

UMA NOVA FORMULAÇÃO MATEMÁTICA PARA UM PROBLEMA DE PLANEJAMENTO DE ORDENS DE MANUTENÇÃO PREVENTIVA DE LONGO PRAZO

João Luiz Marques de Andrade¹, Marcone Jamilson Freitas Souza², Elisângela Martins de Sá¹, Gustavo Campos Menezes¹, Sérgio Ricardo Souza¹

¹Centro Federal de Educação Tecnológica de Minas Gerais - CEFET/MG
Programa de Pós Graduação em Modelagem Matemática e Computacional
Campus Nova Gameleira - CEP 30421-169 - Belo Horizonte - MG, Brasil

²Universidade Federal de Ouro Preto - UFOP
Campus Universitário Morro do Cruzeiro - CEP 35400-000 - Ouro Preto - MG, Brasil
andrad.joao15@gmail.com, marcone@iceb.ufop.br,
elisangelamartins@cefetmg.br, gustavo@cefetmg.br,
sergio@cefetmg.br

RESUMO

Este artigo estuda uma abordagem exata para o Problema de Planejamento de Ordens de Manutenção Preventiva de Longo Prazo (PPOMPLP). O problema consiste em decidir simultaneamente quais ordens de manutenção devem ser executadas, bem como fazer a atribuição das ordens de manutenção às equipes e às máquinas ao longo do horizonte de planejamento e determinar quando cada ordem de manutenção deve ser executada. O objetivo do problema é minimizar o número de equipes de trabalho utilizadas e a soma das penalidades das ordens de manutenção não executadas. Este trabalho propõe uma nova formulação matemática baseada em variáveis indexadas no tempo. Os resultados dos experimentos computacionais mostram que a nova formulação supera substancialmente a formulação da literatura, encontrando novas soluções ótimas e melhores limitantes inferiores.

PALAVRAS CHAVE. Planejamento de manutenção de longo prazo, Sequenciamento, Formulação Indexada no Tempo.

Tópicos: Programação Matemática, Otimização Combinatória

ABSTRACT

This paper studies an exact approach to the Long-Term Preventive Maintenance Order Scheduling Problem. The problem consists of simultaneously deciding which maintenance orders should be executed, as well as assigning the maintenance orders to teams and machines over the planning horizon, and determining when each maintenance order should be executed. The goal of the problem is to minimize the number of work teams used and the sum of penalties for non-executed maintenance orders. This paper proposes a new time-indexed variables formulation. The results of the computational experiments show that the new formulation substantially outperforms the literature formulation by finding new optimal solutions and better lower bounds.

KEYWORDS. Long-term maintenance planning. Scheduling. Time-indexed Formulation.

Paper topics: Mathematical Programming, Combinatorial Optimization

1. Introdução

As intervenções de manutenção em máquinas destinam-se a garantir a sua disponibilidade, confiança e desempenho seguro, buscando maximizar a produção e a qualidade. As manutenções podem ser preventivas, com o objetivo de garantir um desempenho melhor das máquinas e prevenir falhas, ou corretivas, que têm o objetivo de reparar falhas menores ou maiores nas máquinas. Enquanto as manutenções corretivas são reativas, pois são executadas quando ocorrem falhas nas máquinas, as preventivas, por outro lado, são periódicas e programadas de acordo com um calendário pré-determinado provido pelo fabricante, por pesquisas e/ou por profissionais com experiência [Ruiz-Hernández et al., 2020].

Seguindo uma linha de pesquisa de programação de manutenções preventivas, [Aquino et al., 2018a] conduzem um estudo de caso em uma indústria de mineração de ferro para otimizar o planejamento de um conjunto de manutenções preventivas para um horizonte de 52 semanas. Eles introduzem o Problema de Planejamento de Ordens de Manutenção Preventivas de Longo Prazo (PPOMPLP). Esse problema consiste em decidir, simultaneamente, (i) quais ordens de manutenção devem ser executadas, (ii) atribuir estas ordens às equipes e (iii) determinar quando cada ordem de manutenção deve ser executada. Para isso, deve-se considerar que, caso a ordem de manutenção seja executada, ela deve ser executada dentro de uma janela de tempo em uma máquina específica e por uma equipe com habilidade necessária para realizar a atividade. No entanto, caso a ordem de manutenção não seja executada, uma penalidade deve ser contabilizada.

Em [Aquino et al., 2018a], o PPOMPLP é reduzido ao problema de sequenciamento em máquinas paralelas, em que as ordens de manutenção são tarefas e as equipes de trabalho são máquinas paralelas. No restante deste trabalho, as ordens de manutenção serão referenciados como tarefas. O conceito de multiprocessamento de tarefas (em inglês, *multiprocessor tasks* [Brucker, 2004]) é considerado no problema, pois cada tarefa deve ser executada por uma equipe e em uma máquina. O multiprocessamento de tarefas torna o PPOMPLP muito mais desafiador e complexo, uma vez que é necessário garantir que as tarefas associadas a uma máquina e alocadas a equipes diferentes não sejam executadas ao mesmo tempo. Além disso, deve-se sequenciar tarefas que usam máquinas diferentes em uma mesma equipe. De acordo com [Aquino et al., 2018a], com base na notação dos três campos [Graham et al., 1979], o PPOMPLP pode ser denotado como $MPT2_m|r_jM_j|\gamma$, em que $MPT2_m$ representa o multiprocessamento de tarefas pela equipe de trabalho e a máquina; r_j denota que existe uma restrição quanto ao instante em que cada tarefa está liberada para iniciar sua execução; M_j denota que existe uma restrição quanto ao conjunto de equipes que podem executar cada tarefa; e γ representa a função objetivo, que busca minimizar o número de equipes de trabalho e a soma das penalidades das tarefas não executadas. O objetivo é, portanto, alocar o máximo de tarefas ao mínimo de equipes. Consequentemente, não é garantido que todas as tarefas sejam executadas.

