frac{1,67}{(08)}$	$\frac{0,77}{(11)}$	$\frac{0,59}{(01)}$	$\frac{0,37}{(15)}$



Fonte: Autora, 2021.

Constata-se que o *makespan* neste caso apresentou-se na máquina M_1 com valor de $C_1 = 73,22$. Importante ressaltar que esse valor foi o tempo necessário da máquina mais carregada para concluir seu conjunto de tarefas alocadas em M_1 .

O próximo subitem 2.1.3, apresenta mais uma regra de prioridade, denominada como FCFS (*First Come, First Served*), ou melhor, a primeira tarefa a chegar e a primeira a ser servida. E a adaptação da heurística FCFSteste para o problema $R|multifact|Cmax$.

2.2.3 FCFS - (*First Come, First Served*)

A regra de prioridade FCFS (*First Come, First Served*) vem sendo estudada e aplicada em problemas de programação de produção. Este método, pode ser também denominado como FCFS (Primeiro a chegar, primeiro a ser servido) e tem como critério: a primeira tarefa que é finalizada no processo deve ser a primeira a sair, ou melhor, as tarefas devem ser executadas conforme ordem de chegada. Essa regra minimiza o tempo

de finalização e a perda de tempo, já que diminui a permanência das tarefas na máquina ou no próprio espaço de produção.

Wang e Xing (2006) estudaram este método em máquinas paralelas não-relacionadas considerando um problema de sequenciamento Off-line, onde todas as tarefas estão esperando em uma linha e o critério de chegada deve ser seguida. Esta heurística sempre processa a primeira tarefa na lista de tarefas não agendadas na primeira máquina ociosa. Caso a primeira tarefa não programada seja uma tarefa especial e a máquina estiver ocupada, toda linha aguardará até que a máquina esteja ociosa.

Lin (2013) investigou uma heurística e um algoritmo de otimização por enxame de partículas. As duas primeiras partículas são geradas pela regra FCFS (*First Come, First Served*). Para esta regra o pesquisador considerou todas as tarefas não sequenciadas e agendadas, cada uma na primeira máquina disponível.

Yang-Kuei e Chi-Wei (2013) desenvolveram um modelo de programação inteira mista para uma solução ótima para máquinas paralelas não-relacionadas, propuseram várias regras de prioridade para encontrar bons resultados e dentre elas o critério FCFS (*First Come, First Served*).

Li et al. (2021a) compararam quatro heurísticas construtivas clássicas. O critério FCFS usado na avaliação da solução se mostrou confiável e com bons resultados ao resolver o problema para máquinas não-relacionadas.

O presente estudo mostra, com base nos dados fornecidos das Tabelas 2.1, 2.2 e 2.3, o critério utilizado para a regra de prioridade FCFS (*First Come, First Served*) conforme as Tabelas 2.10, 2.11 e 2.12, onde apresentam-se os valores dos tempos de processamento de cada tarefa em cada máquina, não tendo a necessidade de uma pré-ordenação das tarefas, condição necessária por muito dos algoritmos construtivos em problemas de sequenciamento conceituados.

Tabela 2.10 – Tempos de processamento das tarefas segundo regra FCFS - M1.

j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
p_{jM1}	1.11	13.64	5.22	10.83	10.00	27.14	3.08	5.07	7.33	13.33	4.00	21.25	8.57	4.21	3.33	21.67	2.31	2.27	18.87	8.28

Tabela 2.11 – Tempos de processamento das tarefas segundo regra FCFS - M2.

j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
p_{jM2}	0.83	25.00	15.00	14.44	8.33	6.13	1.48	2.92	2.56	10.00	1.54	8.10	5.81	2.35	0.67	1.91	5.45	0.71	28.57	8.24

Tabela 2.12 – Tempos de processamento das tarefas segundo regra FCFS - M3.

j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
p_{jM3}	0.59	12.50	17.14	14.44	4.00	19.00	1.74	1.67	5.79	5.52	0.77	15.45	5.14	2.86	0.37	6.84	7.50	6.00	9.09	3.78

Cada regra de prioridade possui um método para iniciar um sequenciamento. E como detalhado anteriormente nas regras SPT e LPT, em seguida, será proposto uma implementação para o problema $R|multifact|Cmax$ com uma heurística adaptada e denominada como FCFSteste, baseado na regra de prioridade FCFS (*First Come, First Served*). Apresenta-se agora, a demonstração do procedimento do algoritmo 3 nesta etapa com o método proposto FCFSteste.

Algoritmo 3: Alocação Inicial com pré-teste para FCFSteste.

Entrada de Dados;

Solução Inicial - Algoritmo Construtivo

repita

 Posicionar as tarefas (p_j) nas M máquinas

 Aplicar pré-teste nas máquinas:

 Escolher a tarefa (p_j) que proporcione o menor valor de C_i

 Alocar a tarefa escolhida na máquina com menor valor de C_i

 Verificar número da tarefa alocada (j)

 Eliminar do sequenciamento as demais tarefas (p_j) com número (j) igual nas demais máquinas

 Atualizar o valor de C_i das máquinas

até que todas tarefas estejam alocadas;

retorna SoluçãoInicial

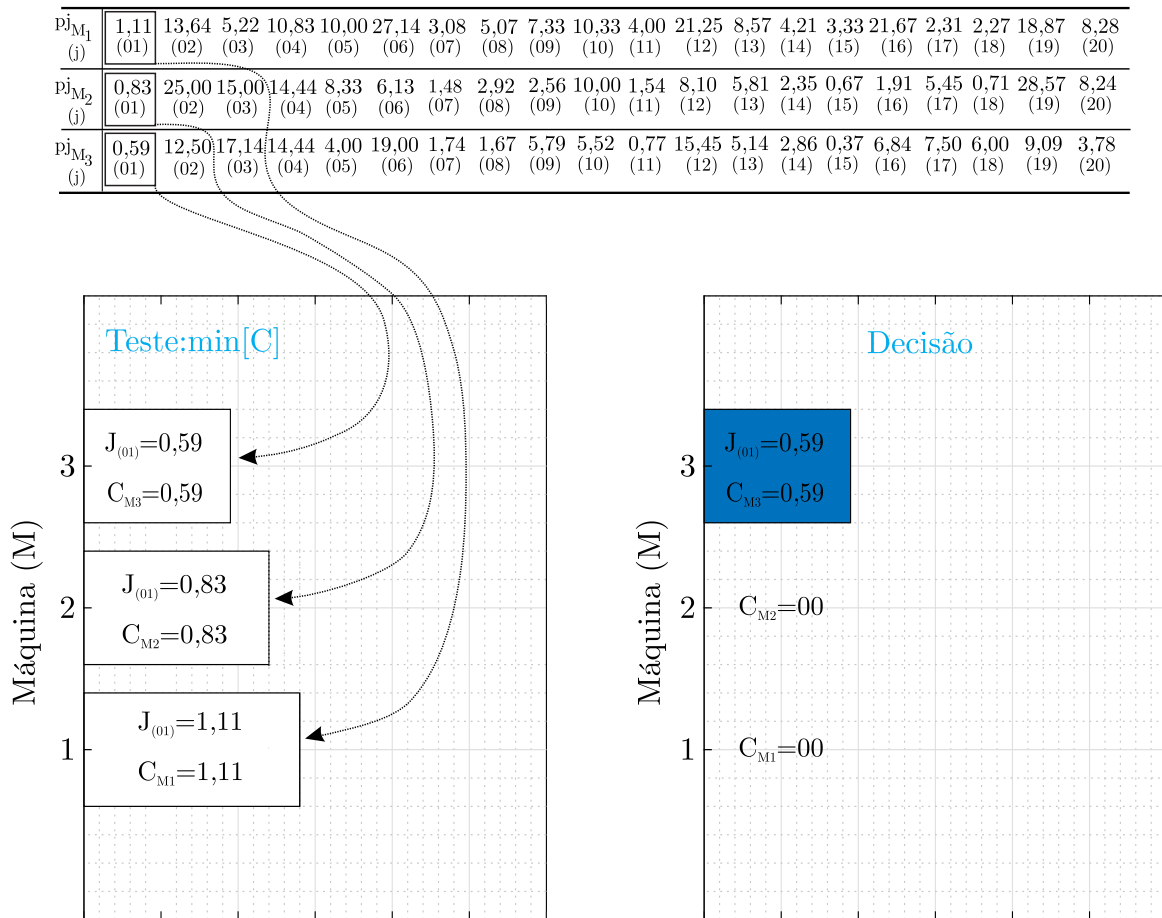
2.2.3.1 Introdução sobre a regra proposta FCFSteste

Fundamentada na regra de prioridade FCFS (*First Come, First Served*) e mencionada por Baker (1974), Morton e Pentico (1993), o algoritmo FCFSteste é sugerido para a alocação inicial do problema de sequenciamento em estudo. Conforme abordado nas regras anteriores, o método utilizado como pré-teste, é descrito da mesma forma para o desenvolvimento desta etapa, denominada também de Fase 1 - Alocação Inicial do procedimento construtivo.

2.2.3.2 FASE 1 - Alocação Inicial - FCFSteste

Usando a regra de prioridade adaptada FCFSteste, a Figura 2.15 demonstra os primeiros passos do pré-teste para início do sequenciamento. As primeiras tarefas candidatas são $\{J_1, J_1, J_1\}$ e os tempos de processamento $p_{jM1} = 1, 11$, $p_{jM2} = 0, 83$ e $p_{jM3} = 0, 59$. Inicialmente, as tarefas são posicionadas de acordo com a máquina relacionada.

Figura 2.15 – Primeiro pré-teste com FCFSteste.



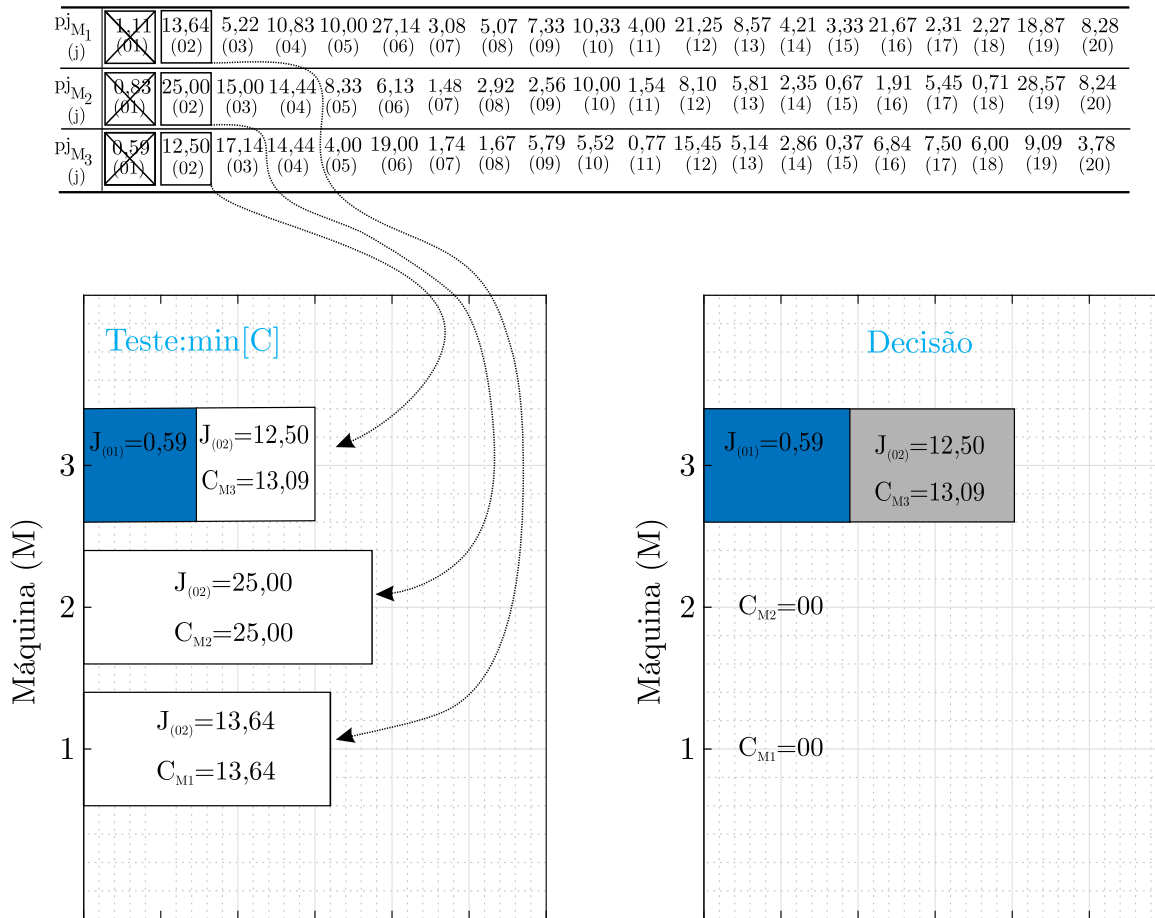
Fonte: Autora, 2021.

Observa-se que na Figura 2.15, as tarefas posicionadas para o pré-teste possuem o mesmo número $\{J_1, J_1, J_1\}$, isso ocorre devido ao critério adotado para esse processo. No método FCFS não há necessidade de uma pré-ordenação dos tempos de processamento das tarefas, ou seja, a primeira tarefa que chega é a primeira a ser processada. Desse modo, os pré-testes seguem de acordo com o posicionamento em que as tarefas encontram-se na fila das suas respectivas máquinas.

Na Figura 2.15 a tarefa $J_1 \{p_{jM_3} = 0,59\}$ é escolhida para alocação por ser a tarefa com menor valor de tempo de processamento para dar início ao sequenciamento e seguindo a regra FCFSteste, as tarefas J_1 das máquinas M_1 e M_2 são eliminadas do pré-teste.

No próximo pré-teste as tarefas a serem posicionadas são $\{J_2, J_2, J_2\}$ e os valores dos tempos de processamento $p_{jM_1} = 13,64$, $p_{jM_2} = 25,00$ e $p_{jM_3} = 12,50$, conforme ilustrado na Figura 2.16:

Figura 2.16 – Segundo pré-teste com FCFSteste.

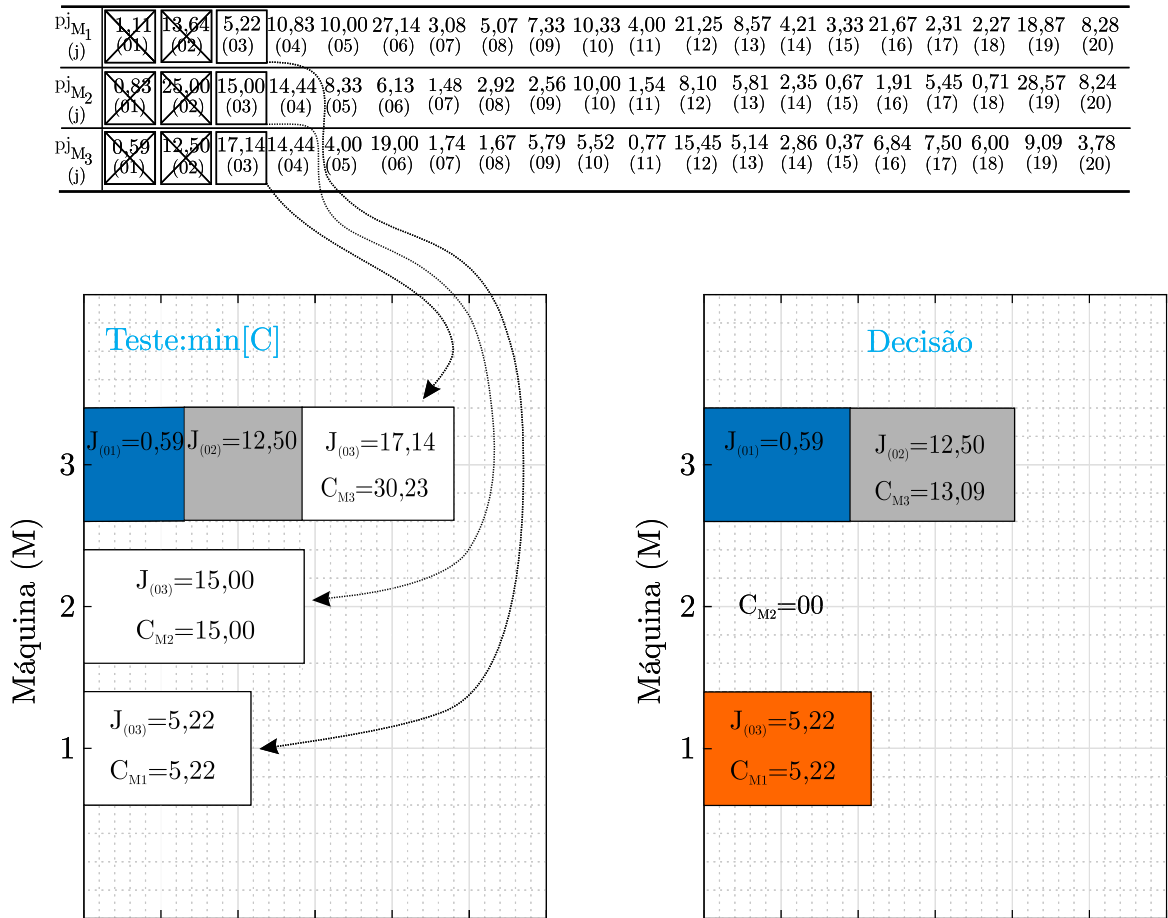


Fonte: Autora, 2021.

Na figura acima, a tarefa escolhida para alocação é $J_2 \{p_{jM_3} = 12,50\}$ na máquina M_3 , em virtude de ter apresentado o menor valor de C_i entre as três máquinas.

A Figura 2.17 abaixo, demonstra o terceiro pré-teste com as tarefas $\{J_3, J_3, J_3\}$ e os tempos de processamento $p_{jM_1} = 5,22$, $p_{jM_2} = 15,00$ e $p_{jM_3} = 17,14$.

Figura 2.17 – Terceiro pré-teste com FCFSteste.

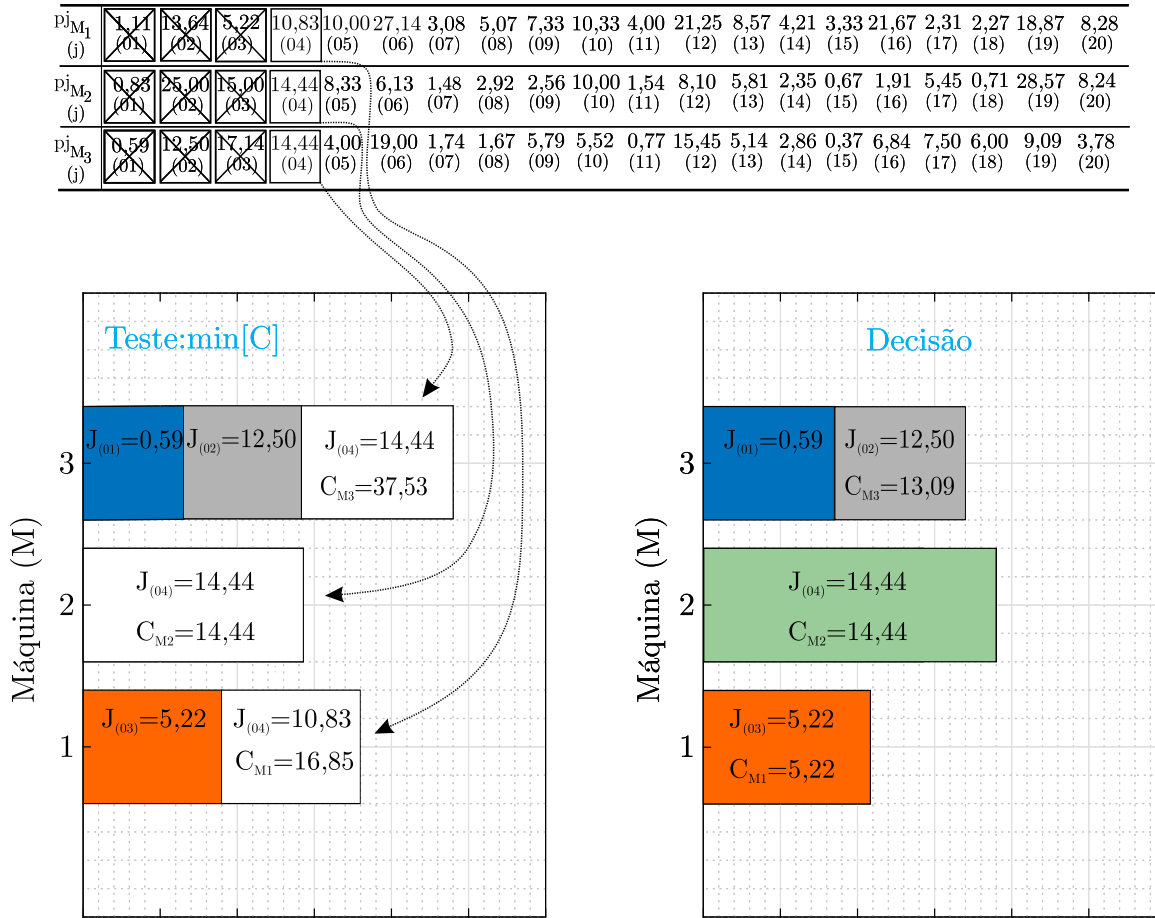


Fonte: Autora, 2021.

A tarefa escolhida para alocação neste pré-teste é $J_3 \{p_{jM_1} = 5, 22\}$ na máquina M_1 , por ter apresentado o menor valor de C_I e as tarefas J_3 são excluídas da fila de sequenciamento.

Prosseguindo os pré-testes FCFSteste, as próximas tarefas candidatas são $\{J_4, J_4, J_4\}$ e os valores dos tempos de processamento $p_{jM_1} = 10, 83$, $p_{jM_2} = 14, 44$ e $p_{jM_3} = 14, 44$, como é demonstrado na Figura 2.18:

Figura 2.18 – Quarto pré-teste com FCFSteste.

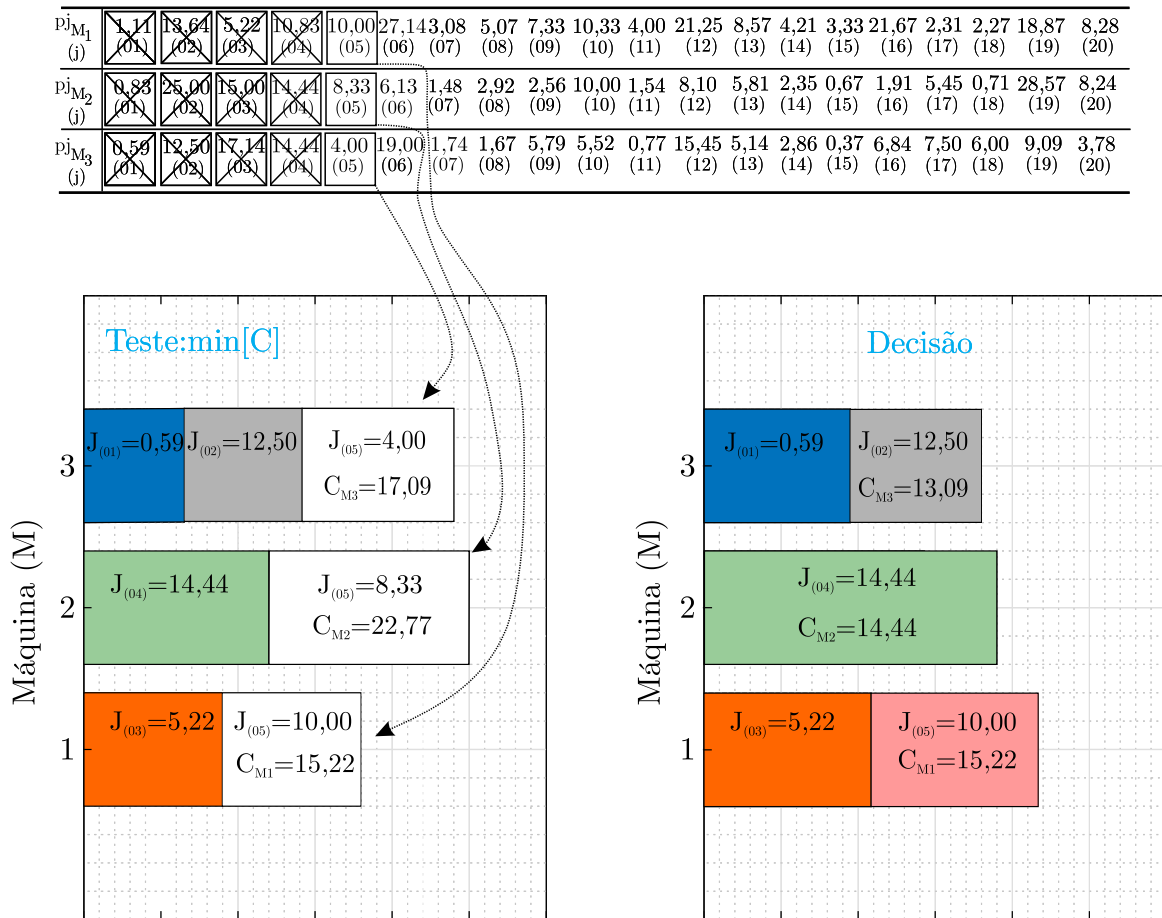


Fonte: Autora, 2021.

Constata-se na Figura 2.17, que o pré-teste vai aproximando os valores de C_i das três máquinas paralelas não-relacionadas, deixando uma distribuição equilibrada de tarefas entre as máquinas. A tarefa definida para alocação neste pré-teste é $J_4 \{p_{j_2} = 14,44\}$ na máquina M_2 .

Os pré-testes seguem até que todas as tarefas sejam escolhidas e alocadas nas máquinas. As tarefas $\{J_5, J_5, J_5\}$ e os tempos de processamento $p_{jM_1} = 10,00$, $p_{jM_2} = 8,33$ e $p_{jM_3} = 4,00$ são as próximas candidatas a serem testadas, de acordo com a Figura 2.19:

Figura 2.19 – Quinto pré-teste com FCFSteste.



