LVI SBPO SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL
DE 04 ATÉ 07 NOVEMBRO 2024
FORTALEZA | CEARÁ

OTIMIZAÇÃO DO SEQUENCIAMENTO DA PRODUÇÃO NO PROCESSO DE TÊMPERA EM UMA SIDERURGIA

Anderson Alves Tzitas
Vallourec
Belo Horizonte, MG - Brasil
anderson.atzitas@vallourec.com

João Victor de Paula Sá
Vallourec
Belo Horizonte, MG - Brasil
joao.sa@vallourec.com

João Vitor Cordeiro Sánchez
Universidade Federal de Minas Gerais
Belo Horizonte, MG - Brasil
jvcsanchez96@gmail.com

Felipe Shoei Otiai
Universidade Federal de Minas Gerais
Belo Horizonte, MG - Brasil
felipeotiai@gmail.com

Maurício Cardoso de Souza
Universidade Federal de Minas Gerais
Belo Horizonte, MG - Brasil
mauriciodesouza@gmail.com

RESUMO

Este trabalho aborda o sequenciamento de ordens de produção em um horizonte de curto prazo, com foco na etapa de têmpera em uma indústria siderúrgica. O problema envolve a minimização do tempo total necessário para completar um conjunto de operações, respeitando condições de contorno específicas do processo. Duas formulações matemáticas adaptadas foram elaboradas, e um *framework* foi desenvolvido para a seleção da formulação mais adequada de acordo com o cronograma a ser otimizado, levando em conta a configuração das datas de disponibilidade dos materiais. Além disso, um algoritmo de inserção de paradas para manutenções planejadas foi proposto visando absorver o ambiente dinâmico do processo produtivo em questão. Os resultados obtidos foram validados pelos responsáveis do setor de planejamento, demonstrando aumentos significativos de capacidade produtiva e assertividade no sequenciamento.

PALAVRAS CHAVE. Sequenciamento, Otimização, Programação Matemática, Indústria.

Tópicos: POI - PO na Indústria, PM - Programação Matemática, OC - Otimização Combinatória, AD&GP – PO na Administração e Gestão da Produção

ABSTRACT

This study deals with a short-term scheduling problem, focusing on the quenching and tempering process in a steel industry. The problem involves minimizing the total time required to complete a set of operations, respecting specific boundary conditions. Two adapted mathematical

DOI: 10.59254/sbpo-2024-193489
https://proceedings.science/p/193489?lang=pt-br

formulations have been built and a *framework* has been developed for selecting the most suitable formulation according to the schedule to be optimized, considering material availability dates. Additionally, an algorithm for the insertion of planned maintenance stops has been proposed aiming to increase the flexibility of using the developed optimization tool. The results obtained have been validated by stakeholders, demonstrating significant increases in production capacity and scheduling accuracy.

KEYWORDS. Scheduling, Optimization, Mathematical Programming, Steel Industry.

Paper Topics: POI – OR in Industry, PM – Mathematical Programming, OC – Combinatorial Optimization, AD&GP – OR in Administration and Production Management

1. Introdução

A busca contínua por decisões mais assertivas e o aumento da produtividade são essenciais para as indústrias que desejam manter sua competitividade no mercado. Com o advento da Indústria 4.0, áreas de tecnologia passaram a ser pilares da transformação industrial, desempenhando papel estratégico na busca pela maximização da eficiência e se consolidando no cotidiano de processos produtivos, como citado por Venâncio [2022]. Neste cenário, a Pesquisa Operacional torna-se uma grande aliada na aderência entre a operação e o planejamento da produção, agregando valor a partir de técnicas de otimização e simulação.

O trabalho em questão aborda os desafios do sequenciamento de ordens de produção em um horizonte de curto prazo de uma indústria siderúrgica. A etapa do processo estudada foi a de têmpera, que pode ser definida como o aquecimento ou resfriamento de materiais, com a finalidade de ajustar suas propriedades mecânicas sem alterar a forma do produto. Tal etapa atualmente é considerada o gargalo produtivo desta indústria, identificado como ponto em que a produção é limitada pela capacidade deste processo.

Diante deste contexto, surge uma necessidade de ganho de capacidade produtiva neste processo a partir de melhores decisões fundamentadas e orientadas a dados. Uma das oportunidades é sequenciar as ordens de produção (*jobs*) de maneira a minimizar o tempo despendido em *setups* existentes entre os materiais subsequentes, caracterizando um problema com *setup* dependente da sequência [Bigras et al. 2008]. O *setup* é calculado em função da diferença de temperatura no processamento de dois materiais consecutivos, pois há um tempo para resfriamento ou aquecimento dos fornos em que são submetidos. O dinamismo da linha produtiva aumenta o potencial ganho de ferramentas de suporte à decisão otimizadas, principalmente quando há um grande volume de ordens a produzir. Atualmente, gasta-se um tempo significativo de um analista de planejamento para que uma sequência boa o suficiente seja encontrada.