Este trabalho contribui com uma nova formulação matemática baseada em variáveis indexadas no tempo para modelar o PPOMPLP. Os resultados dos experimentos computacionais mostraram um desempenho superior da nova formulação quando comparada com a de [Aquino et al., 2018a], em termos de tempo médio de resolução da formulação pelo *solver*, quantidade de instâncias resolvidas na otimalidade, *gap* médio e limite inferior na resolução de instâncias de pequeno porte.

O restante do trabalho está organizado da seguinte forma. A Seção 2 apresenta uma visão geral da literatura relacionada com o problema estudado. A Seção 3 define o problema através de uma descrição detalhada. A Seção 4 apresenta a formulação matemática proposta por [Aquino et al., 2018a]. A Seção 5 introduz a nova formulação proposta. A Seção 6 apresenta uma comparação dos

resultados computacionais da formulação proposta neste trabalho com a formulação apresentada por [Aquino et al., 2018a]. As conclusões são apresentadas na Seção 7.

2. Revisão da literatura

Os problemas de sequenciamento de manutenção preventiva têm aplicações em diversas áreas, com vários tipos de abordagens. Dentre as áreas de aplicação, tem-se, por exemplo, o planejamento de manutenções em sistemas de energia elétrica [Froger et al., 2016] e o planejamento de manutenções em vias ferroviárias [Sedghi et al., 2021]. Uma revisão de literatura sobre otimização de manutenções é apresentada por [de Jonge e Scarf, 2020].

Na literatura referente aos problemas de planejamento de manutenções preventivas há vários trabalhos que consideram janela de tempo para execução das manutenções. [Froger et al., 2016] apresentam uma revisão de literatura que inclui os problemas *Generator Maintenance Scheduling*, que, segundo os autores, frequentemente incluem restrições de janelas de tempo para as manutenções. [Suresh e Kumarappan, 2013; Guedes et al., 2015] e [Rodríguez et al., 2018, 2021] estudam variações do problema *Generator Maintenance Scheduling* considerando que as atividades de manutenção devem ser alocadas dentro de janelas de tempo. Essa restrição é vista com frequência na literatura envolvendo problemas de sequenciamento de manutenção preventiva, pois ela tem uma relação direta com o conceito de manutenção preventiva. Os trabalhos de [Chansombat et al., 2019] e [Zhang et al., 2019] também consideram janelas de tempo e manutenções preventivas na indústria de bens de capital e nas vias ferroviárias. [Geurtsen et al., 2022] revisam problemas que integram sequenciamento de produção com atividades de manutenção. Nesse trabalho existe uma seção dedicada aos problemas de sequenciamento de máquinas paralelas com manutenções em janelas de tempo. Esta classe de problema considera que existe um instante predefinido em que a manutenção preventiva deve ser inicializada.

Em uma linha de pesquisa com relativa proximidade aos artigos citados, o *Order Acceptance and Scheduling Problem* (OASP) busca definir quais tarefas são aceitas para o processamento e determinar um cronograma para elas, com objetivo de maximizar a receita. Para uma revisão de literatura do OASP veja [Slotnick, 2011]. Os estudos sobre OASP têm-se voltado principalmente para os problemas com máquina única [Slotnick e Morton, 2007; Oguz et al., 2010; Chaurasia e Singh, 2017; Silva et al., 2018; Tarhan e Oguz, 2021a,b], enquanto que os problemas com máquinas paralelas [Wang et al., 2015; Emami et al., 2016, 2017; Ou e Zhong, 2017; Wu et al., 2018; Wang e Ye, 2019; Liu e Lu, 2020; Naderi e Roshanaei, 2020] receberam uma menor atenção. Em [Wang et al., 2015; Emami et al., 2016, 2017; Wu et al., 2018; Wang e Ye, 2019] e [Naderi e Roshanaei, 2020] são estudados problemas com o objetivo de maximização da receita líquida total como a diferença entre a soma das receitas das tarefas aceitas e o atraso total ponderado. [Ou e Zhong, 2017] e [Liu e Lu, 2020] se dedicam a problemas com máquinas idênticas, com o objetivo de minimizar o *makespan* de todas as tarefas aceitas adicionado à soma total das penalidades das tarefas não rejeitadas.

Com relação à literatura do PPOMPLP, [Aquino et al., 2018a,b, 2019] tratam o PPOM-PLP, que foi baseado em um estudo de caso de planejamento de manutenções preventivas de uma unidade de beneficiamento de minério de ferro no Brasil. Esses trabalhos tiveram como objetivo a otimização do planejamento de um conjunto de manutenções preventivas em um horizonte de planejamento de 52 semanas. Para solucionar o problema em instâncias de larga escala, foram implementados algoritmos meta-heurísticos. As soluções geradas pelos algoritmos desenvolvidos foram capazes de produzir um planejamento melhor que os empregados pela empresa. [Santos et al., 2022] desenvolveram novos algoritmos meta-heurísticos para solucionar o problema, sendo o associado à implementação da metaheurística *Iterated Local Search* aquele de melhor desempe-

nho. Os autores destacam uma melhora de 40,5% no resultado obtido na instância real em relação ao trabalho de [Aquino et al., 2019]. Em [Andrade et al., 2022] é proposto um algoritmo híbrido ILS-GRASP que encontra novos limites superiores para instâncias de larga escala e para a instância real em um tempo computacional significativamente menor que o de [Aquino et al., 2019].

