

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

Walter Greff Rodrigues Neto

**UMA PROPOSTA DE RESOLUÇÃO PARA O PROBLEMA DE
SEQUENCIAMENTO DE TAREFAS EM MÁQUINAS PARALELAS
CONSIDERANDO DESGASTES DEPENDENTES DA SEQUÊNCIA**

Santa Maria, RS
2020

Walter Greff Rodrigues Neto

**UMA PROPOSTA DE RESOLUÇÃO PARA O PROBLEMA DE SEQUENCIAMENTO DE
TAREFAS EM MÁQUINAS PARALELAS CONSIDERANDO DESGASTES
DEPENDENTES DA SEQUÊNCIA**

Dissertação de Mestrado apresentada ao 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 **Mestre em ENGENHARIA DE PRODUÇÃO.**

ORIENTADOR: Prof. Olinto César Bassi de Araújo

Santa Maria, RS
2020

Rodrigues Neto, Walter Greff
UMA PROPOSTA DE RESOLUÇÃO PARA O PROBLEMA DE
SEQUENCIAMENTO DE TAREFAS EM MÁQUINAS PARALELAS
CONSIDERANDO DESGASTES DEPENDENTES DA SEQUÊNCIA / Walter
Greff Rodrigues Neto.- 2020.
51 p.; 30 cm

Orientador: Olinto César Bassi de Araújo
Dissertação (mestrado) - Universidade Federal de Santa
Maria, Centro de Tecnologia, Programa de Pós-Graduação em
Engenharia de Produção, RS, 2020

1. Máquinas Paralelas 2. Programação Inteira 3. Modelo
Matemático 4. Programação de Tarefas 5. Manutenção I. de
Araújo, Olinto César Bassi II. Título.

Sistema de geração automática de ficha catalográfica da UFSM. Dados fornecidos pelo autor(a). Sob supervisão da Direção da Divisão de Processos Técnicos da Biblioteca Central. Bibliotecária responsável Paula Schoenfeldt Patta CRB 10/1728.

©2020

Todos os direitos autorais reservados a Walter Greff Rodrigues Neto. A reprodução de partes ou do todo deste trabalho só poderá ser feita mediante a citação da fonte.

Endereço: Av. Roraima, 1000, Camobi

End. Eletr.: waltergrneto@gmail.br

Walter Greff Rodrigues Neto

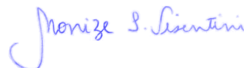
**UMA PROPOSTA DE RESOLUÇÃO PARA O PROBLEMA DE SEQUENCIAMENTO DE
TAREFAS EM MÁQUINAS PARALELAS CONSIDERANDO DESGASTES
DEPENDENTES DA SEQUÊNCIA**

Dissertação de Mestrado apresentada ao 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 **Mestre em ENGENHARIA DE PRODUÇÃO.**

Aprovado em 30 de março de 2020:



Olinto César Bassi de Araújo, Dr. (UFSM)
(Presidente/Orientador)



Monize Sâmara Visentini - Videoconferência, Dra. (UFFS)



Guilherme Dhein - Videoconferência, Dr. (UFSM)

Santa Maria, RS
2020

AGRADECIMENTOS

O fim desta etapa traz a oportunidade de agradecer pessoas que foram importantes não só ao longo desta jornada, mas desde sempre.

Agradeço aos meus avós (in memorian), Olga e Walter, pelo amor e pelos ensinamentos.

Aos meus irmãos, de fato, não de direito, Felipe (Japa), Marcel, Rafael e Romário, minha família de verdade.

À Jeline, pelo amor, paciência, amizade, paciência, conselhos, (eu já disse paciência?) e imenso apoio ao longo de todo este processo, inclusive com revisões de texto intermináveis.

À June, minha sogra/mãe, aos meus padrinhos, Hermes e Jeanine, ao Da Silva, pelos comentários nada tendenciosos durante os jogos do Inter, à Jane Elisa, pelo fornecimento de ambrosia e a toda a galera das famílias Moura e Tellier.

Aos membros do Grupo de Otimização Aplicada (G.Ot.A), com os quais tive um ótimo período de convivência, estudos (não muito) e piadas (até demais).

E, finalmente, agradeço ao Prof. Olinto, pela amizade, pelos conselhos, puxões de orelha e pelo fato de ser o homem com mais fé na face da Terra, afinal, ele sempre acreditou que eu poderia concluir este trabalho.

RESUMO

UMA PROPOSTA DE RESOLUÇÃO PARA O PROBLEMA DE SEQUENCIAMENTO DE TAREFAS EM MÁQUINAS PARALELAS CONSIDERANDO DESGASTES DEPENDENTES DA SEQUÊNCIA

AUTOR: Walter Greff Rodrigues Neto

ORIENTADOR: Olinto César Bassi de Araújo

Problemas de escalonamento de tarefas consideram situações nas quais é necessário definir a sequência em que um conjunto de tarefas será processado em uma ou mais máquinas, de modo a minimizar (maximizar) uma função objetivo. Diferentes restrições oriundas de aplicações reais podem ser incorporadas na formulação de problemas de escalonamento de tarefas, dentre elas os desgastes dependentes da sequência e o tempo de parada programada para manutenção. O foco deste trabalho é o Problema de Sequenciamento de Tarefas em Máquinas Paralelas considerando Desgastes Dependentes da Sequência que tem como objetivo a minimização do tempo necessário para processar a última tarefa programada ($R_m|S_{dd}|C_{max}$). Neste problema, a execução de uma tarefa resulta na deterioração no nível de performance da máquina para a execução das tarefas seguintes. Em específico, este estudo considera o problema $R_m|S_{dd}, Mnt|C_{max}$, no qual eventos de parada programada para manutenção são incluídos. Dois modelos matemáticos, baseados em programação linear inteira mista, são propostos. O primeiro, $ADF+$, é voltado para o problema $R_m|S_{dd}|C_{max}$ e o segundo, $ADFM$, é direcionado para o problema $R_m|S_{dd}, Mnt|C_{max}$. Testes computacionais conduzidos para avaliar o desempenho dos modelos propostos revelaram que ambos modelos produziram melhores resultados, tanto em qualidade de solução como em tempos computacionais, quando comparados com as abordagens propostas na literatura.

Palavras-chave: Máquinas Paralelas. Programação Inteira. Modelo Matemático. Programação de Tarefas. Manutenção.

ABSTRACT

SOLVING THE PARALLEL MACHINE SCHEDULING PROBLEM WITH SEQUENCE-DEPENDENT DETERIORATION EFFECTS

AUTHOR: Walter Greff Rodrigues Neto

ADVISOR: Olinto César Bassi de Araújo

Machine scheduling problems consider situations where it is necessary to define the sequence in which a set of jobs is processed in one or more machines to minimize (maximize) an objective function. It is possible to incorporate several restrictions from real applications into the formulations of these problems, including sequence dependent machine deterioration and maintenance events. This work focus on the parallel machine scheduling problem with sequence dependent deteriorating effects to minimize the makespan ($R_m|S_{dd}|C_{max}$). In this problem, job processing results in performance deterioration of the machine for the processing of the following jobs. Specifically, this study considers the problem $R_m|S_{dd}, Mnt|C_{max}$, in which maintenance events are considered. We propose two mathematical models based in Mixed Integer Linear Programming. The first one, $ADF+$, is applied to solve $R_m|S_{dd}|C_{max}$ and the second, $ADFM$, is applied to solve $R_m|S_{dd}, Mnt|C_{max}$. We conducted computational tests to evaluate the performance of the proposed models, revealing that they returned the best results, both in quality and in computational time, when compared with approaches from the literature.

Keywords: Parallel Machines. Integer Programming. Mathematical Model. Machine Scheduling. Maintenance.

LISTA DE FIGURAS

| | |
|---|----|
| Figura 1 – Exemplos de sequenciamento de tarefas (a) sem considerar desgaste e (b) considerando desgaste dependente da sequência | 18 |
| Figura 2 – Exemplo de busca pela minimização do <i>makespan</i> no sequenciamento de tarefas considerando desgastes dependentes da sequência | 19 |
| Figura 3 – Exemplo de alocação de tarefas em posição predeterminadas | 27 |
| Figura 4 – Exemplo de sequenciamento de tarefas em máquinas paralelas com desgaste dependente da sequência e eventos de parada programada para manutenção | 30 |
| Figura 5 – Comparação de sequenciamento de tarefas com e sem a utilização de manutenção | 32 |
| Figura 6 – Diferenças entre os modelos <i>RPP</i> , <i>ADF</i> e <i>ADF+</i> | 37 |
| Figura 7 – Diferenças entre os modelos de Ruiz-Torres, Paletta e M'Hallah (2017), <i>IM</i> , <i>ADF+</i> e <i>ADFM</i> | 39 |

LISTA DE GRÁFICOS

- Gráfico 1 – Gráfico contendo o percentual de soluções encontradas no menor tempo por cada um dos modelos ADF e $ADF+$. A área "Tempos iguais" representa o percentual de soluções com tempos de processamento idênticos. 42
- Gráfico 2 – Gráfico contendo o percentual de soluções encontradas com menor gap por cada um dos modelos ADF e $ADF+$. A área tempos "Gaps idênticos" representa o percentual de soluções com os mesmos gaps para ambos os modelos. 43
- Gráfico 3 – Gráfico contendo o total de soluções encontradas por $ADFM$, soluções "empatadas" e por IM , respectivamente, 470, 7 e 3 46

LISTA DE TABELAS

| | |
|---|----|
| Tabela 1 – Dados utilizados para a elaboração do exemplo ilustrado na Figura 1 | 17 |
| Tabela 2 – Dados utilizados para a elaboração do exemplo ilustrado na Figura 4 | 30 |
| Tabela 3 – Dados utilizados para a elaboração do exemplo ilustrado na Figura 5 | 31 |
| Tabela 4 – Comparação dos Resultados dos Modelos ADF e $ADF+$ | 40 |
| Tabela 5 – Comparativo entre as soluções obtidas por $ADFM$ e os Resultados apresentados por Mendes e Iori (2019) | 44 |

SUMÁRIO

| | | |
|--------------|---|-----------|
| 1 | INTRODUÇÃO | 9 |
| 1.1 | OBJETIVOS | 10 |
| 1.2 | JUSTIFICATIVA | 11 |
| 2 | REFERENCIAL TEÓRICO | 12 |
| 2.1 | PROGRAMAÇÃO DE TAREFAS | 12 |
| 2.2 | PROBLEMAS DE SEQUENCIAMENTO DE TAREFAS CONSIDERANDO DESGASTE | 14 |
| 2.2.1 | Definição do Problema $R_m S_{dd} C_{max}$ | 16 |
| 2.2.2 | Modelo Matemático em Programação Não Linear Inteira Mista | 20 |
| 2.2.3 | Linearização do Modelo RPP e Ordem de Alocação das Tarefas | 21 |
| 2.3 | PROBLEMAS DE SEQUENCIAMENTO DE TAREFAS CONSIDERANDO EVENTOS DE MANUTENÇÃO | 28 |
| 2.3.1 | Definição do Problema $R_m S_{dd}, Mnt C_{max}$ | 29 |
| 2.3.2 | Modelo Matemático proposto por Mendes e Iori (2019) | 33 |
| 3 | RESULTADOS E DISCUSSÃO | 35 |
| 3.1 | MODELOS PROPOSTOS | 35 |
| 3.1.1 | Proposta de Melhoria do modelo ADF: o Modelo ADF+ | 35 |
| 3.1.2 | Proposta de modelo para o $R_m S_{dd}, Mnt C_{max}$ | 37 |
| 3.2 | RESULTADOS COMPUTACIONAIS | 39 |
| 3.3 | COMPARAÇÃO DOS RESULTADOS OBTIDOS PELOS MODELOS <i>ADF</i> E <i>ADF+</i> | 40 |
| 3.4 | COMPARAÇÃO DOS RESULTADOS OBTIDOS PELO MODELO <i>ADFM</i> E PELA METODOLOGIA PROPOSTA POR MENDES E IORI (2019) | 43 |
| 4 | CONCLUSÃO | 48 |
| | REFERÊNCIAS BIBLIOGRÁFICAS | 49 |

1 INTRODUÇÃO

Problemas de escalonamento de tarefas consideram situações nas quais se deseja definir a sequência em que um conjunto de tarefas deve ser designado para execução em uma ou mais máquinas. Entre os objetivos mais considerados para a definição da melhor configuração está a minimização do *makespan*, ou seja, do tempo necessário para processar todas as tarefas alocadas na máquina mais carregada.

Em aplicações reais, é esperado que algum tipo de desgaste ocorra na máquina devido ao processamento das tarefas. Em algumas atividades, este desgaste é bastante lento, de modo que seus efeitos podem ser desprezados no contexto de um planejamento com horizonte limitado. Já em outros casos, este efeito é mais pronunciado e impacta significativamente o desempenho das máquinas, em específico, no tempo de processamento de duas tarefas em sequência. Desprezar essa deterioração pode tornar soluções de escalonamento inviáveis em uma aplicação real.

A deterioração de máquinas dependente da sequência de tarefas processadas é uma característica que ocorre no chão de fábrica. Um exemplo em que esta deterioração acontece é o processo de corte ou perfuração em materiais resistentes. O desgaste em serras, lâminas ou brocas, decorrentes do processamento de uma tarefa, pode afetar diretamente a tarefa seguinte, causando um aumento no tempo necessário para a sua realização. Dependendo do material envolvido nas tarefas, a sequência de processamento pode levar a diferentes graus de desgaste na máquina. A deterioração no nível de desempenho também pode acontecer com pessoas ou equipes, em decorrência de cansaço após o esforço para executar trabalhos anteriores, por exemplo. Nesse último caso, não é difícil intuir que o grau de desgaste pode ser diretamente relacionado com o sequenciamento das tarefas.

O presente trabalho tem como base o problema de escalonamento de tarefas proposto por Ruiz-Torres, Paletta e Pérez (2013), denominado Problema de Sequenciamento de Tarefas em Máquinas Paralelas considerando Desgastes Dependentes da Sequência. Neste problema, a execução de uma tarefa resulta na deterioração no nível de performance da máquina para a execução das tarefas seguintes. Os autores apresentam um modelo de programação matemática com duas restrições não lineares, além de resultados para dois conjuntos de instâncias. O primeiro conjunto considera instâncias de pequena porte (até 14 tarefas, 2, 3 e 4 máquinas), para as quais são apresentadas soluções ótimas obtidas por força bruta (enumeração completa), o segundo conjunto consta de instâncias de grande porte (até 50 tarefas, 4, 7 e 10 máquinas), para as quais soluções (melhores conhecidas até então) são obtidas através da meta-heurística *Simulated Annealing*.

Conforme a classificação apresentada em Graham et al. (1979), pode-se denotar o problema como $R_m|S_{dd}|C_{max}$, onde o primeiro campo corresponde às características do

maquinário utilizado, o segundo diz respeito aos detalhes do processamento das tarefas e das restrições do problema e o terceiro identifica o critério de otimização adotado. Neste caso, respectivamente, tem-se máquinas paralelas não relacionadas, representadas por R_m , desgastes dependentes da sequência, denotados como S_{dd} , e a minimização do *makespan*, identificado por C_{max} .

Uma linearização do modelo proposto por Ruiz-Torres, Paletta e Pérez (2013) para o problema $R_m|S_{dd}|C_{max}$ foi apresentada por Araújo, Dhein e Fampa (2017), e possibilitou, encontrar a solução ótima de 650 das 900 instâncias do conjunto *benchmark* de grande porte. Isto é, o modelo linear passou a apresentar os melhores resultados conhecidos.

Este trabalho ainda considera o problema proposto por Ruiz-Torres, Paletta e M'Hallah (2017), o Problema de Sequenciamento de Tarefas em Máquinas Paralelas considerando Degastes Dependentes da Sequência e Eventos de Manutenção, $R_m|S_{dd}, Mnt|C_{max}$. Neste, além das considerações existentes em Ruiz-Torres, Paletta e Pérez (2013), é acrescentada a existência de eventos de manutenção que, quando ocorrerem, restauram o nível de desempenho da máquina para seu patamar máximo. Os autores propõem um modelo matemático em programação não linear inteira mista e heurísticas construtivas para a resolução do problema. Mendes e Iori (2019) propõem uma versão linearizada do referido modelo e, com a utilização de meta-heurísticas baseadas em Busca Local Iterada (*Iterated Local Search*), produzem as melhores soluções conhecidas para um conjunto *benchmark* com 480 instâncias.

A estrutura desta dissertação começa com estudo bibliográfico para a compreensão do estado da arte dos problemas de planejamento e sequenciamento de tarefas em máquinas paralelas. A etapa seguinte consiste na definição e formulação de um modelo de programação matemática para o problema $R_m|S_{dd}|C_{max}$. Este mesmo processo será aplicado para o problema que é a proposta central deste estudo, $R_m|S_{dd}, Mnt|C_{max}$.

Por fim, testes computacionais são necessários para validar os modelos propostos. Para tais testes serão utilizadas instâncias da literatura da área de *Scheduling*. A saber, para $R_m|S_{dd}|C_{max}$ será utilizado o conjunto *benchmark* proposto por Ruiz-Torres, Paletta e Pérez (2013), que também foi utilizado em Araújo, Dhein e Fampa (2017). Para o problema $R_m|S_{dd}, Mnt|C_{max}$ o conjunto de instâncias utilizado será o mesmo de Mendes e Iori (2019), gerado em conformidade com as premissas de Ruiz-Torres, Paletta e M'Hallah (2017).