Fonte: Autora, 2021.

Conforme mostrado na Figura acima, o cálculo para encontrar o menor valor de C_i consiste em:

$$M_1 = J_3 \{p_{jM_1} = 5, 22\} + J_5 \{p_{jM_1} = 10, 00\} + J_5 \{p_{jM_1} = 10, 00\} = 15,22.$$

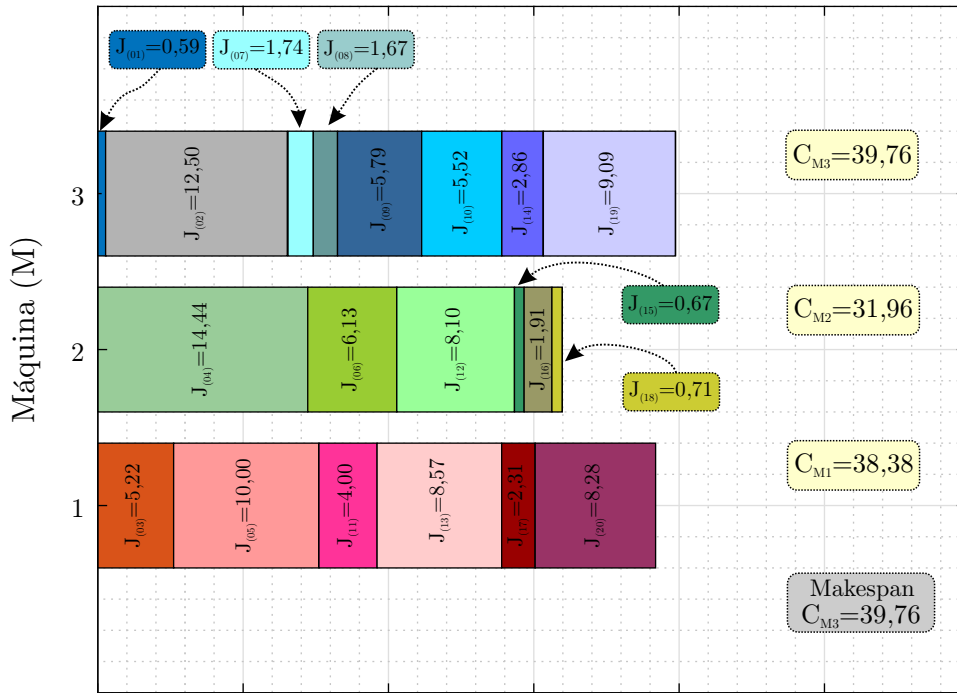
$$M_2 = J_4 \{p_{jM_2} = 14, 44\} + J_5 \{p_{jM_2} = 8, 33\} = 22,77.$$

$$M_3 = J_1 \{p_{jM_3} = 0, 59\} + J_2 \{p_{jM_3} = 12, 50\} + J_5 \{p_{jM_3} = 4, 00\} = 17,09.$$

Ao final dos pré-testes, com o critério adaptado para o atual estudo, FCFSteste, a Figura 2.20 apresenta as 20 tarefas alocadas nas 3 máquinas. Desse modo, obtém-se uma solução inicial para o problema $R||Cmax$.

Figura 2.20 – Solução Inicial com o critério proposto FCFSteste.

Pj_{M_1} (j)	1,11 (01)	18,64 (02)	5,22 (03)	10,83 (04)	10,00 (05)	27,14 (06)	3,08 (07)	5,07 (08)	7,33 (09)	10,33 (10)	2,00 (11)	21,25 (12)	1,57 (13)	1,21 (14)	3,35 (15)	21,67 (16)	2,31 (17)	3,27 (18)	18,8 (19)	8,28 (20)
Pj_{M_2} (j)	0,83 (01)	25,00 (02)	18,00 (03)	14,44 (04)	8,33 (05)	6,13 (06)	1,48 (07)	2,92 (08)	2,56 (09)	10,00 (10)	1,54 (11)	8,10 (12)	8,81 (13)	0,35 (14)	1,91 (15)	0,67 (16)	5,45 (17)	0,71 (18)	28,57 (19)	8,24 (20)
Pj_{M_3} (j)	0,59 (01)	12,50 (02)	17,14 (03)	14,44 (04)	4,00 (05)	19,00 (06)	1,74 (07)	1,67 (08)	5,79 (09)	5,52 (10)	0,77 (11)	15,49 (12)	0,14 (13)	0,86 (14)	0,37 (15)	6,84 (16)	7,50 (17)	6,00 (18)	0,09 (19)	3,78 (20)



Fonte: Autora, 2021.

É possível observar na Figura 2.20 como foram distribuídas as tarefas nas máquinas. De acordo com a finalização do sequenciamento para alocação das tarefas nas máquinas não-relacionadas, a máquina M_3 apontou ser o processador mais carregado, o tempo de finalização de todas as tarefas em M_3 é de 39,76, denominado como *makespan*.

A busca por um critério para iniciar um problema de sequenciamento parte do que pretende-se resolver no estudo. A atual pesquisa, é um problema de sequenciamento em máquinas paralelas não-relacionadas ($R/multifactor/Cmax$) e visa minimizar o tempo máximo de conclusão de todas as tarefas, ou seja, reduzir o *makespan*.

Para cada uma das regras de prioridade SPT, LPT e FCFS, adaptou-se uma heurística nova explicando em cada processo do sequenciamento o passo a passo para a implementação.

O próximo capítulo trata da segunda fase, denominada também como fase de balanceamento, apresenta-se o comportamento das tarefas nas máquinas e o critério proposto para esta etapa do estudo.

3 FASE DE BALANCEAMENTO PARA *R/MULTIFACT/CMAX*

A fase de balanceamento, ou também podemos denominá-la como Fase 2, ocorre após a fase de alocação inicial. Esta fase parte da solução encontrada na Fase 1 e pode ser observada como um procedimento de melhoramento. Busca diminuir o tempo de finalização, mudando tarefas da máquina mais carregada para a máquina menos carregada, até que não haja mais trocas que alcancem este objetivo.

Desse modo, nessa fase, para as tarefas candidatas serem transferidas da máquina mais carregada para a máquina menos carregada, estas são orientadas por um tempo médio de finalização, também chamado C_{medio} .

Os critérios de solução conseguem ser analisados quanto ao seu bom resultado em relação a qualidade da solução e quanto ao seu melhor rendimento em relação ao custo computacional imposto. De um modo geral, referindo-se as heurísticas, como: construtivas, de melhoramento ou as metaheurísticas, não são limitadas em relação a eficiência computacional.

Para o problema abordado, foi proposto um valor médio para a solução do problema do presente estudo:

$$C_{medio} = \sum_{m=1}^M \frac{C_m}{M} \quad (3.1)$$

O valor do C_{medio} servirá como um alvo para decidir em qual máquina será feita a busca da tarefa a ser trocada e qual será essa tarefa. Cujo valor é a função dos tempos de processamento total de cada máquina C_m dividido pelo número de máquinas M .

O C_{medio} representa também um valor médio e a cada iteração esse valor geralmente muda. Caso não ocorra nenhuma melhora para este caso, levando em conta o alvo, inicia-se uma nova procura, levando em consideração uma melhora na solução, por meio de qualquer tarefa que possibilite diminuir o valor do *makespan* inicial.

O algoritmo 4 trás a demonstração do desenvolvimento da fase de balanceamento para um melhor entendimento.

Algoritmo 4: Algoritmo da Fase de Balanceamento

Entrada de dados;

início

inicialização - Balanceamento;

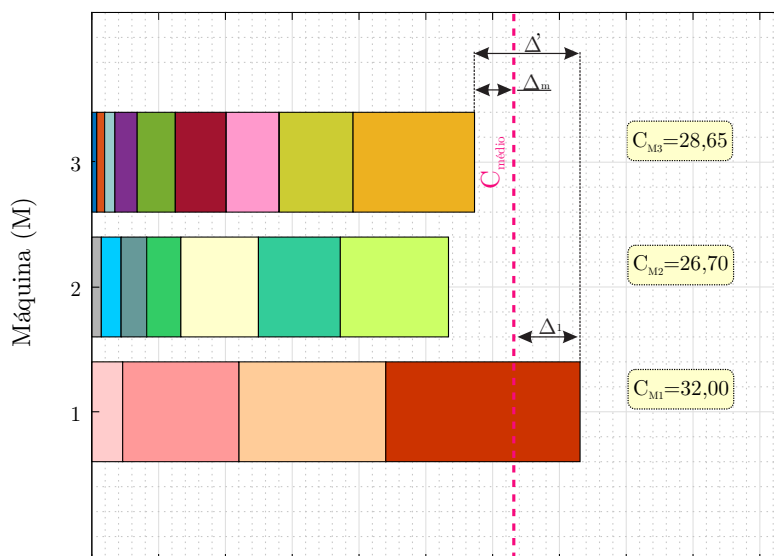
repitaPosicionar as tarefas (p_j) nas M máquinas;Identificar a máquina mais carregada (M_1) e a máquina menos carregada (M_m);Avaliar o valor da tarefa (p_j) ao ser retirada da máquina M_1 ;Verificar o efeito desta tarefa em M_m ;Escolher as tarefas candidatas a sair de M_1 para entrar em M_m ;Encontrar o valor da tarefa (p_j) \leq a Δ_1 ;Encontrar o valor da tarefa (p_j) \leq a Δ_m ;Recalcular o valor de C_{medio} ;**if** C_{medio} for $<$ que valor anterior **then**

| define-se como novo alvo de busca;

else| define-se o valor de Δ' ;**end****até** que se obtenha uma aproximação nos valores de M_1 , M_m ao C_{medio} ;**fim**

A Figura 3.2 demonstra o procedimento da Fase 2 do algoritmo.

Figura 3.1 – Representação dos elementos da Fase 2.



O objetivo nesta fase será a aproximação dos valores de C_{M_1} e de C_{M_m} ao valor de C_{medio} . Em seguida, ao identificar a máquina mais carregada M_1 e a máquina menos carregada M_m , defini-se os valores de:

$$\Delta_1 = C_{M_1} - C_{medio}$$

$$\Delta_m = C_{medio} - C_{M_m}$$

$$\Delta' = C_{M_1} - C_{M_m}$$

Para transferir uma tarefa da máquina mais carregada para a máquina menos carregada, é necessário avaliar o comportamento do valor do p_j produzido pela retirada de uma tarefa da máquina mais carregada e o efeito desta tarefa a máquina menos carregada. A busca pelas tarefas candidatas a sair da máquina mais carregada M_1 para a máquina menos carregada M_m , deve ser uma tarefa menor ou igual ao Δ_1 . O valor de p_j da máquina mais carregada será obtida por:

$$p_{jM_1} \leq \underbrace{C_{M_1} - C_{medio}}_{\Delta_1} \quad (3.2)$$

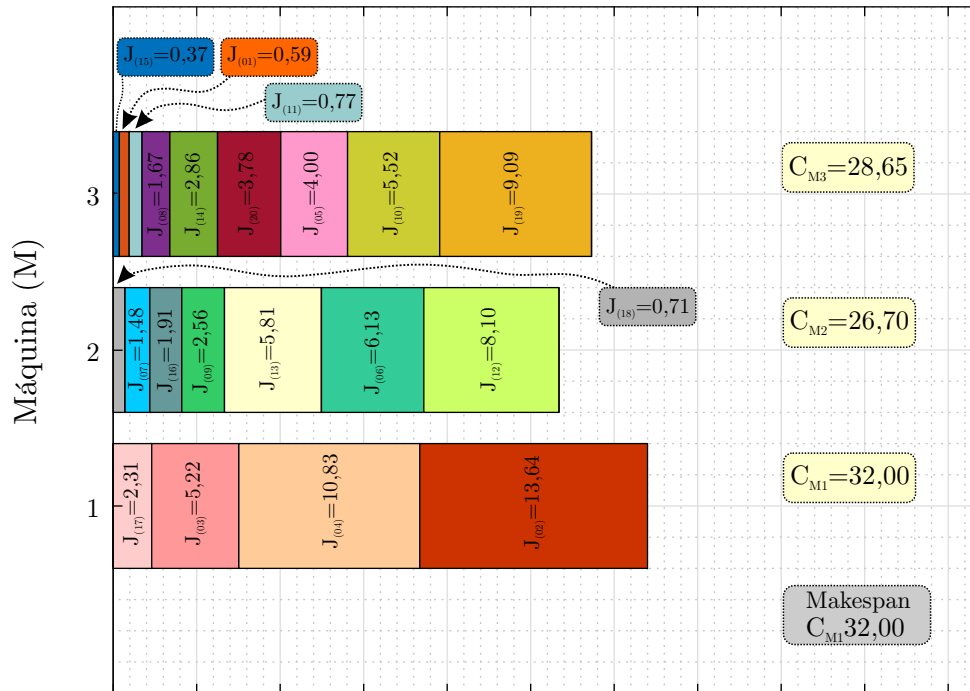
Após a busca pelas tarefas candidatas a entrar na máquina menos carregada, é necessário adequar as velocidades a máquina M_m , em razão do problema tratar de máquinas R/multifact/Cmax. O valor da tarefa candidata da máquina menos carregada deve ser menor ou igual ao Δ_m , então é escolhida a tarefa mais próxima de Δ_m , obtida por:

$$\max \left(p_{jM_m} \leq \underbrace{C_{medio} - C_{M_m}}_{\Delta_m} \right) \quad (3.3)$$

Após a escolha das tarefas candidatas para transferência, o valor de C_{medio} é recalculado. Caso o novo C_{medio} seja menor que o anterior, este será denominado como o novo alvo de busca. Se ocorrer de nenhuma tarefa da máquina mais carregada poder ser transferida para a máquina menos carregada, em função desse alvo, um novo alvo é definido por Δ' .

Para um melhor entendimento desta fase de balanceamento, a seguir, apresenta-se os passos na Fase 2, utilizando o mesmo exemplo da Fase 1.

Figura 3.2 – Exemplo Fase 1 com critério proposto SPTteste.



Fonte: Autora, 2021.

A máquina mais carregada é M_1 e a máquina menos carregada é M_2 . O algoritmo procura transferir uma tarefa alocada em M_1 para M_2 :

$$C_{medio} = 29,12$$

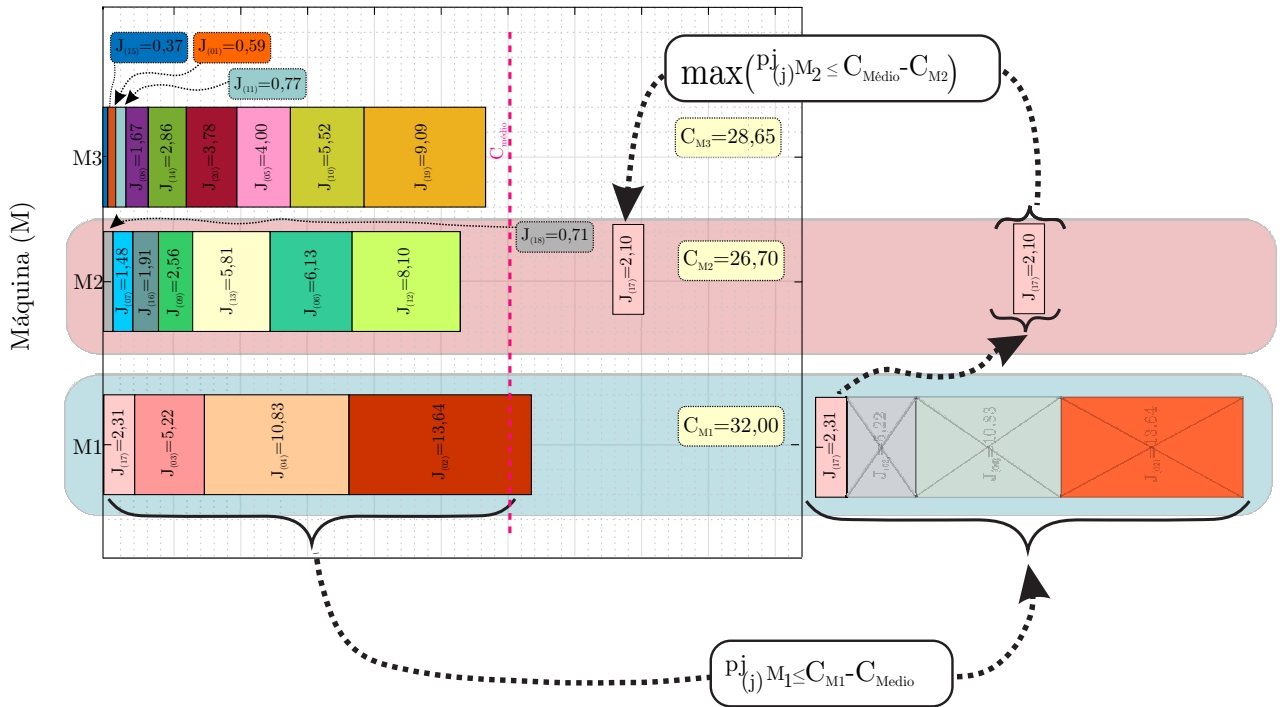
$$\Delta_1 = C_{M1} - C_{medio}$$

$$\Delta_1 = 32,00 - 29,12$$

$$\Delta_1 = 2,88$$

Verifica-se há uma tarefa com valor menor ou igual ao Δ_1 . Localiza-se a tarefa candidata a sair de acordo com a equação 3.2. O resultado é demonstrado na Figura 3.4:

Figura 3.3 – Escolha da primeira tarefa candidata a ser transferida.



Fonte: Autora, 2021.

Nota-se que a tarefa $J_{17} \{p_{jM_1} = 2, 31\}$ na máquina mais carregada, é a tarefa candidata escolhida para ser transferida para máquina menos carregada M_2 . Conforme a equação 3.2, esta tarefa satisfaz o valor de busca, que neste caso, o valor precisa ser menor ou igual ao $\Delta_1 = 2,88$.

Como o estudo trata de máquinas paralelas não-relacionadas em multi-fábricas, em que os tempos de processamento das tarefas variam entre as máquinas, mas não seguem nenhum tipo padrão, o valor de p_{jM_2} muda ao ser transferido para M_2 . Encontra-se este novo valor aplicando a equação 2.1.

$$p_{jM_2} = \frac{2, 31}{1, 10} = 2, 10 \quad (3.4)$$

Divide-se o tempo de processamento da máquina M_1 , $J_{17} \{p_{jM_1} = 2, 31\}$ pela velocidade da tarefa $J_{17} \{S_{ijM_2} = 1, 10\}$, conforme valor da Tabela 2.2. O resultado obtido de $p_j = 2, 10$ satisfaz o valor de busca para entrar em M_2 , que de acordo com a equação 3.3, a tarefa precisa ser menor ou igual ao Δ_m :

$$C_{medio} = 29, 12$$

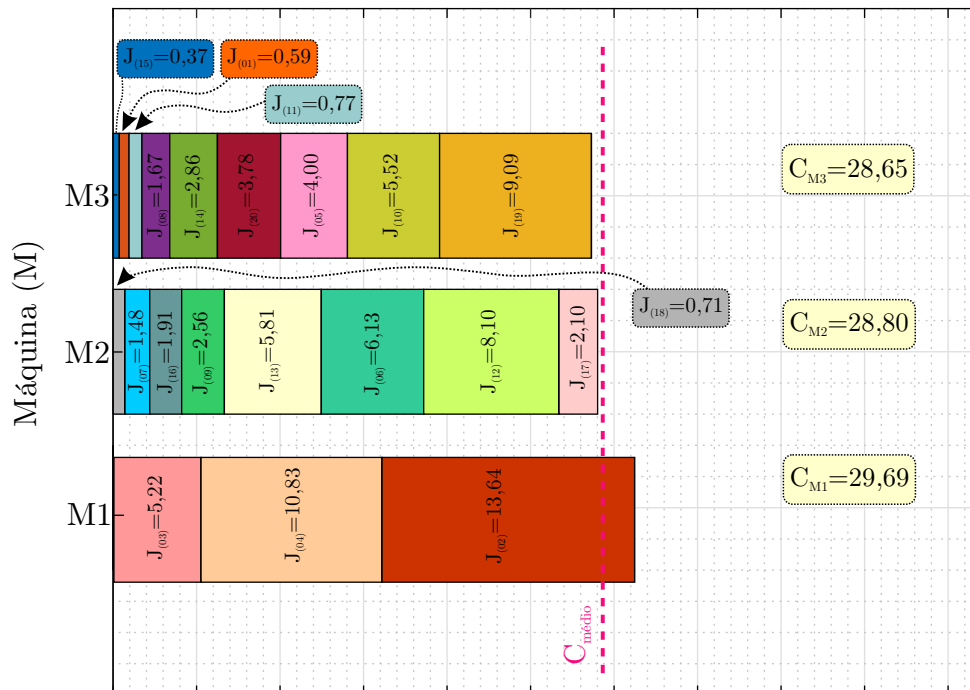
$$\Delta_m = C_{medio} - C_{M2}$$

$$\Delta_m = 29,12 - 26,70$$

$$\Delta_m = 2,42$$

Constata-se que na Figura 3.5 ao ser transferida a primeira tarefa para M_2 , os valores de C_i , ou melhor, os tempos de processamento total das tarefas nas máquinas i ficam balanceadas e aproximam-se do valor de C_{medio} .

Figura 3.4 – Solução Final da Fase 2.



Fonte: Autora, 2021.

Na Figura 3.5, a máquina mais carregada continua sendo M_1 e a máquina menos carregada agora passou a ser M_3 . O algoritmo procura transferir uma tarefa alocada na máquina M_1 para a máquina M_3 .

$$C_{medio} = 29,12$$

$$\Delta_1 = C_{M1} - C_{medio}$$

$$\Delta_1 = 29,69 - 29,12$$

$$\Delta_1 = 0,57$$

Procura-se uma tarefa com valor menor ou igual ao $\Delta_1=0,57$ conforme equação 3.2 na máquina M_1 . Como não se encontra nenhum valor menor ou igual que 0,57 redefine-se o alvo de busca para o valor de Δ' , onde:

$$\Delta' = C_{M1} - C_{Mm}$$

$$\Delta' = 29,69 - 28,65$$

$$\Delta' = 1,04$$

Como nenhuma tarefa é menor ou igual $\Delta' = 1,04$, o algoritmo passa para Fase 3, denominada como Fase de Duplas Trocas para $R|multifact|C_{max}$.

4 FASE DE DUPLAS TROCAS PARA *R/MULTIFACT/CMAX*

A fase de duplas trocas, ou também chamada de fase 3, possibilita que uma tarefa seja transferida da máquina mais carregada para uma outra máquina selecionada entre as máquinas menos carregadas, e obtém desta uma tarefa menor, cuja transferência possibilite um valor de *makespan* menor que o valor inicial, permitindo assim uma diminuição no tempo de finalização da máquina mais carregada.

Esta fase é semelhante a fase 2, porém, a dupla troca é formada por uma série de trocas simples e têm a demonstração de dois métodos de execução: um método para aplicar em um pequeno número de tarefas e outro procedimento para usar em grandes números de tarefas a serem sequenciadas. Para um sequenciamento com grandes números de tarefas, o presente estudo propõe um alvo, não é necessariamente o melhor alvo, mas um valor de referência, que será explicado no decorrer do capítulo.

O algoritmo a seguir demonstra o procedimento nesta fase de duplas trocas com o método proposto de um teste, como descrito nas fases anteriores.