Até onde sabemos, nenhum outro trabalho, com exceção dos trabalhos de [Aquino et al., 2018a,b, 2019; Santos et al., 2022; Andrade et al., 2022], consideraram o problema estudado neste artigo. Os demais trabalhos na literatura estudam problemas com apenas algumas características semelhantes ao PPOMPLP.

3. Descrição do problema

O PPOMPLP pode ser formalmente definido como a seguir. Seja I um conjunto de tarefas que devem ser executados, K um conjunto de equipes de trabalho disponíveis para executar essas tarefas e M um conjunto de máquinas nas quais essas tarefas serão realizadas.

Cada tarefa $i \in I$ pode ser executada apenas por um subconjunto $\Gamma_i \subseteq K$ de equipes que possuem a habilidade hab_i requerida para realizar a tarefa. Além disso, cada tarefa $i \in I$ requer um tempo de execução p_i e deve ser realizada na máquina $m = e_i \in M$ dentro da janela de tempo $[\alpha_i, \beta_i]$. As tarefas podem ser executadas em qualquer instante da janela de tempo, desde que iniciem e finalizem dentro da janela de tempo. Caso a tarefa $i \in I$ não seja finalizada dentro da janela, ela é classificada como não executada e penalizada com o valor w_i .

Cada equipe de trabalho $k \in K$ tem apenas uma habilidade $hab_k \in K$ (mecânica, elétrica, etc.) para realizar as tarefas. Esta habilidade está disponível somente no intervalo de tempo $[0, h_k]$. Além disso, seja o subconjunto $\Omega_k \subseteq I$, que denota o conjunto de tarefas que a equipe de trabalho $k \in K$ tem habilidade para realizar. Por fim, seja o subconjunto $\Phi_m \subseteq I$, que denota o conjunto de tarefas que podem ser realizadas na máquina $m \in M$.

O PPOMPLP tem, como objetivo, minimizar o número de equipes de trabalho ativas e o somatório das penalidades das tarefas não executadas, satisfazendo às seguintes restrições: (i) se executada, cada tarefa deve ser designada simultaneamente a uma equipe de trabalho e a uma máquina; (ii) cada equipe de trabalho pode executar somente uma tarefa por vez; (iii) cada máquina $m \in M$ não pode ter mais de uma tarefa sendo executada simultaneamente; (iv) cada equipe $k \in K$ está disponível somente no intervalo $[0, h_k]$; (v) cada tarefa $i \in I$ deve ser executada dentro de sua janela de tempo $[\alpha_i, \beta_i]$ previamente estabelecida; e (vi) cada tarefa $i \in \Omega_k \subseteq I$ requer uma equipe $k \in \Gamma_i \subseteq K$ que tenha a habilidade específica para realizá-la.

A Figura 1 ilustra, através de um gráfico de Gantt, uma solução factível para o PPOMPLP para uma instância-teste. Os dados de entrada referentes a este exemplo são apresentados nas Tabelas 1 e 2. A Tabela 1 mostra, para cada tarefa i , a habilidade hab_i necessária para executar a tarefa e demais parâmetros, de acordo com a notação apresentada na definição do problema. A Tabela 2 apresenta, para cada equipe k , a habilidade hab_k que a equipe de trabalho possui e o instante final h_k , em horas, da disponibilidade da equipe.

O gráfico de Gantt apresentado na Figura 1 mostra a alocação das tarefas às equipes de trabalho e às máquinas. As barras em cor cinza representam a janela de tempo em que cada equipe pode trabalhar. Observe que a equipe 1 executa, nesta sequência, as tarefas 4 e 3, enquanto que a equipe 2 executa, nesta sequência, as tarefas 2 e 1, e a equipe 3 executa, nesta sequência, as tarefas 5 e 8. Note, também, que as tarefas 4 e 3 estão alocados à equipe 1, pois ela possui a habilidade necessária para executar as duas tarefas. As demais tarefas estão alocadas às equipes 2 e 3, pois requerem a habilidade 2. As três cores diferentes identificam em qual máquina será realizada cada tarefa. Note que as tarefas 2 e 3, nesta sequência, estão alocados à máquina 199; as tarefas 5 e 1 estão alocadas à máquina 213 e as tarefas 4 e 8 estão alocadas à máquina 342. Evidentemente,

Tabela 1: Dados de entrada das tarefas.

i	hab_i	e_i	α_i	β_i	p_i	w_i
1	2	213	2	8	2	9
2	2	199	0	7	4	8
3	1	199	3	7	2	10
4	1	342	0	6	4	9
5	2	213	1	10	5	10
6	1	213	2	7	1	7
7	2	199	3	10	3	7
8	2	342	4	9	3	8

Tabela 2: Dados de entrada das equipes de trabalho.

k	hab_k	h_k
1	1	6
2	2	10
3	2	9

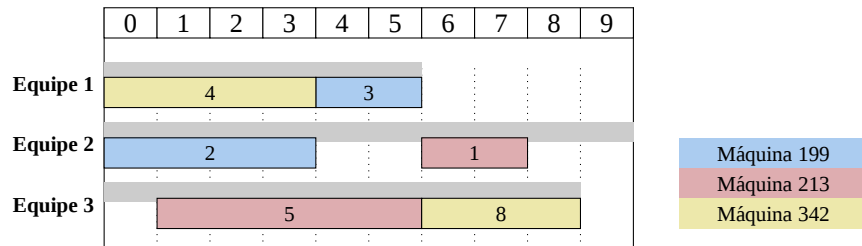


Figura 1: Ilustração de uma solução factível para o PPOMPLP, que representa a alocação das tarefas às equipes e às máquinas, bem como os instantes de início e fim de cada tarefa alocada.

os tempos de início e conclusão das tarefas são os mesmos das equipes de trabalho e máquinas. Por esta figura, note que existe uma sequência de tarefas para cada equipe e cada máquina, que é uma das principais características que aumentam a complexidade do problema. Note, ainda, que na solução apresentada na Figura 1 não foi possível alocar as tarefas 6 e 7 em nenhuma equipe; logo, elas não serão executadas.