1.1 OBJETIVOS

O objetivo geral deste trabalho é propor um modelo matemático linear para a resolução do Problema de Sequenciamento de Tarefas em Máquinas Paralelas considerando Degastes Dependentes da Sequência e Tempo de Parada Programada para Manutenção.

Como objetivos específicos este trabalho pretende:

- Melhorar o modelo proposto por Araújo, Dhein e Fampa (2017) para o problema $R_m|S_{dd}|C_{max}$;
- Adaptar a melhoria supracitada do modelo proposto por Araújo, Dhein e Fampa (2017) para o problema $R_m|S_{dd}, Mnt|C_{max}$.
- Comparar os resultados obtidos para o problema $R_m|S_{dd}, Mnt|C_{max}$ com os resultados obtidos pela abordagem apresentada em Mendes e Iori (2019);

1.2 JUSTIFICATIVA

As propostas de resolução para os problemas de Programação de Tarefas, dada sua dificuldade inerente para encontrar boas soluções, muitas vezes são baseadas em abordagens heurísticas e meta-heurísticas. Resultados como os apresentados em Araújo, Dhein e Fampa (2017), obtidos através de um modelo em Programação Linear Inteira Mista (PLIM), demonstram que formulações fortes podem ser competitivas, tanto pela qualidade das soluções como pelos tempos computacionais despendidos. Tal situação estimula o desafio de tentar melhorar um modelo que já possui resultados interessantes, além de trazer a possibilidade de atingir resultados competitivos ao aplicar uma adaptação do mesmo para problemas com características extras.

O estudo apresentado por Mendes e Iori (2019), que trata do Sequenciamento de Tarefas em Máquinas Paralelas com Desgastes Dependentes da Sequência e Tempo de Parada Programada para Manutenção, traz a proposta de um modelo em PLIM, mas os resultados para a totalidade das instâncias do conjunto teste são obtidos com o uso da meta-heurística de Busca Local Iterada, ou seja, não garantem a otimalidade dos mesmos. Assim, este estudo irá basear-se na proposta de melhoria do modelo constante no já citado estudo de Araújo, Dhein e Fampa (2017) buscando adaptá-lo para resolução do problema abordado por Mendes e Iori (2019), pois proposições de modelagens independentes de heurísticas, para este problema, são escassas na literatura e representam uma aplicação prática de grande utilidade.

2 REFERENCIAL TEÓRICO

Os problemas de Programação de Tarefas (*Machine Scheduling Problems*) contemplam uma infinidade de diferentes situações, como a consideração de máquina única ou máquinas paralelas, produção contínua ou intermitente etc. Há, ainda, possibilidade de levar em conta especificidades como desgastes, manutenções, efeitos de aprendizagem, atribuição de penalidades por atraso ou adiantamento na entrega de uma tarefa, dentre outras (PINEDO, 2016). Devido a esta extensa gama de possibilidades, as subseções a seguir apresentam os principais conceitos desta temática, com ênfase aos que serão diretamente abordados neste estudo.

A Seção 2.1 trata da programação de tarefas, além de conter definições sobre o ambiente das máquinas e sobre os objetivos usualmente considerados nos problemas de otimização. As Seções 2.2 e 2.3 concentram-se, respectivamente, na apresentação de problemas já abordados na literatura sobre sequenciamento de tarefas considerando desgastes e sequenciamento de tarefas e eventos de manutenção.

2.1 PROGRAMAÇÃO DE TAREFAS

Segundo Herrmann (2006), a Programação de Tarefas se refere ao processo de alocação de recursos escassos ao longo do tempo para atingir o(s) objetivo(s) estabelecidos durante o processo de tomada de decisão. Pinedo (2016) descreve a Programação de Tarefas como um processo de tomada de decisão com papel de grande importância nas mais variadas áreas da indústria. Este processo tem como função primordial a otimização de um ou mais objetivos através da alocação de recursos para a execução de tarefas em determinados períodos de tempo.

É necessário conhecer o tipo e a quantidade de cada um dos recursos para, então, determinar a maneira como as tarefas poderão ser cumpridas, de modo a se respeitar a factibilidade da solução. Ao especificar recursos, constroem-se os limites do problema e, adicionalmente, deve-se descrever cada uma das tarefas com informações relativas à sua demanda por estes recursos e citar quaisquer restrições de ordem tecnológica que podem significar um gargalo para sua realização como, por exemplo, a necessidade de precedência entre elas (BAKER; TRIETSCH, 2013).

Dado que a literatura traz uma infinidade de definições e restrições para problemas de programação de tarefas, algumas destas particularidades serão descritas a seguir, como o ambiente de máquinas e os principais objetivos trabalhados, com ênfase em máquinas paralelas.

As máquinas envolvidas no problema podem ter características distintas. Primeira-

mente, podemos considerar o número de máquinas analisadas. Há na literatura estudos que consideram máquina única, duas máquinas ou m máquinas, por exemplo. Os primeiros problemas tratando de máquina única são datados da década de 1950 (GUPTA; KYPARISIS, 1987). Este problema trata de casos mais simples e especializados dentre os ambientes de máquinas, e sua importância se dá tanto porque problemas mais complexos são decompostos em problemas de máquina única, quanto pelo fato de que pode prover uma base para heurísticas aplicáveis em casos mais complicados (PINEDO, 2016).

Dentre os diversos trabalhos que consideram uma única máquina, pode-se citar Gupta e Kyparisis (1987) que realizam uma revisão sobre três décadas de pesquisas relativas ao tema, Gupta, Kunnathur e Dandapani (1987) que propõem um problema na área de gestão financeira, e Browne e Yechiali (1990) e Cai, Cai e Zhu (1998), cujos trabalhos serão abordados mais detalhadamente na Seção 2.2, que tratam de problemas considerando desgastes. Além destes, Biskup (1999) apresenta um problema com efeitos de aprendizagem, Low, Hsu e Su (2008) levam em consideração restrições de disponibilidade da máquina, ao passo que Nesello et al. (2018) apresentam a solução exata para o problema que considera manutenções periódicas e tempos de preparação dependentes da sequência.

Casos envolvendo duas ou mais máquinas paralelas são importantes em modelagens que buscam se aproximar da realidade, uma vez que é uma situação comum no chão de fábrica. As máquinas podem ter características iguais e, neste caso, são ditas idênticas. Lawler et al. (1993) observam que problemas com máquinas paralelas idênticas são os mais estudados nas publicações que consideram múltiplas máquinas. Não seria errado afirmar que este cenário ainda é verdadeiro. No entanto, o foco deste trabalho está em problemas que consideram máquinas paralelas não relacionadas, isto é, cada uma possui características individuais, possibilitando a ocorrência de situações em que o processamento de uma dada tarefa corresponde a poucos segundos em uma determinada máquina enquanto demora vários minutos em outra, por exemplo (PINEDO, 2016).

Sobre o sequenciamento de tarefas em máquinas paralelas não relacionadas, Lenstra, Shmoys e Tardos (1990) apresentam algoritmos aproximativos que utilizam soluções relaxadas obtidas com programação linear. Os autores observam que a obtenção de uma solução linear suficientemente próxima do ótimo não pode ser simplesmente arredondada para os valores inteiros mais próximos, pois proceder desta forma não só não garante que este novo resultado será a solução ótima, como muitas vezes pode acarretar em decréscimo da qualidade da solução ou levar a uma solução infactível. Ainda, caso a solução com utilização de programação linear encontre como pontos extremos somente variáveis com valores inteiros, o arredondamento torna-se desnecessário, tendo em vista que isto significa que esta é a solução ótima.

Srivastava (1998) apresenta uma meta-heurística busca tabu com o objetivo de minimizar o *makespan* do sequenciamento de tarefas em máquinas paralelas não relacio-

nadas. Hsu, Cheng e Yang (2011) apresentam um problema de minimização do tempo de processamento de todas as tarefas, considerando atividades com taxa de modificação (atividades que, quando processadas, alteram a taxa de produção da máquina). Ainda, é possível citar o estudo de Wang e Wang (2014), que considera efeitos de aprendizagem e deterioração de tarefas.

Pinedo (2016) afirma que, relativamente a máquinas paralelas, há três principais objetivos considerados, a saber, a minimização do *makespan*, do tempo total de conclusão das tarefas e do máximo atraso. Respectivamente, estes objetivos correspondem ao tempo necessário para processar todas as tarefas alocadas à máquina mais carregada, o tempo total para finalizar todas as tarefas consideradas no problema e a diferença de tempo da conclusão de uma tarefa em relação ao seu prazo de entrega (BAKER; TRIETSCH, 2013). Em diversos problemas, para chegar-se aos objetivos aqui citados, são consideradas restrições que buscam aproximar as modelagens da realidade como, por exemplo, os desgastes.

2.2 PROBLEMAS DE SEQUENCIAMENTO DE TAREFAS CONSIDERANDO DESGASTE

Entre as décadas de 1960 e 1980, os estudos realizados sobre problemas de sequenciamento de tarefas tiveram aplicação praticamente restrita à área industrial e de manufatura. Uma primeira conotação diferente do usual, para a época, é apresentada em Gupta, Kunnathur e Dandapani (1987), trabalho no qual os autores propõem uma aplicação na área de gestão financeira. Eles estudam o caso de uma organização interessada no desenvolvimento de uma política de crédito (empréstimos) objetivando o maior retorno possível deste investimento.

Os autores seguem, neste mesmo estudo, citando duas diferenças cruciais na abordagem de sua proposta em relação aos tratamentos dados aos problemas até aquele momento. A primeira diz respeito ao fato de que, se uma máquina não puder operar durante determinado período de tempo, sua capacidade de trabalho não pode ser estocada para utilizações futuras, ao contrário do capital, que, caso não esteja alocado em determinado momento, pode ser investido, sendo possível até mesmo gerar retorno proveniente desta aplicação “secundária” em vez de ficar ocioso. A segunda diferença é referente a forma de utilização do dinheiro ou da capacidade de uma máquina em executar determinada tarefa. A máquina será utilizada continuamente ao longo de um determinado período, enquanto o dinheiro é fornecido a um cliente em um momento específico de tempo, sendo ressarcido posteriormente. As ideias provenientes deste estudo funcionaram para impulsionar as pesquisas que consideravam o desgaste como uma das situações presentes nos problemas de escalonamento.

Browne e Yechiali (1990) introduziram um caso em que as tarefas apresentam de-

terioração conforme aguardam seu processamento, aumentando o tempo necessário para sua execução. Os autores atentam para o fato dos tempos de processamento das tarefas não serem invariáveis ao longo do tempo, o que torna o cálculo do *makespan* uma função do sequenciamento das tarefas.

A deterioração nas tarefas envolvendo máquinas paralelas idênticas é considerada por Mosheiov (1998), que abordou o problema objetivando minimizar o *makespan*, provando se tratar de problema NP-difícil até mesmo para duas máquinas, além de apresentar uma proposta de heurística para sua resolução e, ainda, os limitantes inferiores para o problema. Alidaee e Womer (1999) apresentam uma extensa revisão bibliográfica contendo os estudos que consideram deterioração de tarefas, contemplando o período de 1984 a 1989. Neste ínterim, a maioria destes estudos consideravam apenas aplicações em máquina única.

Gawiejnowicz, Kurc e Pankowska (2003) abordam o problema de programação de tarefas considerando máquinas paralelas idênticas e deterioração das tarefas, com o objetivo de minimizar o tempo total de processamento das mesmas. Os autores apresentam uma heurística gulosa para obtenção de soluções. O mesmo problema é abordado por Kang e Ng (2007), e os autores apresentam um esquema de aproximação de tempo completamente polinomial para resolução do mesmo.

Ji e Cheng (2008) analisam o sequenciamento de tarefas em máquinas paralelas em que o tempo de processamento de cada tarefa é função linear do seu respectivo tempo de início, objetivando minimizar o tempo total para completar todas as tarefas. Os mesmos autores ampliaram seus estudos de problemas com máquinas idênticas e deterioração nas tarefas em Ji e Cheng (2009), no qual tratam tanto da função objetivo de seu trabalho anterior quanto da minimização do *makespan*. Ademais, os autores demonstram que o problema pertence a classe NP-difícil e propõem dois algoritmos de aproximação de tempo polinomial completo, quando o número de máquinas é determinado a priori.

Toksari e Güner (2009) apresentam um problema de sequenciamento em máquinas paralelas considerando, de forma simultânea, os efeitos de deterioração e de aprendizagem, sendo que esta última trata de uma característica em que a repetição de tarefas semelhantes resulta em decréscimo nos tempos de processamento subsequentes. A função objetivo deste trabalho é minimizar os adiantamentos e atrasos das tarefas com data de vencimento em comum. Após a proposição de um modelo matemático, os autores apresentam uma heurística construtiva para resolução do problema, de modo a resolver instâncias maiores em tempos mais adequados para aplicações práticas. Assim, o referido algoritmo tem seu desempenho avaliado pela comparação de seus resultados com os valores obtidos através do modelo matemático.

Outros trabalhos de escalonamento consideram a deterioração como uma função da posição que uma tarefa ocupa no sequenciamento de uma máquina, como em Yang (2011) e Mosheiov (2012). Nestes casos, a alocação de uma tarefa em diferentes posições

de uma mesma máquina alterando, portanto, o sequenciamento, pode ocasionar grandes alterações na avaliação da solução final. Especificamente o estudo realizado por Yang (2011) considera concomitantemente eventos de manutenção, tema da Seção 2.3.

Resumidamente, existem problemas em que a tarefa deteriora enquanto aguarda ser processada, ou seja, o tempo de processamento é uma função do tempo de espera, ou problemas em que a deterioração é uma função da posição que a tarefa ocupa no sequenciamento atribuído à máquina em que esta é processada. E, ainda, há a aprendizagem, que significa que a repetição de tarefas semelhantes resultará em decréscimo nos tempos de processamento subsequentes. No presente estudo, é considerado que a deterioração ocorre na máquina, assim o processamento de uma tarefa acarreta na diminuição de seu nível de desempenho, o que leva a um aumento nos tempos de processamento de todas as tarefas alocadas subsequentemente.

O problema de escalonamento com efeitos de deterioração proposto por Ruiz-Torres, Paletta e Pérez (2013) traz uma abordagem que considera a deterioração de cada máquina como função da sequência de tarefas processadas por ela até então, tendo como objetivo a minimização do *makespan*. Neste problema, o tempo de processamento de cada tarefa está diretamente relacionado às tarefas que foram anteriormente processadas na mesma máquina. O estudo demonstra que o problema é NP-difícil, pois pode ser reduzido a um problema de máquinas paralelas idênticas sem considerar desgastes, que já é conhecido como NP-difícil. Como método de resolução é proposto uma meta-heurística *Simulated Annealing* para tratar instâncias geradas pelos autores e classificadas como de grande porte (até 50 tarefas, 4, 7 e 10 máquinas). Neste estudo, o problema é definido conforme segue.

2.2.1 Definição do Problema $R_m|S_{dd}|C_{max}$

Existem n tarefas independentes, pertencentes ao conjunto $N = \{1, \dots, j, \dots, n\}$, a serem sequenciadas e processadas em m máquinas paralelas, que formam o conjunto $M = \{1, \dots, k, \dots, m\}$. As tarefas em N são não-preemptivas, ou seja, a execução de uma dada tarefa não pode ser interrompida antes de sua conclusão. Ainda, cada tarefa deve ser processada somente uma vez e todas estão disponíveis para processamento no tempo inicial (igual a zero). As máquinas do conjunto M podem processar uma única tarefa por vez e são não relacionadas, ou seja, o tempo de processamento da tarefa j é relativo à máquina k a qual ela estiver alocada. Além disso, tem-se o tempo de processamento base representado por p_{jk} , $\forall j \in N$ e $\forall k \in M$, no qual não é considerada a deterioração da máquina.

O processamento de uma tarefa j em uma máquina k gera um desgaste a esta, que corresponde ao decréscimo no nível de performance da mesma. A deterioração na

máquina k decorrente do processamento da tarefa j é representada por $d_{jk} \in [0, 1)$. Como exemplo, se uma máquina k apresenta 100% de nível de performance, após processar uma tarefa j , que produz um desgaste percentual de 10% ($d_{jk} = 0,1$), o novo nível de performance desta máquina passará a ser de 90%. É importante observar que esta deterioração é cumulativa, de modo que o desgaste decorrente do processamento de uma dada tarefa influencia no tempo de processamento de todas as demais tarefas subsequentes alocadas nesta mesma máquina. E, por fim, a função objetivo corresponde à minimização do tempo para processar a última tarefa (*makespan*), representado por C_{max} .