Algoritmo 5: Algoritmo da Fase de Duplas Trocas

Entrada de dados;

início

inicialização - Duplas Trocas;

repita

Posicionar as tarefas (p_j) nas M máquinas;

Identificar a máquina mais carregada (M_1) e a tarefa com menor valor de p_j ;

Testar a tarefa com menor valor de p_j nas máquinas M_m ;

Atualizar os valores de C_i ;

Identificar nas máquinas M_m as tarefas que possibilitem para troca um valor de *makespan* atual < que o inicial;

Escolher as possíveis candidatas nas máquinas M_m ;

Identificar se as candidatas possuem valores \leq a tarefa que saiu de M_1 ;

if *tiver tarefas candidatas para troca* **then**

| aplica-se o teste;

else

| tiver tarefa para troca, muda-se a busca para outra máquina;

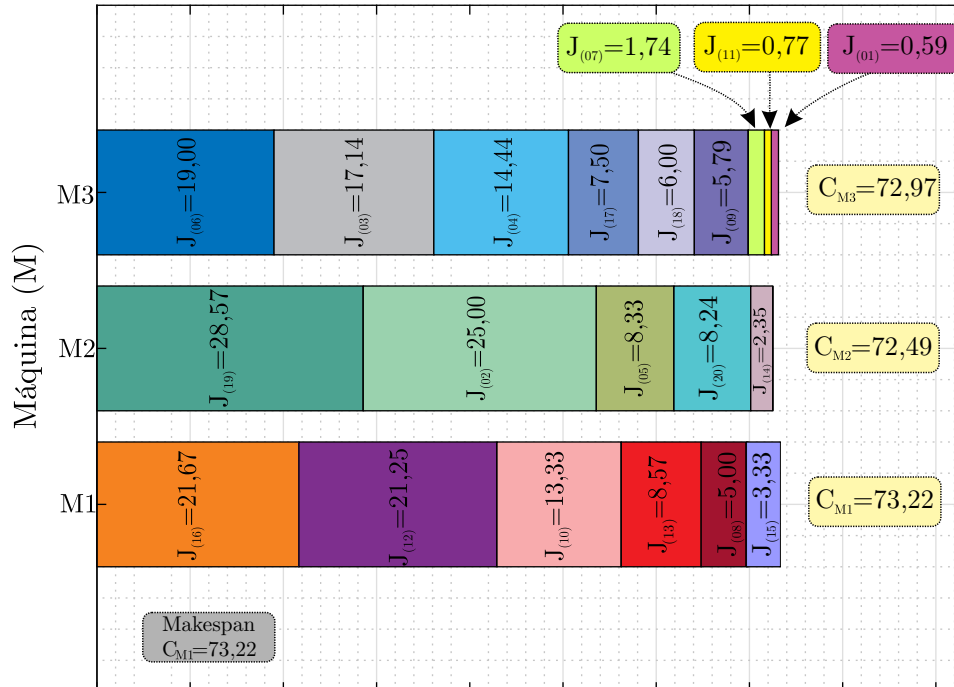
end

até *o processo de busca não encontre mais nenhuma possível candidata para dupla troca.*;

fim

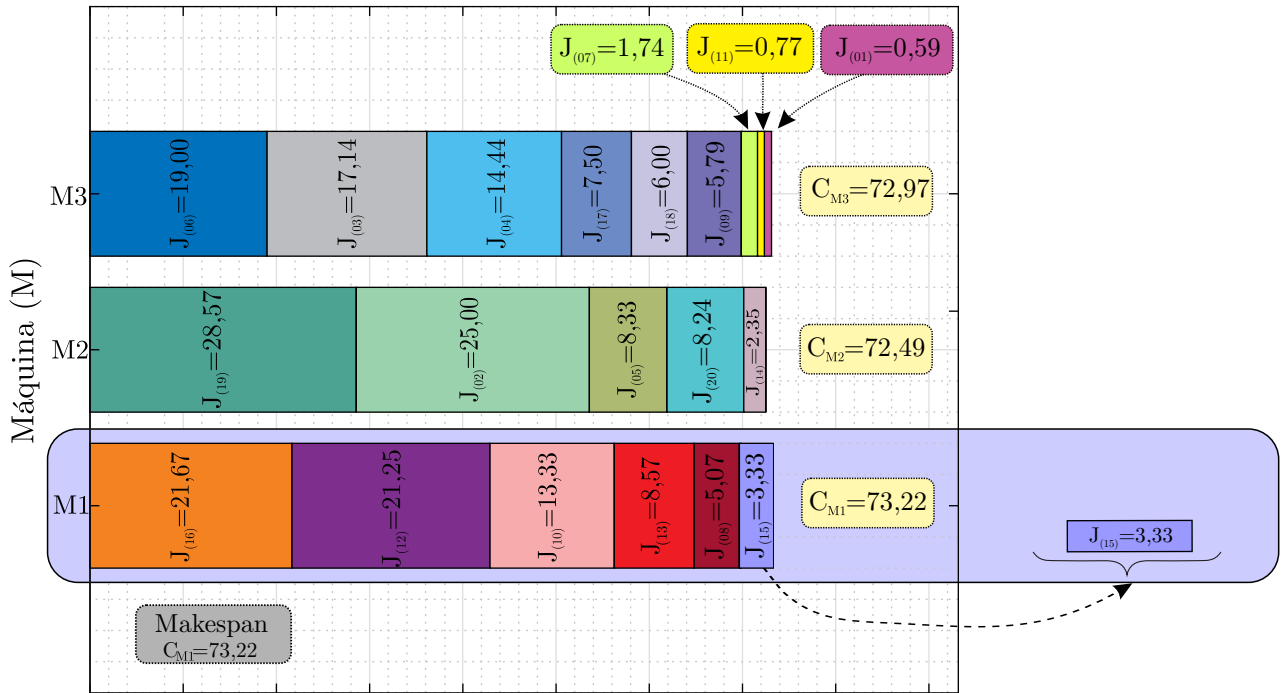
Para uma melhor compreensão da Fase 3, a Figura 4.1 apresenta a solução obtida da heurística proposta LPTteste no final da fase de alocação inicial.

Figura 4.1 – Solução Inicial da Fase 3 de Duplas Trocas.



Fonte: Autora, 2021.

Na solução inicial tem-se M_1 para a máquina mais carregada e M_m para as máquinas menos carregadas, neste caso M_2 e M_3 . A Figura 4.2 apresenta a máquina mais carregada M_1 e a tarefa com menor valor de p_j que será transferida para as demais máquinas.

Figura 4.2 – Máquina mais carregada e menor valor de p_j para sair da máquina.

Fonte: Autora, 2021.

Conforme demonstrado na Figura 4.2, a tarefa de menor valor em M_1 é J_{15} $\{p_{jM_1} = 3,33\}$. Será testada a transferência dessa tarefa para as demais máquinas, as mudanças nas máquinas envolvidas são novamente influenciadas pela velocidade da máquina e pelas duas tarefas, a que entra e a que sai da máquina. Importante observar que se tratam de máquinas paralelas não-relacionadas, a velocidade de cada máquina depende das tarefas executadas, podendo ser diferente para cada máquina.

Assim, considerando a máquina mais carregada M_1 , faz-se o cálculo para identificar o valor que a tarefa receberá em cada M_m . O valor da tarefa de M_1 que irá sair é determinada por:

$$p_{jM_1} = \frac{p_j}{s_{i,j}} \quad (4.1)$$

Após a identificação da tarefa a sair, o valor de C_{M_1} é dada por:

$$C_{M_1} = C_{M_1} - p_{jM_1} \quad (4.2)$$

Para encontrar o valor da tarefa que irá entrar nas máquinas, o cálculo é dado da seguinte maneira:

$$p_{jM_m} = \frac{p_j}{s_{i,j}} \quad (4.3)$$

Logo, ao encontrar o novo valor da tarefa que irá entrar e sair nas máquinas menos carregadas, o valor de C_i é determinado por:

Para entrar em M_m :

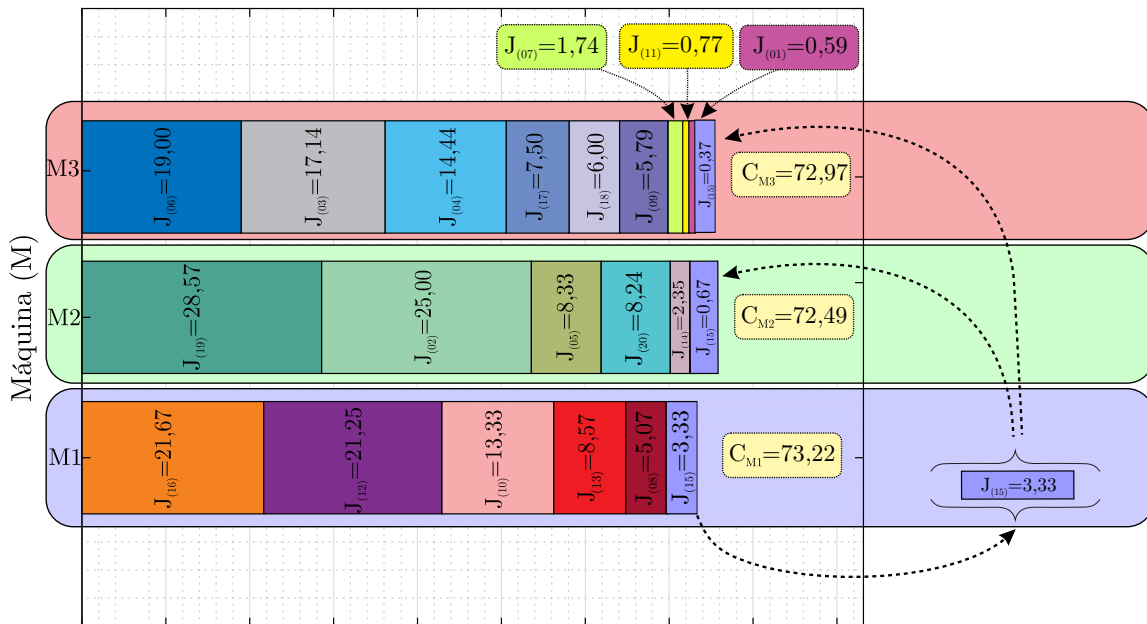
$$C_{Mm} = C_{Mm} + p_{jMm} \quad (4.4)$$

Para sair de M_m :

$$C_{Mm} = C_{Mm} - p_{jMm} \quad (4.5)$$

Ao encontrar os novos valores dos tempos de processamento para fazer a dupla troca, faz-se um teste antes da escolha da tarefa que será transferida (exceto na máquina mais carregada), utiliza-se o mesmo teste que foi aplicado na fase de alocação inicial. A Figura 4.3 apresenta os novos valores de tempos de processamento para alocação nas máquinas menos carregadas, e a saída da menor tarefa de M_1 e entrada das tarefas nas máquinas menos carregadas.

Figura 4.3 – Saída de M_1 e entrada das tarefas em M_2 e M_3 .



Fonte: Autora, 2021.

Para dar continuidade ao sequenciamento das tarefas, o próximo passo é escolher nas máquinas que receberam a tarefa de M_1 , as tarefas cuja troca possibilite um valor de *makespan* menor que o valor inicial, esse teste também tem como finalidade verificar os valores de C_i atualizados. Teste esse determinado por:

$$C_{testeM_1} = C_{inicialM_1} - Tar_{troca} = 69,89 \quad (4.6)$$

$$C_{testeM_2} = C_{inicialM_2} + Tar_{troca} = 73,16 \quad (4.7)$$

$$C_{teste}M_3 = C_{inicial}M_3 + Tar_{troca} = 73,34 \tag{4.8}$$

Após o teste, realiza-se a escolha das possíveis candidatas a serem transferidas das máquinas menos carregadas, através de:

$$(C_{inicial}M_2 + Tar_{troca}) - C_{inicial}M_1 \tag{4.9}$$

$$(72,49 + 0,67) - 73,22 = -0,06$$

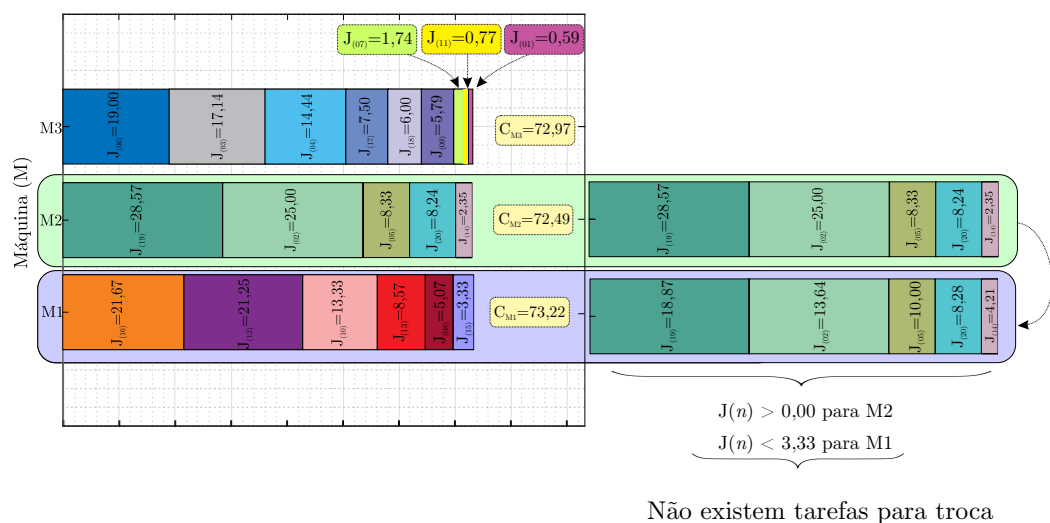
$$(C_{inicial}M_3 + Tar_{troca}) - C_{inicial}M_1 \tag{4.10}$$

$$(72,97 + 0,37) - 73,22 = 0,12$$

Para a tarefa sair de M_2 , é necessário que o valor de p_j seja maior que 0 e menor que o valor de p_j que saiu de M_1 $J_{15} \{p_{jM_1} = 3,33\}$, para entrar em M_1 . Para a máquina M_3 acontece o mesmo critério, a tarefa a sair da máquina precisar ter valor de p_j maior que 0,12 e valor de p_j menor que a tarefa que saiu de M_1 $J_{15} \{p_{jM_1} = 3,33\}$ para entrar em M_1 . Importante salientar que em casos que o resultado do teste apresentar valores negativos, conforme demonstrou a equação 4.6, o valor estipulado é 0.

Na Figura 4.4, todas as tarefas são retiradas da máquina M_2 para verificar a possibilidade de troca, porém, como trata-se de máquinas não-relacionadas, os valores de p_j mudam em cada máquina e ao aplicar o critério da equação 4.6, constata-se que não existem tarefas para troca.

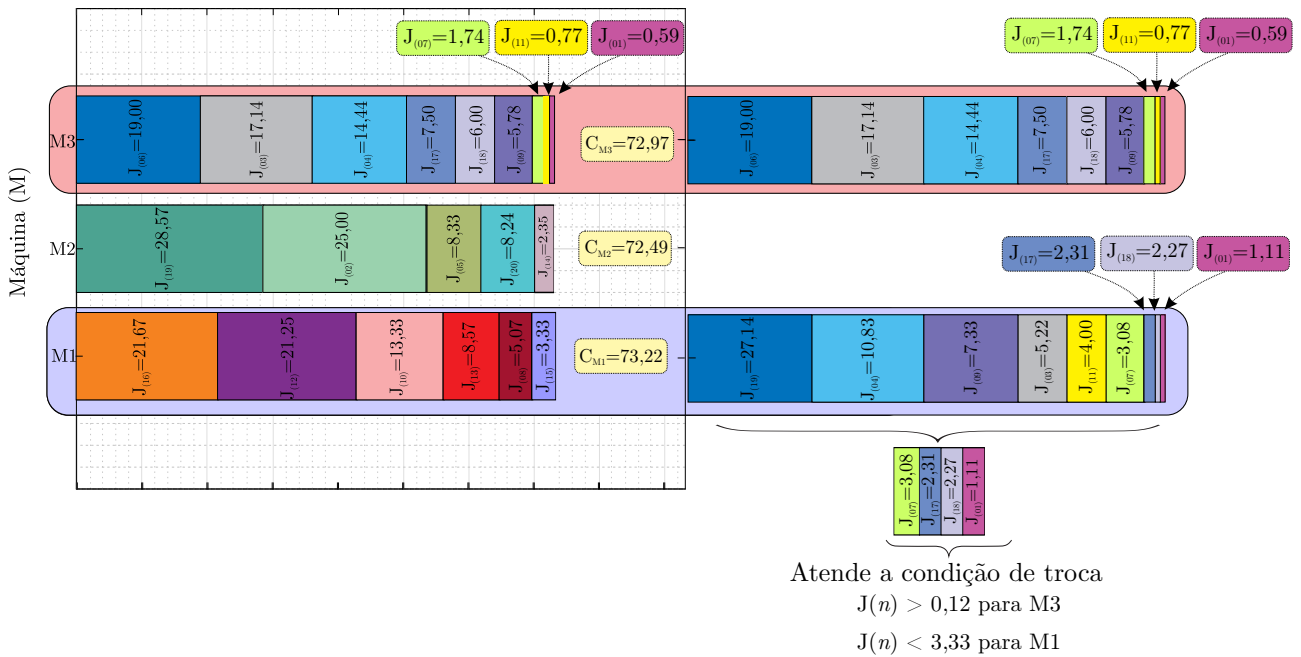
Figura 4.4 – Candidatas a serem escolhidas para transferência de M_2 para M_1 .



Ao observar a Figura 4.4, constata-se que não houve nenhuma possibilidade de troca da máquina M_2 para a máquina M_1 , neste primeiro processo de escolha das possíveis candidatas. Entretanto, o procedimento continua em busca de alguma possibilidade, agora na escolha das possíveis candidatas da máquina M_3 para a máquina M_1 .

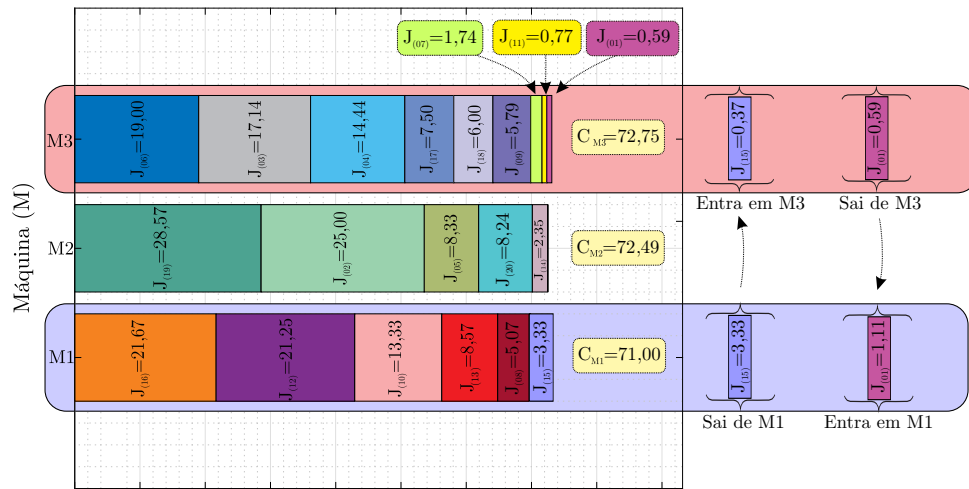
Na Figura 4.5, todas as tarefas de M_3 são retiradas para verificar as possíveis candidatas para troca, e aplicar a equação 4.7, encontrou-se quatro tarefas que atendem a condição para troca.

Figura 4.5 – Candidatas a serem escolhidas para transferência de M_3 para M_1 .



Fonte: Autora, 2021.

Qual das tarefas candidatas escolhe-se para fazer a troca? Opta-se pela aplicação de um teste, descrito a seguir, para encontrar as possíveis candidatas que possibilitem alguma dupla troca. Este método apresentado pode ocorrer quando há um número pequeno de tarefas para alocar e encontrar um novo valor do *makespan*. Observa-se na Figura 4.6, o comportamento do primeiro teste entre a tarefa que sai e entra na máquina mais carregada M_1 e na máquina menos carregada M_3 .

Figura 4.6 – Primeira possibilidade de troca entre M_1 e M_3 .

Fonte: Autora, 2021.

Na primeira possibilidade de troca, é necessário calcular o valor de C_i das máquinas M_1 e M_3 para no final, depois dos testes, verificar se o valor do *makespan* permitiu a minimização no tempo de finalização da máquina mais carregada e identificar qual tarefa candidata será transferida. O cálculo para a máquina M_1 é determinado por:

$$C_{M1} = (C_{inicial}M1 - j_{15}M1) \quad (4.11)$$

$$C_{M1} = (73,22 - 3,33) = 69,89$$

$$(69,89 + J_{01}M1 \{p_j = 1,11\}) = 71,00$$

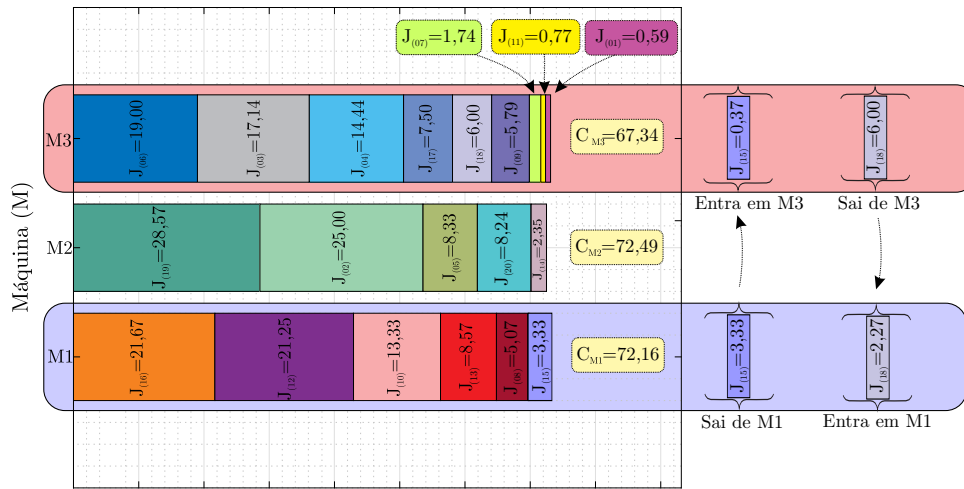
O cálculo para a máquina M_3 , no primeiro teste, é dado por:

$$C_{M3} = (C_{inicial}M3 - j_{01}M3)$$

$$C_{M3} = (72,97 - 0,59) = 72,38$$

$$(72,38 + J_{15}M1 \{p_j = 0,37\}) = 72,75$$

A Figura 4.7, apresenta a segunda possibilidade de troca entre as máquinas M_1 e M_3 , e outros novos valores de C_i .

Figura 4.7 – Segunda possibilidade de troca entre M_1 e M_3 .

Fonte: Autora, 2021.

Na segunda possibilidade de trocas entre M_1 e M_3 , busca-se verificar se este teste apresenta um valor de *makespan* menor que a primeira possibilidade. Assim, os novos valores de C_i para esta possibilidade é determinada por:

$$C_{M1} = (C_{inicial}M1 - j_{15}M1)$$

$$C_{M1} = (73,22 - 3,33) = 69,89$$

$$(69,89 + J_{18}M1 \{p_j = 2,27\}) = 72,16$$

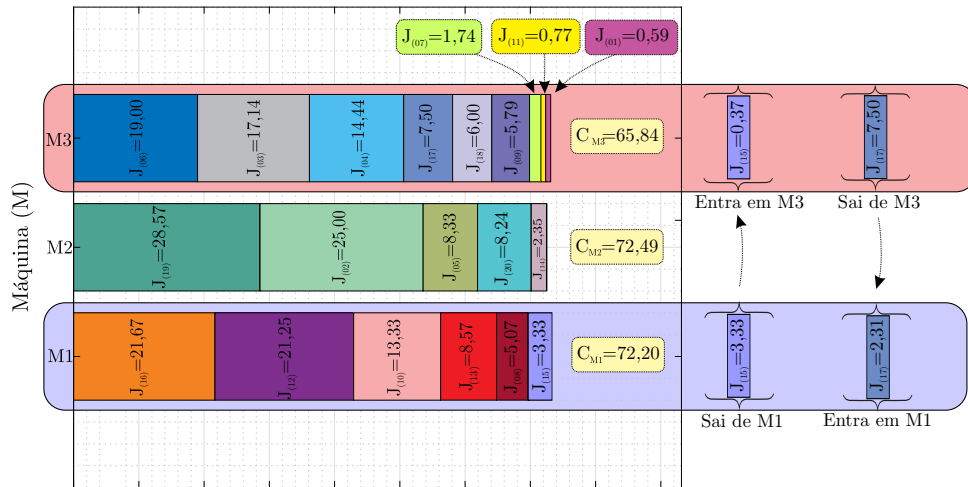
Este segundo teste demonstra o valor de C_{M3} , nesta possibilidade de troca:

$$C_{M3} = (C_{inicial}M3 - j_{18}M3)$$

$$C_{M3} = (72,97 - 6,00) = 66,97$$

$$(66,97 + J_{15}M3 \{p_j = 0,37\}) = 67,34$$

Os testes vão sendo sequenciados até que encontre-se a tarefa candidata a ser escolhida para troca. A terceira possibilidade é apresentada na Figura 4.8.

Figura 4.8 – Terceira possibilidade de troca entre M_1 e M_3 .

Fonte: Autora, 2021.

O cálculo para a máquina M_1 na terceira possibilidade de troca é dada por:

$$C_{M1} = (C_{inicial}M1 - j_{15}M1)$$

$$C_{M1} = (73,22 - 3,33) = 69,89$$

$$(69,89 + J_{17}M1 \{p_j = 2,31\}) = 72,20$$

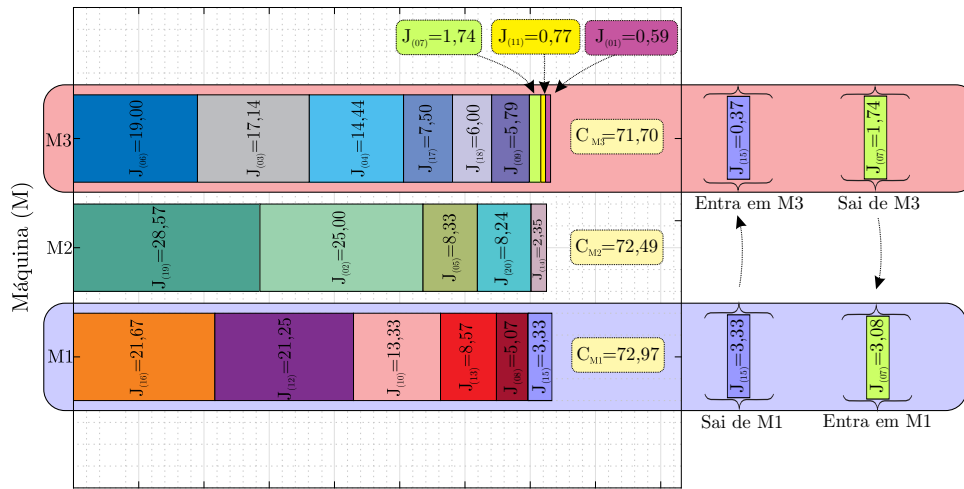
Na máquina M_3 os novos valores são determinados por:

$$C_{M3} = (C_{inicial}M3 - j_{17}M3)$$

$$C_{M3} = (72,97 - 7,50) = 65,47$$

$$(65,47 + J_{15}M3 \{p_j = 0,37\}) = 65,84$$

Na Figura 4.9, demonstra-se o quarto teste para possibilidade de troca e como as tarefas candidatas comportaram-se neste caso.

Figura 4.9 – Quarta possível candidata para troca entre M_1 e M_3 

Fonte: Autora, 2021.