4. Formulação matemática da literatura

A formulação matemática proposta por [Aquino et al., 2018a] é baseada em variáveis em rede [Queyranne e Schulz, 1994]. Nos modelos com variáveis em rede para problemas de máquinas paralelas, as tarefas são representadas por nós da rede e o planejamento para uma máquina é representado por um caminho na rede. Os nós visitados neste caminho representam a sequência nas quais as tarefas são processadas por uma máquina. Esta classe de variáveis tem, como uma das suas vantagens, o baixo tempo de resolução da relaxação linear de sua formulação, e tem, como desvantagem, o fornecimento de limites inferiores fracos [Unlu e Mason, 2010].

A formulação matemática proposta por [Aquino et al., 2018a], denotada por FN, utiliza os conjuntos e os parâmetros apresentados na Seção 3. Além disso, esta formulação utiliza os parâmetros B' e B'' para representar números suficientemente grandes. Sejam, então, as seguintes variáveis de decisão: (i) y_k é uma variável binária, que é igual a 1 se a equipe de trabalho k é ativada e 0, caso contrário; (ii) x_{ik} é uma variável binária, que é igual a 1 se a tarefa i for programada para ser executada pela equipe de trabalho k e 0, caso contrário; (iii) v_{ij}^k é uma variável binária, que é igual a 1 se a tarefa i precede imediatamente a tarefa j programada para ser executada pela equipe

de trabalho k e 0, caso contrário; (iv) r_{ij} é uma variável binária, que é igual a 1 se a tarefa i precede a tarefa j no uso da mesma máquina quando ambas as tarefas são executadas por equipes diferentes e 0, caso contrário; (v) c_{ik} é uma variável maior ou igual a zero que representa o tempo de conclusão da tarefa i pela equipe de trabalho k . Além disso, a formulação considera a tarefa 0, que possui o tempo de processamento igual a zero. Essa tarefa deve ser a primeira tarefa na sequência de qualquer máquina. A formulação matemática proposta por [Aquino et al., 2018a] é dada por:

$$FN : \min z = \sum_{k \in K} y_k + \sum_{i \in I} w_i \left(1 - \sum_{k \in \Gamma_i} x_{ik} \right) \quad (1)$$

$$\text{sujeito a: } \sum_{k \in \Gamma_i} x_{ik} \leq 1 \quad \forall i \in I \quad (2)$$

$$\sum_{i \in \Omega_k \cup \{0\} \setminus \{j\}} v_{ij}^k = x_{jk} \quad \forall j \in I, k \in \Gamma_j \quad (3)$$

$$\sum_{j \in \Omega_k} v_{0j}^k = y_k \quad \forall k \in K \quad (4)$$

$$\sum_{i \in \Omega_k \cup \{0\} \setminus \{l\}} v_{il}^k = \sum_{j \in \Omega_k \cup \{0\} \setminus \{l\}} v_{lj}^k \quad \forall k \in K, l \in \Omega_k \quad (5)$$

$$c_{0k} = 0 \quad \forall k \in K \quad (6)$$

$$c_{jk} \geq c_{ik} + p_j - B'_{ij}(1 - v_{ij}^k) \quad \forall k \in K, i \in \Omega_k \cup \{0\}, j \in \Omega_k \quad (7)$$

$$c_{ik} \geq (\alpha_i + p_i)x_{ik} \quad \forall k \in K, i \in \Omega_k \quad (8)$$

$$c_{ik} \leq \beta_i \quad \forall k \in K, i \in \Omega_k \quad (9)$$

$$c_{jk'} \leq c_{ik} - p_i + B''_{ij}r_{ij} \quad \forall k \in K, \forall k' \in K, i \in \Omega_k, j \in \Omega_{k'}, k \neq k', i < j, e_i = e_j \quad (10)$$

$$c_{jk'} \geq c_{ik} + p_j - B'_{ij}(1 - r_{ij}) \quad \forall k \in K, \forall k' \in K, i \in \Omega_k, i \in \Omega_{k'}, k \neq k', i < j, e_i = e_j \quad (11)$$

$$c_{ik} \leq h_k \quad \forall k \in K, i \in \Omega_k \quad (12)$$

$$v_{ij}^k \in \{0,1\} \quad \forall k \in K, i \in \Omega_k \cup \{0\}, j \in \Omega_k \cup \{0\} \quad (13)$$

$$x_{ik} \in \{0,1\} \quad \forall k \in K, i \in \Omega_k \cup \{0\} \quad (14)$$

$$y_k \in \{0,1\} \quad \forall k \in K \quad (15)$$

$$r_{ij} \in \{0,1\} \quad \forall i \in I, j \in I | i < j, e_i = e_j \quad (16)$$

$$c_{ik} \geq \{0,1\} \quad \forall k \in K, i \in \Omega_k \quad (17)$$

A função objetivo (1) minimiza o número de equipes de trabalho ativadas (primeiro termo) e a soma das penalidades das tarefas não executadas (segundo termo). As restrições (2) garantem que, caso uma tarefa seja realizada, ela deve ser executada por, no máximo, uma equipe. As restrições (3) garantem que, se alguma tarefa i preceder a tarefa j para ser executada pela equipe k é por que a tarefa j terá sido programada para ser executada por aquela equipe. As restrições (4) impõem que uma equipe seja ativada somente se tiver uma ou mais tarefas alocadas a ela. As restrições (5) são responsáveis por manter a sequência das tarefas executadas em cada equipe. As restrições (6) impõem que o tempo de conclusão da tarefa fictícia seja zero em todas as equipes. Esta tarefa é alocada antes da primeira tarefa executada por cada equipe.