Sendo assim, o objetivo do trabalho envolve minimizar o *makespan*, tempo total necessário para completar um conjunto de operações J, respeitando as restrições específicas do processo. A etapa de têmpera será considerada como uma máquina única e as operações deverão ser sequenciais, de maneira com que a próxima tarefa só poderá começar quando a atual for inteiramente realizada. Dessa forma, o problema pode ser caracterizado como o *Single Machine with Setup Time Problem* (SMSP).

Devido à disponibilidade de ferramentaria, é importante a consideração das datas de disponibilidade de cada ordem, denominadas de *release dates* (r_j), determinando o momento em que uma tarefa se torna disponível para a execução. Para tratar o problema, uma revisão bibliográfica foi realizada e foram adaptadas duas formulações matemáticas. Após a implementação, se tornou viável a realização de testes considerando instâncias reais. Observou-se que a performance das formulações dependia da distribuição das *release dates* e, portanto, um *framework* foi desenvolvido

DOI: 10.59254/sbpo-2024-193489
https://proceedings.science/p/193489?lang=pt-br

para verificação de qual formulação será utilizada, visando obter a melhor resposta para uso do analista de planejamento.

Objetivando uma maior vida útil do maquinário, são realizadas paradas de manutenção programadas que precisam ser inseridas no cronograma de produção. Essas paradas têm janelas de tempo específicas para sua realização, assim podem ser tratadas por meio de um *job* artificial com *release dates* e datas de entrega (d_j). Como abordagem secundária, além de abranger essas paradas a partir de restrições adicionadas nas formulações, desenvolveu-se um Algoritmo de Inserção de Paradas (AIP) com o intuito de obter bons resultados.

Visando uma melhor aderência do planejamento otimizado aos processos operacionais, foram consideradas algumas premissas específicas do contexto produtivo. Visto que o planejamento da produção é realizado de forma contínua, é necessário a fixação de um primeiro *job* que represente o estado da máquina. Além disso, algumas ordens de produção, devido às suas características físicas, precisam passar pelo recurso produtivo duas vezes, sendo denominadas ordens de dupla têmpera. Estas ordens são tratadas como dois *jobs* distintos, mas com mesmos tempos de processamento e *setups* para outros *jobs*. Define-se B_i o conjunto de *jobs* que devem ser processados após o *job* i. Cabe notar que devido a restrições operacionais, os *jobs* de Bi não podem ser processados imediatamente após o processamento do *job* i, sendo exigido que pelo menos um *job* não pertencente a Bi seja processado entre eles. A Tabela 1 resume a notação utilizada para declarar os conjuntos e parâmetros utilizados nas formulações propostas.

É importante ressaltar que, para a comparação entre as formulações, considerou-se a qualidade da solução mensurada a partir do *gap* de otimalidade e do tempo de execução, dado que para absorver mudanças na linha produtiva a resposta precisaria ser entregue em um certo limite de tempo. Os resultados foram validados pelos responsáveis da área de sequenciamento do tratamento térmico e os ganhos de capacidade produtiva foram mensurados e comprovados após a fase de implementação da ferramenta.

O artigo está estruturado da seguinte forma. Na Seção 2 é discutido o referencial teórico utilizado. Na Seção 3 são apresentadas as formulações propostas para abordar o problema, assim como o algoritmo de inserção de paradas planejadas para a manutenção. Resultados computacionais em instâncias reais são apresentados na Seção 4 e as conclusões do trabalho na Seção 5.

Tabela 1: Notação dos conjuntos e parâmetros

Notação	**Descrição**
J	Conjunto de *jobs*
J'	Conjunto de *jobs* artificiais (representando as paradas planejadas)
B_i	Conjunto de *jobs* que não podem ser consecutivos ao processamento de $i \in J$
n	Número total de *jobs* ($\|J\|$)
s_{ij}	Tempo de *setup* do *job* i para o *job* j
p_j	Tempo de processamento do *job* $j \in J$
r_j	Data de disponibilidade (*release date*) do *job* $j \in J$
d_j	Data limite para o término (*due date*) do *job* $j \in J'$