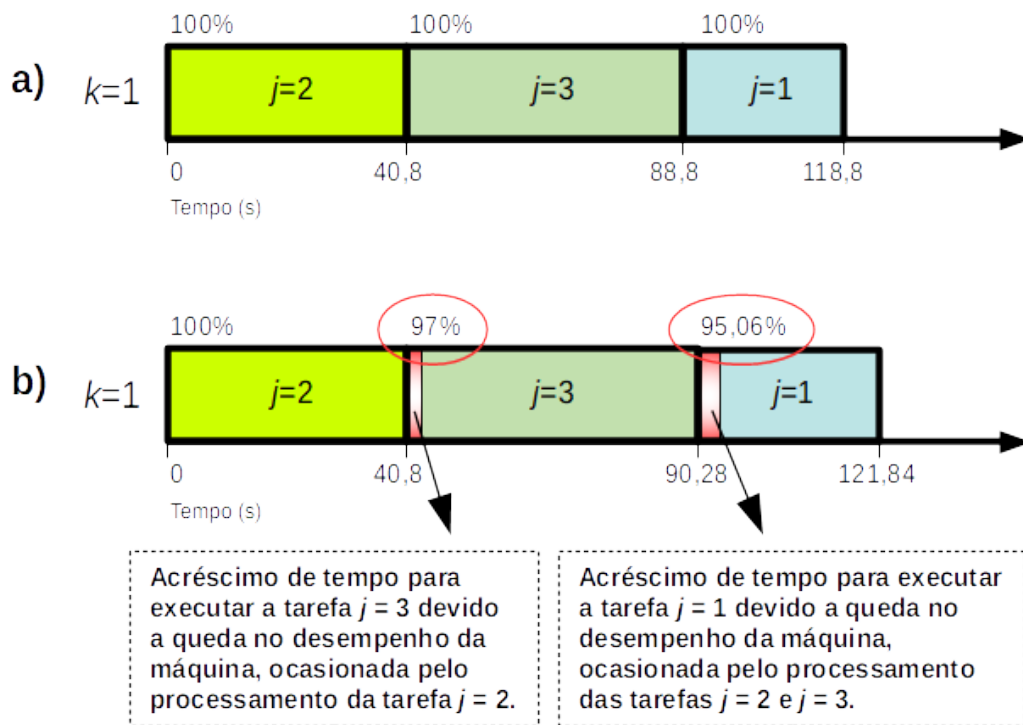
Para um melhor entendimento dos desgastes dependentes da sequência, um exemplo é ilustrado na Figura 1. Neste exemplo, três tarefas são sequenciadas em uma determinada máquina, na qual em (a) é apresentado um sequenciamento em que o desgaste é desprezado e em (b) o mesmo sequenciamento é apresentado considerando desgaste. O eixo horizontal em cada caso representa os tempos decorridos e, na parte superior, há valores percentuais que retratam o nível de performance da máquina. Níveis de atividade inferiores a 100% são destacados com elipses em vermelho. Os retângulos demarcados com borda de maior espessura simbolizam o tempo total das tarefas que estão sendo processadas e, especificamente em (b), há retângulos hachurados em vermelho representando o acréscimo nos tempos de processamento base ocasionados pela queda nos respectivos níveis de atividade da máquina. Os dados utilizados para a elaboração da Figura 1 estão na tabela abaixo.

Tabela 1 – Dados utilizados para a elaboração do exemplo ilustrado na Figura 1

| Tarefa j | Deterioração d_{jk} | Tempo p_{jk} (s) |
|------------|-----------------------|--------------------|
| 1 | 6,00% | 30,0 |
| 2 | 3,00% | 40,8 |
| 3 | 2,00% | 48,0 |

Fonte: Autor.

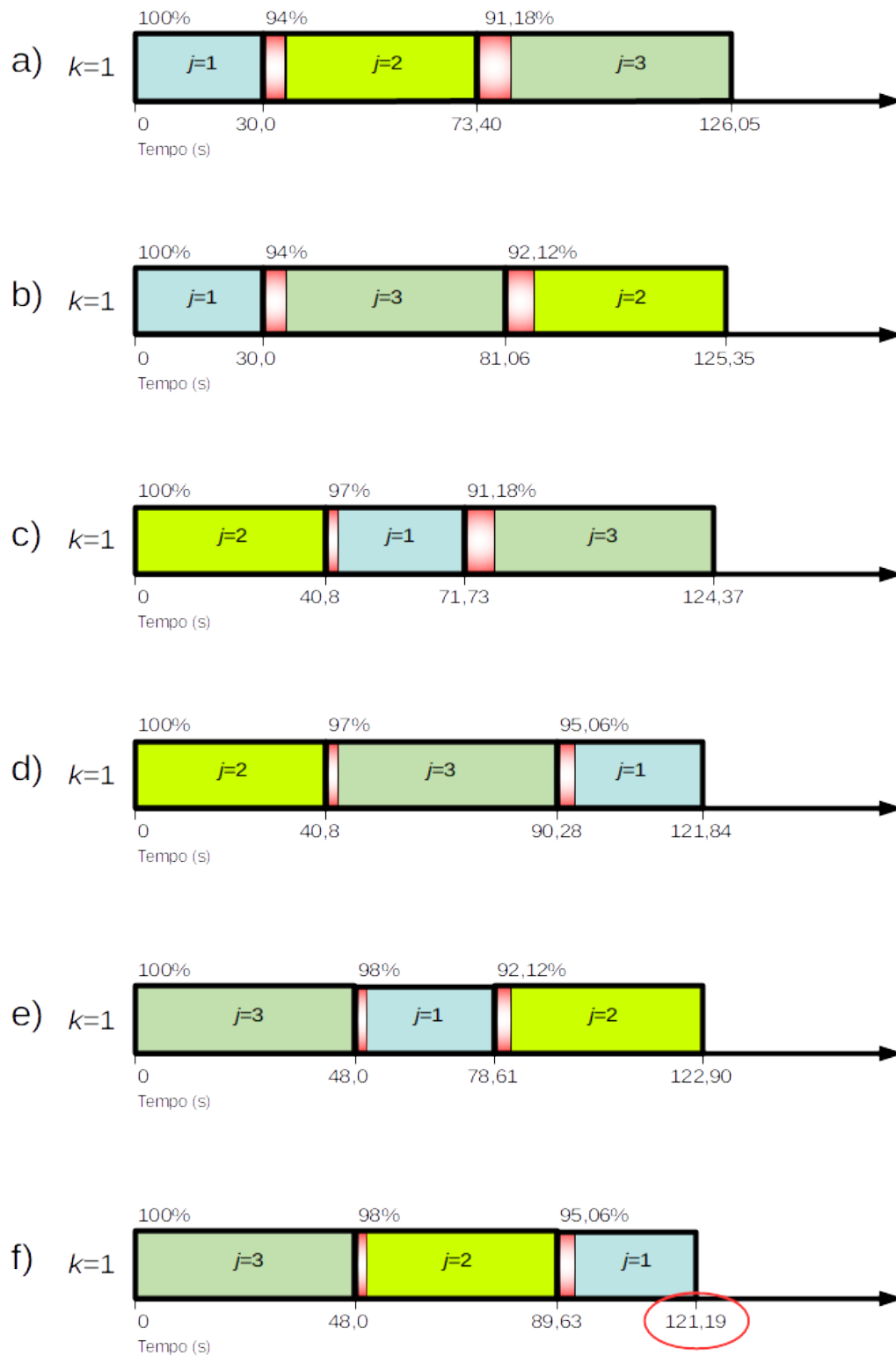
Figura 1 – Exemplos de sequenciamento de tarefas (a) sem considerar desgaste e (b) considerando desgaste dependente da sequência



Fonte: Autor.

É possível perceber que, em (a), independentemente da ordem de alocação das tarefas, o tempo total de processamento seria de 118,8 segundos. Isto não acontece quando desgastes dependentes da sequência são levados em consideração e, como exemplo, a Figura 2 contém todas as possíveis soluções para o sequenciamento já apresentado, porém sob efeito dos referidos desgastes. A solução ótima está destacada com uma elipse vermelha e corresponde ao sequenciamento em (f).

Figura 2 – Exemplo de busca pela minimização do *makespan* no sequenciamento de tarefas considerando desgastes dependentes da sequência



Fonte: Autor.

2.2.2 Modelo Matemático em Programação Não Linear Inteira Mista

Nesta seção, é descrito o modelo matemático em PNLIM proposto por Ruiz-Torres, Paletta e Pérez (2013), referenciado daqui por diante como *RPP*, para o problema $R_m|S_{dd}|C_{max}$. Algumas definições adicionais são necessárias para o entendimento da formulação.

Dado o conjunto $H = \{1, \dots, h, \dots, n\}$, que corresponde a todas as n posições possíveis para o processamento de uma tarefa em uma determinada máquina e, considerando $\tau(h, k)$ como a tarefa alocada à posição h da máquina k , $\forall h \in H$ e $\forall k \in M$, o nível de performance da máquina k para processar a tarefa alocada à posição h é definido como

$$q_{kh} := (1 - d_{\tau(h-1,k)k}) \times q_{k(h-1)}, \quad (2.1)$$

$\forall k \in M$ e $\forall h \in H : h \geq 1$. Ou seja, o nível de performance q_{kh} é dado pelo nível de performance da posição anterior $q_{k(h-1)}$ deteriorado pelo desgaste ocasionado pelo processamento da tarefa alocada à posição $h - 1$, $d_{\tau(h-1,k)k}$, ambos da mesma máquina. Além disso, assume-se que as máquinas não possuem deterioração na posição inicial, sendo $q_{k1} = 1$, $\forall k \in M$, isto é, todas elas possuem nível de performance máximo em sua primeira posição. Por sua vez, o tempo de processamento da tarefa j na posição h da máquina k é dado por p_{jk}/q_{kh} , que consiste na divisão do tempo de processamento base pelo desempenho da máquina naquela mesma posição. Como este desempenho é dado por valores entre 0 e 1, representando os níveis de desempenho mínimo e máximo, respectivamente, torna-se evidente que as tarefas sofrem um incremento no seu tempo de processamento quando a máquina já está desgastada. Com isto, o modelo *RPP* pode ser formalizado conforme segue.

Seus conjuntos e parâmetros são:

p_{jk} : tempo de processamento da tarefa j na máquina k .

d_{jk} : desgaste causado à máquina k pelo processamento da tarefa j .

M : conjunto das máquinas que farão o processamento das tarefas.

N : conjunto das tarefas a serem processadas.

H : conjunto das posições para processamento de tarefas disponíveis em uma máquina.

As variáveis são definidas como:

q_{kh} : nível de performance da máquina k para processar uma tarefa na posição h .

C_{max} : *makespan*.

$$x_{jkh} : \begin{cases} 1, & \text{se a tarefa } j \text{ está alocada à máquina } k \text{ na posição } h; \\ 0, & \text{caso contrário.} \end{cases}$$

$$\text{(RPP)} \quad \min C_{\max}, \quad (2.2)$$

$$\sum_{j \in N} x_{jkh} \leq 1, \quad \forall k \in M, h \in H, \quad (2.3)$$

$$\sum_{k \in M} \sum_{h \in H} x_{jkh} = 1, \quad \forall j \in N, \quad (2.4)$$

$$\sum_{j \in N} \sum_{h \in H} \frac{p_{jk}}{q_{kh}} \times x_{jkh} \leq C_{\max}, \quad \forall k \in M, \quad (2.5)$$

$$x_{jkh} \leq \sum_{\ell \in N} x_{\ell k(h-1)}, \quad \forall j \in N, k \in M, h \in H \setminus \{1\}, \quad (2.6)$$

$$q_{kh} = \sum_{j \in N} (1 - d_{jk}) \times q_{k(h-1)} \times x_{jk(h-1)}, \quad \forall k \in M, h \in H \setminus \{1\}, \quad (2.7)$$

$$q_{k1} = 1, \quad \forall k \in M, \quad (2.8)$$

$$x_{jkh} \in \{0, 1\}, \quad \forall j \in N, k \in M, h \in H. \quad (2.9)$$

A Equação (2.2) é a função objetivo do problema e considera a minimização do *makespan*. A Equação (2.3) garante que no máximo uma tarefa será alocada em cada posição de cada máquina. A Equação (2.4) assegura que cada tarefa será alocada a uma única máquina em uma única posição. A Equação (2.5) determina o tempo máximo de conclusão das tarefas dentre todas as máquinas (*makespan*). A Equação (2.6) exige que a alocação de tarefas seja sequencial, ou seja, a máquina não pode ficar ociosa entre o processamento de duas tarefas consecutivas. As Equações (2.7) e (2.8) definem o nível de performance de cada máquina para cada posição. E, por fim, a Equação (2.9) apresenta a restrição de domínio da variável x_{jkh} .

Araújo, Dhein e Fampa (2017) propõem uma linearização do modelo matemático apresentando por Ruiz-Torres, Paletta e Pérez (2013). Esta proposta, aliada ao uso de desigualdades válidas, permitiu resolver na otimalidade diversas instâncias pela primeira vez na literatura, além de encontrar a melhor solução conhecida de 40% das instâncias utilizadas como *benchmark*, de Ruiz-Torres, Paletta e Pérez (2013).

2.2.3 Linearização do Modelo RPP e Ordem de Alocação das Tarefas

A linearização do modelo *RPP* proposta em Araújo, Dhein e Fampa (2017) é descrita a seguir. Especificamente, existem duas restrições não lineares em *RPP*, a saber, as Equações (2.5) e (2.7), devido a, respectivamente, uma operação de divisão entre as

variáveis x_{jkh} e q_{kh} e uma multiplicação entre as variáveis $q_{k(h-1)}$ e $x_{jk(h-1)}$.

Inicialmente, o conceito do desgaste foi alterado. Ruiz-Torres, Paletta e Pérez (2013) consideram d_{jk} como o desgaste causado à máquina k pelo processamento da tarefa j , e este atua como um valor que diminui percentualmente o desempenho q_{kh} da máquina k . Entretanto, ao utilizar as relações

$$\tilde{d}_{jk} := \frac{d_{jk}}{1 - d_{jk}},$$

$\forall j \in N$ e $k \in M$, ou ainda

$$\frac{1}{1 - d_{jk}} = 1 + \tilde{d}_{jk},$$

Araújo, Dhein e Fampa (2017) alteram o conceito. Com esta modificação é definido o parâmetro \tilde{d}_{jk} , que também se refere ao desgaste sofrido pela máquina k pelo processamento da tarefa j , no entanto, com uma abordagem diferente. Neste caso, considera-se \tilde{d}_{jk} como um valor percentual que representa um aumento que será causado no tempo de processamento base das tarefas subsequentes a j . Este novo conceito permite o uso de outra forma para o cálculo do desempenho da máquina k , estabelecido conforme segue

$$\tilde{q}_{kh} := \frac{1}{q_{kh}},$$

$\forall k \in M$ e $h \in H$. Desta forma, é apresentada a variável \tilde{q}_{kh} , que pode ser definida como taxa acumulada de incremento do tempo de processamento base (TAI). A mesma corresponde a qual o percentual de tempo que a máquina k utilizará para o processamento da tarefa j , em que $\tilde{q}_{kh} \geq 100\%$, sendo que $\tilde{q}_{kh} = 100\%$ significa que a máquina processa uma tarefa em tempo igual ao tempo base desta, ou seja, está com seu nível de performance/desempenho máximo. Como exemplo, para uma situação de uma máquina já deteriorada, se $\tilde{q}_{kh} = 110\%$, uma tarefa j com $p_{jk} = 20\text{seg}$ será processada em 110% deste tempo base, isto é, seu processamento durará vinte e dois segundos. Ou seja, nesta abordagem, valores de TAI acima de 100% não são desejáveis, pois representam acréscimo no tempo de processamento das tarefas.

A partir destes novos conceitos, é possível reescrever a Equação (2.1) na forma seguinte

$$\tilde{q}_{kh} := (1 + \tilde{d}_{jk}) \times \tilde{q}_{k(h-1)},$$

para cada máquina $k \in M$ e cada posição $h > 1$, e $\tilde{q}_{k1} = 1$, $\forall k \in M$. Com estas alterações, a Equação (2.7) pode ser substituída por

$$\tilde{q}_{kh} = \tilde{q}_{k(h-1)} \times \sum_{j \in N} (1 + \tilde{d}_{jk}) \times x_{jk(h-1)}, \quad (2.10)$$

$\forall k \in M$ e $h \in H \setminus \{1\}$.

No entanto, é evidente que a Equação (2.10) ainda é não linear. A linearização ocorre com o uso do método *Big-M*, conforme apresentado nas equações que seguem

$$\begin{aligned}\tilde{q}_{kh} &\geq (1 + \tilde{d}_{jk}) \times \tilde{q}_{k(h-1)} - \mathbb{M}(1 - x_{jk(h-1)}) , & \forall j \in N, k \in M, h \in H \setminus \{1\} , \\ \tilde{q}_{kh} &\geq 0 , & \forall k \in M, h \in H \setminus \{1\} , \\ \tilde{q}_{k1} &= 1 , & \forall k \in M .\end{aligned}$$

Deste modo, \tilde{q}_{kh} passa a ser calculada através de uma equação linear e, especificamente para a posição inicial ($h = 1$) de qualquer máquina k , tem-se $\tilde{q}_{k1} = 1$, o que correspondente à TAI de 100%, ou seja, a tarefa j será processada em um tempo igual ao seu processamento base p_{jk} .

No que se refere à linearização da Equação (2.5), é introduzida a variável u_{jk} , que refere-se ao tempo de processamento da tarefa j alocada à máquina k na posição h , considerando o desgaste sofrido pela máquina k até aquele momento. Em conjunto com a definição de \tilde{q}_{kh} e de uma constante suficientemente grande, a Equação 2.5 pode ser substituída por

$$\begin{aligned}\sum_{j \in N} u_{jk} &\leq C_{max} , & \forall k \in M , \\ u_{jk} &\geq p_{jk} \tilde{q}_{kh} - \mathbb{M}(1 - x_{jkh}) , & \forall j \in N, k \in M, h \in H , \\ u_{jk} &\geq 0 , & \forall j \in N, k \in M .\end{aligned}$$

Desta maneira, o tempo de processamento u_{jk} , que considera o desgaste sofrido pela máquina k , é utilizado para o cálculo do *makespan* através de uma equação linear.

As definições necessárias para a primeira versão da formulação linear proposta por Araújo, Dhein e Fampa (2017), denominada aqui *ADF**, podem ser sumarizadas conforme segue.