Como apresentado nos testes anteriores, os cálculos para esta quarta possibilidade de troca em M_1 é determinado por:

$$C_{M1} = (C_{inicial}M1 - j_{15}M1)$$

$$C_{M1} = (73,22 - 3,33) = 69,89$$

$$(69,89 + J_{07}M1 \{p_j = 3,08\}) = 72,97$$

E na máquina M_3 , os cálculos apresentam as seguinte possibilidade:

$$C_{M3} = (C_{inicial}M3 - j_{07}M3)$$

$$C_{M3} = (72,97 - 1,74) = 71,23$$

$$(71,23 + J_{15}M3 \{p_j = 0,37\}) = 71,60$$

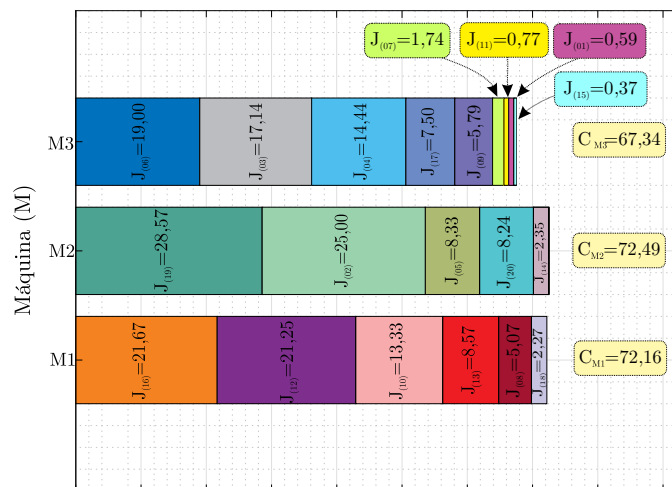
A Tabela 4.1, apresenta o resultado dos valores de C_i nas máquinas M_1 e M_3 com os novos valores dos testes e o valor do *makespan*.

Tabela 4.1 – Apresentação das possibilidades de troca.

<i>Possibilidades</i>	M1	M3	<i>Makespan</i>
1ª Possibilidade	71,00	72,45	72,45
2ª Possibilidade	72,16	67,34	72,16
3ª Possibilidade	72,20	65,84	72,20
4ª Possibilidade	72,97	71,60	72,97

Fonte: Autora, 2021.

No processo de verificação das possibilidades de trocas viáveis para melhoria na qualidade da solução, a segunda possibilidade apresentou um valor de *makespan* menor em relação ao valor inicial da fase. A Figura 4.10 apresenta o diagrama de *Gantt* com os novos valores de C_i .

Figura 4.10 – Solução se a escolha fosse a segunda troca entre M_1 e M_3 

Fonte: Autora, 2021.

Ao observar a Figura 4.10, constatou-se que ainda existem tarefas para fazer dupla troca, assim, o processo continua até que o algoritmo não encontre mais nenhuma possibilidade de duplas trocas. As próximas possibilidades serão apresentadas em formas de tabelas, no entanto, seguem o mesmo critério aplicado nas primeiras figuras explicativas deste capítulo.

Para encontrar novas possibilidade de duplas trocas, o processo continua dos valores de C_i , demonstrados na Figura 4.10. A Tabela 4.2, apresenta os valores da solução inicial e os valores das três máquinas após aplicado o teste para encontrar as possíveis candidatas a serem escolhidas.

Tabela 4.2 – Continuidade para encontrar mais possibilidades de troca.

M	C_i	p_j	Valores com teste
M_1	72,16	4,21	76,21
M_2	72,49	2,35	70,14
M_3	67,34	2,86	70,20

Fonte: Autora, 2021.

Os valores dos testes seguem o mesmo critério das equações 4.6, 4.7 e 4.8, porém, é importante observar qual a máquina mais carregada, para desta diminuir a tarefa com menor valor de tempo de processamento que está sendo retirada e quais as máquinas menos carregadas, para receber esta tarefa somando ao valor da solução inicial em sua respectiva máquina.

Após os testes calcula-se os valores mínimos das tarefas a saírem das máquinas M_1 e M_3 , conforme as equações 4.9 e 4.10. Para este caso, as tarefas precisam ter valores maiores que 3,72 em M_1 e -2,29 em M_3 . Quando o valor apresenta negativo, este é representado por 0, ou seja, na máquina M_3 as tarefas candidatas a sair, serão todas maiores que 0 e para entrar em M_2 nenhuma tarefa poderá ser maior que o valor da tarefa que saiu de M_2 $J_{14} \{p_{jM_2} = 2, 35\}$.

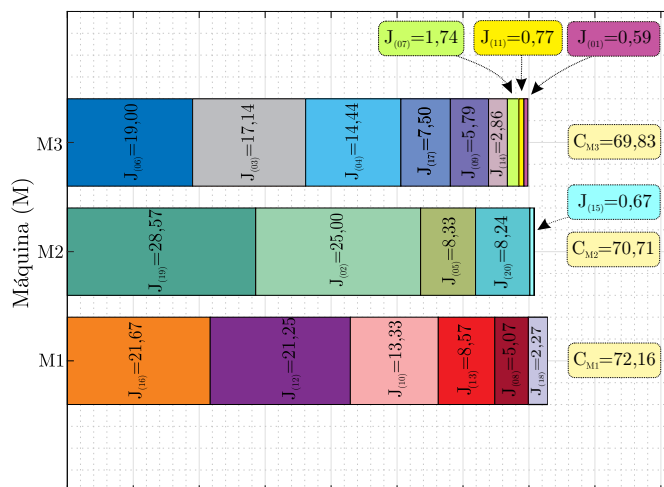
A Tabela 4.3, apresenta o próximo resultado de possibilidades de troca entre M_1 e M_2 , e M_2 e M_3 , em destaque está a possibilidade que apresentou uma dupla troca.

Tabela 4.3 – Exibição das possibilidades de troca entre M_1 e M_2 e entre M_1 e M_3 .

Possibilidades	M1	M2	M3	<i>Makespan</i>
6ª Possibilidade	54,54	72,05	-	72,05
7ª Possibilidade	-	70,81	69,83	70,81
8ª Possibilidade	-	70,97	69,61	70,97
9ª Possibilidade	-	71,68	69,43	71,68
10ª Possibilidade	-	71,62	68,46	71,62
11ª Possibilidade	-	72,45	62,70	72,45

Fonte: Autora, 2021.

A Figura 4.11, demonstra como as tarefas ficaram organizadas após este processo e os novos valores de C_i das três máquinas.

Figura 4.11 – Novos valores de C_i nas três máquinas.

Fonte: Autora, 2021.

Como o algoritmo apresentou uma dupla troca entre M_2 e M_3 , o processo segue em busca de melhores valores de *makespan* em relação ao valor inicial da fase.

O processo continua com os novos valores das três máquinas apresentados na Tabela 4.4.

Tabela 4.4 – Continuidade para encontrar mais possibilidades de troca.

M	C_i	p_j	Valores com teste
M_1	72,16	2,27	69,89
M_2	70,81	0,71	71,52
M_3	69,83	6,00	75,83

Fonte: Autora, 2021.

Para encontrar as novas possibilidades, aplica-se novamente o teste nas três máquinas, ou seja, identifica-se a máquina mais carregada, que neste caso é $M_1 = 72,16$, e as máquinas menos carregadas $M_2 = 70,81$ e $M_3 = 69,83$. Escolhe-se para sair de M_1 a tarefa $J_{18} \{p_{jM_1} = 2,27\}$ com menor tempo de processamento, e testa-se esta tarefa em M_2 $70,81 + J_{18} \{p_{jM_2} = 0,71\} = 71,52$, e em M_3 $69,83 + J_{18} \{p_{jM_3} = 6,00\} = 75,83$.

É necessário aplicar esse teste mencionado anteriormente, pois, é através dele que encontra-se o valor máximo da tarefa candidata que pode entrar em M_1 e os valores mínimos das possíveis candidatas a saírem de M_2 e M_3 , conforme demonstra a Tabela 4.5.

Tabela 4.5 – Valor máximo e valores mínimos para escolha das possíveis candidatas.

M	Valores para encontrar possibilidades de troca
M_1	$72,16 - 69,89 = 2,27$ (max)
M_2	$71,52 - 72,16 = -0,64$ (min)
M_3	$75,83 - 72,16 = 3,67$ (min)

Fonte: Autora, 2021.

Para a tarefa sair de M_2 as possíveis candidatas precisam ser maiores que -0,64, ou seja, maiores que 0 quando apresentar valores negativos, e não podem ultrapassar o valor de 2,27 para entrar em M_1 . Neste caso a Tabela 4.6 demonstra os valores de M_2 e os valores das mesmas tarefas em M_1 , pois, como tratam-se de máquinas paralelas não-relacionadas, os valores de p_j alteram-se de máquina para máquina.

Tabela 4.6 – Possíveis candidatas de M_2 para M_1 .

M	J_{15}	J_{20}	J_{05}	J_{02}	J_{19}
M_1	3,33	8,28	10,00	13,64	18,87
M_2	0,67	8,24	8,33	25,00	28,57

Fonte: Autora, 2021.

Após a verificação das alterações dos valores dos tempos de processamento de uma máquina para outra, constatou-se que, de acordo com as condições determinadas, não existe nenhuma possibilidade para troca entre M_1 e M_2 , assim, o processo de busca passa a ser na máquina M_3 .

Para encontrar as possíveis tarefas candidatas a saírem de M_3 , estas precisam ser maiores que 3,67, conforme Tabela 4.5, e para entrarem em M_1 , não podem ultrapassar o valor de 2,27. A Tabela 4.7 apresenta os valores dessa busca para possíveis possibilidades de troca.

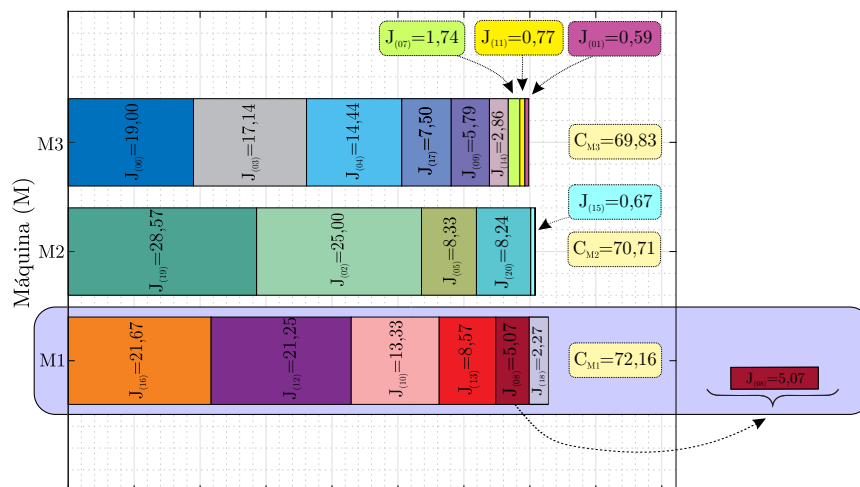
Tabela 4.7 – Possíveis candidatas de M_3 para M_1 .

M	J_{09}	J_{17}	J_{04}	J_{05}	J_{06}
M_1	7,33	2,31	10,83	5,22	27,14
M_3	5,79	7,50	14,44	17,14	19,00

Fonte: Autora, 2021.

Para as condições determinadas, e conforme a Tabela 4.7, notou-se que a máquina M_3 também não apresentou nenhuma possibilidade de troca.

Nesse caso, como nenhuma máquina apresentou possibilidade de troca para a primeira tarefa J_{18} $\{p_{jM_1} = 2,27\}$, com menor tempo de processamento, como demonstrou em M_1 , esta é eliminada da possibilidade da troca, o código escolhe a segunda menor tarefa J_{08} $\{p_{jM_1} = 5,07\}$ como a Figura 4.12 apresenta, e refaz o processo dos cálculos apresentado anteriormente.

Figura 4.12 – Novos valores de C_i nas três máquinas.

Fonte: Autora, 2021.

O processo continua com os valores de C_i demonstrados na Figura 4.11. A Tabela 4.8, mostra os valores de C_i e a aplicação do teste para encontrar o valor máximo e os valores mínimos das possíveis tarefas candidatas para troca., ressaltando que a máquina M_1 é a mais carregada e as máquinas M_2 e M_3 são as máquinas menos carregadas.

Tabela 4.8 – Valor máximo e valores mínimos para escolha das possíveis candidatas.

M	Valores para encontrar possibilidades de troca
M_1	$72,16 - 67,09 = 5,07$ (max)
M_2	$73,73 - 72,16 = 1,57$ (min)
M_3	$71,50 - 72,16 = -0,66$ (min)

Fonte: Autora, 2021.

As possíveis tarefas candidatas a saírem de M_2 precisam ser maiores que 1,57 e para entrarem em M_1 , não podem ultrapassar o valor de 5,07, de acordo com Tabela 4.8. Nesse caso, a Tabela 4.9 apresenta os valores dos tempos de processamento em M_2 e como ficam em relação a máquina M_1 .

Tabela 4.9 – Possíveis candidatas de M_2 para M_1 .

M	J_{20}	J_{05}	J_{02}	J_{19}
M_1	8,28	10,00	13,64	18,87
M_2	8,24	8,33	25,00	28,57

Fonte: Autora, 2021.

Constatou-se que de acordo com as condições determinadas, a máquina M_2 não possui nenhuma possibilidade de troca com a máquina M_1 . A busca por alguma possibilidade de troca segue na máquina M_3 , e para que seja possível as possíveis tarefas candidatas para saírem de M_3 precisam ser maiores que -0,66, ou melhor, maiores que 0 e para entrarem em M_1 não podem ter valor maior que 5,07. A Tabela 4.10 apresenta os valores das tarefas em M_3 e como seus valores ficam em relação a máquina M_1 .

Tabela 4.10 – Possíveis candidatas de M_3 para M_1 .

M	J_{01}	J_{11}	J_{07}	J_{09}	J_{14}	J_{17}	J_{04}	J_{03}	J_{06}
M_1	1,11	4,00	3,08	7,33	4,21	2,31	10,83	5,22	27,14
M_3	0,59	0,77	1,74	5,79	2,86	7,50	14,44	17,14	19,00

Fonte: Autora, 2021.

Ao fazer a verificação na alteração dos valores dos tempos de processamento da máquina M_3 para a máquina M_1 , notou-se que apresentaram cinco possibilidades de trocas, agora é necessário encontrar qual destas possibilidades será a troca mais viável para que o valor do *makespan* atual seja menor que o valor do *makespan* inicial.

Após os testes aplicados como demonstra no processo das figuras iniciais desta fase, a Tabela 4.11 apresenta o resultado das possibilidades de trocas entre M_1 e M_3 , em destaque encontra-se a possibilidade que apresentou uma dupla troca.

Tabela 4.11 – Apresentação das possibilidades de troca M_1 e M_3 .

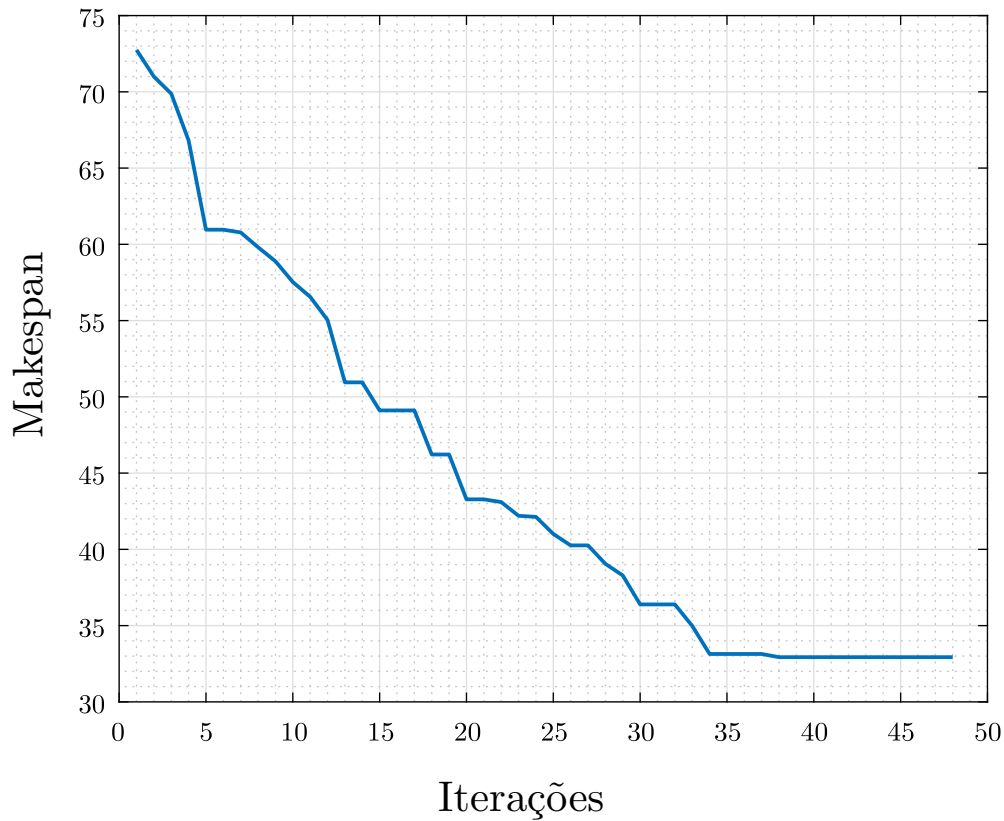
<i>Possibilidades</i>	M1	M3	<i>Makespan</i>
12^a Possibilidade	68,20	70,91	70,91
13^a Possibilidade	71,09	70,93	71,09
14^a Possibilidade	70,17	69,76	70,17
15^a Possibilidade	71,30	57,06	71,30
16^a Possibilidade	69,40	64,00	69,40

Fonte: Autora, 2021.

A 16^o possibilidade apresentou maior redução de *makespan* em relação ao valor do *makespan* inicial = 72,16. No entanto, o processo de busca, por mais possibilidades de trocas, continua até que o algoritmo não encontre mais nenhuma possível candidata.

O gráfico a seguir, demonstra o comportamento do algoritmo na busca por duplas trocas e como o *makespan* reduziu com o método aplicado.

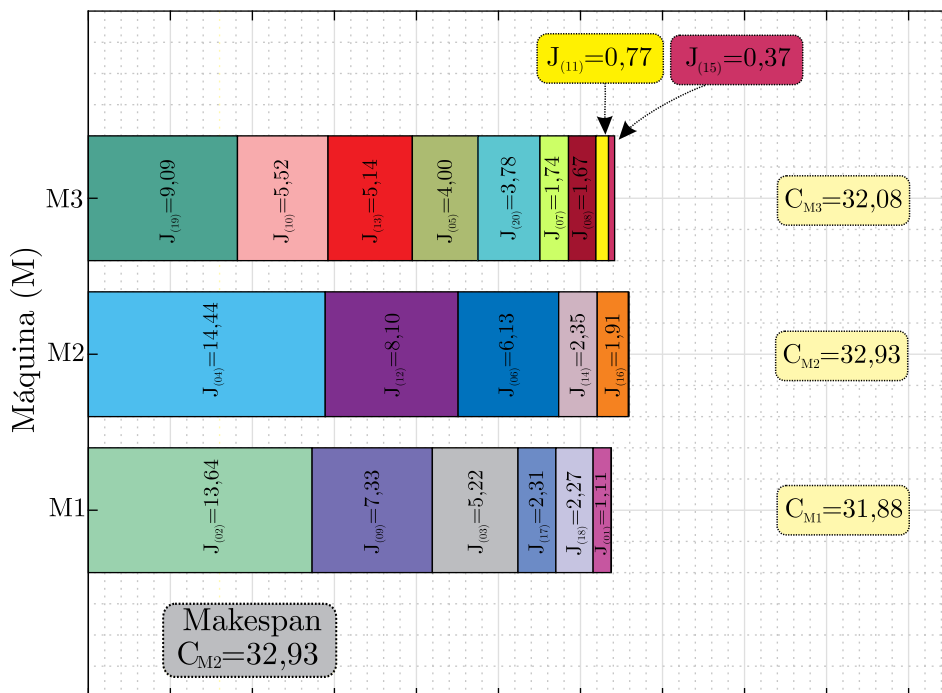
Figura 4.13 – Comportamento do algoritmo na busca por duplas trocas.



Fonte: Autora, 2021.

Após 38 iterações o algoritmo não encontrou mais nenhuma possibilidade de duplas trocas entre a máquina mais carregada e as demais máquinas, para este caso, partindo então para a próxima fase de triplas trocas. A Figura 4.14, apresenta os valores de C_i das máquinas ao final da busca por possibilidades de trocas nesta fase e a organização das tarefas em suas respectivas máquinas, também como o valor do *makespan* atual em relação ao valor do *makespan* inicial.

Figura 4.14 – Final das buscas por possibilidades de trocas para este caso.

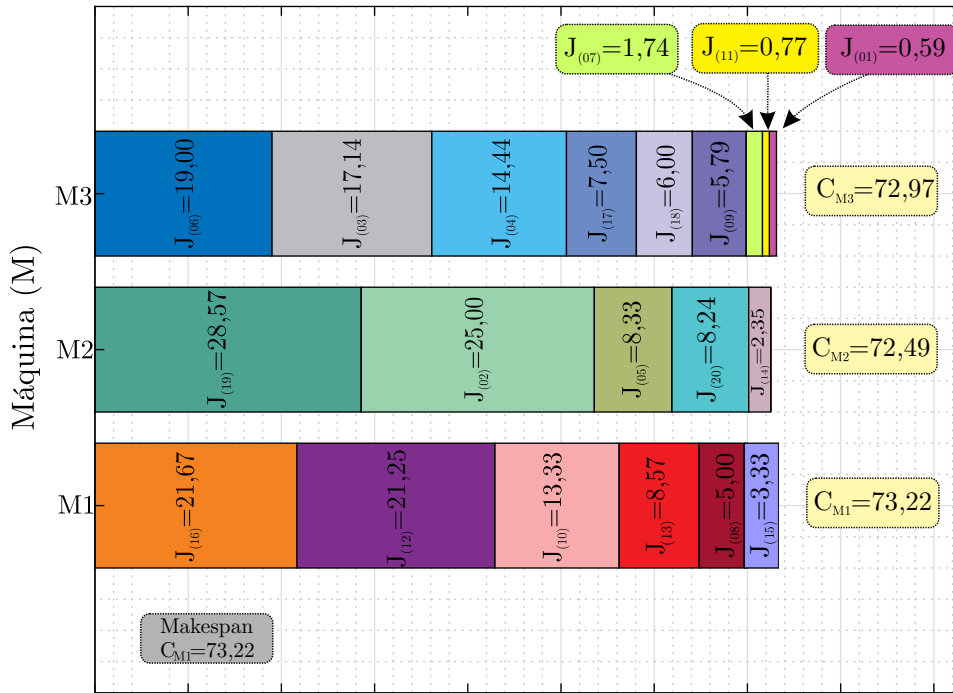


Fonte: Autora, 2021.

Quando existir um número grande de tarefas e possibilidades para trocas, faz-se necessário adotar um critério para encontrar as possíveis candidatas. No presente estudo, é proposto um método denominado alvo, não necessariamente é o melhor alvo, mas um valor de referência. Este alvo tem a finalidade de, em uma faixa de tarefas de possíveis candidatas, escolher qual será a melhor opção para troca.

Para melhor compreensão do método aplicado, utilizou-se como exemplo na Figura 4.15, os valores da Figura 4.1.

Figura 4.15 – Exemplo utilizado para busca de um alvo.



Fonte: Autora, 2021.

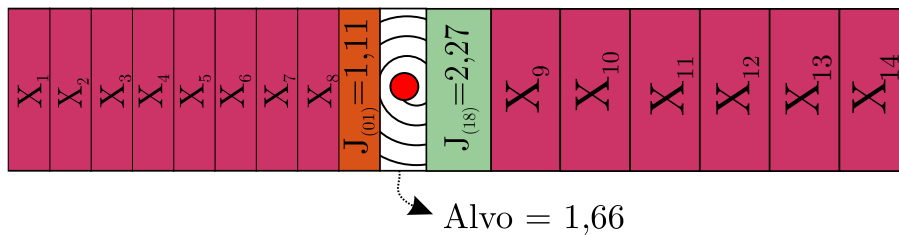
O alvo é determinado por: tarefa com menor tempo de processamento que saí da máquina mais carregada $J_{15} \{p_{jM1} = 3,33\}$, dividido pelo número de máquinas disponíveis para os testes, que neste caso são duas.

$$Alvo = \frac{J_{15} \{p_{jM1} = 3,33\}}{2} \tag{4.12}$$

$$Alvo = \frac{3,33}{2} = 1,66$$

Na Figura 4.16, demonstra-se o alvo e sua posição após a equação 4.12, quando o número de tarefas e possibilidades forem grandes.

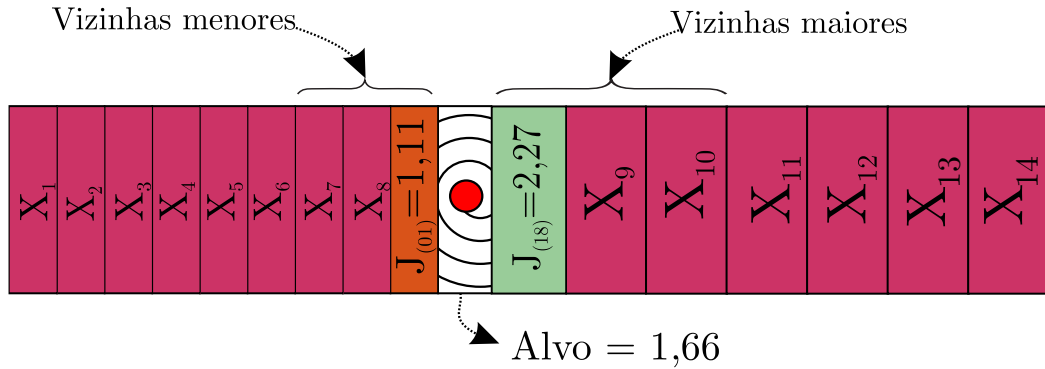
Figura 4.16 – Escolha das vizinhanças com um alvo para números grandes de tarefas.



Fonte: Autora, 2021.

Após determinar o valor de p_j do alvo, escolhe-se as tarefas que estão na vizinhança do alvo, se determinar $n=1$, importante salientar que, tais valores de vizinhanças são todas as tarefas com tempos de processamento determinadas para esta máquina em específico, conforme demonstra a Figura 4.17.