2. Referencial Teórico

A literatura sobre sequenciamento é ampla e diversificada. Existem abordagens para o problema de tempos de *setup* dependentes em uma máquina que diferem principalmente na estrutura e no objetivo. Inicialmente, foi realizado um estudo baseado no trabalho de Nogueira et al. [2019] que apresenta diferentes formulações para o SMSP. Outra referência estudada foi a proposta por Keha et al. [2009], que também explora este problema com ênfase em diferentes funções objetivo,

DOI: 10.59254/sbpo-2024-193489
https://proceedings.science/p/193489?lang=pt-br

sendo elas, o número de atividades em atraso, atraso máximo, soma das datas de término ponderadas e atraso ponderado. Estes trabalhos foram fundamentais para entender as variadas estratégias de otimização aplicáveis ao problema de sequenciamento.

A segunda parte da revisão bibliográfica teve foco em um arcabouço de formulações provenientes do problema do Caixeiro Viajante (TSP), onde a função objetivo visa minimizar o somatório dos custos dos arcos utilizados. Devido à maturidade dos estudos relacionados ao problema do TSP, é de interesse aproveitar a estrutura e o conhecimento desse tipo de problema, como citado em Öncan et al. [2009]. No entanto, a presença de possíveis datas de disponibilidade para as tarefas nos impede de utilizar tal função objetivo de minimizar os custos dos arcos, pois resulta em ociosidade na linha, nos afastando assim da estrutura padrão para problemas do TSP. Por isto, o objetivo passa a ser minimizar o *makespan* (C_{max}) e, desta forma, o aumento da capacidade de produção da têmpera será o foco, conforme introduzido anteriormente. Bigras et al. [2008] propõe e adapta uma formulação TSP considerando janelas de tempo, no contexto do SMSP.

Nogueira et al. [2019] apresenta quatro formulações: *Completion Time and Precedence* (CTP), *Assigned Positional Variables* (APD), *Time Index* (TI) e *Arc Time Index* (ATI). De acordo com os resultados apresentados por Nogueira et al. [2019], as formulações TI e ATI demonstram uma melhor relaxação linear e uma dominância por proposições em relação às demais formulações. No entanto, nestas formulações, as variáveis são indexadas pelo tempo. No contexto em que uma discretização do tempo em minutos é necessária, a implementação desses modelos se torna desafiadora devido ao número de variáveis de decisão, considerando um horizonte de planejamento que varia de dois a quinze dias. De acordo com Nogueira et al. [2019], a formulação CTP apresenta um melhor desempenho e predominância por proposição sobre a APD e, portanto, opta-se por implementá-la. A função objetivo foi adaptada para minimizar o *makespan* e a variável y_{ij} foi utilizada para representar a precedência indireta da tarefa i em relação à tarefa j. A modelagem é válida apenas para problemas que possuem igualdade triangular na matriz de *setups*, o que ocorre no problema estudado para todas as instâncias testadas. Caso essa igualdade não seja atendida, a variável x_{ij} é considerada para representar a precedência direta. As formulações utilizando ambas as variáveis foram implementadas e testadas ao longo deste trabalho.

Bigras et al. [2008] apresenta uma formulação adaptada ao TSP que permite a consideração da desigualdade triangular por meio da inclusão de variáveis de precedência direta. Essa característica viabiliza o bloqueio de arcos de solução, respeitando a regra de ordens de dupla têmpera. A escalabilidade para instâncias com muitas ordens, com a implementação de planos de corte, foram aspectos positivos considerados em nossa análise, motivando assim a seleção desta formulação. No entanto, um aspecto negativo reside na complexidade associada ao tratamento de ponderações dos tempos de conclusão e atrasos na função objetivo.

3. Formulações Matemáticas

3.1. Formulação de precedência

A formulação denominada FA1 utiliza a variável binária y_{ij}, que assume 1 se o *job* i precede o *job* j na sequência e 0, caso contrário. A variável contínua C_j informa o tempo em que a atividade j é finalizada. A formulação FA1 é apresentada abaixo:

$$Min\, C_{max} \tag{1}$$

$$y_{ij} + y_{ji} = 1 \quad \forall\, i, j \in J \tag{2}$$

$$C_j \geq C_i + s_{ij} + p_j - M_{ij}(1 - y_{ij}), \quad \forall\, i, j \in J \tag{3}$$