Parâmetros:

\tilde{d}_{jk} : desgaste causado à máquina k pelo processamento da tarefa j .

\mathbb{M} : número suficientemente grande.

Variáveis:

u_{jk} : tempo de processamento atual da tarefa j na posição h da máquina k , considerando o desgaste sofrido pela máquina k até aquele momento.

\tilde{q}_{kh} : taxa acumulada de incremento do tempo de processamento base da máquina k na posição h .

Em conjunto com as definições já apresentadas na Seção 2.2.2, tem-se o modelo linear seguinte.

$$\begin{aligned}
(\mathbf{ADF}^*) \quad & \min C_{\max} , \\
& \sum_{j \in N} x_{jkh} \leq 1 , & \forall k \in M, h \in H , \\
& \sum_{k \in M} \sum_{h \in H} x_{jkh} = 1 , & \forall j \in N , \\
& \sum_{j \in N} u_{jk} \leq C_{\max} , & \forall k \in M , \\
& u_{jk} \geq p_{jk} \tilde{q}_{kh} - \mathbb{M}(1 - x_{jkh}) , & \forall j \in N, k \in M, h \in H , \\
& u_{jk} \geq 0 , & \forall j \in N, k \in M , \\
& x_{jkh} \leq \sum_{\ell \in N} x_{\ell k(h-1)} , & \forall j \in N, k \in M, h \in H \setminus \{1\} , \\
& \tilde{q}_{kh} \geq (1 + \tilde{d}_{jk}) \times \tilde{q}_{k(h-1)} - \mathbb{M}(1 - x_{jk(h-1)}) , & \forall j \in N, k \in M, h \in H \setminus \{1\} , \\
& \tilde{q}_{kh} \geq 0 , & \forall j \in N, k \in M, h \in H \setminus \{1\} , \\
& \tilde{q}_{k1} = 1 , & \forall k \in M , \\
& x_{jkh} \in \{0, 1\} , & \forall j \in N, k \in M, h \in H .
\end{aligned}$$

De forma a melhorar o modelo ADF^* , Araújo, Dhein e Fampa (2017) incluíram na formulação o seguinte Lema, proposto por Ruiz-Torres, Paletta e Pérez (2013).

Lema 1 Considerando $n_k \leq n$ como o número de tarefas alocadas à máquina k . Considerando $\tau(h, k)$ como a tarefa alocada à posição h da máquina k , $\forall h \in H$ e $k \in M$. Então, se o sequenciamento de tarefas satisfaz:

$$\frac{p_{\tau(1,k)k}(1 - d_{\tau(1,k)k})}{d_{\tau(1,k)k}} \geq \frac{p_{\tau(2,k)k}(1 - d_{\tau(2,k)k})}{d_{\tau(2,k)k}} \geq \dots \geq \frac{p_{\tau(n_k,k)k}(1 - d_{\tau(n_k,k)k})}{d_{\tau(n_k,k)k}} , \quad (2.11)$$

o tempo total para conclusão de todas as tarefas alocadas à máquina k é mínimo.

De acordo com o Lema 1, uma vez determinada a alocação de tarefas para as máquinas, a sequência de processamento ótima das mesmas pode ser facilmente estabelecida. De fato, seja $n_k = \{j_1, j_2, \dots, j_n\}$ o conjunto de tarefas designado para uma dada máquina $k \in M$, então a sequência ótima é dada pela permutação $(j_k(1), j_k(2), \dots, j_k(n))$, considerando o seguinte critério de ordenação

$$\frac{p_{j_k(1)k}(1 - d_{j_k(1)k})}{d_{j_k(1)k}} \geq \frac{p_{j_k(2)k}(1 - d_{j_k(2)k})}{d_{j_k(2)k}} \geq \dots \geq \frac{p_{j_k(n)k}(1 - d_{j_k(n)k})}{d_{j_k(n)k}} .$$

Tanto RPP quanto ADF^* consideram três índices na definição das variáveis de decisão que estabelecem a atribuição das tarefas a uma determinada máquina em uma determinada posição, x_{jkh} , $\forall j \in N$, $k \in M$ e $h \in H$. Araújo, Dhein e Fampa (2017)

utilizam o resultado do Lema 1 para propor um modelo aprimorado, denominado daqui por diante como *ADF*, que substitui x_{jkh} por uma nova variável x_{jk} , $\forall j \in N, k \in M$. Neste novo contexto, quando $x_{jk} = 1$, assume-se que a tarefa j será processada na máquina k na posição h , que corresponde a posição da tarefa j na sequência ótima de processamento definida pelo Lema 1, caso todas as tarefas fossem alocadas a essa máquina. Nota-se, também, que a substituição de x_{jkh} por x_{jk} diminui o número de variáveis de decisão do problema de $n^2 \times m$ para $n \times m$.

Deste modo, considerando a variável

$$x_{jk} : \begin{cases} 1, & \text{se a tarefa } j \text{ está alocada na máquina } k; \\ 0, & \text{caso contrário.} \end{cases}$$

o Modelo *ADF* é formulado conforme segue.

$$\text{(ADF)} \quad \min C_{\max} . \quad (2.12)$$

$$\sum_{k \in M} x_{jk} = 1 , \quad \forall j \in N , \quad (2.13)$$

$$\sum_{j \in N} u_{jk} \leq C_{\max} , \quad \forall k \in M , \quad (2.14)$$

$$u_{j_k(h)k} \geq p_{j_k(h)k} \tilde{q}_{kh} - \mathbb{M} (1 - x_{j_k(h)k}) \quad \forall k \in M, h \in H , \quad (2.15)$$

$$\tilde{q}_{kh} \geq (1 + \tilde{d}_{j_k(h-1)k}) \times \tilde{q}_{k(h-1)} - \mathbb{M} (1 - x_{j_k(h-1)k}) , \quad \forall k \in M, h \in H \setminus \{1\} , \quad (2.16)$$

$$\tilde{q}_{kh} \geq \tilde{q}_{k(h-1)} , \quad \forall k \in M, h \in H \setminus \{1\} , \quad (2.17)$$

$$u_{jk} \geq 0 , \quad \forall j \in N, k \in M , \quad (2.18)$$

$$\tilde{q}_{k1} = 1 , \quad \forall k \in M , \quad (2.19)$$

$$\tilde{q}_{kh} \geq 1 , \quad \forall k \in M , \quad (2.20)$$

$$x_{jk} \in \{0, 1\} , \quad \forall j \in N, k \in M . \quad (2.21)$$

A função objetivo (2.12) considera a minimização do *makespan*, C_{\max} . A Equação (2.13) garante que todas as tarefas serão alocadas exatamente uma única vez em uma única máquina. A Equação (2.14) assegura que o *makespan*, na solução ótima, será igual ao tempo total de processamento da máquina mais carregada. A Equação (2.15) define o tempo de processamento da tarefa j alocada na posição h da máquina k , considerando o desgaste da máquina. A Equação (2.16) determina a TAI da máquina k na posição h , levando em consideração o desgaste e a TAI da posição anterior ($h - 1$), caso uma tarefa tenha sido alocada exatamente na posição ($h - 1$). A Equação (2.17) garante que a

TAI da máquina k no processamento de uma tarefa j alocada na posição h considerará o desgaste acumulado pela máquina até aquele momento, mesmo que a tarefa j não possua uma tarefa predecessora imediata na posição $(h - 1)$. A Equação (2.18) assegura que o tempo de processamento de uma tarefa nunca será menor que zero. A Equação (2.19) assegura que qualquer máquina k terá $TAI = 100\%$ ao processar uma tarefa j na posição $h = 1$. As Equações (2.20) e (2.21) definem, respectivamente, os domínios das variáveis \tilde{q}_{kh} e x_{kh} .

Conforme explicado anteriormente, ao definir as posições das tarefas em cada máquina, estas terão seus tempos de processamento minimizados, pois a relação (2.11) no Lema 1 certamente será satisfeita. O importante é observar que, com esta técnica, cada tarefa possui uma posição "predeterminada" para cada máquina. Em *ADF*, esta situação é evidenciada nas restrições (2.15), (2.16) e (2.17).

Por exemplo, em uma situação hipotética em que existam duas máquinas disponíveis ($k = 1$ e 2), quatro tarefas a serem processadas ($j = 1, 2, 3, 4$) e o resultado da aplicação do Lema 1 determine a sequência $(2, 1, 4, 3)$, para $k = 1$, e $(3, 1, 4, 2)$, para $k = 2$ e, supondo que a solução ótima seja alocar $\{2, 4\}$ à máquina 1 e $\{1, 3\}$ à máquina 2, o sequenciamento das tarefas fica conforme ilustrado na Figura 3.

Na mesma figura, pode-se observar em (a) a forma como a modelagem foi realizada, possibilitando alocação de tarefas em suas posições predeterminadas, contendo posições vagas entre um e outro processamento, quando necessário. Nestes casos, o status da máquina até o momento anterior à posição vaga (tempo de processamento, taxa acumulada de incremento do tempo de processamento base, etc) é propagado para a posição seguinte. É possível perceber que este comportamento ocorre em virtude das restrições (2.16) e (2.17). A Equação (2.16) determina o valor da variável \tilde{q}_{kh} , caso haja uma tarefa alocada à posição $h - 1$. Quando isto ocorre, automaticamente a Equação (2.17) é satisfeita. Além do mais, é plausível notar que, caso não haja tarefa alocada à posição $h - 1$, o resultado da restrição (2.16) será zero. Sendo assim, devido à Equação (2.17), \tilde{q}_{kh} assumiria um valor igual ao de $\tilde{q}_{k(h-1)}$, propagando, de fato, a *TAI* de $h - 1$ para h .

Já em (b) é apresentado o comportamento efetivo do sequenciamento, em que é considerado o processamento das tarefas como se fossem imediatamente subsequentes, ignorando as posições sem ocupação.

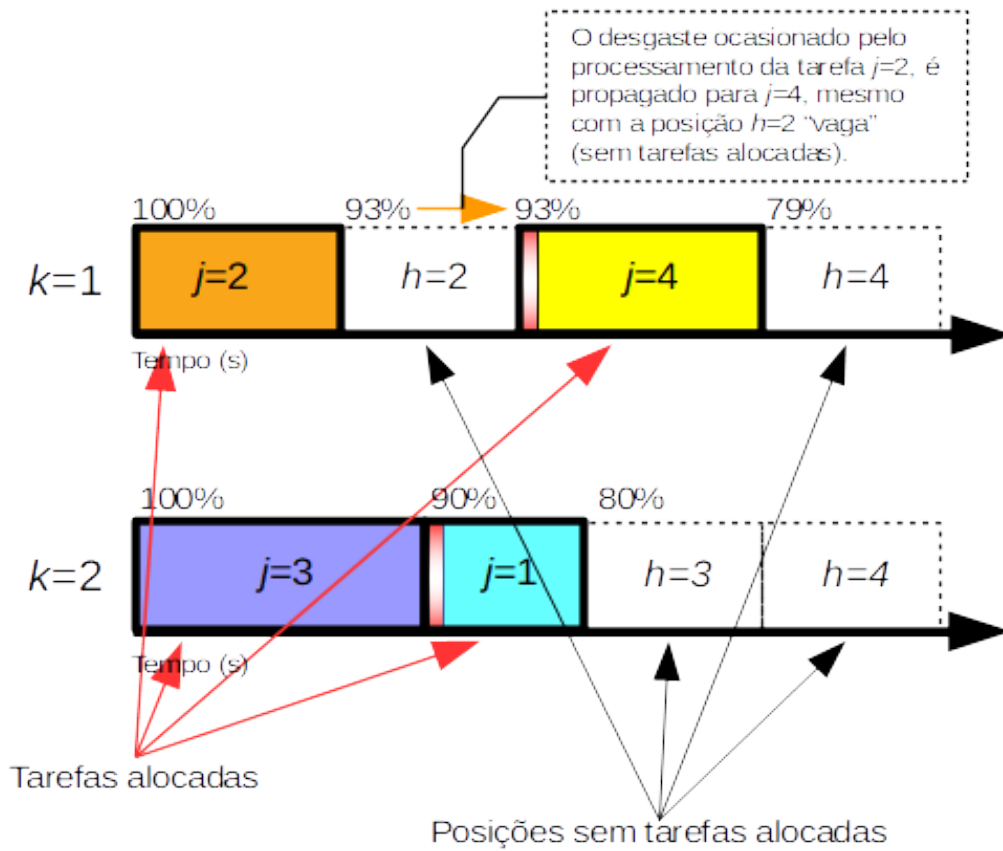
Por fim, Araújo, Dhein e Fampa (2017) observam que as constantes \mathbb{M} das restrições (2.15) e (2.16) podem ser calculadas de forma a melhorar a qualidade da relaxação linear. A constante da restrição (2.16) pode assumir um valor dependente da máquina e da posição,

$$\mathbb{M}_2(k, h) := \prod_{\ell=1}^{h-1} \left(1 + \tilde{d}_{j_k(\ell)k}\right),$$

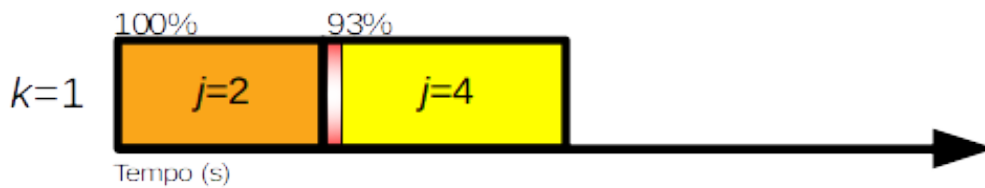
da mesma forma que a constante da restrição (2.15),

Figura 3 – Exemplo de alocação de tarefas em posição predeterminadas

a) Modelagem: tarefas alocadas em posições predeterminadas:



b) Processamento "efetivo": as posições "vagas" são ignoradas e as tarefas são processadas sequencialmente, como se ocupassem posições subsequentes:



Fonte: Autor.

$$M_1(k, h) := p_{j_k(h)k} \times M_2(k, h) .$$

Além das considerações trazidas até aqui, existem problemas que podem trazer características adicionais como, por exemplo, eventos de manutenção.

2.3 PROBLEMAS DE SEQUENCIAMENTO DE TAREFAS CONSIDERANDO EVENTOS DE MANUTENÇÃO

Adiri et al. (1989) estudam uma variação dos problemas de programação de tarefas em uma única máquina. Os autores propõem um problema onde a máquina está sujeita à ocorrência de uma ou múltiplas panes durante o processamento. Quando isto ocorre, a máquina fica indisponível até que o reparo aconteça. Lee e Liman (1992) ampliam os estudos do caso determinístico apresentado por Adiri et al. (1989), em que se conhece de antemão o momento em que a pane ocorrerá. Especificamente para esta situação, utilizam o termo "manutenção programada", para referirem-se a estes reparos.

Conforme constatado por Leon e Wu (1992), grande parte da literatura que trata do sequenciamento de tarefas assume que as máquinas estão continuamente disponíveis durante todo o horizonte de tempo de processamento, entretanto esta suposição não é válida na prática, uma vez que as máquinas que executam as tarefas podem permanecer indisponíveis durante um determinado período de tempo pelas mais diversas razões, como manutenção, trocas de turno ou intervalo dos operadores, modificações no ferramental, panes, etc. Os autores propõem um algoritmo *branch and bound* para a resolução do problema com máquina única.

Suresh e Chaudhuri (1996) abordam, pela primeira vez, o problema de programação de tarefas em máquinas paralelas não relacionadas com restrições de disponibilidade, considerando dois casos distintos em que, no primeiro, já se conhece a disponibilidade das máquinas de antemão e, no segundo, a disponibilidade tem de ser verificada entre os processamentos de uma e outra tarefas. Os autores propõem a solução do problema através da utilização de um algoritmo guloso e de heurísticas de melhoramento passagem múltipla.

O problema de programação de tarefas em máquinas paralelas idênticas considerando a deterioração das tarefas e manutenção programada foi abordado pela primeira vez em Lee e Wu (2008). Os autores apresentam limitantes inferiores e um algoritmo heurístico para encontrar soluções para o problema, utilizando a metodologia *largest deterioration first* (em que as tarefas com maior deterioração são alocadas primeiramente) para encontrar as soluções iniciais. Adaptando a pesquisa anterior, Yang (2011) estuda o sequenciamento de tarefas em máquinas paralelas com desgastes dependentes da posição e atividades de manutenção, em que a máquina pode ser reparada exatamente uma vez ao longo do horizonte de planejamento. O autor define que a ocorrência de uma manutenção em uma máquina deteriorada reverte o desempenho desta para sua condição inicial. Yang et al. (2012), por sua vez, trazem uma abordagem na qual cada máquina pode sofrer eventos de manutenção múltiplas vezes, e sua duração pode variar de uma máquina para outra.

O problema proposto por Ruiz-Torres, Paletta e M'Hallah (2017) trata da programação de tarefas objetivando a minização do *makespan*, considerando máquinas paralelas

com desgastes dependentes da sequência e tempo de parada programada para manutenção. Neste estudo, a parada programada para manutenção inicia somente após a tarefa ser totalmente concluída, ou seja, uma tarefa iniciada deve ser concluída antes que a parada programada para manutenção tenha início. Da mesma forma, a máquina que está sofrendo manutenção não pode receber novas tarefas até sua conclusão, além de a manutenção poder ocorrer simultaneamente em mais de uma máquina. Ainda, os eventos de manutenção ocorrem somente se for lucrativo para a função objetivo do problema e não de maneira aleatória ou a partir de um determinado nível de desgaste atingido pela máquina. Por fim, segundo os autores, trata-se de um problema NP-difícil, uma vez que, ao desconsiderarmos os eventos de manutenção e os efeitos da deterioração, o problema é reduzido a $R_m||C_{max}$, que é NP-difícil. Neste estudo, os autores definem o problema conforme segue.