Figura 4.17 – Escolha de um alvo.



Fonte: Autora, 2021.

Definido o alvo para este caso, $p_j = 1,66$, as tarefas candidatas vizinhas J_{01} $\{p_{jM1} = 1, 11\}$ e J_{18} $\{p_{jM1} = 2, 27\}$, passam novamente pelo processo de teste, para verificar qual tarefa será escolhida e qual minimizará o valor do *makespan* inicial. Dessa forma, o primeiro teste para a troca na máquina M_1 neste passo é determinado por:

$$C_{M1} = (C_{inicialM1} - j_{15}M1)$$

$$C_{M1} = (73, 22 - 3, 33) = 69, 89$$

$$(69, 89 + J_{01}M1 \{p_j = 1, 11\}) = 71, 00$$

E para a máquina M_3 , é dado por:

$$C_{M3} = (C_{inicialM3} - j_{01}M3)$$

$$C_{M3} = (72, 97 - 0, 59) = 72, 38$$

$$(72, 38 + J_{15}M1 \{p_j = 3, 33\}) = 75, 71$$

O segundo teste aplicado para a tarefa vizinha J_{18} $\{p_{jM1} = 2, 27\}$ na máquina M_1 , é estabelecido por:

$$C_{M1} = (C_{inicial}M1 - j_{15}M1)$$

$$C_{M1} = (73,22 - 3,33) = 69,89$$

$$(69,89 + J_{18}M1 \{p_j = 2, 27\}) = 72,16$$

E na máquina M_3 , é determinada por:

$$C_{M3} = (C_{inicial}M3 - j_{18}M3)$$

$$C_{M3} = (72,97 - 6,00) = 66,97$$

$$(66,97 + J_{15}M3 \{p_j = 0, 37\}) = 67,34$$

Observa-se na Tabela 4.12, o comportamento das tarefas vizinhas na escolha da candidata para troca com o alvo, e entre as duas tarefas qual proporcionou a diminuição no valor do *makespan* inicial = 73,22.

Tabela 4.12 – Vizinhas candidatas para troca.

Vizinhanças	M1	M3	<i>Makespan</i>
1ª Vizinha	71,00	75,38	75,38
2ª Vizinha	72,16	67,34	72,16

Fonte: Autora, 2021.

Com a aplicação do método denominado como alvo, o teste demonstrou que a segunda vizinha possui possibilidade de troca, apresentando uma redução no *makespan* inicial, diminuindo de 73,22 para 72,16. Importante ressaltar que o alvo, somente é possível ser aplicado para números grandes de tarefas a serem sequenciadas.

Esta fase de duplas trocas demonstrou alguns critérios que foram determinados para que houvessem possibilidades de trocas entre a máquina mais carregada e uma máquina menos carregada, com o objetivo de minimizar o valor do *makespan* inicial. Como, para este caso, após 38 iterações, o valor do *makespan* obteve redução e não foi mais possível realizar duplas trocas, o presente estudo passa para a próxima etapa, denominada como Fase de Triplas Trocas para máquinas paralelas não-relacionadas em multi-fábricas.

5 FASE DE TRIPLAS TROCAS PARA *R/MULTIFACT/CMAX*

A Fase 4, conceituada como Fase de Triplas Trocas, será a última fase apresentada neste estudo, adaptada para máquinas paralelas não-relacionadas em multi-fábricas. Esta fase assemelha-se ao método adotado na Fase 3, porém, utiliza-se sequências de três realocações, em que uma tarefa é transferida da máquina mais carregada para outra máquina e tem desta duas tarefas com valores de p_j menores, cuja transferência permita um valor de *makespan* atual menor que valor o inicial, possibilitando alcançar a minimização no tempo de execução da máquina mais carregada.

Apresenta-se a seguir, mais detalhadamente, o algoritmo 6 da fase de triplas trocas, um passo a passo, de como o método proposto como teste é aplicado antes da alocação das tarefas.

Algoritmo 6: Algoritmo da Fase de Triplas Trocas

Entrada de dados;

início

inicialização - Triplas Trocas;

repita

Posicionar as tarefas (p_j) nas M máquinas;

Identificar a máquina mais carregada (M_1) e a tarefa com menor valor de p_j ;

Testar a tarefa com menor valor de p_j nas demais máquinas M_m ;

Identificar o valor máximo a entrar em M_1 e os valores mínimos a saírem de M_m ;

Verificar se alguma soma das combinações apresentou possibilidade de tripla troca;

if *houver combinação possível para troca* **then**

| faz-se o teste para verificar qual combinação será escolhida;

else

| faz-se o processo em outra máquina

end

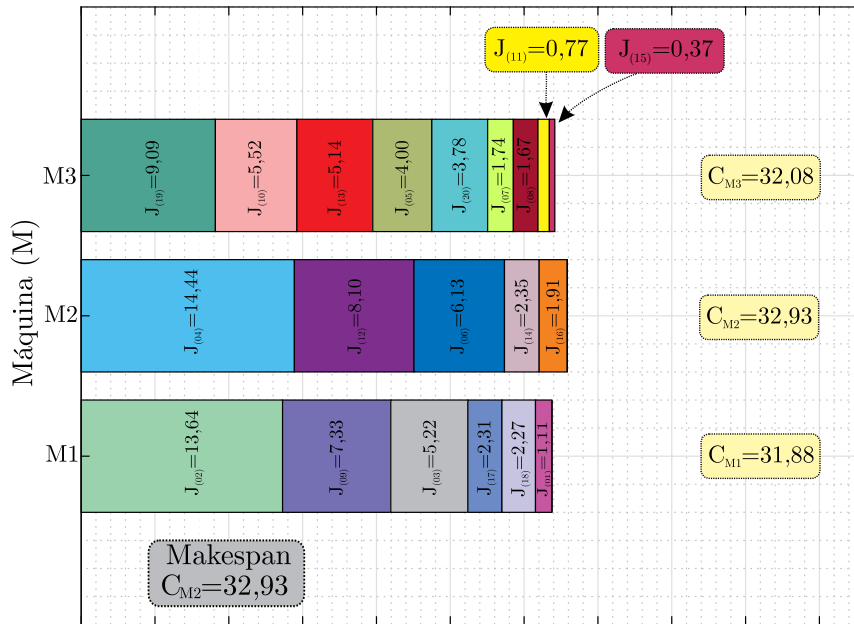
Verificar qual a combinação que obteve o $<$ valor de *makespan* atual em relação ou inicial;

até que não existam mais combinações possíveis para tripla troca;

fim

A Figura 5.1, demonstra o diagrama de *Gantt* da solução final da fase 3, e que será utilizado na fase 4 para melhor compreensão do problema.

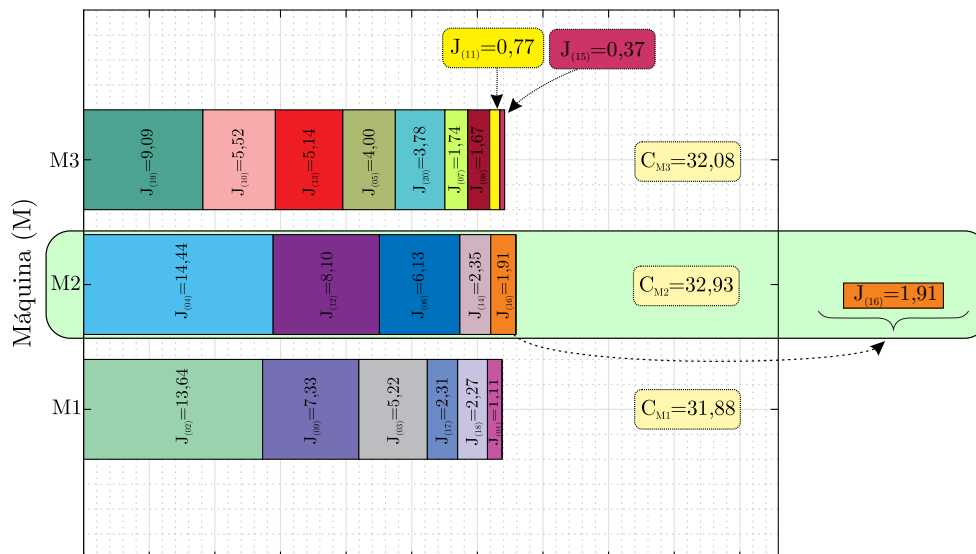
Figura 5.1 – Início da fase de triplas trocas.



Fonte: Autora, 2021.

O primeiro passo nesta fase é identificar a máquina mais carregada, que neste caso, conforme Figura 5.1, M_2 demonstra ser a máquina mais carregada e as máquinas menos carregadas são M_1 e M_3 . No segundo passo, verifica-se na máquina M_2 a tarefa de menor valor de p_j para transferência, de acordo a Figura 5.2 demonstra:

Figura 5.2 – Identificação da máquina mais carregada e menor valor de p_j .



Fonte: Autora, 2021.

Após a identificação da tarefa com menor tempo de processamento em M_2 , neste caso, $J_{16} \{p_{jM_2} = 1, 91\}$, faz-se o mesmo teste aplicado na fase anterior. Calcula-se os valores mínimos a saírem das máquinas M_1 e M_3 e o valor máximo a entrar em M_2 , conforme demonstra a Tabela 5.1.

Tabela 5.1 – Valor máximo e valores mínimos para escolha das possíveis candidatas.

M	Valores para encontrar possibilidades de troca
M_1	53, 55 – 32, 93 = 20, 62 (min)
M_2	32, 93 – 31, 02 = 1, 91 (max)
M_3	38, 92 – 32, 93 = 5, 99 (min)

Fonte: Autora, 2021.

Nota-se que as possíveis combinações de tarefas candidatas a saírem de M_1 , precisam ser maiores que o valor de 20,62, conforme as condições determinadas. Neste caso, os valores a saírem da máquina M_1 , será resultado da soma das combinações de duas tarefas da máquina citada, conforme demonstra-se a seguir:

$J_1+J_2=14, 75$	$J_1+J_3=6, 33$	$J_1+J_{18}=3, 38$	$J_1+J_{17}=3, 42$	$J_1+J_{19}=8, 44$
-	$J_2+J_3=18, 86$	$J_2+J_{18}=15, 91$	$J_2+J_{17}=15, 95$	$J_2+J_9=20, 97$
-	-	$J_3+J_{18}=7, 49$	$J_3+J_{17}=7, 53$	$J_3+J_9=12, 55$
-	-	-	$J_{18}+J_{17}=4, 58$	$J_{18}+J_9=9, 60$
-	-	-	-	$J_{17}+J_9=9, 64$

Através da soma das combinações, apenas uma combinação apresentou possibilidade de tripla troca, porém, como tratam-se de máquinas paralelas não-relacionadas, as possibilidades de entrada devem ser recalculadas, é necessário adequar os tempos de processamento conforme a máquina M_2 , e para as possíveis combinações das tarefas candidatas entrarem em M_2 , não podem exceder o valor de 1,91. A Tabela 5.2 demonstra se existe essa possibilidade de tripla troca entre M_1 e M_2 , para este caso.

Tabela 5.2 – Possibilidade de tripla troca entre M_1 e M_2 .

M	Valores para encontrar possibilidades de tripla troca
M_1	$J_2 + J_9 = 20, 97$ (saem de M_1)
M_2	$J_2 + J_9 = 27, 56$ (entram de M_2)

Fonte: Autora, 2021.

Constatou-se que não houve possibilidade de tripla troca entre M_1 e M_2 , para este caso. O próximo passo é verificar se a máquina M_3 , apresenta alguma possibilidade de tripla troca. Mas, para saírem de M_3 as possíveis combinações candidatas devem ser maiores que 5,99. A seguir, a soma das combinações em M_3 :

$J_{10}+J_{19}=14, 61$	$J_{10}+J_5=9, 52$	$J_{10}+J_{11}=6, 29$	$J_{10}+J_{20}=9, 30$	$J_{10}+J_7=7, 26$	$J_{10}+J_{13}=10, 66$	$J_{10}+J_8=7, 19$	$J_{10}+J_{15}=5, 89$
-	$J_{19}+J_5=13, 09$	$J_{19}+J_{11}=9, 86$	$J_{19}+J_{20}=12, 87$	$J_{19}+J_7=10, 83$	$J_{19}+J_{13}=14, 23$	$J_{19}+J_8=10, 76$	$J_{19}+J_{15}=9, 46$
-	-	$J_5+J_{11}=4, 77$	$J_5+J_{20}=7, 78$	$J_5+J_7=5, 74$	$J_5+J_{13}=9, 14$	$J_5+J_8=5, 67$	$J_5+J_{15}=4, 37$
-	-	-	$J_{11}+J_{20}=4, 55$	$J_{11}+J_7=2, 51$	$J_{11}+J_{13}=5, 91$	$J_{11}+J_8=2, 44$	$J_{11}+J_{15}=1, 14$
-	-	-	-	$J_{20}+J_7=5, 52$	$J_{20}+J_{13}=8, 92$	$J_{20}+J_8=5, 45$	$J_{20}+J_{15}=4, 15$
-	-	-	-	-	$J_7+J_{13}=6, 88$	$J_7+J_8=3, 41$	$J_7+J_{15}=2, 11$
-	-	-	-	-	-	$J_{13}+J_8=6, 81$	$J_{13}+J_{15}=5, 51$
-	-	-	-	-	-	-	$J_8+J_{15}=2, 04$

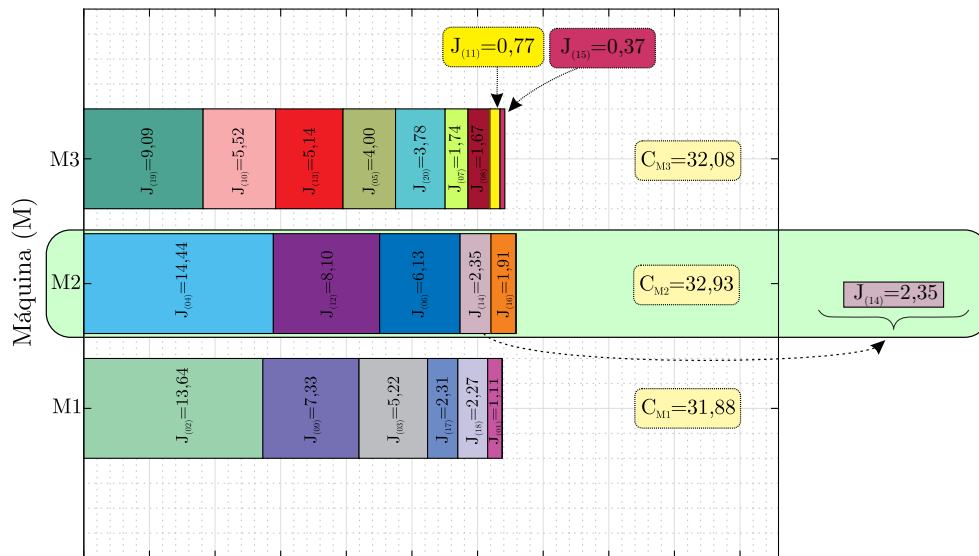
A matriz da soma de combinações da máquina M_3 apresentou dezenove possíveis possibilidades para tripla troca, porém, é necessário alterar os valores para entrarem em M_2 . A matriz a seguir demonstra os valores das combinações na máquina M_3 e como os valores ficam em M_2 .

$J_{10}+J_{19}=38, 57$	$J_{10}+J_5=18, 33$	$J_{10}+J_{11}=11, 54$	$J_{10}+J_{20}=18, 24$	$J_{10}+J_7=11, 48$	$J_{10}+J_{13}=15, 81$	$J_{10}+J_8=12, 92$
$J_{19}+J_5=36, 90$	$J_{19}+J_{11}=30, 11$	$J_{19}+J_{20}=36, 81$	$J_{19}+J_7=30, 05$	$J_{19}+J_{13}=34, 38$	$J_{19}+J_8=31, 49$	$J_{19}+J_{15}=29, 24$
-	-	$J_5+J_{20}=16, 57$	-	$J_5+J_{13}=14, 14$	-	-
-	-	-	-	$J_{20}+J_{13}=14, 05$	-	-
-	-	-	-	$J_7+J_{13}=7, 29$	-	-
-	-	-	-	$J_{13}+J_8=8, 73$	-	-

Para ocorrer possibilidade de tripla troca de M_3 para M_2 , a combinação da soma de duas tarefas, não pode ultrapassar o valor de 1,91 para entrar em M_2 . Neste caso, verificou-se que não existe nenhuma possibilidade de tripla troca de M_3 para M_2 .

Quando isso ocorre, se não houver nenhuma possibilidade de tripla troca em nenhuma máquina, o código escolhe a segunda tarefa com menor tempo de processamento da máquina mais carregada e reinicia o processo, conforme demonstrado na Figura 5.3.

Figura 5.3 – Máquina mais carregada e o segundo menor valor de p_j em M_2 .



Fonte: Autora, 2021.

A máquina mais carregada continua M_2 e a segunda tarefa com menor p_j é J_{14} $\{p_{jM_2} = 2,35\}$, esta é testada nas demais máquinas, conforme demonstra a Tabela 5.3.

Tabela 5.3 – Continuidade para encontrar mais possibilidades de troca.

M	C_i	p_j	Valores com teste
M_1	31,88	4,21	36,09
M_2	32,93	2,35	30,58
M_3	32,08	2,86	34,94

Fonte: Autora, 2021.

Esse teste possibilita encontrar o valor máximo da possível combinação da soma de duas tarefas para entrar em M_2 , e os valores mínimos das possíveis combinações candidatas a saírem de M_1 e M_3 . A Tabela 5.4 apresenta esse processo.

Tabela 5.4 – Valor máximo e valores mínimos para escolha das possíveis candidatas.

M	Valores para encontrar possibilidades de troca
M_1	$36,09 - 32,93 = 3,16$ (min)
M_2	$32,93 - 30,58 = 2,35$ (max)
M_3	$34,94 - 32,93 = 2,01$ (min)

Fonte: Autora, 2021.

As possíveis somas das combinações para saírem da máquina M_1 precisam ser maiores que 3,16. A matriz a seguir apresenta a soma das combinações em M_1 :

$J_1+J_2=14,75$	$J_1+J_3=6,33$	$J_1+J_{18}=3,38$	$J_1+J_{17}=3,42$	$J_1+J_{19}=8,44$
-	$J_2+J_3=18,86$	$J_2+J_{18}=15,91$	$J_2+J_{17}=15,95$	$J_2+J_9=20,97$
-	-	$J_3+J_{18}=7,49$	$J_3+J_{17}=7,53$	$J_3+J_9=12,55$
-	-	-	$J_{18}+J_{17}=4,58$	$J_{18}+J_9=9,60$
-	-	-	-	$J_{17}+J_9=9,64$

Todas as possíveis combinações em M_1 são candidatas a saírem para uma tripla troca, no entanto, os valores precisam ser modificados ao serem transferidos para máquina M_2 . Para as possíveis combinações candidatas entrarem em M_2 , estas não podem ultrapassar o valor de 2,35. E para este caso, esse processo apresentou uma possibilidade de tripla troca entre M_1 e M_2 . A Tabela 5.5, demonstra a combinação das duas tarefas candidatas:

Tabela 5.5 – Possibilidade de tripla troca entre M_1 e M_2 .

M	Valores para encontrar possibilidades de tripla troca
M_3	$J_1 + J_{18} = 3,38$ (saem de M_1)
M_2	$J_1 + J_{18} = 1,54$ (entram de M_2)

Fonte: Autora, 2021.

Após encontrar uma possibilidade de tripla troca entre M_1 e M_2 , o próximo passo é verificar se na máquina M_3 existe alguma possibilidade de tripla troca e se é possível o valor do *makespan* atual ser menor que o valor do *makespan* inicial. Para as combinações

da soma de duas tarefas saírem de M_3 , estas necessitam ser maiores que 2,01. A seguir a matriz com a soma das combinações em M_3 .

$J_{10}+J_{19}=14,61$	$J_{10}+J_5=9,52$	$J_{10}+J_{11}=6,29$	$J_{10}+J_{20}=9,30$	$J_{10}+J_7=7,26$	$J_{10}+J_{13}=10,66$	$J_{10}+J_8=7,19$	$J_{10}+J_{15}=5,89$
-	$J_{19}+J_5=13,09$	$J_{19}+J_{11}=9,86$	$J_{19}+J_{20}=12,87$	$J_{19}+J_7=10,83$	$J_{19}+J_{13}=14,23$	$J_{19}+J_8=10,76$	$J_{19}+J_{15}=9,46$
-	-	$J_5+J_{11}=4,77$	$J_5+J_{20}=7,78$	$J_5+J_7=5,74$	$J_5+J_{13}=9,14$	$J_5+J_8=5,67$	$J_5+J_{15}=4,37$
-	-	-	$J_{11}+J_{20}=4,55$	$J_{11}+J_7=2,51$	$J_{11}+J_{13}=5,91$	$J_{11}+J_8=2,44$	$J_{11}+J_{15}=1,14$
-	-	-	-	$J_{20}+J_7=5,52$	$J_{20}+J_{13}=8,92$	$J_{20}+J_8=5,45$	$J_{20}+J_{15}=4,15$
-	-	-	-	-	$J_7+J_{13}=6,88$	$J_7+J_8=3,41$	$J_7+J_{15}=2,11$
-	-	-	-	-	-	$J_{13}+J_8=6,81$	$J_{13}+J_{15}=5,51$
-	-	-	-	-	-	-	$J_8+J_{15}=2,04$

Ao observar a matriz com os valores das combinações na máquina M_3 , constatou-se que apenas uma combinação não atende a condição para tripla troca. Porém, ao serem alterados os valores das combinações para a máquina M_2 , houve a possibilidade de tripla troca entre M_3 e M_2 , com apenas uma soma de combinação de duas tarefas, como a Tabela 5.6 demonstra:

Tabela 5.6 – Possibilidade de tripla troca entre M_3 e M_2 .

M	Valores para encontrar possibilidades de tripla troca
M_3	$J_7 + J_{15} = 2,11$ (saem de M_3)
M_2	$J_7 + J_{15} = 2,15$ (entram de M_2)

Fonte: Autora, 2021.

Como houve possibilidade de troca, o processo continua até que não existam mais possibilidades de triplas trocas entre as máquinas. Ao encontrar uma possibilidade de troca entre M_1 e M_2 e M_3 e M_2 , verifica-se qual destas obteve o menor valor do *makespan* atual em relação ao valor do *makespan* inicial.

Tabela 5.7 – Apresentação das possibilidades de troca entre M_1 e M_2 e entre M_3 e M_2 .

Possibilidades	M1	M2	M3	Makespan
1ª Possibilidade	32,71	32,12	-	32,12
2ª Possibilidade	-	32,73	32,83	32,73

Fonte: Autora, 2021.

Constatou-se que a primeira possibilidade de tripla troca apresentou a menor redução em relação ao valor do *makespan* inicial = 32,93. No entanto, os valores de C_i das três máquinas são recalculados e atualizados devido a primeira tripla troca entre M_1 e M_2 . Os novos valores ficaram da seguinte maneira:

- $M_1 = 32,71$
- $M_2 = 32,12$
- $M_3 = 32,08$

O processo de triplas trocas continua, e a máquina mais carregada passa a ser M_1 , novamente escolhe-se a tarefa com menor tempo de processamento $J_{14} \{p_{jM1} = 4, 21\}$ para sair e ser testada nas demais máquinas, M_2 e M_3 , de acordo com a Tabela 5.8.

Tabela 5.8 – Valores de C_i com teste.

M	C_i	p_j	Valores com teste
M_1	32,71	2,31	30,40
M_2	32,12	5,45	37,57
M_3	32,08	7,50	39,58

Fonte: Autora, 2021.

Como mencionada anteriormente, esse teste possibilita encontrar o valor máximo da combinação da soma de duas tarefas para entrarem em M_1 , e os valores mínimos a saírem de M_2 e M_3 .

Tabela 5.9 – Valor máximo e valores mínimos para escolha das possíveis candidatas.

M	Valores para encontrar possibilidades de troca
M_1	$32,71 - 30,40 = 2,31$ (max)
M_2	$37,57 - 32,71 = 4,86$ (min)
M_3	$39,58 - 32,71 = 6,87$ (min)

Fonte: Autora, 2021.

A matriz seguinte apresenta os valores das combinações da soma de duas tarefas na máquina M_2 .

$J_4+J_6=20, 57$	$J_4+J_{12}=22, 54$	$J_4+J_{16}=16, 35$	$J_4+J_1=15, 27$	$J_4+J_{18}=15, 15$
-	$J_6+J_{12}=14, 23$	$J_6+J_{16}=8, 04$	$J_6+J_1=6, 96$	$J_6+J_{18}=6, 84$
-	-	$J_{12}+J_{16}=10, 01$	$J_{12}+J_1=8, 93$	$J_{12}+J_{18}=8, 81$
-	-	-	$J_{16}+J_1=2, 74$	$J_{16}+J_{18}=2, 62$
-	-	-	-	$J_1+J_{18}=1, 54$

As combinações candidatas a saírem de M_2 precisam ser maiores que 4,86, conforme condições determinadas na Tabela 5.9. Como os valores se alteram quando mudam de máquina, para este caso, doze combinações são candidatas a sair e para entrar em M_1 não podem exceder o valor de 2,31. A próxima matriz demonstra os valores das combinações na máquina M_1 .