DOI: 10.59254/sbpo-2024-193489
https://proceedings.science/p/193489?lang=pt-br

$$C_j \geq r_j + p_j \quad \forall\, j \in J \tag{4}$$

$$\sum_{k \in J, k!=(i,j)} y_{ik} - y_{jk} \geq 1 \quad \forall\, i \in J, \forall j \in B_i \tag{5}$$

$$y_{ij} = 1 \quad \forall\, i \in J, \forall j \in B_i \tag{6}$$

$$y_{0j} = 1 \quad \forall\, j \in J, \tag{7}$$

$$C_{max} \geq C_j \quad \forall\, j \in J, \tag{8}$$

A expressão (1) representa a função objetivo de minimização do *makespan*. A restrição (2) estabelece a relação de precedência ou subsequência entre dois *jobs*. As restrições (3) e (4) definem e calculam C_j , ou seja, o tempo de término de todas as atividades, definido a partir das *release dates*, tempos de processamento e tempos de *setups*. Conforme apresentado em Nogueira et al. [2019], define-se M_{ij} como:

$$M_{ij} = M_i - r_j + s_{ij} \quad \forall\, j \in J$$

$$M_i = max\{r_i, \max_{l \in J, l \neq i,j} r_l + \sum_{l \in J, l \neq i,j} p_l + \sum_{l \in J, l \neq i,j} \max_{k \in J, k \neq l,j} s_{lk}\} + p_i$$

Na restrição (7), garante-se a fixação de um *job* na primeira posição para representar o estado da máquina. A restrição (5) estabelece que a atividade j deve ocorrer após a atividade i, porém de maneira não consecutiva, assegurando sua separação. A introdução da restrição (6) fortaleceu os limites duais do problema e melhorou o desempenho do modelo em sua relaxação.

Define-se J' como o conjunto de *jobs* fictícios que representam uma parada planejada. Sendo assim, a restrição (9) foi adicionada na formulação. Esta restrição é adicionada ao modelo apenas quando as paradas precisam ser consideradas.

$$C_j \leq d_j \quad \forall\, j \in J' \tag{9}$$

3.2. Formulação arco-posição

A formulação denominada FA2 aborda o problema como uma variante do TSP. É definido um grafo orientado $G(J, A)$, onde J representa o conjunto de *jobs* e A, os arcos, indicando a precedência entre eles. Para essa formulação, o arco $(i, j)^t$ é interpretado como a decisão de visitar o arco (i, j) na posição t, e 0 representa um *job* artificial fixado na posição $t = 0$. Sendo assim, define-se as seguintes variáveis de decisão, conforme Tabela 2.

Tabela 2: Variáveis de Decisão da FA2

Variável	Descrição
x_{ij}^t	1, se utilizamos o arco (i, j) na posição t, 0 senão
C_t	Data de término da atividade na posição t

DOI: 10.59254/sbpo-2024-193489
https://proceedings.science/p/193489?lang=pt-br

A partir das variáveis e parâmetros definidos anteriormente, apresenta-se a FA2:

$$Min\ C_n \tag{10}$$

$$\sum_{i \in J} \sum_{t=0...n} x_{ij}^t = 1 \quad \forall\, j \in J, \tag{11}$$

$$\sum_{i \in J} \sum_{t=0...n} x_{ji}^t = 1 \quad \forall\, j \in J, \tag{12}$$

$$\sum_{j \in J} x_{0j}^0 = 1, \tag{13}$$

$$\sum_{i \in J} x_{ij}^t - \sum_{i \in J} x_{ji}^{t+1} = 0 \quad \forall\, j \in J, t = 1...n-1, \tag{14}$$

$$\sum_{i \in J} x_{i0}^n = 1 \tag{15}$$

$$C_0 = 0 \tag{16}$$

$$C_t \geq C_{t-1} + \sum_{i \in J} \sum_{j \in J} (s_{ij} + p_j) x_{ij}^{t-1} \quad \forall t = 1...n \tag{17}$$

$$C_t \geq \sum_{i \in J} \sum_{j \in J} (r_j + p_j) x_{ij}^{t-1} \quad \forall t = 1...n \tag{18}$$

$$x_{ij}^t \in \{0, 1\} \quad \forall\, i \in J, j \in J, t = 0...n, \tag{19}$$

$$C_t \geq 0 \quad \forall t = 0...n \tag{20}$$

$$\sum_{t=0...n} x_{ij}^t = 0 \quad \forall\, i \in J, \forall j \in B_i \tag{21}$$

$$\sum_{t=0...n} x_{ji}^t = 0 \quad \forall\, i \in J, \forall j \in B_i \tag{22}$$

A expressão (10) representa a função objetivo de minimização do *makespan*. As equações (11) a (15) asseguram que qualquer solução em x_{ij}^t seja um sequenciamento válido para o problema. As restrições (16), (17) e (18) definem e calculam C_t , ou seja, o tempo de término de todas as ordens, definido a partir das *release dates*, tempos de processamento e tempos de *setups*.

