

Introdução

Um grande desafio presente em toda a indústria de transformação, em vários outros setores da economia, é a programação de rotinas de manutenção que obedeçam a política de manutenção da empresa. Paradas mal planejadas podem gerar perdas produtivas significativas, aumento de custos operacionais e comprometimento de indicadores estratégicos de desempenho (KPIs). Nesse contexto, técnicas de Programação Inteira Mista (MILP) têm se destacado na solução de problemas complexos de agendamento, permitindo a consideração simultânea de restrições operacionais, capacidade produtiva e metas de desempenho.

Este trabalho propõe um modelo de otimização baseado em MILP para o agendamento das rotinas de manutenção de 3 processos distribuídos por 10 máquinas. O modelo incorpora restrições de não sobreposição de atividades e restrições soft associadas a KPIs industriais, permitindo equilibrar eficiência operacional e qualidade do plano gerado. A abordagem pretende reduzir impactos na produção, já que a Função Objetivo busca minimizar o tempo total indisponível de cada máquina.

Metodologia

Na formulação matemática são considerados 4 conjuntos de variáveis, $y_{k,i,t}$, $h_{k,i,t}$, $p_{k,i,t}$ e $slack_{ov,t}$, que representam máquinas, tempo decorrido desde o último processo, o atraso acumulado e o número de sobreposições de processos, respectivamente. Os índices k , i e t representam os índices da máquina, processo e tempo. A função objetivo utilizada considera 3 termos principais, penalidade atraso, custo fixo do processo e penalidade por sobreposição de processos, conforme o mostrado na Equação 1. Sendo orientada a minimizar tais penalidades.

$$\min Z = w_{late} \sum_{k \in K} \sum_{i \in I} \sum_{t \in T} p_{k,i,t} + w_{init} \sum_{k \in K} \sum_{i \in I} \sum_{t \in T} y_{k,i,t} + w_{overlap} \sum_{t \in T} slack_{ov,t} \quad (1)$$

Foram consideradas no problema 4 restrições principais. Primeiramente não se pode ocorrer o processo mais de uma vez simultaneamente na mesma máquina, restrição representada pela Equação 2

$$\sum_{r=t-d_k+1}^t y_{k,i,r} \leq 1, \quad \forall k \in K, i \in I, t \in T \quad (2)$$

onde d_k é a duração do processo k . Essa restrição é responsável para que o processo não possa ser iniciado novamente até que o processo iniciado seja concluído.

Para evitar que mais de uma máquina pare simultaneamente, foi adicionado o termo de penalidade apresentado na Equação 1 junto da restrição mostrada na Equação 3, que torna possível acrescido de uma penalidade.

$$\sum_{k \in K} \sum_{i \in I} y_{k,i,t} \leq 1 + slack_{ov,t}, \quad \forall t \in T \quad (3)$$

Dessa forma, é possível dar mais flexibilidade ao modelo e penalizar fortemente paradas simultâneas, que por sua vez podem causar impacto na produção.

Foi utilizada a abordagem *rolling horizon*, em que o problema é solucionado em janelas. Ou seja, ao observar 168 horas adiante, o problema foi quebrado em janelas de 24 horas, e o resultado de cada janela serve de entrada para a janela seguinte. O intuito de utilizar essa abordagem é equilibrar custo computacional com qualidade do resultado.

Para solução do problema foi utilizado o pacote Pyomo, um pacote voltado para problemas de otimização utilizando a linguagem python com o solver HiGHS.

Para avaliação dos resultados foram utilizados gráficos de Gantt, que por sua vez permitem a visualização do planejamento dos processos ao longo do tempo de forma direta, facilitando a tomada de decisões estratégicas com celeridade.

Resultados

O modelo foi avaliado em três cenários com condições iniciais distintas — Neutro, Próximo do Limite e Atrasado — ao longo de 7 dias (168 horas), utilizando rolling horizon com janelas de 24 horas. O solver HiGHS resolveu todas as instâncias com gap inferior a 1%.



Cenário	Custo função objetivo	Tempo total	Tempo médio/janela	Sobreposições
Neutro	8,4M	88,6s	12,7s	0
Próximo do Limite	18,5M	109,7s	15,7s	0
Atrasado	32,8M	154,7s	22,1s	3

Nos cenários Neutro e Próximo do Limite, o modelo agendou todas as manutenções **sem sobreposição**. No cenário Atrasado, houve sobreposição em **3 das 7 janelas (dias 4, 5 e 6)**. O crescimento do custo da função objetivo entre cenários confirma que o modelo **penaliza corretamente atrasos acumulados**.

Conclusões

Este trabalho desenvolveu um modelo MILP com o objetivo solucionar o desafio de otimização da programação de agendamentos de processos de manutenções de máquinas em contexto industrial. Foram utilizadas restrições *soft* e penalidades para desestimular sobreposição de agendamentos. Além disso a técnica de *rolling horizon*, garantiu o bom desempenho computacional.

A solução final obtida em cada cenário de simulação, entrega um resultado ótimo de agendamentos em tempo computacional aceitável. O aumento no custo da Função Objetivo nos cenários mais restritivos, está de acordo com o esperado. Foi verificado 3 sobreposições apenas no cenário Atrasado.

Bibliografia

- Pinedo, Michael L. 2022. *Scheduling: Theory, Algorithms, and Systems*. 6th ed. Springer Cham.
- Hart, William E., Jean-Paul Watson, and David L. Woodruff. 2011. "Pyomo: Modeling and Solving Mathematical Programs in Python." *Mathematical Programming Computation* 3(3): 219–260.