As restrições (7) garantem que o tempo de conclusão da tarefa i seja menor ou igual do

que o tempo de conclusão da tarefa j , caso a tarefa i preceda a tarefa j na mesma equipe. As restrições (8) asseguram que o tempo de conclusão de cada tarefa seja maior ou igual que o início da janela mais o tempo de processamento. As restrições (9) impõem que o tempo de conclusão de cada tarefa seja menor ou igual que o final da janela. As restrições (10) e (11) são restrições disjuntivas, responsáveis por garantir que cada máquina não tenha mais de uma tarefa sendo executada simultaneamente. As restrições (12) asseguram que o tempo total alocado para cada equipe seja menor ou igual ao tempo total disponível para aquela equipe. As restrições (13)-(17) definem os domínios das variáveis.

5. Formulação matemática baseada em variáveis indexadas no tempo

A formulação baseada em variáveis indexadas no tempo para problemas de sequenciamento de máquinas foi introduzida por [Sousa e Wolsey, 1992], com a aplicação em um problema com uma única máquina. Neste tipo de formulação, o horizonte de planejamento é dividido em períodos. De uma forma geral, o problema consiste em definir em qual período cada tarefa deve iniciar o seu processamento, sequenciando, deste modo, as tarefas em cada máquina. A principal vantagem das formulações com variáveis indexadas no tempo é que sua relaxação linear fornece um limite inferior forte (no caso dos problemas de minimização), enquanto sua maior desvantagem é o grande número de variáveis, especialmente quando o horizonte de planejamento é grande e os tempos de processamento são muito variados. Dentre os trabalhos que utilizam formulações com variáveis indexadas no tempo, pode-se citar [Akker et al., 2000] e [Bigras et al., 2008], que propuseram uma formulação para o problema com uma única máquina, e [Berghman et al., 2021], que abordaram o problema de sequenciamento de máquinas paralelas não relacionadas por meio de uma formulação baseada em variáveis indexadas no tempo.

Neste trabalho, o conjunto de períodos é denotado por T . A formulação matemática baseada em variáveis indexadas no tempo para o PPOMPLP, denotada por FTI, utiliza os conjuntos e parâmetros apresentados na Seção 3 e as variáveis de decisão y_k e x_{ik} apresentadas na Seção 4. Para garantir que cada tarefa seja executada dentro da sua janela e do tempo de disponível das equipes, foram adicionadas limitações no índice $t \in T$ na formulação das restrições. Para isto, a formulação utiliza $\bar{T}_i^k \subseteq T$ para denotar o subconjunto de períodos em que a tarefa i pode começar a ser executada na equipe k , de forma a respeitar as janelas de tempo da tarefa e a disponibilidade da equipe, ou seja, $\bar{T}_i^k = \{t \in T : t \leq h_k - p_i, \alpha_i \leq t \leq \beta_i - p_i\}$. Além disso, será utilizado o parâmetro $\bar{f}_{ik}^t = \min\{\bar{t} \in \bar{T}_i^k : \bar{t} \geq t - p_i + 1\}$ para denotar o menor período de início de execução da tarefa i na máquina k que pode estar em execução no período t . A formulação também utiliza a variável binária u_{ik}^t que é igual a 1 se a tarefa i for programada para começar a ser executada no período t pela equipe k e 0, caso contrário, para $i \in I, k \in \Gamma_i, t \in \bar{T}_i^k$.

A FTI é baseada na versão clássica proposta por [Sousa e Wolsey, 1992]. No entanto, como a versão clássica considera que todas as tarefas devem ser processadas, foram necessárias adaptações no modelo. Além desta adaptação, foram feitas outras modificações para alocação simultânea das tarefas às equipes e às máquinas, para considerar as janelas de execução. A FTI proposta é então:

$$FTI : \min \quad z = \sum_{k \in K} y_k + \sum_{i \in I} w_i \left(1 - \sum_{k \in \Gamma_i} x_{ik} \right) \quad (18)$$

$$\text{sujeito a:} \quad (2), \quad (14) - (15) \quad (19)$$

$$x_{ik} \leq y_k \quad \forall i \in I, \forall k \in \Gamma_i \quad (20)$$

$$\sum_{t \in \bar{T}_i^k} u_{ik}^t \geq x_{ik} \quad \forall i \in I, \forall k \in \Gamma_i \quad (21)$$

$$\sum_{k \in \Gamma_i} \sum_{t \in \bar{T}_i^k} u_{ik}^t \leq 1 \quad \forall i \in I \quad (22)$$

$$\sum_{i \in \Omega_k} \sum_{f = \bar{f}_{ik}^t}^t u_{ik}^f \leq 1 \quad \forall k \in K, \forall t \in T \quad (23)$$

$$\sum_{i \in \Phi_m} \sum_{k \in \Gamma_i} \sum_{f = \bar{f}_{ik}^t}^t u_{ik}^f \leq 1 \quad \forall m \in M, \forall t \in T \quad (24)$$

$$u_{ik}^t \in \{0,1\} \quad \forall i \in I, \forall k \in \Gamma_i, \forall t \in \bar{T}_i^k \quad (25)$$