Em (19) e (20) temos os domínios das variáveis de decisão. As restrições (21) e (22) se referem à regra de ordens de dupla têmpera, que consiste em impedir que os *jobs* sejam consecutivos. É importante ressaltar que não há a necessidade da relação de precedência entre estas ordens, dado que os parâmetros de tempo de processamento e de *setup* para outros *jobs* são iguais na primeira e na segunda passadas na etapa de têmpera.

Objetivando a inserção das paradas para manutenção, a restrição (23) foi adicionada na formulação. Esta restrição é adicionada ao modelo apenas quando as paradas precisam ser

DOI: 10.59254/sbpo-2024-193489
https://proceedings.science/p/193489?lang=pt-br

consideradas. Novamente, necessitou-se da utilização do conjunto de *jobs* fictícios J' para representá-las. Além disso, introduziu-se o parâmetro M, que é calculado a partir da expressão: $M = \sum_{i \in J} p_i + \max_{(i \in J)} r_i + \sum_{i \in J} \max_{(j \in J)} s_{ij}$.

$$C_t \leq M(1 - x_{ij}^{t-1}) + d_j \quad \forall\, j \in J' \,\forall\, i \in J \,\forall\, t = 1...n \tag{23}$$

3.3. Algoritmo de Inserção de Paradas para manutenção (AIP)

Visando uma maior flexibilidade com relação à adição de paradas para manutenção, uma heurística foi desenvolvida. O AIP substitui a necessidade da adição dos *jobs* artificiais nas formulações apresentadas anteriormente, assim como a utilização das restrições (9) e (22). Este é um método alternativo e visa inserir essas paradas no cronograma de modo a minimizar a ociosidade e os tempos de *setup*. A seguir, o pseudocódigo do algoritmo é demonstrado.

Algorithm 1: Algoritmo de Inserção de Paradas (AIP)

Input: X_0: Cronograma inicial, J': lista de paradas ordenada por *release date*, C_i: data de término do *job* $i \in J$
Output: X: Cronograma com as paradas de manutenção inseridas

```
X ← X_0
for j̄ in J' do
    for i in X do
        if C_i ≤ r_j̄ then
            k ← job que antecede i
            Idleness ← R_j̄ − C_k
            X ← Inserir a parada j̄ entre k e i
            Busca Candidatos ← True
        end
    end
    while Busca Candidatos = True and Idleness >0 do
        L ← lista de jobs que podem ocupar a ociosidade, ordenados por impacto no
          setup
        for l in L do
            if p_l + s_kl ≤ Idleness then
                X ← Inserir o job l entre k e j̄
                Idleness ← Idleness + p_l + s_kl
                k ← l
                Busca Candidatos ← True
                Break
            end
            else
                Busca Candidatos ← False
            end
        end
    end
end
```

4. Resultados Computacionais

As formulações matemáticas e o AIP foram implementados na linguagem de programação *python* e ambiente de desenvolvimento *Spyder*. Como *solver*, foi utilizado o *Coin Branch and Cut*

DOI: 10.59254/sbpo-2024-193489
https://proceedings.science/p/193489?lang=pt-br

(CBC) Solver, sendo de código aberto e escrito na linguagem *C++*. Além disso, para facilitar a implementação das formulações, o pacote de otimização *PulP* foi utilizado.

Para a realização dos testes, foram considerados dez cronogramas reais de produção, disponibilizados pelos responsáveis do sequenciamento da têmpera. Os horizontes de planejamento variam entre dois e quinze dias. Todas as execuções foram geradas em um recurso computacional de *16,0 GB* de memória *RAM* e processador *Intel(R) Core (TM) i5-8250U CPU @ 1.60GHz*. Com relação às configurações do *solver*, assumimos um tempo máximo de execução de uma hora para todas as instâncias.

4.1. Análise dos resultados sem considerar paradas para manutenção

Inicialmente as paradas planejadas para manutenção não foram consideradas e, portanto, as Tabelas 4, 5 e 6 apresentam resultados sem os *jobs* artificiais nas formulações. Os cronogramas gerados pelo programador também desconsideram essas paradas. A maneira em que ele trabalha já segue duas etapas separadas, sendo a primeira a geração do cronograma sem paradas e, em seguida, o mesmo as insere manualmente entre as ordens de produção.