2.3.1 Definição do Problema $R_m|S_{dd}, Mnt|C_{max}$

O problema de sequenciamento de tarefas em máquinas paralelas considerando desgastes dependentes da sequência e tempo de parada programada para manutenção utilizado neste estudo tem como base a definição apresentada em Ruiz-Torres, Paletta e M'Hallah (2017). Nesta seção, é explicado como são operacionalizados os eventos de parada programada para manutenção. As demais características do problema $R_m|S_{dd}, Mnt|C_{max}$ são as mesmas de $R_m|S_{dd}|C_{max}$, já apresentadas na Seção 2.2.1.

Os eventos de manutenção podem ocorrer em uma ou mais máquinas, de forma simultânea ou não. A ocorrência destes eventos acarreta em duas importantes consequências, a primeira é a restauração do nível de performance/desempenho da máquina que recebeu a manutenção para o nível máximo, pois a deterioração causada pelo processamento das tarefas já concluídas é zerada. A segunda consequência é o custo despendido para a realização de uma manutenção, que nada mais é do que um incremento no tempo total de processamento da máquina que sofreu a ação.

A Figura 4 ilustra uma programação de tarefas para o problema $R_m|S_{dd}, Mnt|C_{max}$. Os eventos de manutenção estão identificados pelos retângulos com bordas pontilhadas, hachurados em amarelo e contendo a sigla Mnt . É possível perceber que após a realização das manutenções, o valor percentual referente ao desempenho da máquina retorna ao patamar de 100%. Quando isto ocorre, a representação deste nível na figura é destacado com uma elipse vermelha. Assim, o tempo de processamento das tarefas subsequentes a este evento desconsidera o desgaste ocasionado pelas tarefas processadas anteriormente. Por exemplo, na máquina $k = 1$ a tarefa $j = 3$ não considera o desgaste ocasionado pelo processamento de $j = 6$, pois, neste ínterim, houve uma manutenção. Todavia, $j = 3$ sofre com a deterioração $d_{9,1}$, causada pelo processamento de $j = 9$, uma vez que

entre ambas as tarefas não foi realizada manutenção e a máquina $k = 1$ sofre desgaste. Já em $k = 2$, como houve manutenção imediatamente antes de $j = 8$, o tempo de processamento desta não sofre acréscimo pela queda de desempenho causada pelos desgastes ocasionados tanto por $j = 5$ quanto por $j = 4$. Os dados utilizados para a elaboração da Figura 4 estão na tabela abaixo.

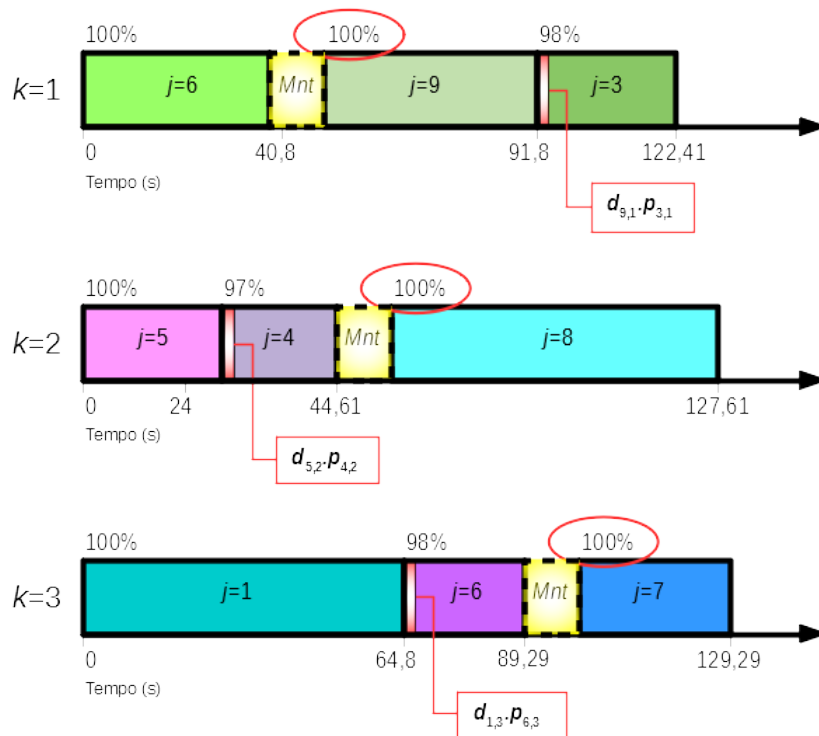
Tabela 2 – Dados utilizados para a elaboração do exemplo ilustrado na Figura 4

| Máquina k | Tarefa j | Deterioração d_{jk} | Tempo p_{jk} (s) |
|-------------|------------|-----------------------|--------------------|
| 1 | 6 | 9,00% | 40,8 |
| | 9 | 2,00% | 48,0 |
| | 3 | 3,00% | 30,0 |
| 2 | 5 | 3,00% | 24,0 |
| | 4 | 6,00% | 20,0 |
| | 8 | 6,00% | 80,0 |
| 3 | 1 | 2,00% | 64,8 |
| | 2 | 8,00% | 24,8 |
| | 7 | 7,00% | 37,0 |

Manutenção = 3 segundos

Fonte: Autor.

Figura 4 – Exemplo de sequenciamento de tarefas em máquinas paralelas com desgaste dependente da sequência e eventos de parada programada para manutenção



Fonte: Autor.

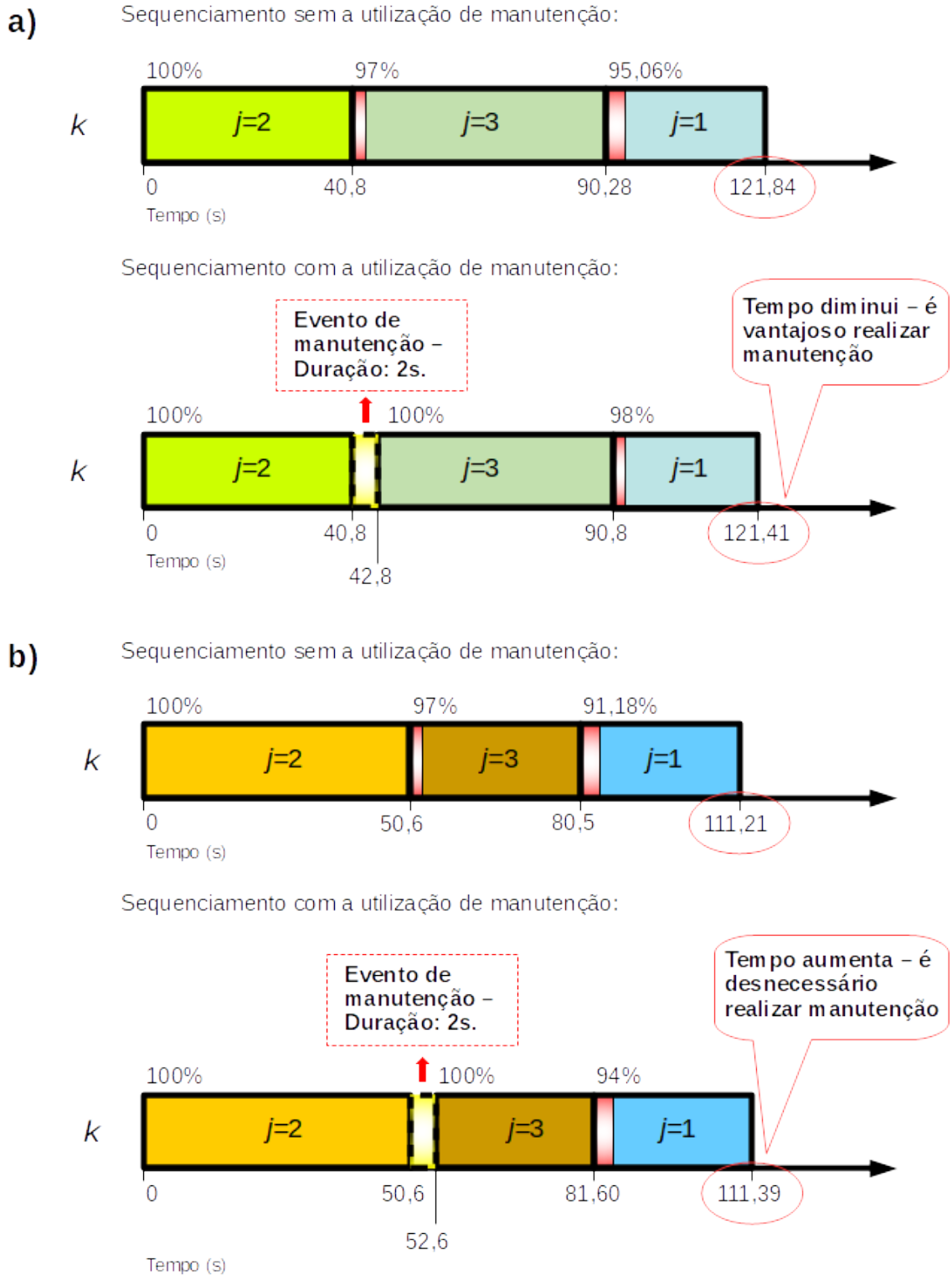
Por fim, na Figura 5 são ilustradas duas situações hipotéticas acerca da ocorrência de eventos de manutenção, considerando que os mesmos possuem dois segundos de duração. Na situação (a), pode-se observar o sequenciamento de tarefas com e sem a realização de manutenção. Ao analisar os valores do tempo total de processamento da máquina, para ambos os casos, resta claro que, em (a), a utilização de manutenção melhora o resultado encontrado. Em contrapartida, a situação (b) apresenta um cenário em que efetuar uma manutenção na máquina k , para aquele sequenciamento de tarefas, não é vantajoso, uma vez que o tempo total de processamento aumenta em relação ao obtido através do sequenciamento sem manutenções.

Tabela 3 – Dados utilizados para a elaboração do exemplo ilustrado na Figura 5

| Situação | Tarefa j | Deterioração d_{jk} | Tempo p_{jk} (s) |
|-------------------------|------------|-----------------------|--------------------|
| (a) | 1 | 6,00% | 30,0 |
| | 2 | 3,00% | 40,8 |
| | 3 | 2,00% | 48,0 |
| (b) | 1 | 6,00% | 28,0 |
| | 2 | 3,00% | 50,6 |
| | 3 | 6,00% | 29,0 |
| Manutenção = 2 segundos | | | |

Fonte: Autor.

Figura 5 – Comparação de sequenciamento de tarefas com e sem a utilização de manutenção



Fonte: Autor.

Finalmente, Mendes e Iori (2019) propõem uma versão linearizada do modelo apre-

sentado por Ruiz-Torres, Paletta e M'Hallah (2017). Além do modelo linearizado, os autores apresentam uma meta-heurística de Busca Local Iterada (*Iterated Local Search* - ILS), encontrando as melhores soluções conhecidas para quase 80% das instâncias, em tempos computacionais razoáveis. O modelo apresentado pelos autores é exposto na seção seguinte.

2.3.2 Modelo Matemático proposto por Mendes e Iori (2019)

O modelo matemático proposto por Mendes e Iori (2019) para o problema $R_m|S_{dd}, Mnt|C_{max}$ será aqui denominado *IM*. Este modelo possui a mesma função objetivo dos modelos anteriores, assim como diversos parâmetros iguais. Os conjuntos de máquinas, tarefas e posições são idênticos aos de *ADF*. Cabe ressaltar que, embora a linearização tenha sido similar a utilizada por Araújo, Dhein e Fampa (2017), os autores utilizaram técnicas ligeiramente diferentes, além de não utilizar o Lema 1, que possibilita prever a ordenação da alocação de tarefas em cada máquina. As notações adicionais serão descritas a seguir.

Os novos parâmetros deste modelo são:

t_k : tempo necessário para realização de um evento de manutenção na máquina k .

M_i^a, M_i^b : Números suficientemente grandes.

O modelo possui as seguintes novas variáveis:

u_{jkh} : tempo de processamento da tarefa j na posição h da máquina k , considerando o desgaste sofrido pela máquina k até aquele momento.

$$s_{kh} : \begin{cases} 1, & \text{se houver manutenção na posição } h \text{ da máquina } k; \\ 0, & \text{caso contrário.} \end{cases}$$

$$(IM) \quad \min C_{\max} \quad (2.22)$$

$$\sum_{j \in N} x_{jkh} + s_{kh} \leq 1, \quad \forall k \in M, h \in H, \quad (2.23)$$

$$\sum_{k \in M} \sum_{h \in H} x_{jkh} = 1, \quad \forall j \in N, \quad (2.24)$$

$$u_{jkh} \leq M_i^a x_{jkh}, \quad \forall k \in M, j \in N, h \in H, \quad (2.25)$$

$$u_{jkh} \geq p_{jk} \tilde{q}_{kh} - M_i^a (1 - x_{jkh}), \quad \forall j \in N, \forall k \in M, h \in H, \quad (2.26)$$

$$\sum_{j \in N} \sum_{h \in H} u_{jkh} + \sum_{h \in H} tk_k s_{kh} - C_{max} \leq 0, \quad \forall k \in M, \quad (2.27)$$

$$\tilde{q}_{kh} \geq \tilde{d}_{jk} \tilde{q}_{k,h-1} - \mathbb{M}_i^b (s_{k,h-1} + 1 - x_{jk,(h-1)}), \quad \forall j \in N, \forall k \in M, h \in H \setminus \{1\}, \quad (2.28)$$

$$u_{jkh} \geq 0, \quad \forall j \in N, k \in M, \quad (2.29)$$

$$\tilde{q}_{k1} = 1, \quad \forall k \in M, \quad (2.30)$$

$$\tilde{q}_{kh} \geq 1, \quad \forall k \in M, \quad (2.31)$$

$$x_{jkh} \in \{0, 1\}, \quad \forall j \in N, k \in M, h \in H, \quad (2.32)$$

$$s_{kh} \in \{0, 1\}, \quad \forall k \in M, h \in H. \quad (2.33)$$

A função objetivo (2.22) considera a minimização do *makespan*. A Equação (2.23) diz que cada posição pode receber ou uma tarefa, ou um evento de manutenção, ou permanecer vazia. A Equação (2.24) afirma que todas as tarefas serão alocadas exatamente uma única vez em uma única posição de uma única máquina. As Equações (2.25) e (2.26) definem o tempo de processamento total da tarefa j , alocada na posição h da máquina k . A Equação (2.27) garante que o *makespan*, na solução ótima, será igual ao tempo total de processamento da máquina mais carregada, incluindo os tempos totais dos eventos de manutenção realizados naquela máquina. A Equação (2.28) caracteriza a TAI da máquina k na posição h . A Equação (2.29) assegura que o tempo de processamento de uma tarefa nunca será menor que zero. A Equação (2.30) assegura que qualquer máquina k terá $TAI = 100\%$ ao processar uma tarefa j na posição $h = 1$. As Equações (2.31), (2.32) e (2.33) definem, respectivamente, o domínio das variáveis \tilde{q} , x , e s .

3 RESULTADOS E DISCUSSÃO

O presente capítulo será dividido em duas seções. A seção 3.1 trata dos modelos propostos neste estudo, enquanto 3.2 apresenta os resultados computacionais obtidos.

3.1 MODELOS PROPOSTOS

Esta seção possui duas subseções, sendo que na primeira é proposta uma melhoria para o modelo ADF . A segunda subseção traz o segundo modelo proposto neste estudo, que estende a formulação apresentada em 3.1.1, incluindo restrições para tratar eventos de manutenção.

3.1.1 Proposta de Melhoria do modelo ADF : o Modelo $ADF+$

O primeiro modelo proposto neste estudo, que será referenciado como $ADF+$, é um aprimoramento do modelo ADF . A ideia subjacente reside no artifício de tornar a formulação mais parecida com a de um problema de alocação de tarefas em máquinas paralelas não relacionadas sem desgaste ($R_m || C_{max}$). Com isto, é possível obter um melhor limitante inferior derivado da relaxação linear.

A partir disso, a concepção desta proposta de melhoria levou em consideração restrições que calculam separadamente o tempo de processamento base de cada tarefa (p_{jk}) e seus respectivos tempos adicionais ocasionados pela deterioração no desempenho da máquina (\hat{u}_{jk}). Para tanto, torna-se necessário modificar as Equações (2.14) e (2.15).

Inicialmente a Equação (2.14), ou seja, a restrição referente ao cálculo do *makespan*, passa a ser formulada da seguinte maneira

$$\sum_{j \in N} (\hat{u}_{jk} + p_{j_k(h)k} x_{jk}) \leq C_{max},$$

$\forall k \in M$.

A alteração na Equação (2.14) demanda uma nova formulação que possibilite calcular o valor de \hat{u}_{jk} . Para tanto, a Equação (2.15), que até então determina o tempo de processamento da tarefa j na máquina k , passa a ter a forma a seguir

$$\hat{u}_{j_k(h)k} \geq p_{j_k(h)k} \times (\tilde{q}_{kh} - 1) - \mathbb{M}_1(k, h) (1 - x_{j_k(h)k})$$

$\forall k \in M$ e $h \in H$. Isto é, $\hat{u}_{j_k(h)k}$ consiste no tempo adicional de processamento da tarefa j

ao ser alocada à posição h da máquina k , ocasionado pelo desgaste sofrido pela máquina k até aquele instante.