$J_4+J_6=37, 97$	$J_4+J_{12}=32, 08$	$J_4+J_{16}=32, 50$	$J_4+J_1=11, 94$	$J_4+J_{18}=13, 10$
-	$J_6+J_{12}=48, 39$	$J_6+J_{16}=48, 81$	$J_6+J_1=28, 25$	$J_6+J_{18}=29, 41$
-	-	$J_{12}+J_{16}=42, 92$	$J_{12}+J_1=22, 36$	$J_{12}+J_{18}=23, 52$

Constatou-se que não houve nenhuma possibilidade de tripla troca entre M_1 e M_2 , para este caso. Desse modo, o algoritmo passa a buscar alguma possibilidade de tripla troca entre M_1 e M_3 . Os valores das combinações de M_3 são:

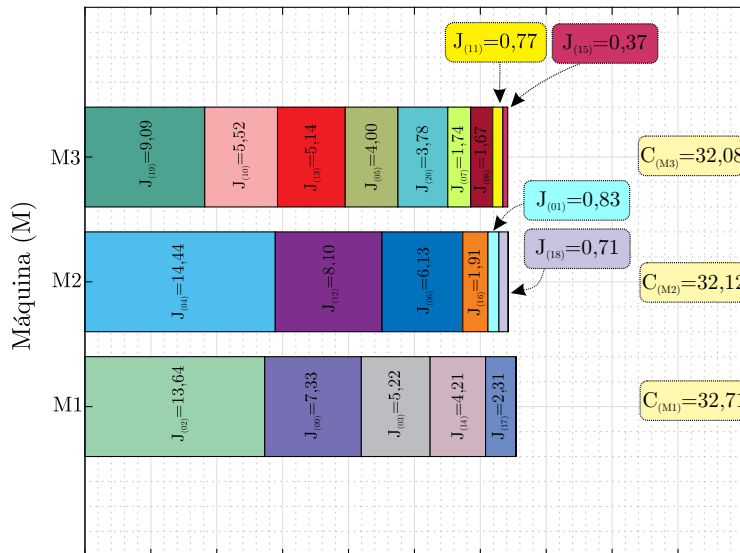
$J_{10}+J_{19}=14, 61$	$J_{10}+J_5=9, 52$	$J_{10}+J_{11}=6, 29$	$J_{10}+J_{20}=9, 30$	$J_{10}+J_7=7, 26$	$J_{10}+J_{13}=10, 66$	$J_{10}+J_8=7, 19$	$J_{10}+J_{15}=5, 89$
-	$J_{19}+J_5=13, 09$	$J_{19}+J_{11}=9, 86$	$J_{19}+J_{20}=12, 87$	$J_{19}+J_7=10, 83$	$J_{19}+J_{13}=14, 23$	$J_{19}+J_8=10, 76$	$J_{19}+J_{15}=9, 46$
-	-	$J_5+J_{11}=4, 77$	$J_5+J_{20}=7, 78$	$J_5+J_7=5, 74$	$J_5+J_{13}=9, 14$	$J_5+J_8=5, 67$	$J_5+J_{15}=4, 37$
-	-	-	$J_{11}+J_{20}=4, 55$	$J_{11}+J_7=2, 51$	$J_{11}+J_{13}=5, 91$	$J_{11}+J_8=2, 44$	$J_{11}+J_{15}=1, 14$
-	-	-	-	$J_{20}+J_7=5, 52$	$J_{20}+J_{13}=8, 92$	$J_{20}+J_8=5, 45$	$J_{20}+J_{15}=4, 15$
-	-	-	-	-	$J_7+J_{13}=6, 88$	$J_7+J_8=3, 41$	$J_7+J_{15}=2, 11$
-	-	-	-	-	-	$J_{13}+J_8=6, 81$	$J_{13}+J_{15}=5, 51$
-	-	-	-	-	-	-	$J_8+J_{15}=2, 04$

Para as possíveis combinações candidatas de M_3 saírem, estas precisam ser maiores que 6,87 e para entrarem em M_1 não podem ultrapassar o valor de 2,31. Ao serem alterados os valores para a máquina M_1 , apresentou-se dezessete possíveis combinações para troca, porém, para este caso, não houve nenhuma possibilidade de tripla troca, conforme é demonstrado a seguir:

-	$J_{10}+J_{19}=32, 20$	$J_{10}+J_5=23, 33$	$J_{10}+J_{20}=21, 61$	$J_{10}+J_7=16, 41$	$J_{10}+J_{13}=21, 90$	$J_{10}+J_8=18, 40$
$J_{19}+J_5=28, 87$	$J_{19}+J_{11}=22, 87$	$J_{19}+J_{20}=27, 15$	$J_{19}+J_7=21, 95$	$J_{19}+J_{13}=27, 44$	$J_{19}+J_8=23, 94$	$J_{19}+J_{15}=22, 20$
-	-	-	-	-	$J_5+J_{20}=18, 28$	$J_5+J_{13}=18, 57$
-	-	-	-	-	-	$J_{20}+J_{13}=16, 85$
-	-	-	-	-	-	$J_7+J_{13}=11, 65$

Como não houve possibilidades de trocas nas demais tarefas, considera-se o processo de otimização de triplas trocas concluído para este caso. A Figura 5.4, apresenta como ficaram ordenadas as tarefas nas máquinas M_1 , M_2 e M_3 e os novos valores de C_i após uma tripla troca encontrada entre a máquina M_1 e M_2 no decorrer do processo.

Figura 5.4 – Valores atualizados de C_i das três máquinas.



Fonte: Autora, 2021.

Verifica-se que o critério proposto e aplicado atingiu o objetivo da fase quatro. Objetivo esse de minimizar o tempo de finalização máxima das tarefas da máquina mais carregada (*makespan*) e ter a possibilidade de fazer uma tripla troca, também obtivemos uma aproximação nos valores de C_i , conforme resultado apresentado acima.

No capítulo seguinte será apresentado os resultados computacionais obtidos e uma análise entre os três algoritmos propostos de 4 fases e duas heurísticas dos autores Behnamian e Ghomi (2013), AGBG e HBG.

6 RESULTADOS COMPUTACIONAIS

O capítulo 6 traz a apresentação dos resultados computacionais a fim de avaliar a eficácia dos algoritmos propostos. Para isso, 540 instâncias foram geradas aleatoriamente segundo sugerido por Behnamian e Ghomi (2013) e todas as implementações dos algoritmos foram feitas em Linguagem Julia e executados por um computador com processador Intel(R) Core(TM)i5-8400, CPU @ 2.80 GHz 281 GHz, 8 GB de memória RAM, Sistema Operacional Microsoft Windows 10.

6.1 GERAÇÃO DE DADOS

Para analisar os algoritmos desenvolvidos para este problema, os dados foram gerados utilizando o formato proposto por Behnamian e Ghomi (2013), porém, as instâncias não foram as mesmas devido ao fato dos autores não terem disponibilizado para o presente estudo. É importante salientar que os dados dos autores foram utilizados como referência em termos de média (RPD), não considerando os tempos computacionais, pelo motivo de terem sido processados em computadores diferentes.

Diversas classes de instâncias foram definidas para o teste nos algoritmos desenvolvidos e foram geradas aleatoriamente 540 instâncias, ou seja, 10 amostras para cada conjunto de parâmetros, sendo todos testados, e para cada classe houve uma mudança em um dos parâmetros.

Para cada conjunto de dados, o sequenciamento foi distribuído com os seguintes parâmetros:

Tabela 6.1 – Parâmetros de entrada de dados.

Parâmetros	Valores
Número de tarefas	50, 100, 200
Número de fábricas	2, 3, 5
Número de máquinas por fábrica	2, 5
Número de máquinas por fábrica randomizadas	(2, 5)
Tempos de processamento	50-70, 70-100

Fonte: Autora, 2021.

A Tabela 6.2 demonstra para uma melhor visualização a distribuição dos parâmetros de entrada:

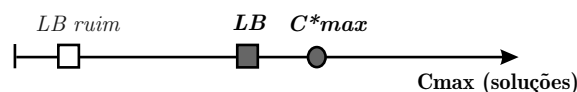
Tabela 6.2 – Distribuição dos parâmetros de entrada dos dados.

Problema	Número de Fábricas	Máquinas por Fábrica	p_j
1	2	2	(50, 70)
2	2	5	(50, 70)
3	2	(2, 5)	(50, 70)
4	2	2	(70, 100)
5	2	5	(70, 100)
6	2	(2, 5)	(70, 100)
7	3	2	(50, 70)
8	3	5	(50, 70)
9	3	(2, 5)	(50, 70)
10	3	2	(70, 100)
11	3	5	(70, 100)
12	3	(2, 5)	(70, 100)
13	5	2	(50, 70)
14	5	5	(50, 70)
15	5	(2, 5)	(50, 70)
16	5	2	(70, 100)
17	5	5	(70, 100)
18	5	(2, 5)	(70, 100)

6.1.1 Limite Inferior (Lower Bound)

Para cada problema do conjunto, os valores dos limitantes inferiores foram calculados de acordo com o *Limite Inferior*(LB) proposto por Martello, Soumis e Toth (1997) e são definidos nas equações 6.1, 6.2 e 6.3.

Este limitante inferior(LB) pode ser denominado como um valor de referência calculado de maneira independente do sequenciamento, garantindo que seja menor ou igual ao *makespan* ótimo. Sendo assim, quanto maior o valor do limitante inferior melhor, de acordo com a Figura 6.1:

Figura 6.1 – Limitante inferior para o *makespan*.

Fonte: Adpatado de (MESQUITA, 2015).

Conforme o conceito, quanto mais aproximado do valor ótimo do *makespan*, melhor será o limitante inferior. No entanto, à princípio não se tem o conhecimento do valor do

ótimo, pois é exatamente em seu lugar que será aplicado o limitante inferior. Dado que, o significado do limitante inferior atesta que seu valor ótimo será no máximo igual ao *makespan*, ou melhor, será sempre à sua esquerda na reta dos valores possíveis para o *makespan* demonstrado na Figura 6.1. Portanto, quanto mais à direita do valor do limitante inferior, mais perto será o *makespan* ótimo, e como ressaltado, quanto maior o seu valor melhor será sua referência (MESQUITA, 2015).

Um limite inferior pode ser obtido determinando $j = 1, \dots, n; pmin_j = \min_i \{p_{ij}\}$. Dado que o tempo de processamento não pode ser inferior a $\sum_{j=1}^n pmin_j$, um limite inferior válido em C_{max} é:

$$L' = \left[\frac{1}{m} \sum_{j=1}^n pmin_j \right] \quad (6.1)$$

Uma vez que cada tarefa deve ser agendada, um segundo limite é proposto:

$$L'' = \max_j \{pmin_j\}, \quad (6.2)$$

Assim, um limite inferior (LB) válido para o C_{max} é:

$$L'' = \max(L', L''). \quad (6.3)$$

Desse modo, a seguir será demonstrado na Tabela 6.3 os valores para a média, usando 10 execuções diferentes, aplicados a heurística proposta SPTteste:

Tabela 6.3 – Conjunto dos valores de LB, do problema 1, com 50 tarefas do SPTteste.

Amostra	Limite inferior	Makespan	RPD(%)	Tempo(s)
1	689,3841	696,7923	0,0055	0,3833
2	674,5293	680,2157	0,0083	0,2035
3	678,9325	683,0114	0,0038	0,1860
4	687,2542	699,5812	0,0150	0,1663
5	677,3747	683,0109	0,0024	0,2251
6	680,1264	692,3253	0,0134	0,1591
7	686,7790	693,4086	0,0029	0,1580
8	681,7128	686,8380	0,0059	0,1534
9	674,1313	680,4633	0,0064	0,1508
10	680,599	680,1726	0,0017	0,1682
Média	681,0823	687,5819	0,0065	0,1954

A Tabela 6.3 demonstra a média calculada para o problema 1, contendo um conjunto de 10 amostras. Para este primeiro problema, foram distribuídas 50 tarefas para serem alocadas em 2 fábricas, com 2 máquinas por fábrica. Observou-se que para este conjunto de amostras o limite inferior apresentou um valor médio de 681,0823 , o *makespan* uma média de 687.5819, o RPD obteve a média de 0,0065%, já o tempo médio para estas amostras foram de 0.1954(s).

6.1.2 Análise dos Resultados

Para avaliar os resultados alcançados nos testes computacionais, foram analisados o desvio percentual relativo (RPD), no que se refere a solução inicial e ao Limite Inferior (LB) e o tempo médio computacional(T), medido em segundos(s).

Nesta análise, utilizou-se a fórmula proposta por Behnamian e Ghomi (2013), calculado da seguinte forma:

$$RPD = \frac{ALG_{SOL} - LB_{SOL}}{LB_{SOL}} \times 100\%$$

Onde ALG_{sol} é o Cmax obtido para um determinado algoritmo e instância, e LB_{SOL} é o limite inferior obtido para cada instância. É importante salientar que quanto menor o valor do RPD, melhor será a qualidade da solução.

6.1.3 Análise dos Resultados entre as Cinco Heurísticas com 50 tarefas.

Neste subitem é apresentado os resultados obtidos das três heurísticas propostas: 4Fases-SPTteste, 4Fases-LPTteste e 4Fases-FCFSteste e os resultados dos dois algoritmos publicados pelos autores Behnamian e Ghomi (2013), utilizados como referência, são eles: Algoritmo Genético(AGBG) e outra denominada como Heurística(HBG).

Foram geradas 10 amostras para cada conjunto de parâmetros, neste caso, os problemas ao qual nos referimos, possuem 18 instâncias em cada conjunto, totalizando 540 instâncias geradas aleatoriamente. Portanto, as tabelas a seguir demonstram os dados gerados através dos respectivos testes, considerando o método descrito de Behnamian e Ghomi (2013) com 50, 100 e 200 tarefas.

Serão apresentados também os valores do desvio percentual médio(RPD) e os Tempos(s), que correspondem, respectivamente, ao tempo despendido para encontrar uma solução de boa qualidade. Dessa forma, pressupõe-se que uma vez que tudo é distribuído uniformemente na geração das heurísticas dos autores Behnamian e Ghomi (2013),

supõe-se que na média, para o presente estudo, os resultados também sejam semelhantes, pois, é importante salientar que esta referência se dá somente entre as médias, devido os autores não terem disponibilizados os códigos para o presente estudo. No que se refere aos Tempos(s), não houve nenhuma comparação devido ao fato de que o processamento dos algoritmos foram executados em computadores diferentes.

Behnamian e Ghomi (2013) geraram seus resultados em um computador Intel Pentium IV Dual Core 2.00 GHz com 1022 MB de RAM, no MATLAB 7, em um ambiente Microsoft Windows Vista.

Em anexo, são demonstrados as tabelas com os resultados obtidos, dos valores do LB e do *makespan* para cada problema processado, os tempos de processamento uniformemente distribuídos, [50, 70] e [70, 100], com 2, 3 e 5 fábricas, para [2, 5] máquinas por fábrica, tendo uma distribuição aleatória de máquinas entre [2, 5] para cada um dos conjuntos de 50, 100 e 200 tarefas.

A Tabela 6.4, apresenta 18 instâncias ou conjuntos, cada instância representa 10 problemas com os parâmetros de cada um dos 18 conjuntos, para 50 tarefas. Os melhores resultados obtidos através das referências entre os cinco algoritmos, estão destacados em negrito.

Tabela 6.4 – Valores das soluções propostas obtidas e os resultados apresentado por Behnamian e Ghomi (2013), com 50 tarefas.

Pbs	4Fases-SPTteste		4Fases-LPTteste		4Fases-FCFSteste		AGBG	HBG
	RPD(%)	T(s)	RPD(%)	T(s)	RPD(%)	T(s)	RPD(%)	RPD(%)
1	0,91	0,20	2,09	0,25	1,74	0,24	4,40	2,40
2	1,80	0,20	2,76	0,26	2,44	0,24	6,90	2,80
3	2,03	0,20	3,79	0,27	3,05	0,24	10,20	2,20
4	0,65	0,20	1,06	0,25	0,81	0,24	0,20	2,80
5	1,68	0,21	2,70	0,26	1,76	0,24	4,00	1,10
6	1,17	0,20	2,51	0,62	2,31	0,23	10,70	2,30
7	0,54	0,22	1,46	0,25	1,42	0,23	3,20	6,90
8	1,53	0,23	3,91	0,27	1,81	0,24	30,30	13,60
9	0,85	0,23	1,33	0,26	0,80	0,23	26,60	3,10
10	0,90	0,20	1,84	0,25	1,30	0,23	1,00	3,40
11	1,22	0,22	2,60	0,27	1,70	0,24	33,30	17,00
12	5,87	0,22	6,77	0,27	6,46	0,24	30,70	6,80
13	1,63	0,23	2,20	0,28	1,07	0,24	1,80	5,90
14	5,02	0,25	6,15	0,31	5,80	0,24	46,60	3,00
15	4,56	0,24	5,26	0,28	5,04	0,23	74,10	24,60
16	1,63	0,22	1,98	0,27	3,48	0,23	1,30	1,80
17	4,04	0,24	5,66	0,30	5,17	0,24	48,30	3,40
18	1,21	0,24	2,00	0,28	1,78	0,24	78,90	22,70

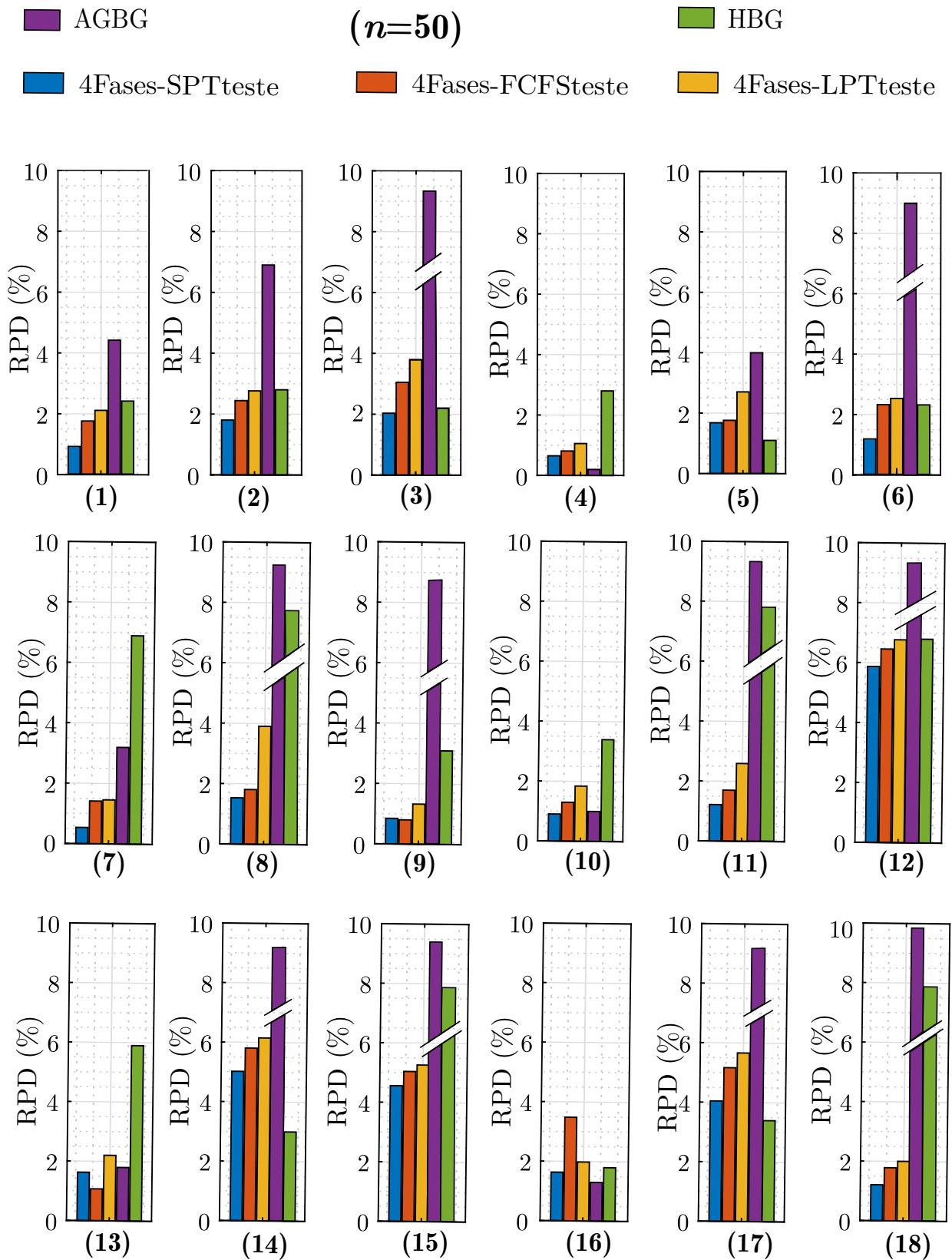
Fonte: Autora, 2021.

Observa-se que, comparativamente aos resultados do RPD médio, apresentado por Behnamian e Ghomi (2013) na Tabela 6.4, o modelo proposto SPTteste obteve, em 11 instâncias valores com melhor qualidade. Para esta análise, dentre as 180 instâncias processadas, a heurística SPTteste encontrou 110 soluções novas, ou seja, 61,11% do total das 180 instâncias processadas para este conjunto de 50 tarefas, além de necessitar de 2,37 segundos para chegar a soluções de boa qualidade.

A demonstração gráfica possibilita de forma mais clara a observação e compreensão do desempenho dos algoritmos propostos e dos algoritmos AGBG e HBG. Para elaboração dos gráficos foi usado o conjunto com 50 tarefas e o desvio relativo percentual (RPD %).

A Figura 6.2, demonstra o desempenho das soluções encontradas pela heurística SPTteste, e o comportamento das outras quatro heurísticas. Como a representação do desenvolvimento dos valores de RPD médio é de até 10%, na Figura 6.2, optou-se por demonstrar com um corte os algoritmos que obtiveram valores maiores que o estipulado para apresentação dos gráficos.

Figura 6.2 – Desempenho dos cinco algoritmos para 50 tarefas.



Fonte: Autora, 2021.

Como pode ser observado na Figura 6.2, dentre as cinco heurísticas apresentadas e 180 instâncias processadas, o algoritmo SPTteste alcançou em 11 problemas, 61,11% dos resultados com boa qualidade, ou melhor, conseguiu valores de RPD médio menores, para este comparativo de 18 instâncias, em seguida, ficaram as heurísticas HBG com 16,67%, o FCFSteste e o AGBG com 11,11% cada, por fim, o LPTteste que para este conjunto não apresentou nenhum melhor valor em destaque.

Ao tratar dos tempos de execução(T) dos cinco algoritmos, o SPTteste obteve os menores tempos gastos em 14 problemas, totalizando 2,98 segundos para realização do processamento, porém, em 2 problemas, o SPTteste e a heurística FCFSteste obtiveram valores iguais, com 0,24 segundos cada.

6.1.4 Análise dos Resultados entre as Cinco Heurísticas com 100 tarefas.

Na Tabela 6.5, são apresentados os resultados obtidos entre as cinco heurísticas, para 100 tarefas, com 18 problemas, ou seja, 180 instâncias.

Tabela 6.5 – Valores dos modelos propostos e os resultados apresentado por Behnamian e Ghomi (2013), com 100 tarefas.

Pbs	4Fases-SPTteste		4Fases-LPTteste		4Fases-FCFSteste		AGBG	HBG
	RPD(%)	T(s)	RPD(%)	T(s)	RPD(%)	T(s)	RPD(%)	RPD(%)
1	0,20	0,25	0,27	0,31	0,29	0,27	0,10	0,10
2	0,67	0,26	5,32	0,34	2,79	0,26	5,50	6,10
3	1,20	0,25	3,57	0,32	1,73	0,27	7,40	3,60
4	0,20	0,25	0,50	0,31	0,29	0,28	0,10	1,60
5	0,56	0,27	0,71	0,34	0,60	0,27	61,20	0,50
6	0,29	0,27	0,92	0,31	0,27	0,27	5,60	2,00
7	0,38	0,26	0,61	0,31	0,97	0,27	0,80	1,70
8	0,72	0,30	1,39	0,36	0,87	0,27	39,30	4,10
9	1,28	0,26	2,21	0,32	1,51	0,25	24,00	3,60
10	0,26	0,26	0,64	0,32	0,56	0,28	1,20	1,30
11	0,59	0,28	0,91	0,35	0,68	0,27	24,30	4,60
12	5,11	0,26	7,09	0,33	6,29	0,26	19,50	0,00
13	0,63	0,28	0,71	0,34	1,09	0,27	2,30	5,80
14	2,14	0,33	7,33	0,42	2,66	0,27	24,20	6,40
15	6,05	0,31	6,64	0,38	6,46	0,27	30,20	6,00
16	0,49	0,28	5,11	0,34	3,94	0,27	2,30	4,00
17	1,97	0,33	5,13	0,41	4,73	0,27	19,00	5,30
18	0,52	0,34	1,32	0,37	0,81	0,27	0,10	8,70

Fonte: Autora, 2021.

Na Tabela 6.5, foram apresentados os valores do desvio relativo percentual (RPD) e os tempos (T) em segundos das heurísticas propostas: 4Fases-SPTteste, 4Fases-LPTteste e 4Fases-FCFSteste, e as heurísticas AGBG e HBG, com 100 tarefas, para 18 instâncias, e com 10 amostras cada, totalizando 180 instâncias processadas. Dentre as 180 instâncias, para esse conjunto de tarefas, o modelo aqui proposto SPTteste, encontrou as melhores soluções para 120 instâncias, ou melhor, obteve resultados menores em 66,66% dos problemas gerados, tendo 3,37 segundos de tempo gasto para concluir o sequenciamento em 12 problemas.

Neste mesmo conjunto de tarefas, as heurísticas AGBG e HBG, publicadas por Behnamian e Ghomi (2013), obtiveram 13,89% cada, de valores de RPD menores que as demais heurísticas, ou seja, cada uma alcançou menores resultados em 2 conjuntos processados com 10 problemas cada, porém, na primeira instância ambas obtiveram valores iguais de RPD médio. O algoritmo FCFSteste apresentou na Tabela 6.4, um valor em destaque, representando menor resultado na média em apenas 1 instância, 5,56% das 18 instâncias apresentadas. O LPTteste, apesar de ter resultados significativos dentre as heurísticas propostas, nesse conjunto não alcançou nenhum valor de RPD médio em destaque.