A função objetivo (18) é igual à função objetivo da FN. As restrições (20) impõem que uma equipe de trabalho será ativada somente se tiver uma ou mais tarefas alocadas. As restrições (21) associam as variáveis u_{ik}^t e x_{ik} e garantem que a tarefa é iniciada apenas na equipe em que ela é alocada. As restrições (22) garantem que, caso uma tarefa seja alocada a uma equipe, ela deverá ser iniciada em algum momento e exatamente uma vez e em um período. As restrições (23) garantem que, se a tarefa i é executada no período t na equipe k , nenhuma outra tarefa pode ser executada entre os períodos $t - p_i + 1$ e t na mesma equipe. As restrições (24) garantem que, se a tarefa i é executada no período t , nenhuma outra tarefa associada à máquina m pode ser executada entre os períodos $t - p_i + 1$ e t , garantindo, assim, que tarefas associadas à mesma máquina não sejam executadas ao mesmo tempo. As restrições (25) definem o domínio do restante das variáveis.

A Tabela 3 fornece uma comparação dos números de restrições e de variáveis das duas formulações apresentadas. Pode-se notar que os termos que mais afetam o número de restrições da FN e da FTI são, respectivamente, $|I|^2 \cdot |K|^2$ e $|K| \cdot (2|I| + |T|)$. Para o número de variáveis, os termos que exercem maior influência em FN e em FTI são, respectivamente, $|I|^2(|K| + 1)$ e $|K| \cdot |I| \cdot |T|$. Diferentemente da FN, a FTI depende de um número pseudo-polinomial de variáveis e restrições. No entanto, o número de variáveis e restrições dependem diretamente do número de períodos.

Tabela 3: Dimensões das formulações FN e FTI.

Formulação	#restrições	#variáveis
FN	$ K \cdot I \cdot (K \cdot I - K + 6) + 2 K + I $	$ K \cdot (I ^2 + I + 1) + I ^2 - I $
FTI	$ K \cdot (2 I + T) + T \cdot M + 2 I $	$ K \cdot (I \cdot T + I + 1)$

6. Experimentos computacionais

Esta seção apresenta os resultados dos experimentos computacionais para comparar e analisar o desempenho da formulação de [Aquino et al., 2018a] com a formulação proposta neste trabalho. Os experimentos computacionais foram executados em um computador equipado com um processador Intel i3-10100T 3.00 GHz, 16 GB de RAM e sistema operacional Ubuntu 18.04.6 LTS. As duas formulações apresentadas foram implementadas na linguagem C++ utilizando a biblioteca *Concert Technology* e solucionadas pelo solver CPLEX 12.9, com limite de execução fixado em 3600 segundos (1 hora). A formulação de [Aquino et al., 2018a] foi reimplementada.

Os experimentos foram realizados com as instâncias de [Aquino et al., 2018a], disponíveis em www.decom.ufop.br/prof/marcone/projects/LTPMSP.html. Para este trabalho, foram utilizadas as instâncias com o número de tarefas variando entre 20 e 80, o número de equipes entre 2 e

Tabela 4: Resultados dos experimentos computacionais com FN e FTI.

$ M $	FN				FTI			
	$t(s)$	$gap(\%)$	z_{rl}	$\#opt$	$t(s)$	$gap(\%)$	z_{rl}	$\#opt$
2	1615,71	26,05	0,00	14	670,93	0,09	278,48	21
3	2076,16	25,27	0,20	12	1397,94	3,14	826,68	16
4	3200,58	35,86	0,20	3	1603,29	11,71	466,68	16
5	3170,82	24,53	12,68	4	1611,46	3,92	510,88	16
	2515,82	27,88	3,27	33	1320,91	4,72	520,68	69

14 e o número de máquinas entre 2 e 5. No total são consideradas 100 instâncias. Este estudo considera apenas as instâncias de pequena escala por que solucionar instâncias de maior escala com a abordagem exata empregada é impraticável.

A Tabela 4 apresenta os resultados computacionais das formulações citadas neste artigo. Nesta tabela, as instâncias são separadas pelo número de máquinas. Assim, são divididas em quatro partes, com 25 instâncias cada parte. Além disso, a coluna $|M|$ indica o número de máquinas consideradas nas instâncias. Para as instâncias com mesmo número de máquinas, a coluna $t(s)$ indica a média do tempo computacional de execução do *solver*; a coluna $gap(\%)$ indica a média do *gap* entre o melhor limitante superior (*ub*) e o melhor limitante inferior (*lb*) encontrado pelo *solver*, sendo dado por: $gap = 100(ub - lb)/ub$; a coluna z_{rl} indica a média dos limites inferiores da relaxação linear; e a coluna $\#opt$ indica o número de soluções ótimas encontradas. A última linha da tabela associada às colunas $t(s)$, $gap(\%)$ e z_{rl} representam as médias para todas as instâncias, e a da coluna $\#opt$ indica o número total de soluções ótimas encontradas.

Os resultados da Tabela 4 mostram um desempenho superior da FTI em relação à FN. A FTI foi capaz de encontrar soluções ótimas em 69% das instâncias, enquanto que a FN alcançou a otimalidade em apenas 33% das instâncias. Além disso, as médias de *gaps* da tabela mostram uma diferença nas soluções encontradas pelas formulações. Os baixos valores de *gap* da FTI estão vinculados aos bons limitantes inferiores gerados pela relaxação linear da formulação, como mostrado na coluna z_{rl} . Além de obter um número maior de soluções ótimas, a FTI gera bons limitantes inferiores para todas as instâncias testadas.