Considerando o exposto, é possível apresentar $ADF+$. Este modelo possui a mesma função objetivo e os mesmos parâmetros, conjuntos e variáveis de ADF , a não ser por \hat{u}_{jk} , que é uma ressignificação do conceito atribuído à variável u_{jk} , a saber:

\hat{u}_{jk} : tempo de processamento adicional da tarefa j na posição h da máquina k .

$$\text{(ADF+)} \quad \min C_{\max} . \quad (3.1)$$

$$\sum_{k \in M} x_{jk} = 1 , \quad \forall j \in N , \quad (3.2)$$

$$\sum_{j \in N} (\hat{u}_{jk} + p_{jk(h)k} x_{jk}) \leq C_{\max} , \quad \forall k \in M , \quad (3.3)$$

$$\hat{u}_{jk(h)k} \geq p_{jk(h)k} \times (\tilde{q}_{kh} - 1) - \mathbb{M}_1(k, h) (1 - x_{jk(h)k}) , \quad \forall k \in M, h \in H , \quad (3.4)$$

$$\tilde{q}_{kh} \geq \tilde{q}_{k(h-1)} \times (1 + \tilde{d}_{jk(h-1)k}) - \mathbb{M}_2(k, h) (1 - x_{jk(h-1)k}) , \quad \forall k \in M, h \in H \setminus \{1\} , \quad (3.5)$$

$$\tilde{q}_{kh} \geq \tilde{q}_{k(h-1)} , \quad \forall k \in M, h \in H \setminus \{1\} , \quad (3.6)$$

$$\hat{u}_{jk} \geq 0 , \quad \forall j \in N, k \in M . \quad (3.7)$$

$$\tilde{q}_{kh} \geq 1 , \quad \forall k \in M , h \in H , \quad (3.8)$$

$$\tilde{q}_{k1} = 1 , \quad \forall k \in M , \quad (3.9)$$

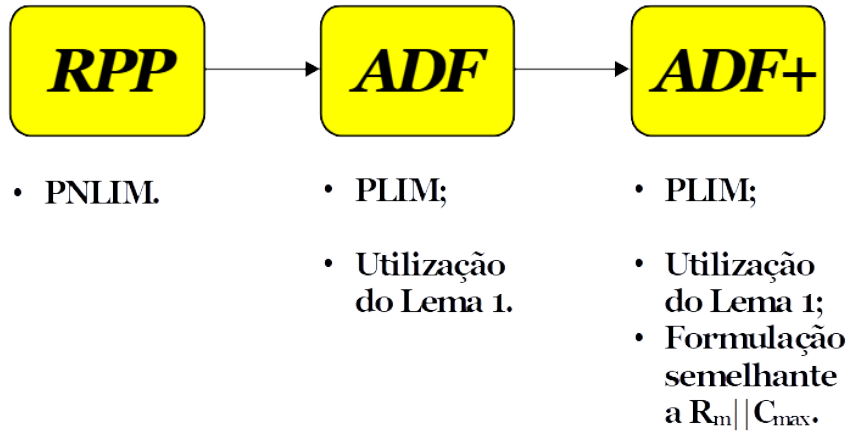
$$x_{jk} \in \{0, 1\} , \quad \forall j \in N, k \in M . \quad (3.10)$$

A função objetivo (3.1) considera a minimização do *makespan*. A Equação (3.2) garante que todas as tarefas serão alocadas exatamente uma única vez em uma única posição de uma única máquina. A Equação (3.3) assegura que o *makespan*, na solução ótima, será igual ao tempo total de processamento da máquina mais carregada. A Equação (3.4) define o tempo de processamento adicional da tarefa j alocada na posição h da máquina k , ocasionado pelo desgaste da máquina k até aquele momento. A Equação (3.5) determina a TAI da máquina k na posição h , levando em consideração o desgaste e a TAI da posição anterior $(h - 1)$, caso uma tarefa tenha sido alocada exatamente na posição $(h - 1)$. A Equação (3.6) garante que a TAI da máquina k no processamento de uma tarefa j alocada na posição h considerará o desgaste acumulado pela máquina até aquele momento, caso a tarefa j não possua uma predecessora imediata na posição $(h - 1)$. A Equação (3.7) atesta que o tempo de processamento adicional de uma tarefa, ocasionado pelo desgaste da máquina em que a mesma está alocada, nunca será menor que zero.

A Equação (3.8) assegura que qualquer máquina k terá $TAI = 100\%$ ao processar uma tarefa j na posição $h = 1$. As Equações (3.9) e (3.10) definem, respectivamente, o domínio das variáveis \tilde{q}_{kh} e x_{jk} .

Na Figura 6 é possível perceber de maneira simplificada as diferenças nas características dos modelos anteriormente apresentados, até $ADF+$.

Figura 6 – Diferenças entre os modelos RPP , ADF e $ADF+$



Fonte: Autor.

3.1.2 Proposta de modelo para o $R_m|S_{dd}, Mnt|C_{max}$

O segundo modelo proposto neste trabalho, denominado de $ADFM$, é uma adaptação de $ADF+$ para que este considere a ocorrência de eventos de manutenção. $ADFM$ possui a mesma função objetivo dos modelos anteriores, assim como os demais parâmetros. Os conjuntos de máquinas, tarefas e posições são idênticos aos de ADF . O modelo é apresentado a seguir.

$$\text{(ADFM)} \quad \min C_{max}, \quad (3.11)$$

$$\sum_{k \in M} x_{jk} = 1, \quad \forall j \in N, \quad (3.12)$$

$$\sum_{j \in N} tk_k s_{jk} + \sum_{j \in N} (\hat{u}_{jk} + p_{jk} x_{jk}) \leq C_{max}, \quad \forall k \in M, \quad (3.13)$$

$$\hat{u}_{jk(h)k} \geq p_{jk(h)k} \times (\tilde{q}_{kh} - 1) - \mathbb{M}_1(k, h) (1 - x_{jk(h)k}), \quad \forall k \in M, h \in H, \quad (3.14)$$

$$\tilde{q}_{kh} \geq \tilde{q}_{k(h-1)} \times (1 + \tilde{d}_{jk(h-1)k}) - \mathbb{M}_2(k, h) (1 - x_{jk(h-1)k} + s_{jk(h)k}), \quad \forall k \in M, h \in H \setminus \{1\}, \quad (3.15)$$

$$\tilde{q}_{kh} \geq \tilde{q}_{k(h-1)} - \mathbb{M}_3(k, h) (s_{jk(h)k}), \quad \forall k \in M, h \in H \setminus \{1\}, \quad (3.16)$$

$$\hat{u}_{jk} \geq 0, \quad \forall j \in N, k \in M, \quad (3.17)$$

$$\tilde{q}_{k1} = 1, \quad \forall k \in M, \quad (3.18)$$

$$\tilde{q}_{kh} \geq 1, \quad \forall k \in M, h \in H, \quad (3.19)$$

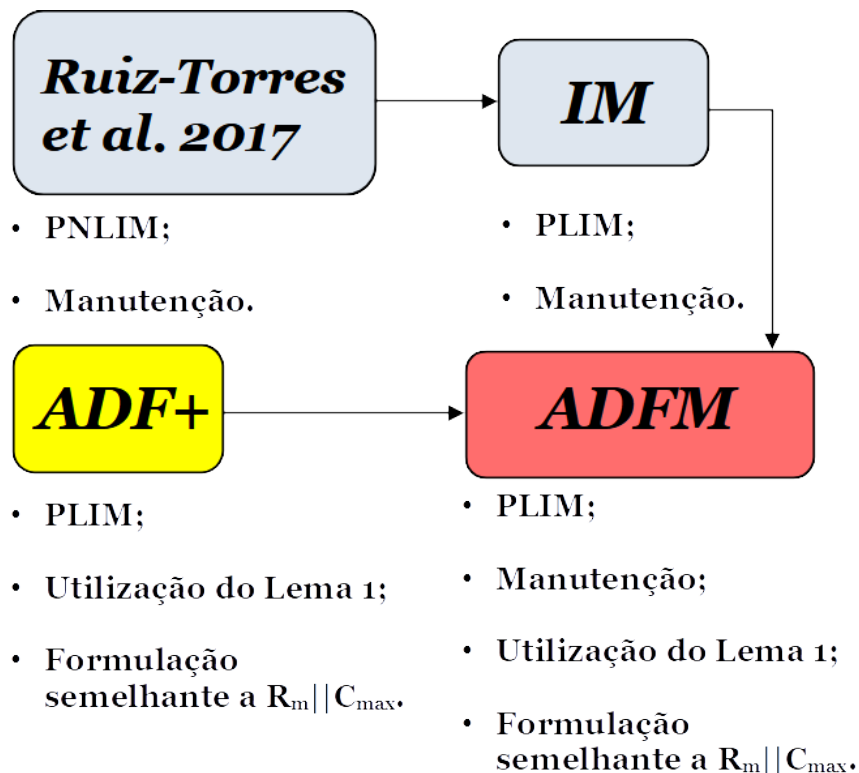
$$x_{jk} \in \{0, 1\}, \quad \forall j \in N, k \in M, \quad (3.20)$$

$$s_{jk} \in \{0, 1\}, \quad \forall k \in M, h \in H. \quad (3.21)$$

A função objetivo (3.11) considera a minimização do *makespan*. A Equação (3.12) garante que todas as tarefas serão alocadas exatamente uma única vez em uma única posição de uma única máquina. A Equação (3.13) assegura que o *makespan*, na solução ótima, será igual ao tempo total de processamento da máquina mais carregada, incluindo os tempos totais dos eventos de manutenção realizados naquela máquina. A Equação (3.14) define o tempo de processamento adicional da tarefa j alocada na posição h da máquina k , ocasionado pelo desgaste da máquina k até aquele momento. A Equação (3.15) demonstra a TAI da máquina k na posição h , levando em consideração o desgaste e a TAI da posição anterior $(h - 1)$, caso uma tarefa tenha sido alocada exatamente na posição $(h - 1)$, ou um evento de manutenção tenha ocorrido na posição h . A Equação (3.16) garante que a TAI da máquina k no processamento de uma tarefa j alocada na posição h considerará o desgaste acumulado pela máquina até aquele momento, caso a tarefa j não possua uma tarefa predecessora imediata na posição $(h - 1)$, tampouco um evento de manutenção na posição h . A Equação (3.17) afirma que o tempo de processamento adicional de uma tarefa, ocasionado pelo desgaste da máquina em que a mesma está alocada, nunca será menor que zero. A Equação (3.18) assegura que qualquer máquina k terá $TAI = 100\%$ ao processar uma tarefa j na posição $h = 1$. As Equações (3.19), (3.20) e (3.21) definem, respectivamente, o domínio das variáveis \tilde{q} , x e s .

Na Figura 7, é possível identificar de forma sintetizada as diferentes características dos modelos anteriormente apresentados, até a exposição de $ADF+$.

Figura 7 – Diferenças entre os modelos de Ruiz-Torres, Paletta e M'Hallah (2017), *IM*, *ADF+* e *ADFM*



Fonte: Autor.

3.2 RESULTADOS COMPUTACIONAIS

Os testes computacionais consideram dois cenários: o primeiro, referente ao problema $R_m | S_{dd} | C_{max}$, trata da comparação entre os modelos *ADF+* e *ADF* quando processados por um resolvidor que representa o estado da arte dos resolvidores comerciais de programação matemática; o segundo diz respeito ao problema $R_m | S_{dd}, Mnt | C_{max}$ e apresenta uma análise dos resultados obtidos através do modelo *ADFM*, considerando os resultados publicados em Mendes e Iori (2019).

Os testes foram realizados em um computador com processador Intel ES-2680, com velocidade de *clock* de 2.7 GHz e 64 GB RAM. O resolvidor comercial escolhido para o processamento dos modelos foi o IBM ILOG CPLEX Optimization Studio v12.8.0.0, com configuração padrão, máximo de 4 *threads* e tempo limite escolhido arbitrariamente em uma hora. Os modelos matemáticos foram implementados na linguagem de modelagem algébrica ZIMPL v. 3.3.8.

3.3 COMPARAÇÃO DOS RESULTADOS OBTIDOS PELOS MODELOS ADF E $ADF+$

As instâncias utilizadas nos testes computacionais correspondem àquelas propostas por Ruiz-Torres, Paletta e Pérez (2013) para o problema $R_m|S_{dd}|C_{max}$, disponíveis em <http://ruiz-torres.uprrp.edu/dm/>. A configuração destas instâncias considera quatro parâmetros base, a saber, o número de máquinas m , o número de tarefas n , o intervalo de tempo de processamento p_{range} e o intervalo de deterioração d_{range} . Foram definidos dois conjuntos de instâncias, um deles considerado de pequeno porte e o outro de grande porte. Os conjuntos apresentam 2, 3 ou 4 máquinas e 8, 11 ou 14 tarefas (pequeno porte), e 4, 7 ou 10 máquinas e 20, 35 ou 50 tarefas (grande porte). Os valores para o tempo de processamento p_{jk} foram selecionados aleatoriamente a partir de um dos intervalos com distribuição uniforme p_{range} , [1,100] e [100,200]. O mesmo se aplica para os valores da deterioração d_{jk} , que foram selecionados de forma aleatória dentro de dois intervalos com distribuição uniforme d_{range} , [1% a 5%] e [5% a 10%]. Ao todo, foram geradas 1800 instâncias, sendo 900 instâncias de pequeno porte e 900 instâncias de grande porte.

Os testes comparativos com os modelos ADF e $ADF+$ foram realizados somente com as instâncias de grande porte, uma vez que Araújo, Dhein e Fampa (2017) resolveram todas as instâncias de pequeno porte na otimalidade com tempos computacionais muito baixos. Os resultados de ADF , constantes neste estudo, foram reproduzidos com a mesma plataforma de *hardware* supracitada, que foi utilizada para o processamento dos modelos aqui propostos.

Na Tabela 4, que apresenta a comparação dos resultados obtidos pelos modelos ADF e $ADF+$, as colunas p_{range} , d_{range} , m e n , representam, respectivamente, a faixa de tempo de processamento, a faixa de deterioração, o número de máquinas e o número de tarefas de cada instância. As colunas F.O, Tempo e *Gap* contém, respectivamente, o valor encontrado para a solução objetivo, em unidades de tempo, o tempo de execução do resolvidor e o *Gap* fornecido pelo CPLEX, devidamente separadas por modelo. Ainda, cada linha contém a média dos resultados encontrados para vinte e cinco instâncias de cada configuração possível. Ao final, a última linha contém as médias dos resultados apresentados em cada coluna da Tabela.

Tabela 4 – Comparação dos Resultados dos Modelos ADF e $ADF+$

(continua)

| p_{range} | d_{range} | m | n | Modelo ADF | | | Modelo $ADF+$ | | |
|-------------|-------------|-----|-----|--------------|-----------|---------|---------------|-----------|---------|
| | | | | F.O. | Tempo (s) | Gap (%) | F.O. | Tempo (s) | Gap (%) |
| | | | 20 | 121,16 | 0,1 | 0,00 | 121,16 | 0,1 | 0,00 |
| | | 4 | 35 | 210,54 | 0,5 | 0,00 | 210,54 | 0,4 | 0,00 |
| | | | 50 | 301,34 | 19,6 | 0,00 | 301,34 | 18,0 | 0,00 |
| | | | 20 | 53,26 | 0,1 | 0,00 | 53,26 | 0,1 | 0,00 |
| | [1%, 5%] | 7 | 35 | 80,87 | 0,7 | 0,00 | 80,87 | 0,4 | 0,00 |

| | | | | (conclusão) | | | | | |
|-------------|-------------|----------|---------|--------------|----------|---------|---------------|----------|---------|
| P_{range} | d_{range} | m | n | Modelo ADF | | | Modelo $ADF+$ | | |
| | | | | F.O. | Time (s) | Gap (%) | F.O. | Time (s) | Gap (%) |
| [1, 100] | | 10 | 50 | 111,72 | 18,7 | 0,00 | 111,72 | 7,4 | 0,00 |
| | | | 20 | 34,17 | 0,1 | 0,00 | 34,17 | 0,0 | 0,00 |
| | | | 35 | 47,79 | 0,9 | 0,00 | 47,79 | 0,5 | 0,00 |
| | | 4 | 50 | 58,11 | 7,7 | 0,00 | 58,11 | 3,0 | 0,00 |
| | | | 20 | 658,91 | 12,0 | 0,00 | 658,91 | 1,0 | 0,00 |
| | | | 35 | 1172,31 | 1734,7 | 0,29 | 1172,31 | 1564,4 | 0,24 |
| | [5%, 10%] | 7 | 50 | 1733,77 | 3600,0 | 4,70 | 1734,09 | 3600,0 | 4,26 |
| | | | 20 | 350,77 | 14,8 | 0,00 | 350,77 | 0,9 | 0,00 |
| | | | 35 | 613,35 | 3453,1 | 2,13 | 613,35 | 175,4 | 0,28 |
| | | 10 | 50 | 909,04 | 3600,0 | 6,84 | 908,98 | 3600,0 | 6,27 |
| | | | 20 | 239,77 | 25,5 | 0,00 | 239,77 | 0,3 | 0,00 |
| | | | 35 | 443,73 | 3600,0 | 9,52 | 443,48 | 1291,7 | 1,49 |
| [100, 200] | | 4 | 50 | 592,69 | 3600,0 | 4,22 | 592,69 | 1846,4 | 0,30 |
| | | | 20 | 127,93 | 0,1 | 0,00 | 127,93 | 0,1 | 0,00 |
| | | | 35 | 226,62 | 23,5 | 0,00 | 226,62 | 20,4 | 0,00 |
| | | [1%, 5%] | 50 | 346,63 | 3574,6 | 6,77 | 346,51 | 3568,3 | 6,70 |
| | | | 20 | 53,51 | 0,1 | 0,00 | 53,51 | 0,1 | 0,00 |
| | | | 35 | 84,02 | 5,3 | 0,00 | 84,02 | 2,8 | 0,00 |
| | [5%, 10%] | 7 | 50 | 113,88 | 1675,0 | 1,23 | 113,84 | 1603,4 | 0,88 |
| | | | 20 | 35,00 | 0,1 | 0,00 | 35,00 | 0,0 | 0,00 |
| | | | 35 | 46,56 | 1,3 | 0,00 | 46,56 | 0,7 | 0,00 |
| | | 10 | 50 | 63,77 | 312,4 | 0,03 | 63,77 | 93,2 | 0,00 |
| | | | 20 | 723,91 | 217,1 | 0,12 | 723,91 | 192,4 | 0,12 |
| | | | 35 | 1409,02 | 3600,0 | 13,56 | 1408,38 | 3600,0 | 13,48 |
| [5%, 10%] | 4 | 50 | 2271,50 | 3600,0 | 28,17 | 2271,79 | 3600,0 | 27,09 | |
| | | 20 | 374,67 | 397,9 | 0,00 | 374,67 | 9,1 | 0,00 | |
| | | 35 | 668,11 | 3600,0 | 8,60 | 668,16 | 3600,0 | 7,46 | |
| | 7 | 50 | 1065,35 | 3600,0 | 20,21 | 1065,12 | 3600,0 | 18,66 | |
| | | 20 | 248,11 | 168,6 | 0,00 | 248,11 | 0,6 | 0,00 | |
| | | 35 | 474,33 | 3600,0 | 14,63 | 474,23 | 3600,0 | 13,71 | |
| [5%, 10%] | 10 | 50 | 649,96 | 3600,0 | 12,44 | 649,87 | 3600,0 | 11,00 | |
| | | Médias | | 464,34 | 1324,01 | 3,71 | 464,31 | 1088,92 | 3,11 |

Fonte: Autor.