No que se refere ao tempo gasto para concluir o processamento, neste conjunto com 100 tarefas, a heurística FCFSteste apresentou os menores tempos em 9 instâncias, e apresentando em 4 instâncias, tempos semelhantes ao algoritmo SPTteste.

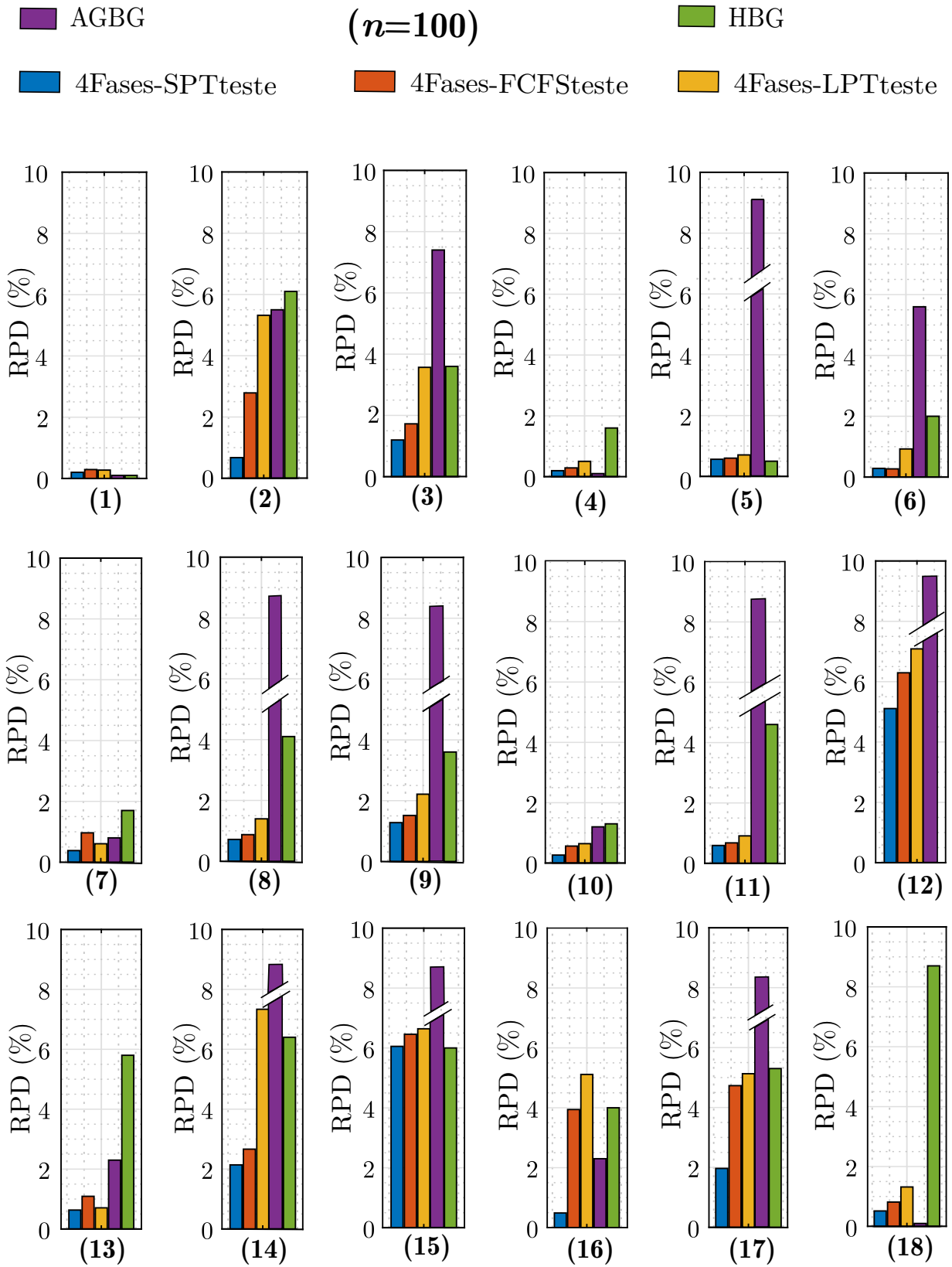
Nota-se que para 360 instâncias, com 50 e 100 tarefas, as heurísticas SPTteste, LPTteste e FCFSteste propostas, alcançaram valores mais próximos do limitante inferior(LB), pois, em comparação aos algoritmos AGBG e HBG, as heurísticas sugeridas neste estudo, atingiram resultados de boa qualidade em relação a média. O resultado que demonstrou melhor RPD médio entre as cinco heurísticas, foi o algoritmo HBG, com valor de 0,00% em relação ao limitante inferior(LB), na instância 12 com 100 tarefas.

Com base nos parâmetros de entrada do problema, para gerar a solução inicial dos algoritmos e obter os resultados apresentados, cada heurística proposta teve como base um método diferente de ordenação na implementação. Dessa forma, a ideia seria demonstrar que as heurísticas propostas: 4Fases-SPTteste, 4Fases-LPTteste e 4Fases-FCFSteste alcançassem soluções iniciais de melhor qualidade do que as heurísticas AGBG e HBG mostraram, nas médias.

A Figura 6.3, demonstra para uma melhor compreensão, o comportamento das heurísticas propostas, em comparação com as heurísticas AGBG e HBG, para 100 tarefas, e para 2 e 5 máquinas, ocorrendo uma distribuição aleatória para cada conjunto de 2, 3 e 5 fábricas. Para analisar os resultados das tabelas 6.3, 6.4 e 6.5, em relação ao limitante inferior, geraram-se as tabelas que encontram-se em anexo. Todas as tabelas contém o número de problemas, número de fábricas, o número de máquinas por fábrica, os tempo

de processamento, o *makespan*, o valor mínimo da solução gerada, o RPD médio de cada instância e o tempo que cada instância gastou para finalizar o processo.

Figura 6.3 – Desempenho dos cinco algoritmos para 100 tarefas.



Para problemas com 2 máquinas, as cinco heurísticas tiveram desempenhos de valores menores. Os maiores valores de RPD médio ocorreram em problemas com 5 máquinas, e para este conjunto de 100 tarefas, a heurística AGBG demonstrou o pior valor de RPD médio = 61,20%, na instância 5. Importante salientar que, conforme o aumento do número de tarefas, os desvios relativos foram obtendo valores maiores.

6.1.5 Análise dos Resultados entre as Cinco Heurísticas com 200 tarefas.

A seguir, a Tabela 6.6 demonstra os valores de RPD médio, utilizado como valores de referência das cinco heurísticas em relação ao limitante inferior, com 200 tarefas, para 18 instâncias.

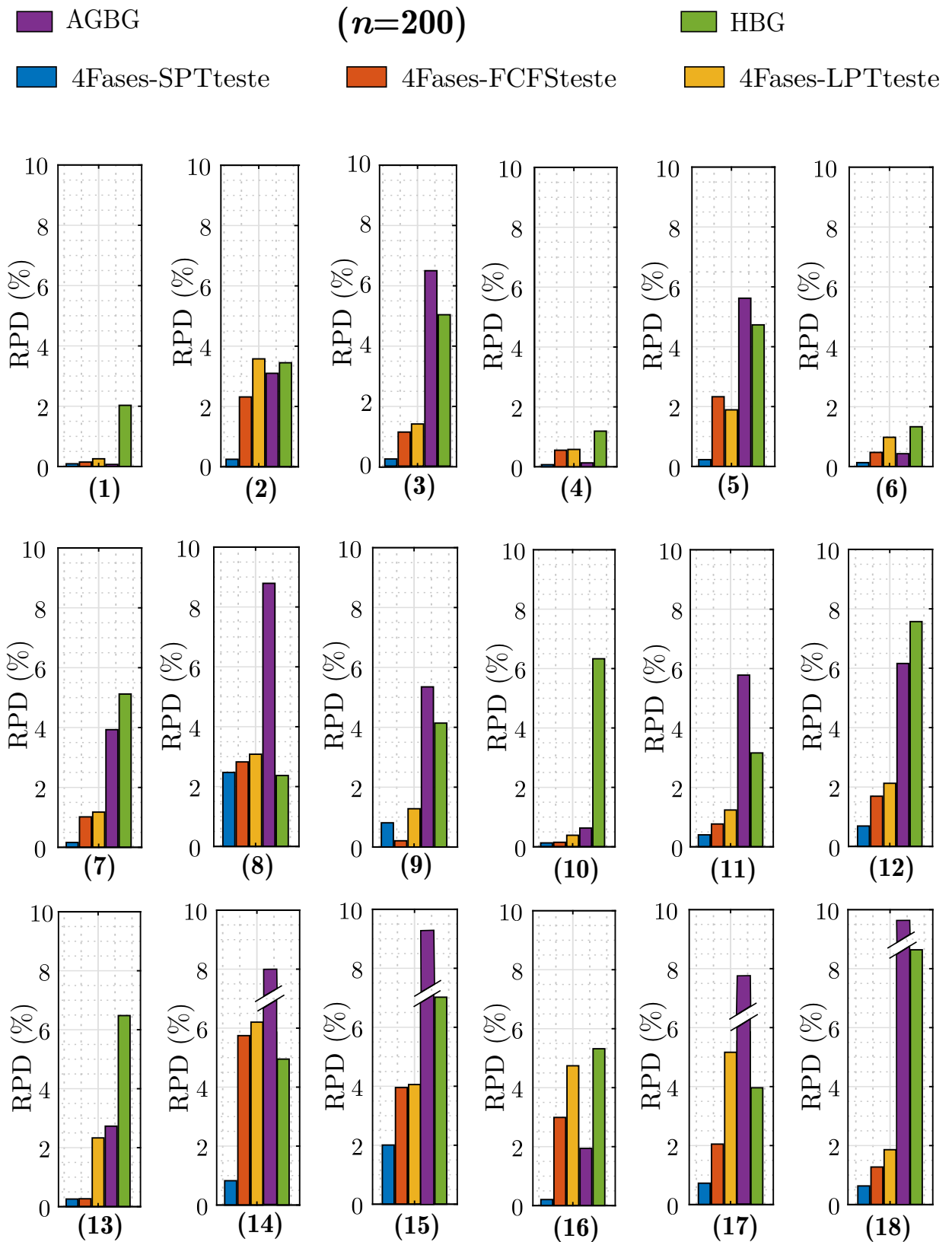
Tabela 6.6 – Valores dos modelos propostos e os resultados apresentado por Behnamian e Ghomi (2013), com 200 tarefas.

Pbs	4Fases-SPTteste		4Fases-LPTteste		4Fases-FCFSteste		AGBG	HBG
	RPD(%)	T(s)	RPD(%)	T(s)	RPD(%)	T(s)	RPD(%)	RPD(%)
1	0,09	0,36	0,26	0,45	0,15	0,39	0,07	2,03
2	0,25	0,41	3,58	0,54	2,32	0,34	3,10	3,45
3	0,24	0,41	1,40	0,59	1,13	0,38	6,48	5,02
4	0,07	0,41	0,58	0,57	0,55	0,41	0,13	1,19
5	0,23	0,25	1,89	0,68	2,33	0,35	5,62	4,73
6	0,13	0,39	0,98	0,57	0,47	0,41	0,43	1,33
7	0,16	0,44	1,18	0,46	1,02	0,43	3,93	5,12
8	2,47	0,41	3,08	0,52	2,82	0,34	8,79	2,37
9	0,81	0,40	1,28	0,51	0,21	0,34	5,35	4,14
10	0,13	0,38	0,39	0,50	0,15	0,47	0,63	6,33
11	0,40	0,40	1,24	0,54	0,77	0,35	5,78	3,16
12	0,69	0,39	2,13	0,58	1,69	0,35	6,16	7,57
13	0,26	0,47	2,33	0,57	0,27	0,33	2,73	6,48
14	0,83	0,50	6,20	0,71	5,75	0,33	15,66	4,95
15	2,01	0,44	4,06	0,54	3,96	0,35	42,74	7,02
16	0,21	0,44	4,74	0,63	2,99	0,33	1,94	5,32
17	0,72	0,51	5,16	0,67	2,05	0,35	14,32	3,96
18	0,63	0,44	1,86	0,55	1,27	0,35	33,04	8,64

Fonte: Autora, 2021.

Na Tabela 6.6, a heurística de 4Fases-SPTteste conseguiu alcançar valores menores em 15 instâncias, para 200 tarefas, isso corresponde a 83,32% das instâncias resolvidas com boa qualidade. A Figura 6.4, demonstra o comportamento das cinco heurísticas em relação ao limitante inferior.

Figura 6.4 – Desempenho dos cinco algoritmos para 200 tarefas.



De acordo com os valores da Tabela 6.6 e a demonstração do comportamento das cinco heurísticas, nas 18 instâncias com 200 tarefas e 2 máquinas, as heurísticas de 4Fases-SPTteste, 4Fases-LPTteste e FCFSteste e a AGBG, apresentaram menores desvios relativos, somente a heurística HBG apresentou em 4 instâncias com valores menores de RPD médio. Já para 5 máquinas, os valores do RPD médio aumentaram significativamente nas três heurísticas propostas e na AGBG. No entanto, a HBG manteve 4 instâncias com valores menores na média. Para o número de distribuição aleatória de máquinas por fábrica, os valores de RPD demonstraram ser menores em 88,88% dos resultados nas três heurísticas sugeridas, e as heurísticas AGBG e HBG em 22,22%.

Tendo como referência as médias dos outros conjuntos de tarefas (50, 100), os desvios relativos para 200 tarefas ficaram com valores menores, e novamente o algoritmo de 4Fases-SPTteste obteve os resultados de boa qualidade em 15 instâncias, além de necessitar de 6,96 segundos para concluir o processo.

A Tabela 6.6, demonstra a média total das 54 instâncias que cada heurística alcançou em relação ao RPD%, para 50, 100 e 200 tarefas.

Tabela 6.7 – RPD% total alcançados por cada heurística, para 50, 100 e 200 tarefas.

n	SPTteste	LPTteste	FCFSteste	AGBG	HBG
50 tarefas	2,06	3,11	2,66	22,91	6,98
100 tarefas	1,29	2,35	2,03	14,83	3,63
200 tarefas	0,57	2,35	1,66	8,71	4,60
Geral	1,31	2,60	2,11	15,49	5,07

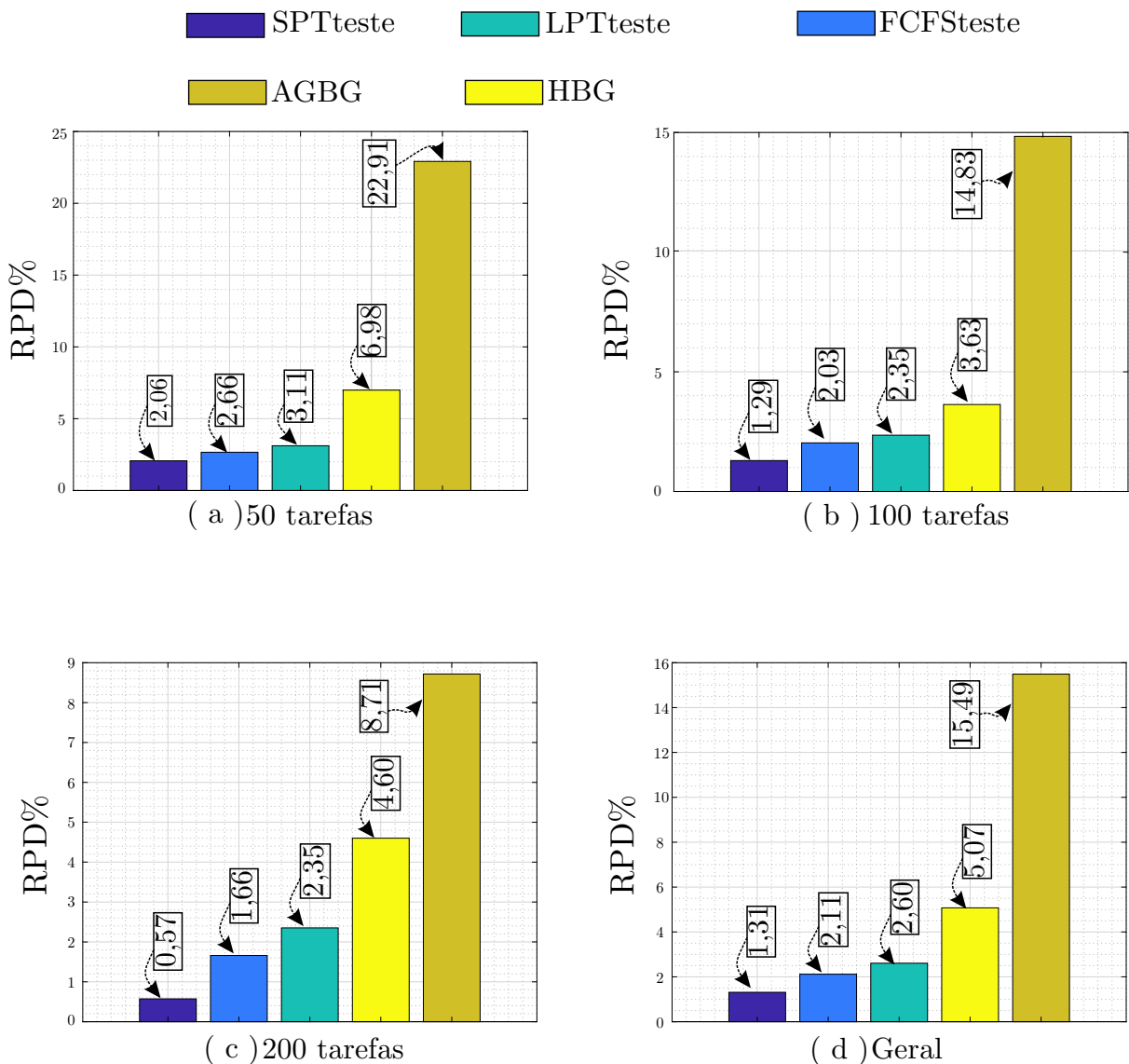
Fonte: Autora, 2021.

Demonstra-se, assim, que de acordo com as 540 instâncias geradas com o método descrito no artigo dos autores Behnamian e Ghomi (2013), a heurística de 4Fases-SPTteste tem grandes indícios da qualidade ser boa na solução do problema de máquinas paralelas não-relacionadas em multi-fábricas.

Desse modo, de acordo com os resultados expostos, bem como a análise realizada, a heurística de 4Fases-SPTteste alcançou bons resultados nas médias no geral, chegando muito próximo ao limitante inferior, destacando-se com o RPD médio geral = 1,31% de boa qualidade. Em seguida, a heurística 4Fases-FCFSteste obteve bons resultados com RPD médio geral = 2,11% e após a heurística proposta 4Fases-LPTteste com RPD médio geral = 2,60%, e por fim, as heurísticas HBG com RPD médio geral = 5,07% e AGBG com RPD médio geral = 15,49%.

A Figura 6.5 demonstra através de gráficos, o RPD médio para as 50, 100 e 200 tarefas e como se comportaram os valores das médias de RPD de modo geral.

Figura 6.5 – Gráficos das médias gerais das cinco heurísticas.



Observou-se que em todos os casos, conforme o número de tarefas aumentava, uma melhoria nos valores do RPD ocorria, e os melhores resultados deram-se com 200 tarefas. Na Figura (d), de um modo geral, as heurísticas propostas nesse estudo conseguiram valores muito bons e significativos em relação ao RPD médio, usando como referência as heurísticas AGBG e HBG dos autores Behnamian e Ghomi (2013).

Ao que se refere as multi-fábricas, as heurísticas propostas de 4Fases-SPTteste, 4Fases-LPTteste e 4Fases-FCFSteste e seus respectivos critérios de ordenação, são utilizados para lidar com a atribuição de tarefas às fábricas e a decisão de programação em cada uma delas, respectivamente.

7 CONCLUSÃO

Nesta dissertação foi abordado o problema de sequenciamento em máquinas paralelas não-relacionadas em multi-fábricas, problema esse denotado por $R|multifact|C_{max}$, e considerando as informações sugeridas por Behnamian e Ghomi (2013). O problema tem como objetivo encontrar uma sequência de tarefas para cada máquina com a finalidade de minimizar o tempo máximo de conclusão, denominado como *makespan*.

Após uma busca na literatura, por problemas que aplicaram as Regras de Prioridade em máquinas paralelas não-relacionadas, foram desenvolvidos, o procedimento da solução inicial, as três heurísticas propostas: SPTteste, LPTteste e FCFSteste, aplicadas em 4 fases: Fase de Alocação Inicial, Fase de Balanceamento, Fase de Duplas Trocas e Fase de Triplas Trocas, e os limitantes inferiores que foram utilizados na análise dos resultados.

O presente estudo apresentou algumas características com diferentes métodos de aplicação para máquinas paralelas não-relacionadas em multi-fábricas, portanto, considera-se um problema importante por fazer parte da área de diferentes indústrias de manufatura e serviço, e por pertencer a classe de problemas **NP-difícil**.

Os autores Behnamian e Ghomi (2013) trataram do mesmo problema, porém, um novo conjunto de dados apresentados nesse estudo foram gerados da mesma maneira que encontra-se na pesquisa dos autores, para que pudesse ser comparada a média, ou melhor o RPD médio.

Para resolver este problema foram propostos três algoritmos baseados nas Regras de Prioridade. As heurísticas foram adaptadas das Regras de Prioridade: SPT(*Shortest Processing Time*), LPT(*Longest Processing Time*) e FCFS(*First Come, First Served*), e denominados como: SPTteste, LPTteste e FCFSteste, aplicadas para quatro fases distintas.

De acordo com os experimentos realizados, as três heurísticas propostas apresentaram diferenças significativas em relação ao RPD médio, na qualidade das soluções geradas, e mostraram-se eficientes computacionalmente, tendo como referência as heurísticas AGBG e HBG. Além disso, a heurística 4FASES-SPTteste obteve o melhor desempenho entre as cinco heurísticas. As diferenças nos resultados ocorreram ao aplicar os critérios de cada regra de prioridade, isso mostra a importância de uma regra de prioridade na qualidade da solução de um problema de sequenciamento.

Em anexo encontram-se as tabelas com as informações apresentadas na presente pesquisa, e as tabelas com os dados sugeridos pelos autores Behnamian e Ghomi (2013). Sugere-se como trabalho futuro, novos métodos de solução para o problema de sequenciamento em máquinas não-relacionadas em multi-fábricas, com outros parâmetros de entrada, como por exemplo, tempos de transporte, tempos de preparação dependente da sequência, prazos de entrega, entre outros.

REFERÊNCIAS

- ALHARKAN, I. M. Algorithms for sequencing and scheduling. **Industrial Engineering Department, King Saud University, Riyadh, Saudi Arabia**, 2005.
- BAKER, K. R. **Introduction to sequencing and scheduling**. [S.l.]: John Wiley & Sons, 1974.
- BARBOSA, M. M. Relações entre classes de problemas computacionais: um estudo sobre classes de complexidade quânticas. Florianópolis, SC., 2020.
- BARGAOUI, H.; DRISS, O. B.; GHÉDIRA, K. Minimizing makespan in multi-factory flow shop problem using a chemical reaction metaheuristic. In: IEEE. **2016 IEEE Congress on Evolutionary Computation (CEC)**. [S.l.], 2016. p. 2919–2926.
- BEHNAMIAN, J.; GHOMI, S. F. The heterogeneous multi-factory production network scheduling with adaptive communication policy and parallel machine. **Information Sciences**, Elsevier, v. 219, p. 181–196, 2013.
- BEHNAMIAN, J.; GHOMI, S. M. T. F. Multi-objective multi-factory scheduling. **RAIRO Oper Res**, 2020.
- EZUGWU, A. E.; ADELEKE, O. J.; VIRIRI, S. Symbiotic organisms search algorithm for the unrelated parallel machines scheduling with sequence-dependent setup times. **PloS one**, Public Library of Science San Francisco, CA USA, v. 13, n. 7, p. e0200030, 2018.
- FU, Y. et al. Distributed scheduling problems in intelligent manufacturing systems. **Tsinghua Science and Technology**, TUP, v. 26, n. 5, p. 625–645, 2021.
- GARY, M. R.; JOHNSON, D. S. **Computers and Intractability: A Guide to the Theory of NP-completeness**. [S.l.]: WH Freeman and Company, New York, 1979.
- GRAHAM, R. L. et al. Optimization and approximation in deterministic sequencing and scheduling: a survey. In: **Annals of discrete mathematics**. [S.l.]: Elsevier, 1979. v. 5, p. 287–326.
- KANYALKAR, A.; ADIL, G. An integrated aggregate and detailed planning in a multi-site production environment using linear programming. **International Journal of Production Research**, Taylor & Francis, v. 43, n. 20, p. 4431–4454, 2005.
- KARIMI, N.; DAVOUDPOUR, H. A knowledge-based approach for multi-factory production systems. **Computers & Operations Research**, Elsevier, v. 77, p. 72–85, 2017.
- LAWLER, E. L. et al. **Sequencing and scheduling: Algorithms and complexity**. Amsterdam, 1989.