O desempenho inferior da FN pode ser confirmado pelas médias de tempo computacional, que estão próximas do tempo limite de execução, pelas médias dos *gaps*, que são superiores a 24%, e o baixo número de soluções ótimas. A maior parte das soluções ótimas encontradas pela FN são nas instâncias com $|M|$ igual a 2 e a 3, e a menor parte são nas instâncias com $|M|$ igual a 4 e a 5.

7. Conclusões e trabalhos futuros

Este artigo apresenta uma formulação matemática baseada em variáveis indexadas no tempo (FTI) para o problema de planejamento de ordens de manutenção de longo prazo (PPOM-PLP). A formulação proposta foi comparada com a de [Aquino et al., 2018a] em instâncias que variam entre 20 e 80 tarefas. Entre as duas formulações, a FTI foi a que obteve o melhor desempenho, sendo capaz de encontrar soluções ótimas para 69% das instâncias. Além disso, essa formulação foi capaz de gerar bons limites inferiores para todas as instâncias consideradas.

Uma sugestão para trabalho futuro é o desenvolvimento de uma matheurística, combinando a formulação desenvolvida com uma heurística. Outra sugestão é o estudo do PPOMPLP com incertezas. Para isso, pode ser aplicada a abordagem de otimização robusta, considerando incerteza sobre os tempos de execução das tarefas.

Agradecimentos

Os autores agradecem à Coordenação de Aperfeiçoamento de Pessoal de Ensino Superior (CAPES, código de financiamento 001), ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq, processo 303266/2019-8), à Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG, processo PPM-CEX 676/17), ao Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG) e à Universidade Federal de Ouro Preto (UFOP).

Referências

- Akker, J., Hurkens, C., e Savelsbergh, M. (2000). Time-indexed formulations for machine scheduling problems: Column generation. *INFORMS Journal on Computing*, 12:111–124.
- Andrade, J. L. M. d., Souza, M. J. F., Sá, E. M. d., Menezes, G. C., e de Souza, S. R. (2022). Um algoritmo GRASP-ILS para um problema de planejamento de ordens de manutenções de longo prazo. In *Anais do LVIII Simpósio Brasileiro de Pesquisa Operacional*, Juiz de Fora, MG, 08 a 11 de novembro. Galoá. Disponível em: <https://proceedings.science/sbpo/sbpo-2022/trabalhos/um-algoritmo-grasp-ils-para-um-problema-de-planejamento-de-ordens-de-manutencoes?lang=pt-br>.
- Aquino, R. D., Chagas, J. B. C., e Souza, M. J. F. (2018a). A mixed-integer linear programming model and a simulated annealing algorithm for the long-term preventive maintenance scheduling problem. In Abraham, A., Muhuri, P. K., Muda, A. K., e Gandhi, N., editors, *Intelligent Systems Design and Applications (ISDA 2017)*, volume 736 of *Advances in Intelligent Systems and Computing*, p. 144–153. Springer International Publishing, Cham.
- Aquino, R. D., Chagas, J. B. C., e Souza, M. J. F. (2018b). A variable neighborhood search algorithm for the long-term preventive maintenance scheduling problem. In *Proceedings of the 20th International Conference on Enterprise Information Systems - Volume 1: ICEIS*, p. 303–310, Funchal, Madeira, Portugal. INSTICC, SciTePress.
- Aquino, R. D., Chagas, J. B. C., e Souza, M. J. F. (2019). Abordagem exata e heurísticas para o problema de planejamento de ordens de manutenção de longo prazo: Um estudo de caso industrial de larga escala. *Pesquisa Operacional para o Desenvolvimento*, 11(3):159–182.
- Berghman, L., Spieksma, F. C. R., e T’kindt, V. (2021). Solving a time-indexed formulation for an unrelated parallel machine scheduling problem by preprocessing and cutting planes. *RAIRO - Operations Research*, 55:S1747 – S1765.
- Bigras, L.-P., Gamache, M., e Savard, G. (2008). Time-indexed formulations and the total weighted tardiness problem. *INFORMS Journal on Computing*, 20(1):133–142.
- Brucker, P. (2004). *Scheduling Algorithms*. Springer Berlin Heidelberg. ISBN 9783540205241.
- Chansombat, S., Pongcharoen, P., e Hicks, C. (2019). A mixed-integer linear programming model for integrated production and preventive maintenance scheduling in the capital goods industry. *International Journal of Production Research*, 57(1):61–82.
- Chaurasia, S. N. e Singh, A. (2017). Hybrid evolutionary approaches for the single machine order acceptance and scheduling problem. *Applied Soft Computing*, 52:725–747. ISSN 1568-4946.
- de Jonge, B. e Scarf, P. A. (2020). A review on maintenance optimization. *European Journal of Operational Research*, 285(3):805–824. ISSN 0377-2217.