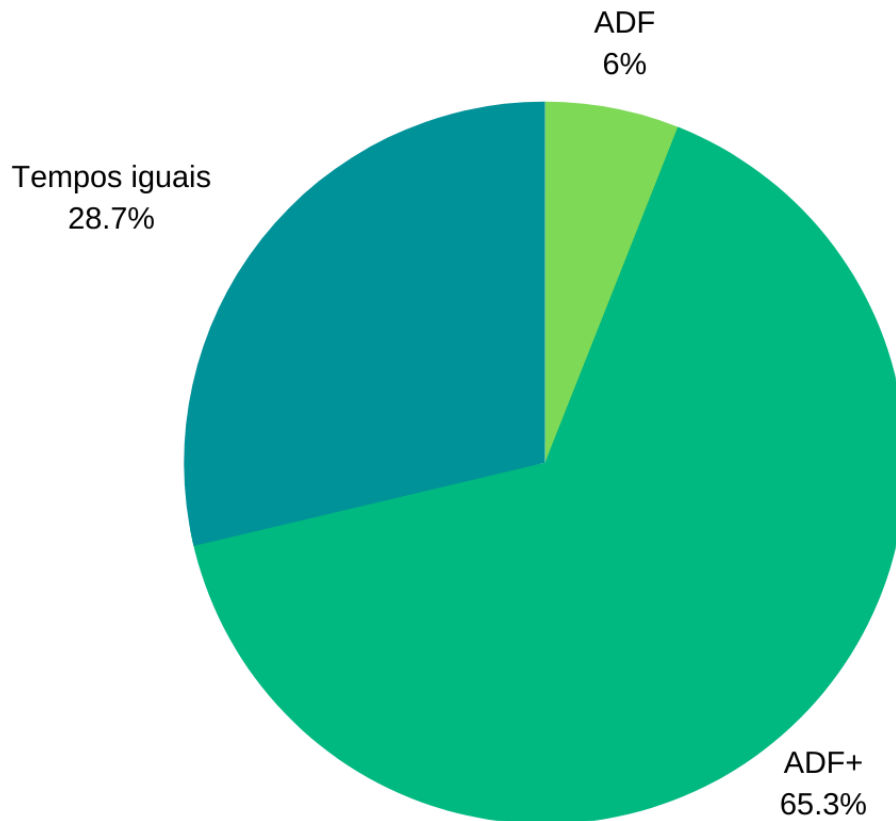
Dentre as 900 instâncias processadas, $ADF+$ resolveu 642 delas na otimalidade, contra 589 de ADF , um incremento de 9%. Além disso, $ADF+$ conseguiu tempos de processamento menores em 588 instâncias, 65% do total, e obteve tempos idênticos à ADF em outras 258. Os casos em que os tempos para ambos os modelos foram iguais, são relativos às instâncias em que o tempo máximo de processamento (uma hora) foi atingido e, nesta situação, $ADF+$ encontrou soluções com *gaps* menores em 88,37% dos cenários. Em média, os *gaps* das soluções obtidas por $ADF+$ foram 16,12% menores

que os de ADF , enquanto os tempos de processamento das instâncias foram, em média, 17,76% menores.

Testes adicionais foram realizados com ambos os modelos para analisar a qualidade da relaxação linear. Ao ser relaxado, ADF retornou o valor de zero para todas as instâncias averiguadas. Isto acontece porque a relaxação permite que as variáveis u_{jk} , através da restrição (2.15), assumam valor zero, resultando, portanto, em uma função objetiva nula. Entretanto, ao testar a relaxação linear de $ADF+$ foi verificado que os valores para a sua relaxação são iguais aos valores das soluções do modelo para o problema $R_m || C_{max}$.

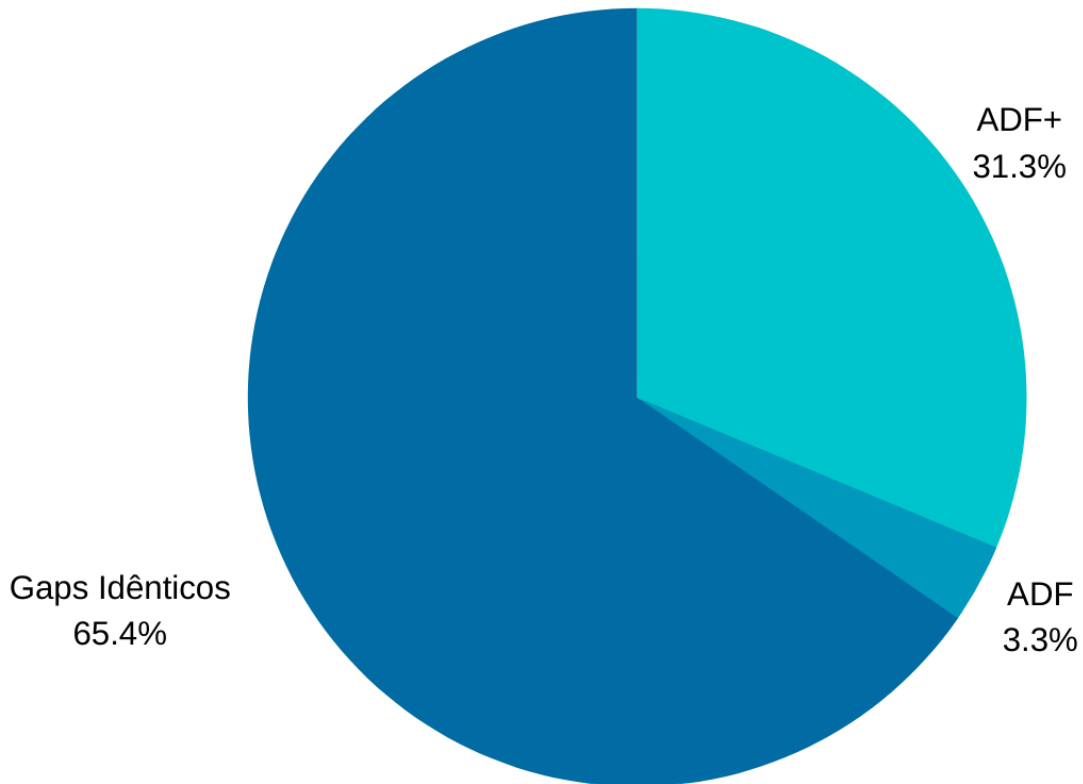
Os resultados disponíveis na Tabela 4, além das análises acima descritas, tornam evidente que o Modelo $ADF+$ conseguiu obter soluções com Gap e tempos menores ou no mínimo iguais à ADF para as instâncias avaliadas e, inclusive, independentemente da complexidade das mesmas. Precisamente, nenhuma das soluções encontradas por ADF foi superior em qualidade e em tempo de processamento, simultaneamente, em relação aos resultados obtidos por $ADF+$. Os resultados completos para os gaps e os tempos de processamento podem ser conferidos nos gráficos abaixo.

Gráfico 1 – Gráfico contendo o percentual de soluções encontradas no menor tempo por cada um dos modelos ADF e $ADF+$. A área "Tempos iguais" representa o percentual de soluções com tempos de processamento idênticos.



Fonte: Autor.

Gráfico 2 – Gráfico contendo o percentual de soluções encontradas com menor gap por cada um dos modelos ADF e $ADF+$. A área tempos "Gaps idênticos" representa o percentual de soluções com os mesmos gaps para ambos os modelos.



Fonte: Autor.

3.4 COMPARAÇÃO DOS RESULTADOS OBTIDOS PELO MODELO $ADFM$ E PELA METODOLOGIA PROPOSTA POR MENDES E IORI (2019)

O problema $R_m|S_{dd}, Mnt|C_{max}$, definido em Mendes e Iori (2019), considera que as máquinas são paralelas e não relacionadas, no entanto, no conjunto de instâncias utilizado no estudo as tarefas possuem tempo de processamento igual em todas as máquinas, enquanto os desgastes são diferentes e relativos a cada máquina. Segundo os autores, esta é uma configuração semi não relacionada. A concepção das instâncias em Mendes e Iori (2019) observou o procedimento utilizado por Ruiz-Torres, Paletta e M'Hallah (2017). Os tempos de processamento das tarefas foram gerados a partir do intervalo com distribuição uniforme $[1,100]$. Os valores de desgaste correspondem a dois intervalos d com distribuição uniforme, respectivamente, 1% a 6%, para $d = 1$, e 5% a 10%, para $d = 2$. Para o tempo de manutenção mt , dois intervalos são considerados, com duração de 1 a 3 e de 1 a 9 unidades de tempo. Ao todo, foram geradas 480 instâncias.

Neste problema, quando um evento de manutenção ocorre na posição h de uma

máquina k , o nível de performance desta máquina volta ao seu patamar máximo, ou seja, a deterioração sofrida pela mesma é zerada após a manutenção. Ou, considerando a abordagem tanto de IM como de $ADFM$, após a realização de um evento de manutenção, a TAI retorna ao patamar de 100%. Ainda, as manutenções demandam um custo igual ao tempo gasto para a realização das mesmas, sendo que este tempo pode ser diferente de uma máquina para outra.

A Tabela 5 apresenta a comparação entre os resultados obtidos com o Modelo $ADFM$ e os resultados publicados por Mendes e Iori (2019). As colunas à esquerda tratam das características das instâncias, sendo m o número de máquinas, n/m a proporção de tarefas por máquina, d a faixa de deterioração e mt o intervalo de tempos de manutenção. Cada linha da Tabela 5 representa a média dos resultados de dez instâncias aleatórias que possuem o mesmo conjunto de parâmetros. Apenas os valores apresentados na coluna NMSC (novas melhores soluções conhecidas) não são médias, e sim uma soma representando o número de novas melhores soluções conhecidas obtidas por $ADFM$. Por exemplo, na referida coluna, uma célula com o valor 9 significa que $ADFM$ encontrou nove novas melhores soluções conhecidas dentre as dez instâncias avaliadas. Sobre as demais colunas, F.O., Tempo e Gap correspondem, respectivamente, ao valor da função objetivo (em unidades de tempo - ut), ao tempo total de processamento da instância pelo resolvidor e ao valor do Gap fornecido pelo CPLEX. Já a coluna Incumbente contém o tempo para encontrar a solução incumbente e a coluna Tempo Alvo refere-se ao tempo despendido por $ADFM$ para encontrar a melhor solução apresentada por Mendes e Iori (2019). A coluna F.O. Alvo contém o valor da referida solução da literatura. Ainda, a coluna Gap Alvo apresenta o resultado da equação $GapAlvo = \left\{ \frac{F.O. - F.O.Alvo}{\min(F.O., F.O.Alvo)} \right\} \times 100$, em que valores menores que zero significam que a solução encontrada por $ADFM$ foi melhor que a de Mendes e Iori (2019). A penúltima linha contém o somatório dos valores da coluna NMSC, que corresponde à totalidade de novas melhores soluções obtidas por $ADFM$ dentre todas as instâncias analisadas. Por fim, a última linha da tabela apresenta as médias de cada coluna, excluindo NMSC.

Tabela 5 – Comparativo entre as soluções obtidas por $ADFM$ e os Resultados apresentados por Mendes e Iori (2019)

| (continua) | | | | | | | | | | | |
|------------|---------------|-----|------|----------------|--------------|--------------|-------------------|---------------------|-------------------|------|-------------------|
| m | $\frac{n}{m}$ | d | mt | F.O. (u.t.) | Tempo (s) | Gap (%) | Incumbente (s) | F.O. Alvo (u.t.) | Tempo Alvo (s) | NMSC | Gap Alvo (%) |
| | 10 | 1 | 3 | 499,94 | 38,2 | 0,00 | 16,4 | 508,81 | 0,1 | 10 | -1,79 |
| | | | 9 | 490,80 | 96,0 | 0,00 | 46,2 | 498,69 | 0,2 | 10 | -1,60 |
| | | 2 | 3 | 524,26 | 59,1 | 0,00 | 11,3 | 528,60 | 0,1 | 10 | -0,85 |
| | | | 9 | 552,02 | 830,0 | 0,13 | 367,8 | 561,25 | 7,6 | 9 | -1,65 |
| 2 | 15 | 1 | 3 | 793,05 | 3600,0 | 0,70 | 3267,1 | 803,42 | 0,1 | 10 | -1,30 |
| | | | 9 | 836,54 | 3600,0 | 1,09 | 2023,4 | 855,79 | 0,2 | 10 | -2,28 |
| | | 2 | 3 | 829,51 | 3600,0 | 0,67 | 1243,4 | 839,27 | 0,2 | 9 | -1,17 |
| | | | 9 | 816,92 | 3600,0 | 1,66 | 2707,5 | 825,59 | 1,7 | 9 | -1,04 |