Na primeira fase de testes, o objetivo foi comparar a performance das formulações apresentadas nas seções 3.1 e 3.2. Nesta etapa do trabalho, ainda não era necessário considerar as *release dates*. A Tabela 3 mostra os resultados obtidos.

Tabela 3: Comparação entre FA1 e FA2 sem *release dates*

Instância	***Jobs***	**FA1 Tempo(s)**	**FA1 *gap*(%)**	**FA2 Tempo(s)**	**FA2 *gap*(%)**
A_28j	28	3600	113,53	41	0
B_50j	50	3600	128,78	3600	2,1
C_36j	36	3600	72,77	3600	4,5
D_16j	16	3600	7,75	5	0
E_16j	16	3600	21,39	60	0
F_33j	33	3600	120,91	3600	1,31
G_45j	45	3600	312,03	3600	1,22
H_26j	26	3600	104,19	3600	1,01
I_42j	42	3600	436,46	3600	9,97
J_48j	48	3600	184,38	3600	3,56

A FA2 apresentou melhores resultados em todas as instâncias, justificando utilizarmos este modelo nesta etapa do trabalho.

Em seguida, incorporou-se as *releases dates*, pois em alguns casos, o sequenciamento proposto estava sofrendo alterações manuais em razão da indisponibilidade de ferramentaria para produzir as ordens. Ao inserir esse parâmetro e executar os modelos FA1 e FA2, novos resultados foram gerados para as mesmas instâncias, conforme a Tabela 4.

Nesta análise, não houve unanimidade com relação aos bons resultados obtidos pela FA2. A FA1 apresentou uma solução significativamente melhor para as instâncias B_50j, H_26j, I_42j e J_48j. Sendo assim, foi realizado um estudo para análise do comportamento das *releases dates* das instâncias testadas. Normalizamos e ordenamos essas datas para fins de comparação, sendo parametrizadas entre 0 e 1, $1 = \sum_{j \in J} p_i + \sum_{i \in J} \max_{(j \in J)} s_{ji}$.

De acordo com a Figura 1, ao examinarmos as instâncias que alcançaram melhor desempenho na FA1, foi observado que elas apresentam as maiores *release dates* máximas parametrizadas. Outro ponto analisado foi a diversidade das *release dates*, ou seja, a quantidade de datas diferentes em uma instância. Na Figura 2, observamos que quanto maior é este número, melhor o desempenho da FA1 em comparação com a FA2.

DOI: 10.59254/sbpo-2024-193489
https://proceedings.science/p/193489?lang=pt-br

Tabela 4: Comparação entre FA1 e FA2 com *release dates*

Instância	**Jobs**	**FA1 Tempo(s)**	**FA1 gap(%)**	**FA2 Tempo(s)**	**FA2 gap(%)**
A_28j	28	3600	110	3600	0,24
B_50j	50	3600	3,50	3600	47,27
C_36j	36	3600	5,57	3600	4,91
D_16j	16	3600	8,82	5	0
E_16j	16	3600	21,39	60	0
F_33j	33	3600	45,49	3600	2,76
G_45j	45	3600	268,89	3600	2,41
H_26j	26	70	0	3600	53,2
I_42j	42	9	0	3600	65,70
J_48j	48	3600	6,75	3600	29,24