- Emami, S., Moslehi, G., e Sabbagh, M. (2017). A benders decomposition approach for order acceptance and scheduling problem: a robust optimization approach. *Computational and Applied Mathematics*, 36(4):1471–1515. ISSN 1807-0302.
- Emami, S., Sabbagh, M., e Moslehi, G. (2016). A lagrangian relaxation algorithm for order acceptance and scheduling problem: a globalised robust optimisation approach. *International Journal of Computer Integrated Manufacturing*, 29(5):535–560.
- Froger, A., Gendreau, M., Mendoza, J. E., Éric Pinson, e Rousseau, L.-M. (2016). Maintenance scheduling in the electricity industry: A literature review. *European Journal of Operational Research*, 251(3):695–706. ISSN 0377-2217.
- Geurtsen, M., Didden, J. B., Adan, J., Atan, Z., e Adan, I. (2022). Production, maintenance and resource scheduling: A review. *European Journal of Operational Research*. ISSN 0377-2217.
- Graham, R., Lawler, E., Lenstra, J., e Kan, A. (1979). Optimization and approximation in deterministic sequencing and scheduling: a survey. In Hammer, P., Johnson, E., e Korte, B., editors, *Discrete Optimization II*, volume 5 of *Annals of Discrete Mathematics*, p. 287–326. Elsevier.
- Guedes, L., Vieira, D., Lisboa, A., e Saldanha, R. (2015). A continuous compact model for cascaded hydro-power generation and preventive maintenance scheduling. *International Journal of Electrical Power & Energy Systems*, 73:702–710. ISSN 0142-0615.
- Liu, P. e Lu, X. (2020). New approximation algorithms for machine scheduling with rejection on single and parallel machine. *Journal of Combinatorial Optimization*, 40(4):929–952.
- Naderi, B. e Roshanaei, V. (2020). Branch-relax-and-check: A tractable decomposition method for order acceptance and identical parallel machine scheduling. *European Journal of Operational Research*, 286(3):811–827. ISSN 0377-2217.
- Oguz, C., Sibel Salman, F., e Bilgintürk Yalçın, Z. (2010). Order acceptance and scheduling decisions in make-to-order systems. *International Journal of Production Economics*, 125(1):200–211.
- Ou, J. e Zhong, X. (2017). Bicriteria order acceptance and scheduling with consideration of fill rate. *European Journal of Operational Research*, 262(3):904–907. ISSN 0377-2217.
- Queyranne, M. e Schulz, A. S. (1994). *Polyhedral approaches to machine scheduling*. TU, Fachbereich 3, Berlin.
- Rodríguez, J. A., Anjos, M. F., Côté, P., e Desaulniers, G. (2018). MILP formulations for generator maintenance scheduling in hydropower systems. *IEEE Transactions on Power Systems*, 33(6): 6171–6180.
- Rodríguez, J. A., Anjos, M. F., Côté, P., e Desaulniers, G. (2021). Accelerating benders decomposition for short-term hydropower maintenance scheduling. *European Journal of Operational Research*, 289(1):240–253. ISSN 0377-2217.
- Ruiz-Hernández, D., Pinar-Pérez, J. M., e Delgado-Gómez, D. (2020). Multi-machine preventive maintenance scheduling with imperfect interventions: A restless bandit approach. *Computers & Operations Research*, 119:104927. ISSN 0305-0548.

- Santos, A. A., Martins, A. X., e Souza, M. J. F. (2022). Um algoritmo GRASP-ILS para um problema de planejamento de ordens de manutenções de longo prazo. In *Anais do Simpósio Brasileiro de Pesquisa Operacional*, Juiz de Fora, MG, 08 a 11 de novembro. Galoá. Disponível em: <https://proceedings.science/sbpo/sbpo-2022/trabalhos/modelos-heuristicos-e-meta-heuristicos-para-a-resolucao-de-um-problema-de-sequen?lang=pt-br#>.
- Sedghi, M., Kauppila, O., Bergquist, B., Vanhatalo, E., e Kulahci, M. (2021). A taxonomy of railway track maintenance planning and scheduling: A review and research trends. *Reliability Engineering & System Safety*, 215:107827. ISSN 0951-8320.
- Silva, Y. L. T., Subramanian, A., e Pessoa, A. A. (2018). Exact and heuristic algorithms for order acceptance and scheduling with sequence-dependent setup times. *Computers & Operations Research*, 90:142–160. ISSN 0305-0548.
- Slotnick, S. A. (2011). Order acceptance and scheduling: A taxonomy and review. *European Journal of Operational Research*, 212(1):1–11. ISSN 0377-2217.
- Slotnick, S. A. e Morton, T. E. (2007). Order acceptance with weighted tardiness. *Computers & Operations Research*, 34(10):3029–3042. ISSN 0305-0548.
- Sousa, J. P. e Wolsey, L. A. (1992). A time indexed formulation of non-preemptive single machine scheduling problems. *Mathematical Programming*, 54(1):353–367.
- Suresh, K. e Kumarappan, N. (2013). Hybrid improved binary particle swarm optimization approach for generation maintenance scheduling problem. *Swarm and Evolutionary Computation*, 9: 69–89. ISSN 2210-6502.
- Tarhan, I. e Oguz, C. (2021a). Generalized order acceptance and scheduling problem with batch delivery: Models and metaheuristics. *Computers & Operations Research*, 134:105414.
- Tarhan, I. e Oguz, C. (2021b). A matheuristic for the generalized order acceptance and scheduling problem. *European Journal of Operational Research*, 299.
- Unlu, Y. e Mason, S. J. (2010). Evaluation of mixed integer programming formulations for non-preemptive parallel machine scheduling problems. *Computers & Industrial Engineering*, 58(4): 785–800. ISSN 0360-8352.
- Wang, S. e Ye, B. (2019). Exact methods for order acceptance and scheduling on unrelated parallel machines. *Computers & Operations Research*, 104:159–173. ISSN 0305-0548.
- Wang, X., Huang, G., Hu, X., e Cheng, T. C. E. (2015). Order acceptance and scheduling on two identical parallel machines. *Journal of the Operational Research Society*, 66(10):1755–1767.
- Wu, G.-H., Cheng, C.-Y., Yang, H.-I., e Chena, C.-T. (2018). An improved water flow-like algorithm for order acceptance and scheduling with identical parallel machines. *Applied Soft Computing*, 71:1072–1084. ISSN 1568-4946.
- Zhang, C., Gao, Y., Yang, L., Kumar, U., e Gao, Z. (2019). Integrated optimization of train scheduling and maintenance planning on high-speed railway corridors. *Omega*, 87(C):86–104.