| (conclusão) | | | | | | | | | | | | | | |
|-------------|---------------|-----|------|----------------|--------------|------------|-------------------|---------------------|-------------------|---------|-----------------|-------|-------|-------|
| m | $\frac{n}{m}$ | d | mt | F.O. (u.t.) | Tempo (s) | Gap (%) | Incumbente (s) | F.O. Alvo (u.t.) | Tempo Alvo (s) | NMSC | Gap Alvo (%) | | | |
| | | 1 | 3 | 1016,25 | 3600,0 | 1,55 | 3370,2 | 1023,11 | 1,2 | 7 | -0,70 | | | |
| | | | 9 | 1067,37 | 3600,0 | 2,60 | 3362,5 | 1085,66 | 0,6 | 10 | -1,73 | | | |
| | | 2 | 3 | 1053,08 | 3600,0 | 1,66 | 3188,4 | 1066,09 | 0,4 | 10 | -1,24 | | | |
| | | | 9 | 1056,96 | 3600,0 | 3,20 | 3437,4 | 1065,80 | 1,2 | 7 | -0,84 | | | |
| 5 | 10 | 1 | 3 | 522,22 | 3600,0 | 2,19 | 3389,7 | 534,55 | 2,2 | 10 | -2,40 | | | |
| | | | 9 | 495,75 | 3600,0 | 3,00 | 3378,7 | 516,26 | 0,6 | 10 | -4,19 | | | |
| | | 2 | 3 | 520,62 | 3600,0 | 2,96 | 3341,5 | 533,83 | 0,6 | 9 | -2,51 | | | |
| | | | 9 | 539,66 | 3600,0 | 5,23 | 3384,6 | 552,36 | 3,7 | 10 | -2,34 | | | |
| | | 15 | 1 | 3 | 772,67 | 3600,0 | 2,29 | 3460,9 | 786,20 | 1,4 | 10 | -1,74 | | |
| | | | 9 | 776,59 | 3600,0 | 3,60 | 3502,1 | 801,53 | 4,3 | 10 | -3,24 | | | |
| | | 2 | 3 | 801,79 | 3600,0 | 2,91 | 3384,0 | 817,99 | 1,3 | 10 | -2,03 | | | |
| | | | 9 | 785,16 | 3600,0 | 6,07 | 3455,3 | 811,44 | 334,2 | 10 | -3,35 | | | |
| 10 | 20 | 1 | 3 | 993,45 | 3600,0 | 2,51 | 3474,5 | 1011,58 | 4,8 | 10 | -1,83 | | | |
| | | | 9 | 1076,04 | 3600,0 | 3,93 | 3518,5 | 1106,26 | 15,7 | 10 | -2,84 | | | |
| | | 2 | 3 | 1053,59 | 3600,0 | 3,20 | 3444,8 | 1075,01 | 2,3 | 10 | -2,03 | | | |
| | | | 9 | 1069,84 | 3600,0 | 6,54 | 3437,7 | 1085,82 | 14,4 | 10 | -1,53 | | | |
| | | 10 | 15 | 1 | 3 | 507,48 | 3600,0 | 2,49 | 3467,9 | 531,48 | 4,7 | 10 | -4,73 | |
| | | | | 9 | 517,33 | 3600,0 | 3,52 | 3536,7 | 554,58 | 5,5 | 10 | -7,25 | | |
| | | 2 | 3 | 524,62 | 3600,0 | 3,21 | 3408,4 | 549,36 | 8,0 | 10 | -4,73 | | | |
| | | | 9 | 541,98 | 3600,0 | 5,44 | 3466,1 | 573,56 | 5,7 | 10 | -5,83 | | | |
| 10 | 15 | 1 | 3 | 777,13 | 3600,0 | 2,73 | 3533,1 | 808,17 | 11,0 | 10 | -4,01 | | | |
| | | | 9 | 763,95 | 3600,0 | 4,19 | 3576,7 | 800,09 | 16,6 | 10 | -4,75 | | | |
| | | 2 | 3 | 771,61 | 3600,0 | 3,25 | 3454,5 | 805,13 | 14,0 | 10 | -4,35 | | | |
| | | | 9 | 798,10 | 3600,0 | 6,35 | 3524,0 | 832,13 | 22,5 | 10 | -4,27 | | | |
| | | | | 2 | 1 | 3 | 1057,41 | 3600,0 | 2,66 | 3549,1 | 1082,40 | 32,8 | 10 | -2,36 |
| | | | | | 9 | 1037,35 | 3600,0 | 4,59 | 3504,9 | 1083,80 | 18,6 | 10 | -4,47 | |
| 20 | 20 | 1 | 3 | 1024,36 | 3600,0 | 3,47 | 1803,4 | 1057,23 | 26,7 | 10 | -3,21 | | | |
| | | | 9 | 1095,08 | 3600,0 | 6,84 | 3157,2 | 1123,47 | 679,4 | 10 | -2,59 | | | |
| | | 10 | 15 | 1 | 3 | 525,19 | 3600,0 | 2,83 | 3563,9 | 549,68 | 50,2 | 10 | -4,66 | |
| | | | | 9 | 536,76 | 3600,0 | 4,22 | 3528,6 | 580,74 | 31,2 | 10 | -8,19 | | |
| | | | | 2 | 3 | 519,69 | 3600,0 | 3,57 | 3488,6 | 542,77 | 73,4 | 10 | -4,46 | |
| | | | | | 9 | 534,13 | 3600,0 | 6,46 | 2872,2 | 567,28 | 62,3 | 10 | -6,22 | |
| 20 | 15 | 1 | 3 | 790,10 | 3600,0 | 2,97 | 3554,5 | 825,93 | 105,7 | 10 | -4,54 | | | |
| | | | 9 | 783,06 | 3600,0 | 4,80 | 3555,4 | 831,55 | 117,4 | 10 | -6,19 | | | |
| | | 2 | 3 | 783,42 | 3600,0 | 2,47 | 634,3 | 827,80 | 214,1 | 10 | -5,67 | | | |
| | | | 9 | 787,34 | 3600,0 | 4,42 | 1099,3 | 843,60 | 209,2 | 10 | -7,16 | | | |
| | | | | 2 | 1 | 3 | 1033,95 | 3600,0 | 3,12 | 2188,2 | 1068,44 | 514,3 | 10 | -3,34 |
| | | | | | 9 | 1069,25 | 3600,0 | 5,23 | 2254,6 | 1113,89 | 436,8 | 10 | -4,17 | |
| | | 2 | 3 | 1045,60 | 3600,0 | 1,67 | 1280,0 | 1110,26 | 300,7 | 10 | -6,19 | | | |
| | | | 9 | 1038,64 | 3600,0 | 3,10 | 1773,2 | 1123,02 | 360,8 | 10 | -8,13 | | | |
| Total | | | | | | | | | | 470 | | | | |
| Média | | | | 787,66 | 3322,9 | 3,15 | 2791,4 | 814,19 | 77,51 | | -3,49 | | | |

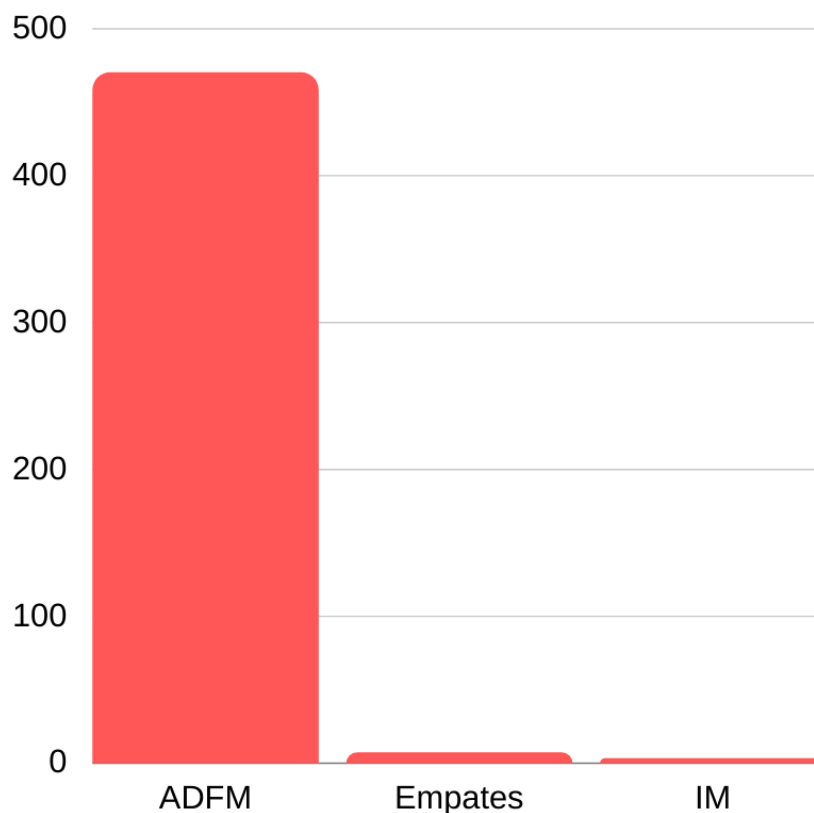
Fonte: Autor.

Mendes e Iori (2019) geraram seus resultados através de um computador com sistema operacional Ubuntu 18.04 LTS, 32 GB RAM e processador Intel Xeon E3-1245 v5

8-core de 3.5GHz. O resolvidor comercial utilizado foi o IBM ILOG CPLEX Optimization Studio v12.6, com configuração padrão e tempo limite de uma hora para o modelo IM. Para a meta-heurística ILS foram utilizados limites de até 30 iterações sem melhora na solução incumbente ou 30 minutos de execução.

Comparativamente aos resultados apresentados por Mendes e Iori (2019), o Modelo *ADFM* obteve, em média, soluções 3,49% mais próximas da otimalidade. Dentre as 480 instâncias processadas, o modelo aqui proposto encontrou as melhores soluções conhecidas para 477 delas e, dentre estas, 470 foram soluções inéditas na literatura, ou seja, 97,9% do total (conforme consta no Gráfico 3), além de necessitar, em média, de 77,51 segundos para chegar à solução alvo fornecida por Mendes e Iori (2019). Finalmente, além de *ADFM* atingir resultados substancialmente melhores, a eficiência do modelo chama atenção sobremaneira pelo fato destes serem logrados através de programação matemática, enquanto as soluções apresentadas tanto no estudo de Ruiz-Torres, Paletta e M'Hallah (2017) quanto no trabalho que serviu para esta comparação, de Mendes e Iori (2019), salienta-se, serem fruto da utilização de meta-heurísticas.

Gráfico 3 – Gráfico contendo o total de soluções encontradas por *ADFM*, soluções "empatadas" e por *IM*, respectivamente, 470, 7 e 3



Fonte: Autor.

Assim, resta claro nos resultados expostos na Tabela 5, bem como na análise realizada acima, que *ADFM* alcançou melhores resultados em 99,375% das instâncias em

comparação com Mendes e Iori (2019) e nas três instâncias em que isso não ocorreu, os valores encontrados ficaram com diferenças menores do que 1,5 %, independentemente da complexidade das mesmas. Sobretudo, a melhora na qualidade das soluções aumenta para as instâncias mais complexas, o que demonstra sua qualidade para, com os solucionadores comerciais disponíveis, apresentar resultados satisfatórios em tempos computacionais aceitáveis.

4 CONCLUSÃO

O presente trabalho abordou um problema de sequenciamento de tarefas em máquinas paralelas considerando desgastes dependentes da sequência e tempo de parada programada para manutenção. O problema considerado tem por objetivo a minimização do *makespan*. Os desgastes aqui referidos são dependentes da sequência da alocação das tarefas às máquinas e acarretam em um aumento no tempo de processamento das tarefas subsequentes. Os eventos de manutenção são utilizados para restabelecer o nível de performance das máquinas, zerando o desgaste sofrido.

No que se refere ao objetivo principal aqui proposto, foi possível apresentar um modelo matemático linear para a resolução de $R_m|S_{dd}, Mnt|C_{max}$. A forma utilizada para cumprimento do referido objetivo ocorreu através da conclusão de dois dos objetivos específicos. Primeiramente, apresentou-se uma versão melhorada do modelo de Araújo, Dhein e Fampa (2017), esta chamada *ADF+*. A partir do mesmo, realizou-se uma adaptação para possibilitar também a resolução de problemas que considerem eventos de manutenção, *ADFM*.

Pode-se concluir que os modelos propostos, neste estudo, obtiveram resultados de melhor qualidade em comparação com os disponíveis na literatura. *ADF+*, além de conseguir se sobressair frente a um modelo que já possuía resultados interessantes, provou possuir uma melhor relaxação linear. *ADFM*, por sua vez, proporcionou a obtenção de soluções ótimas e várias melhores soluções conhecidas, assim, atingindo todos os objetivos propostos neste estudo. Ainda, com a utilização destes dois modelos, que foram capazes de resolver de maneira satisfatória instâncias de porte considerável, acredita-se que em trabalhos futuros a combinação dos mesmos com abordagens heurísticas pode viabilizar a solução de instâncias de tamanhos maiores, de modo a aproximar a modelagem ainda mais de situações reais. Por fim, outra linha para estudos futuros seria adaptar estes modelos para considerar outras funções objetivos diferentes do *makespan*.

REFERÊNCIAS BIBLIOGRÁFICAS

- ADIRI, I. et al. Single machine flow-time scheduling with a single breakdown. **Acta Informatica**, v. 26, n. 7, p. 679–696, 1989.
- ALIDAEI, B.; WOMER, N. K. Scheduling with time dependent processing times: review and extensions. **Journal of the Operational Research Society**, v. 50, p. 711–720, 1999.
- ARAÚJO, O. C. B. de; DHEIN, G.; FAMPA, M. Minimizing the makespan on parallel machines with sequence dependent deteriorating effects. In: SIMPOSIO BRASILEIRO DE PESQUISA OPERACIONAL, 49., Blumenau. **Anais do XLIX SBPO**. [S.l.], 2017.
- BAKER, K. R.; TRIETSCH, D. **Principles of Sequencing and Scheduling**. New York: John Wiley & Sons, 2013. 637 p.
- BISKUP, D. Single-machine scheduling with learning considerations. **European Journal of Operational Research**, v. 115, p. 173–178, 1999.
- BROWNE, S.; YECHIALI, U. Scheduling deteriorating jobs on a single processor. **Operations Research**, v. 38, n. 3, p. 495–498, 1990.
- CAI, J.-Y.; CAI, P.; ZHU, Y. On a scheduling problem of time deteriorating jobs. **Journal of Complexity**, v. 14, n. 2, p. 190–209, 1998.
- GAWIEJNOWICZ, S.; KURC, W.; PANKOWSKA, L. Minimizing time-dependent total completion time on parallel identical machines. In: INTERNATIONAL CONFERENCE ON PARALLEL PROCESSING AND APPLIED MATHEMATICS. Berlin, Heidelberg: Springer, 2003.
- GRAHAM, R. L. et al. Optimization and approximation in deterministic sequencing and scheduling: a survey. **Annals of discrete mathematics**, v. 5, p. 287–326, 1979.
- GUPTA, S.; KYPARISIS, J. Single machine scheduling research. **Omega**, v. 15, p. 207–227, 1987.
- GUPTA, S. K.; KUNNATHUR, A. S.; DANDAPANI, K. Optimal repayment policies for multiple loans. **Omega**, v. 15, n. 4, p. 323–330, 1987.
- HERRMANN, J. W. **Handbook of production scheduling**. Nova Iorque: Springer, 2006. 319 p.
- HSU, C.-J.; CHENG, T. C. E.; YANG, D.-L. Unrelated parallel-machine scheduling with rate-modifying activities to minimize the total completion time. **Information Sciences**, v. 181, n. 20, p. 4977–4803, 2011.
- JI, M.; CHENG, T. C. E. Parallel-machine scheduling with simple linear deterioration to minimize total completion time. **European Journal of Operational Research**, v. 188, n. 2, p. 342–347, 2008.
- _____. Parallel-machine scheduling of simple linear deteriorating jobs. **Theoretical Computer Science**, v. 410, n. 38, p. 3761–3768, 2009.
- KANG, L.; NG, C. T. A note on a fully polynomial-time approximation scheme for parallel-machine scheduling with deteriorating jobs. **International Journal of Production Economics**, v. 109, n. 1-2, p. 180–184, 2007.

LAWLER, E. L. et al. Sequencing and scheduling: Algorithms and complexity. **Handbooks in operations research and management science**, v. 4, p. 445–522, 1993.

LEE, C.-Y.; LIMAN, S. D. Single machine flow-time scheduling with scheduled maintenance. **Acta Informatica**, v. 29, n. 4, p. 375–382, 1992.

LEE, W.-C.; WU, C.-C. Multi-machine scheduling with deteriorating jobs and scheduled maintenance. **Applied Mathematical Modelling**, v. 32, p. 362–373, 2008.

LENSTRA, J. K.; SHMOYS, D. B.; TARDOS, E. Approximation algorithms for scheduling unrelated parallel machines. **Mathematical programming**, v. 46, n. 1-3, p. 259–271, 1990.

LEON, J.; WU, D. On scheduling with ready-times, due-dates and vacations. **Naval Research Logistics (NRL)**, v. 39, n. 1, p. 53–65, 1992.

LOW, C.; HSU, C.-J.; SU, C.-T. Minimizing the makespan with an availability constraint on a single machine under simple linear deterioration. **Computers & Mathematics with Applications**, v. 56, p. 257–265, 2008.

MENDES, N. F. M.; IORI, M. A mathematical model and a metaheuristic for a job and maintenance machine scheduling problem with sequence dependent deterioration. In: SÍMPO-SIO BRASILEIRO DE PESQUISA OPERACIONAL, 51., Limeira. **Anais do LI SBPO**. [S.l.], 2019.

MOSHEIOV, G. Multi-machine scheduling with linear deterioration. **INFOR: Information Systems and Operational Research**, v. 36, n. 4, p. 205–214, 1998.

_____. A note: Multi-machine scheduling with general position-based deterioration to minimize total load. **International Journal of Production Economics**, v. 135, n. 1, p. 523–525, 2012.

NESELLO, V. et al. Exact solution of the single-machine scheduling problem with periodic maintenances and sequence-dependent setup time. **European Journal of Operational Research**, v. 266, n. 2, p. 498–507, 2018.

PINEDO, M. L. **Scheduling: theory, algorithms, and systems**. [S.l.]: Springer, 2016. 670 p.

RUIZ-TORRES, A. J.; PALETTA, G.; M'HALLAH, R. Makespan minimisation with sequence-dependent machine deterioration and maintenance events. **International Journal of Production Research**, v. 55, n. 2, p. 462–479, 2017.

RUIZ-TORRES, A. J.; PALETTA, G.; PÉREZ, E. Parallel machine scheduling to minimize the makespan with sequence dependent deteriorating effects. **Computers & Operations Research**, v. 40, n. 8, p. 2051–2061, 2013.

SRIVASTAVA, B. An effective heuristic for minimising makespan on unrelated parallel machines. **Journal of the Operational Research Society**, v. 49, n. 8, p. 886–894, 1998.

SURESH, V.; CHAUDHURI, D. Scheduling of unrelated parallel machines when machine availability is specified. **Production Planning & Control**, v. 7, n. 4, p. 393–400, 1996.

TOKSARI, M. D.; GÜNER, E. Parallel machine earliness/tardiness scheduling problem under the effects of position based learning and linear/nonlinear deterioration. **Computers & Operations Research**, v. 36, n. 8, p. 2394–2417, 2009.

WANG, X.-Y.; WANG, J.-J. Scheduling deteriorating jobs with a learning effect on unrelated parallel machines. **Applied Mathematical Modelling**, v. 38, n. 21-22, p. 5231–5238, 2014.

YANG, D.-L. et al. Unrelated parallel-machine scheduling with aging effects and multi-maintenance activities. **Computers & Operations Research**, v. 39, n. 7, p. 1458–1464, 2012.

YANG, S.-J. Parallel machines scheduling with simultaneous considerations of position-dependent deterioration effects and maintenance activities. **Journal of the Chinese Institute of Industrial Engineers**, v. 28, n. 4, p. 270–280, 2011.