Figura 1: *Release dates* máximas parametrizadas de todas as instâncias

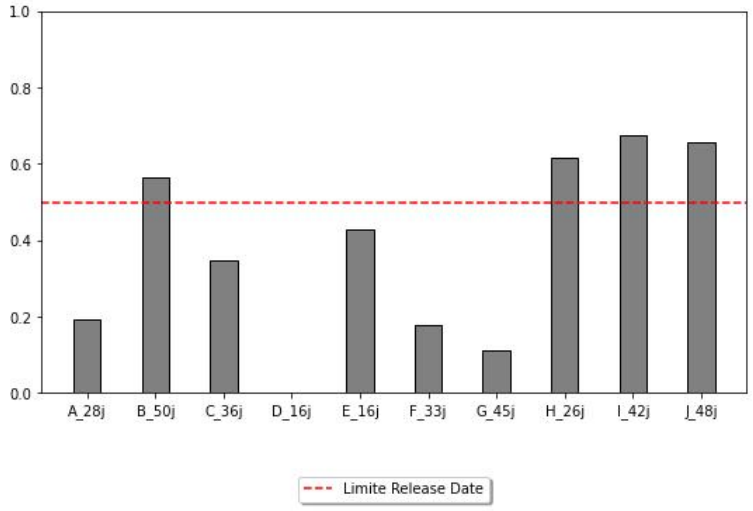
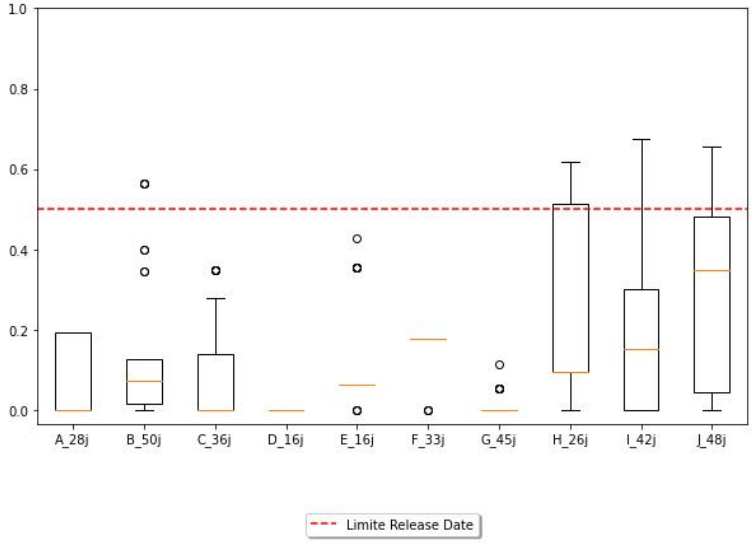


Figura 2: Distribuição de *release dates*



DOI: 10.59254/sbpo-2024-193489
https://proceedings.science/p/193489?lang=pt-br

Um *framework* que leva em consideração a instância para indicar qual modelo deve ser utilizado foi desenvolvido. Definimos então o limite de 0.5 para critério de seleção. Se a instância exceder o limite, a formulação utilizada será a FA1, caso contrário, seleciona-se a FA2. Tal limite é ilustrado através do tracejado em vermelho nas Figuras 1 e 2. O valor do limite máximo foi determinado empiricamente.

Visando mensurar o ganho obtido pelo *framework*, comparamos o *makespan* de cada uma das sequências montadas pelo programador da têmpera e os obtidos pela ferramenta de otimização, como apresentado na Tabela 5.

Tabela 5: Comparação entre ferramenta desenvolvida e programador da têmpera

Instância	Makespan Programador	Makespan Otimizador	Ganho Capacidade
A_28j	03/04 12:14	03/04 07:12	5:02h
B_50j	12/04 13:03	12/04 05:06	7:57h
C_36j	12/04 09:43	12/04 07:55	1:48h
D_16j	05/04 04:27	05/04 02:46	1:41h
E_16j	15/04 23:59	15/04 19:27	3:28h
F_33j	22/04 18:53	22/04 11:54	6:59h
G_45j	28/04 07:19	27/04 23:07	8:12h
H_26j	05/05 12:05	05/05 05:09	6:56h
I_42j	08/04 7:33	08/04 00:45	6:48h
J_48j	21/04 02:22	20/04 19:19	7:03h

A aplicação do *framework* demonstrou-se eficaz a partir dos ganhos de capacidade. Os resultados obtidos foram satisfatórios, especialmente ao integrar os dois modelos. Em quatro instâncias (B_50j, H_26j, I_42j, J_48j), os *gaps* de otimalidade da FA2 foram 47,27%, 53,20%, 65,70% e 29,24% respectivamente, e após a aplicação do *framework*, a FA1 foi selecionada para executar estas instâncias e observou-se uma redução significativa nos *gaps*, alcançando o ótimo em duas delas. A aplicação da ferramenta resultou em melhorias significativas que não teriam sido alcançadas com uma abordagem única. As análises evidenciam a complementariedade das formulações nos experimentos conduzidos.

4.2. Avaliação das inserções de paradas para manutenção

Na segunda fase do trabalho, o desempenho do AIP foi analisado. Os testes foram realizados em três instâncias, pois eram as únicas que havia planejamento de paradas para manutenção. O tempo limite de execução continuou fixado em uma hora. A Tabela 6 compara o *makespan* obtido pela FA1 inserindo a solução gerada como entrada para o AIP e o *makespan* do modelo FA1 considerando os *jobs* artificiais, adicionando a restrição (9). A Tabela 7 apresenta a mesma comparação, mas considerando a restrição (22) no modelo FA2. Observa-se que para a instância com menor número de *jobs* (H_26j) o algoritmo apresentou, em média, uma piora de aproximadamente 4 horas no *makespan* em comparação com as formulações. Em contrapartida, a utilização do algoritmo se justificou para o caso das instâncias (B_50j e C_36j), dado que houve, em média, melhorias no *makespan* de aproximadamente 47 horas para a instância B_50j e quase 9 dias no caso da instância C_36j para a formulação FA2. Nota-se que para instâncias maiores, o AIP encontra melhores resultados em razão da diminuição do número de *jobs*, pois os artificiais não são considerados no modelo, reduzindo o número de variáveis e restrições. Para estes casos, dado que o tempo de execução é pré-determinado, justifica-se a utilização do algoritmo. Além disso, outro fator que contribui para a complexidade dos modelos com paradas planejadas são as restrições (9) e (22), que