- LEI, D. et al. An imperialist competitive algorithm with memory for distributed unrelated parallel machines scheduling. **International Journal of Production Research**, Taylor & Francis, v. 58, n. 2, p. 597–614, 2020.
- LI, W. et al. An integrated production and transportation scheduling method in hybrid flow shop. 2021.
- LI, Y. et al. A discrete artificial bee colony algorithm for distributed hybrid flowshop scheduling problem with sequence-dependent setup times. **International Journal of Production Research**, Taylor & Francis, v. 59, n. 13, p. 3880–3899, 2021.
- LIMBERGER, S. J. **Algoritmos Heurísticos para Solução de Problemas de Sequenciamento em Processadores Paralelos Uniformes**. 77 p. Dissertação (Mestrado) — Universidade Federal de Santa Maria, Santa Maria, 1997.
- LIN, Y.-K. Particle swarm optimization algorithm for unrelated parallel machine scheduling with release dates. **Mathematical Problems in Engineering**, Hindawi, v. 2013, 2013.
- LOHMER, J.; LASCH, R. Production planning and scheduling in multi-factory production networks: a systematic literature review. **International Journal of Production Research**, Taylor & Francis, v. 59, n. 7, p. 2028–2054, 2021.
- MARTELLO, S.; SOUMIS, F.; TOTH, P. Exact and approximation algorithms for makespan minimization on unrelated parallel machines. **Discrete applied mathematics**, Elsevier, v. 75, n. 2, p. 169–188, 1997.
- MENG, T.; PAN, Q.-K.; WANG, L. A distributed permutation flowshop scheduling problem with the customer order constraint. **Knowledge-Based Systems**, Elsevier, v. 184, p. 104894, 2019.
- MESQUITA, F. N. Meta-heurística baseada em simulated annealing para programação da produção em máquinas paralelas com diferentes datas de liberação e tempos de setup. Universidade Federal de Goiás, 2015.
- MORTON, T.; PENTICO, D. W. **Heuristic scheduling systems: with applications to production systems and project management**. [S.l.]: John Wiley & Sons, 1993. v. 3.
- MÜLLER, F. M. **Algoritmos heurísticos e exatos para resolução do problema de sequenciamento em processadores paralelos**. Tese (Doutorado) — University of Campinas, Brazil, 1993. Disponível em: <<http://www.bibliotecadigital.unicamp.br/document/?code=vtls000069995>>.
- MÜLLER, F. M.; DIAS, O. B. Algoritmo para o problema de seqüenciamento em máquinas paralelas não-relacionadas. **Revista Produção**, SciELO Brasil, v. 12, n. 2, p. 7, 2002.

- RAMEZANIAN, R.; SAIDI-MEHRABAD, M. Multi-product unrelated parallel machines scheduling problem with rework processes. **Scientia Iranica**, Elsevier, v. 19, n. 6, p. 1887–1893, 2012.
- SANLAVILLE, E.; SCHMIDT, G. Machine scheduling with availability constraints. **Acta Informatica**, Springer, v. 35, n. 9, p. 795–811, 1998.
- SHAO, W.; SHAO, Z.; PI, D. Effective constructive heuristics for distributed no-wait flexible flow shop scheduling problem. **Computers & Operations Research**, Elsevier, v. 136, p. 105482, 2021.
- WACHTEL, G.; ELALOUF, A. Efficient approximation scheme for job assignment in a multi-factory environment. **Journal of Industrial and Production Engineering**, Taylor & Francis, v. 37, n. 7, p. 313–320, 2020.
- WANG, Z.; XING, W. Parallel machine scheduling with special jobs. **Tsinghua Science & Technology**, Elsevier, v. 11, n. 1, p. 107–110, 2006.
- YANG-KUEI, L.; CHI-WEI, L. Dispatching rules for unrelated parallel machine scheduling with release dates. **The International Journal of Advanced Manufacturing Technology**, Springer, v. 67, n. 1-4, p. 269–279, 2013.
- ZHANG, G.; XING, K. Differential evolution metaheuristics for distributed limited-buffer flowshop scheduling with makespan criterion. **Computers & Operations Research**, Elsevier, v. 108, p. 33–43, 2019.

ANEXOS

Anexo A – Tabelas de resultados computacionais

Tabela A.1 – Limitante Inferior (LB) para os 18 problemas avaliados.

N ^o	F	M_F	p_j	$n_{pj} = 50$	$n_{pj} = 100$	$n_{pj} = 200$
1	2	2	(50,70)	681,4109	1347,2513	2685,9731
2	2	5	(50,70)	259,0445	519,0160	1038,6879
3	2	(2,5)	(50,70)	390,6167	864,7629	1406,1416
4	2	2	(70,100)	953,6460	1892,8477	3735,6463
5	2	5	(70,100)	366,8559	732,1715	1452,4156
6	2	(2,5)	(70,100)	618,6449	1077,5902	2262,9599
7	3	2	(50,70)	446,4735	886,8959	1765,3184
8	3	5	(50,70)	191,9665	347,1262	694,0323
9	3	(2,5)	(50,70)	257,1302	482,2966	1061,1869
10	3	2	(70,100)	629,0662	1250,2843	2479,8265
11	3	5	(70,100)	276,6559	487,0308	973,1043
12	3	(2,5)	(70,100)	397,6271	688,8247	1411,9966
13	5	2	(50,70)	259,7245	519,8144	1036,3735
14	5	5	(50,70)	100,6716	202,7030	406,3194
15	5	(2,5)	(50,70)	176,9564	284,7886	612,5571
16	5	2	(70,100)	367,3948	731,4836	1458,0305
17	5	5	(70,100)	142,2134	284,7402	570,6065
18	5	(2,5)	(70,100)	216,9955	396,9511	870,5207

Fonte: Autora, 2021.

Tabela A.2 – Conjunto de 18 problemas, com 50 tarefas, para 4Fases-SPTteste.

Nº	F	M_F	p_j	Makespan	Min	RPD(%)	T(s)
1	2	2	(50,70)	687,5819	680,1726	0,9056	0,1954
2	2	5	(50,70)	263,7129	260,3414	1,8022	0,2018
3	2	(2,5)	(50,70)	398,5604	265,2911	2,0336	0,2007
4	2	2	(70,100)	959,8410	950,7364	0,6496	0,2018
5	2	5	(70,100)	373,0219	368,2035	1,6808	0,2063
6	2	(2,5)	(70,100)	625,8816	430,9921	1,1698	0,1990
7	3	2	(50,70)	448,8696	461,4023	0,5367	0,2175
8	3	5	(50,70)	194,9063	205,1520	1,5314	0,2313
9	3	(2,5)	(50,70)	259,3096	209,5037	0,8476	0,2290
10	3	2	(70,100)	634,7401	647,2240	0,9020	0,1967
11	3	5	(70,100)	280,0188	284,6095	1,2156	0,2189
12	3	(2,5)	(70,100)	420,9760	356,3906	5,8721	0,2227
13	5	2	(50,70)	263,9507	262,0509	1,6272	0,2277
14	5	5	(50,70)	105,7242	105,0904	5,0189	0,2512
15	5	(2,5)	(50,70)	185,0240	153,9192	4,5591	0,2408
16	5	2	(70,100)	373,3938	366,1768	1,6328	0,2232
17	5	5	(70,100)	147,9629	145,8587	4,0429	0,2392
18	5	(2,5)	(70,100)	219,6145	214,5542	1,2069	0,2380

Fonte: Autora, 2021.

Tabela A.3 – Conjunto de 18 problemas, com 100 tarefas, para 4Fases-SPTteste.

Nº	F	M_F	p_j	Makespan	Min	RPD(%)	T(s)
1	2	2	(50,70)	1349,9516	1312,4546	0,2004	0,2465
2	2	5	(50,70)	522,5131	520,6760	0,6738	0,2628
3	2	(2,5)	(50,70)	875,1250	609,0272	1,1983	0,2476
4	2	2	(70,100)	1896,5645	1864,9967	0,1964	0,2479
5	2	5	(70,100)	736,2993	728,3335	0,5638	0,2658
6	2	(2,5)	(70,100)	1080,6774	934,1105	0,2865	0,2678
7	3	2	(50,70)	890,2807	882,5271	0,3816	0,2627
8	3	5	(50,70)	349,6170	357,9156	0,7175	0,2951
9	3	(2,5)	(50,70)	488,4606	459,2376	1,2781	0,2638
10	3	2	(70,100)	1253,5953	1243,4361	0,2648	0,2612
11	3	5	(70,100)	489,9043	501,4165	0,5900	0,2815
12	3	(2,5)	(70,100)	724,0153	568,9616	5,1088	0,2596
13	5	2	(50,70)	523,1051	518,7088	0,6331	0,2810
14	5	5	(50,70)	207,0387	205,7616	2,1389	0,3330
15	5	(2,5)	(50,70)	302,0276	256,3351	6,0533	0,3148
16	5	2	(70,100)	735,0514	732,0027	0,4877	0,2790
17	5	5	(70,100)	290,3380	289,1564	1,9659	0,3260
18	5	(2,5)	(70,100)	399,0049	360,4259	0,5174	0,3394

Fonte: Autora, 2021.

Tabela A.4 – Conjunto de 18 problemas, com 200 tarefas, para 4Fases-SPTteste.

Nº	F	M_F	p_j	Makespan	Min	RPD(%)	T(s)
1	2	2	(50,70)	2688,4162	2657,9000	0,0910	0,3550
2	2	5	(50,70)	1041,2771	1036,9302	0,2493	0,4111
3	2	(2,5)	(50,70)	1409,4738	1039,2718	0,2370	0,4075
4	2	2	(70,100)	3738,2448	3629,6066	0,0696	0,4127
5	2	5	(70,100)	1455,7308	1445,3595	0,2283	0,2470
6	2	(2,5)	(70,100)	2265,9455	1469,7152	0,1319	0,3910
7	3	2	(50,70)	1768,1739	1751,6462	0,1618	0,4430
8	3	5	(50,70)	711,1903	708,4841	2,4722	0,4065
9	3	(2,5)	(50,70)	1069,7791	761,9687	0,8097	0,3951
10	3	2	(70,100)	2483,0560	2467,4975	0,1302	0,3840
11	3	5	(70,100)	997,0200	994,2513	0,4024	0,3970
12	3	(2,5)	(70,100)	1421,7057	998,0568	0,6876	0,3881
13	5	2	(50,70)	1039,0530	1035,1135	0,2585	0,4691
14	5	5	(50,70)	409,6740	408,0738	0,8256	0,5030
15	5	(2,5)	(50,70)	624,8473	557,6425	2,0064	0,4350
16	5	2	(70,100)	1461,0396	1451,8476	0,2064	0,4445
17	5	5	(70,100)	574,7392	572,8276	0,7243	0,5052
18	5	(2,5)	(70,100)	896,0194	703,8086	0,6317	0,4373

Fonte: Autora, 2021.

Tabela A.5 – Conjunto de 18 problemas, com 50 tarefas, para 4Fases-LPTteste.

Nº	F	M_F	p_j	Makespan	Min	RPD(%)	T(s)
1	2	2	(50,70)	695,6739	682,5032	2,0932	0,2504
2	2	5	(50,70)	266,2010	262,2132	2,7627	0,2631
3	2	(2,5)	(50,70)	405,4299	265,6158	3,7923	0,2663
4	2	2	(70,100)	963,7325	955,3042	1,0577	0,2463
5	2	5	(70,100)	376,7767	367,3496	2,7043	0,2639
6	2	(2,5)	(70,100)	634,1850	436,0035	2,5120	0,6180
7	3	2	(50,70)	452,9789	462,7728	1,4571	0,2514
8	3	5	(50,70)	199,4706	205,3197	3,9091	0,2746
9	3	(2,5)	(50,70)	260,5460	212,8101	1,3284	0,2618
10	3	2	(70,100)	640,6509	649,0534	1,8416	0,2532
11	3	5	(70,100)	283,8368	286,8476	2,5956	0,2710
12	3	(2,5)	(70,100)	424,5522	353,8975	6,7714	0,2695
13	5	2	(50,70)	265,4478	262,5549	2,2036	0,2756
14	5	5	(50,70)	106,8632	105,6442	6,1503	0,3144
15	5	(2,5)	(50,70)	186,2686	154,6223	5,2624	0,2847
16	5	2	(70,100)	374,6763	369,0818	1,9819	0,2662
17	5	5	(70,100)	150,2695	148,1193	5,6648	0,2975
18	5	(2,5)	(70,100)	221,3426	220,0749	2,0033	0,2782

Fonte: Autora, 2021.

Tabela A.6 – Conjunto de 18 problemas, com 100 tarefas, para 4Fases-LPTteste.

Nº	F	M_F	p_j	Makespan	Min	RPD(%)	T(s)
1	2	2	(50,70)	1350,9298	1329,2540	0,2730	0,3059
2	2	5	(50,70)	546,6291	520,6489	5,3203	0,3394
3	2	(2,5)	(50,70)	895,6325	613,4585	3,5697	0,3194
4	2	2	(70,100)	1902,3791	1857,0594	0,5035	0,3143
5	2	5	(70,100)	737,3454	728,5806	0,7067	0,3386
6	2	(2,5)	(70,100)	1087,5534	943,8275	0,9246	0,3083
7	3	2	(50,70)	892,2755	882,7072	0,6066	0,3137
8	3	5	(50,70)	351,9661	358,1869	1,3943	0,3604
9	3	(2,5)	(50,70)	492,9696	461,1817	2,2130	0,3236
10	3	2	(70,100)	1258,3213	1239,1428	0,6428	0,3156
11	3	5	(70,100)	491,4537	505,3908	0,9081	0,3536
12	3	(2,5)	(70,100)	737,6391	571,4528	7,0866	0,3254
13	5	2	(50,70)	523,4882	519,6239	0,7068	0,3389
14	5	5	(50,70)	217,5703	206,1567	7,3345	0,4194
15	5	(2,5)	(50,70)	303,6990	256,8594	6,6402	0,3754
16	5	2	(70,100)	768,8899	734,3293	5,1138	0,3366
17	5	5	(70,100)	299,3414	289,9439	5,1279	0,4141
18	5	(2,5)	(70,100)	402,1815	365,5003	1,3176	0,3686

Fonte: Autora, 2021.

Tabela A.7 – Conjunto de 18 problemas, com 200 tarefas, para 4Fases-LPTteste.

Nº	F	M_F	p_j	Makespan	Min	RPD(%)	T(s)
1	2	2	(50,70)	2692,8676	2616,3097	0,2567	0,4464
2	2	5	(50,70)	1075,8558	1036,4137	3,5784	0,5398
3	2	(2,5)	(50,70)	1425,7828	1035,8652	1,3968	0,5880
4	2	2	(70,100)	3757,2817	3643,5119	0,5792	0,5678
5	2	5	(70,100)	1479,8820	1440,0474	1,8911	0,6847
6	2	(2,5)	(70,100)	2285,0903	1466,2428	0,9779	0,5747
7	3	2	(50,70)	1786,0808	1770,0044	1,1761	0,4575
8	3	5	(50,70)	715,4176	712,1085	3,0813	0,5219
9	3	(2,5)	(50,70)	1074,7658	765,3757	1,2796	0,5103
10	3	2	(70,100)	2489,4382	2447,1306	0,3876	0,5007
11	3	5	(70,100)	985,1556	995,4356	1,2384	0,5370
12	3	(2,5)	(70,100)	1442,0151	1018,4688	2,1260	0,5757
13	5	2	(50,70)	1060,5372	1032,8545	2,3316	0,5661
14	5	5	(50,70)	431,5251	409,1235	6,2034	0,7145
15	5	(2,5)	(50,70)	637,4124	560,1978	4,0576	0,5393
16	5	2	(70,100)	1527,1759	1444,7624	4,7424	0,6318
17	5	5	(70,100)	600,0731	575,4674	5,1641	0,6655
18	5	(2,5)	(70,100)	886,7180	710,1499	1,8606	0,5486

Fonte: Autora, 2021.

Tabela A.8 – Conjunto de 18 problemas, com 50 tarefas, para 4Fases-FCFSteste.

Nº	F	M_F	p_j	Makespan	Min	RPD(%)	T(s)
1	2	2	(50,70)	693,2941	679,1257	1,7439	0,2350
2	2	5	(50,70)	265,3721	262,0919	2,4427	0,2403
3	2	(2,5)	(50,70)	402,5279	263,5667	3,0493	0,2402
4	2	2	(70,100)	961,3687	954,8390	0,8098	0,2419
5	2	5	(70,100)	373,3197	367,3865	1,7619	0,2380
6	2	(2,5)	(70,100)	632,9101	438,2421	2,3059	0,2318
7	3	2	(50,70)	452,7910	465,0515	1,4150	0,2315
8	3	5	(50,70)	195,4428	204,0952	1,8109	0,2353
9	3	(2,5)	(50,70)	259,1926	212,6045	0,8021	0,2334
10	3	2	(70,100)	637,2272	648,5401	1,2973	0,2334
11	3	5	(70,100)	281,3602	286,2912	1,7004	0,2401
12	3	(2,5)	(70,100)	423,3112	355,5207	6,4593	0,2421
13	5	2	(50,70)	262,5081	262,9553	1,0718	0,2384
14	5	5	(50,70)	106,5147	104,5158	5,8041	0,2364
15	5	(2,5)	(50,70)	185,8772	153,6413	5,0412	0,2349
16	5	2	(70,100)	380,1851	370,3187	3,4814	0,2318
17	5	5	(70,100)	149,5601	146,9419	5,1660	0,2383
18	5	(2,5)	(70,100)	220,8673	214,3815	1,7843	0,2364

Fonte: Autora, 2021.

Tabela A.9 – Conjunto de 18 problemas, com 100 tarefas, para 4Fases-FCFSteste.

Nº	F	M_F	p_j	Makespan	Min	RPD(%)	T(s)
1	2	2	(50,70)	1351,1695	1326,1860	0,2908	0,2688
2	2	5	(50,70)	533,4780	520,5487	2,7864	0,2616
3	2	(2,5)	(50,70)	879,6809	612,5859	1,7251	0,2697
4	2	2	(70,100)	1898,2916	1868,9975	0,2876	0,2815
5	2	5	(70,100)	736,5674	732,4755	0,6004	0,2686
6	2	(2,5)	(70,100)	1080,4946	942,3200	0,2695	0,2692
7	3	2	(50,70)	895,4588	887,6368	0,9655	0,2679
8	3	5	(50,70)	350,1612	358,4831	0,8743	0,2728
9	3	(2,5)	(50,70)	489,5988	459,2320	1,5140	0,2548
10	3	2	(70,100)	1257,3174	1242,2109	0,5625	0,2805
11	3	5	(70,100)	490,3244	502,0053	0,6763	0,2744
12	3	(2,5)	(70,100)	732,1811	576,6118	6,2943	0,2595
13	5	2	(50,70)	525,4896	520,1168	1,0918	0,2654
14	5	5	(50,70)	208,1039	206,9305	2,6644	0,2715
15	5	(2,5)	(50,70)	303,1796	256,8901	6,4578	0,2699
16	5	2	(70,100)	760,3181	732,4775	3,9419	0,2697
17	5	5	(70,100)	298,2070	289,8481	4,7295	0,2738
18	5	(2,5)	(70,100)	400,1805	366,1633	0,8136	0,2679

Fonte: Autora, 2021.

Tabela A.10 – Conjunto de 18 problemas, com 200 tarefas, para 4Fases-FCFS teste.

Nº	F	M_F	p_j	Makespan	Min	RPD(%)	T(s)
1	2	2	(50,70)	2689,9560	2656,5461	0,1483	0,3924
2	2	5	(50,70)	1062,7437	1064,9763	2,3160	0,3445
3	2	(2,5)	(50,70)	1422,0119	1066,2139	1,1286	0,3775
4	2	2	(70,100)	3756,2849	3654,8981	0,5525	0,4054
5	2	5	(70,100)	1486,2450	1470,4931	2,3292	0,3485
6	2	(2,5)	(70,100)	2273,6659	1499,0296	0,4731	0,4074
7	3	2	(50,70)	1783,2689	1747,4244	1,0168	0,4282
8	3	5	(50,70)	713,6356	710,3889	2,8246	0,3422
9	3	(2,5)	(50,70)	1063,4088	763,4833	0,2094	0,3398
10	3	2	(70,100)	2483,5638	2442,7984	0,1507	0,4729
11	3	5	(70,100)	980,5733	993,8558	0,7675	0,3473
12	3	(2,5)	(70,100)	1435,8725	1001,0945	1,6909	0,3540
13	5	2	(50,70)	1039,1481	1041,3244	0,2677	0,3349
14	5	5	(50,70)	429,6671	410,3643	5,7461	0,3290
15	5	(2,5)	(50,70)	636,7917	559,3162	3,9563	0,3489
16	5	2	(70,100)	1501,6360	1492,7161	2,9907	0,3309
17	5	5	(70,100)	582,3030	576,5230	2,0498	0,3529
18	5	(2,5)	(70,100)	881,6061	704,8652	1,2734	0,3534

Fonte: Autora, 2021.

Tabela A.11 – Conjunto de 18 problemas com 50 tarefas.

No.	F	M_f	p_j	LB	Algoritmo Genético			Heurística	
					Makespan	ARPD	Min	Makespan	HRPD
1	2	2	(50,70)	678,236	707,968	4,4	701,657	694,422	2,4
2	2	5	(50,70)	265,283	283,646	6,9	265,283	272,702	2,8
3	2	(2,5)	(50,70)	366,992	404,596	10,2	376,009	375,210	2,2
4	2	2	(70,100)	992,675	994,370	0,2	993,004	1020,200	2,8
5	2	5	(70,100)	424,236	441,016	4,0	432,313	428,904	1,1
6	2	(2,5)	(70,100)	568,244	629,209	10,7	585,755	581,356	2,3
7	3	2	(50,70)	476,924	492,183	3,2	486,537	509,625	6,9
8	3	5	(50,70)	183,561	239,261	30,3	210,997	208,555	13,6
9	3	(2,5)	(50,70)	301,604	381,691	26,6	354,639	310,904	3,1
10	3	2	(70,100)	633,060	639,284	1,0	634,239	654,445	3,4
11	3	5	(70,100)	264,431	352,506	33,3	324,340	309,347	17,0
12	3	(2,5)	(70,100)	427,553	558,958	30,7	513,883	456,459	6,8
13	5	2	(50,70)	265,654	270,330	1,8	267,417	281,404	5,9
14	5	5	(50,70)	107,954	158,294	46,6	142,851	111,165	3,0
15	5	(2,5)	(50,70)	131,192	228,466	74,1	214,831	163,480	24,6
16	5	2	(70,100)	405,320	410,527	1,3	408,672	412,751	1,8
17	5	5	(70,100)	152,802	226,667	48,3	220,269	158,042	3,4
18	5	(2,5)	(70,100)	199,031	355,985	78,9	337,570	244,290	22,7

Fonte: Adaptado de Behnamian e Ghomi (2013).

Tabela A.12 – Conjunto de 18 problemas com 100 tarefas.

No.	F	M_f	p_j	LB	Algoritmo Genético			Heurística	
					Makespan	ARPD	Min	Makespan	HRPD
1	2	2	(50,70)	1349,90	1350,817	0,1	1350,368	1350,700	0,1
2	2	5	(50,70)	558,549	589,399	5,5	558,549	592,474	6,1
3	2	(2,5)	(50,70)	841,564	903,659	7,4	867,708	872,115	3,6
4	2	2	(70,100)	1871,30	1872,580	0,1	1871,624	1901,900	1,6
5	2	5	(70,100)	841,198	1356,008	61,2	1154,348	845,432	0,5
6	2	(2,5)	(70,100)	1126,70	1190,213	5,6	1143,007	1148,700	2,0
7	3	2	(50,70)	955,759	963,691	0,8	957,538	972,321	1,7
8	3	5	(50,70)	361,164	502,925	39,3	438,439	376,145	4,1
9	3	(2,5)	(50,70)	579,505	718,326	24,0	629,326	600,646	3,6
10	3	2	(70,100)	1297,90	1313,241	1,2	1303,043	1314,500	1,3
11	3	5	(70,100)	528,867	657,610	24,3	607,162	553,406	4,6
12	3	(2,5)	(70,100)	911,827	1089,721	19,5	992,061	911,882	0,0
13	5	2	(50,70)	562,788	575,914	2,3	568,970	595,479	5,8
14	5	5	(50,70)	216,190	268,402	24,2	253,742	230,027	6,4
15	5	(2,5)	(50,70)	339,409	442,077	30,2	403,983	359,829	6,0
16	5	2	(70,100)	768,333	786,033	2,3	780,047	799,225	4,0
17	5	5	(70,100)	305,671	363,900	19,0	314,218	321,831	5,3
18	5	(2,5)	(70,100)	466,117	1350,817	0,1	1350,368	506,574	8,7

Fonte: Adaptado de Behnamian e Ghomi (2013).

Tabela A.13 – Conjunto de 18 problemas com 200 tarefas.

No.	F	M_f	p_j	LB	Algoritmo Genético			Heurística	
					Makespan	ARPD	Min	Makespan	HRPD
1	2	2	(50,70)	2801,6	2803,513	0,07	2801,797	2858,4	2,03
2	2	5	(50,70)	1110,8	1145,181	3,1	1110,800	1149,1	3,45
3	2	(2,5)	(50,70)	1760,7	1874,805	6,48	1825,341	1849,0	5,02
4	2	2	(70,100)	3735,6	3740,409	0,13	3736,719	3,78E+03	1,19
5	2	5	(70,100)	1623,5	1714,778	5,62	1688,918	1700,3	4,73
6	2	(2,5)	(70,100)	2376,1	2386,38	0,43	2381,588	2407,8	1,33
7	3	2	(50,70)	1847,3	1919,882	3,93	1912,082	1941,9	5,12
8	3	5	(50,70)	759,82	826,6264	8,79	798,236	777,8241	2,37
9	3	(2,5)	(50,70)	1103,2	1162,253	5,35	1143,987	1148,9	4,14
10	3	2	(70,100)	2570,4	2586,55	0,63	2571,692	2733,2	6,33
11	3	5	(70,100)	1053,1	1113,93	5,78	1087,638	1086,4	3,16
12	3	(2,5)	(70,100)	1538,9	1633,655	6,16	1589,087	1655,4	7,57
13	5	2	(50,70)	1112,5	1142,881	2,73	1136,739	1184,6	6,48
14	5	5	(50,70)	443,32	512,7672	15,66	497,541	465,2646	4,95
15	5	(2,5)	(50,70)	634,40	905,5642	42,74	798,945	678,9621	7,02
16	5	2	(70,100)	1551,6	1581,666	1,94	1561,619	1634,2	5,32
17	5	5	(70,100)	619,86	708,6224	14,32	680,495	644,4225	3,96
18	5	(2,5)	(70,100)	907,36	1207,134	33,04	1051,132	985,7475	8,64

Fonte: Adaptado de Behnamian e Ghomi (2013).