DOI: 10.59254/sbpo-2024-193489
https://proceedings.science/p/193489?lang=pt-br

envolvem datas de término. Entretanto, ressalta-se que caso as formulações com paradas encontrem a solução ótima, não há possibilidade do AIP obter um *makespan* menor.

Tabela 6: Comparação entre FA1 considerando *jobs* fictícios e FA1 com AIP

Instância	FA1 + AIP Makespan	FA1 + (9) Makespan
B_50j	14/04 07:37	16/04 02:15
C_36j	13/04 22:17	13/04 22:17
H_26j	06/05 20:48	06/05 16:22

Tabela 7: Comparação entre FA2 considerando *jobs* fictícios e FA2 com AIP

Instância	FA2 + AIP Makespan	FA2 + (23) Makespan
B_50j	16/04 01:59	18/04 12:17
C_36j	13/04 22:17	22/04 15:25
H_26j	06/05 11:52	06/05 08:43

5. Conclusão

Este trabalho abordou estratégias de resolução para o problema de sequenciamento de máquina única dependente de tempo de *setup* (TMSP), dado um horizonte de curto prazo na etapa de têmpera de uma indústria siderúrgica. O objetivo principal consistiu em minimizar o *makespan* para promover uma eficiência maior do processo que atualmente é identificado como um dos gargalos produtivos. A partir de técnicas de Pesquisa Operacional, duas formulações matemáticas propostas na literatura foram adaptadas para absorver as condições de contorno específicas da operação.

Após a realização de testes utilizando cronogramas reais, concluiu-se que a performance das formulações dependia da configuração das *release dates* dos *jobs*. Neste cenário, foi desenvolvida uma ferramenta de suporte à decisão que, com base na característica da instância, define-se qual formulação será utilizada. Além disso, em alguns cronogramas existem as janelas de tempo para manutenção e, por esse motivo, um algoritmo complementar de inserção de paradas foi proposto, apresentando um bom desempenho para cronogramas maiores.

A ferramenta desenvolvida foi testada e validada pelos tomadores de decisão da indústria em questão e apresentou uma alta relevância prática na eficiência do sequenciamento. Além disso, o tempo para se obter uma solução de qualidade foi bem menor em comparação com os métodos anteriores.

Como trabalhos futuros, objetiva-se otimizar um cronograma de horizonte de médio prazo (um a dois meses de produção). Para isso, o número de *jobs* aumenta significativamente e, consequentemente, o tempo de execução do modelo pode se tornar insatisfatório. Pretende-se utilizar alguma abordagem heurística para que a partir de uma solução inicial seja possível obter um *upper bound* melhor para as nossas formulações.

DOI: 10.59254/sbpo-2024-193489
https://proceedings.science/p/193489?lang=pt-br

Referências

Bigras, L.-P., Gamache, M., e Savard, G. (2008). The time-dependent traveling salesman problem and single machine scheduling problems with sequence dependent setup times. *Discrete Optimization*, 5(4):685–699.

Keha, A. B., Khowala, K., e Fowler, J. W. (2009). Mixed integer programming formulations for single machine scheduling problems. *Computers & Industrial Engineering*, 56(1):357–367.

Nogueira, T. H., Carvalho, C. R. V. d., Ravetti, M. G., e Souza, M. C. d. (2019). Analysis of mixed integer programming formulations for single machine scheduling problems with sequence dependent setup times and release dates. *Pesquisa Operacional*, 39:109–154.

Öncan, T., Altınel, I. K., e Laporte, G. (2009). A comparative analysis of several asymmetric traveling salesman problem formulations. *Computers & Operations Research*, 36(3):637–654.

Venâncio, C., V. e Machado (2022). Cartilha de convergência ot-it. Instituto Brasileiro de Petróleo e Gás. Comissão de Tecnologia e Inovação – Rio de Janeiro: IBP. Acessado: 25/03/2024.

DOI: 10.59254/sbpo-2024-193489
https://proceedings.science/p/193489?lang=pt-br