LII Simpósio Brasileiro de Pesquisa Operacional
João Pessoa - PB, 3 a 5 de novembro de 2020

Modelos e algoritmos para o problema de sequenciamento de pedidos de clientes para minimização do atraso total

Massimo Pinto Antonioli
Mestrado Acadêmico em Modelagem e Métodos Quantitativos
Campus do Pici - Centro de Ciências - Bloco 910 - Fortaleza-CE, 60440-900
antonioli.massimo16@yahoo.com

Carlos Diego Rodrigues
Mestrado Acadêmico em Modelagem e Métodos Quantitativos
Campus do Pici - Centro de Ciências - Bloco 910 - Fortaleza-CE, 60440-900
diego@lia.ufc.br

Bruno de Athayde Prata
Universidade Federal do Ceará
Campus do Pici - Centro de Ciências - Bloco 910 - Fortaleza-CE, 60440-900
baprata@ufc.br

RESUMO

Neste trabalho, abordamos o problema de sequenciamento de pedidos, no qual os pedidos dos clientes (compostos por diferentes tarefas individuais) são agendados, em que a função objetivo a ser minimizada é o atraso total dos pedidos. Na literatura revisada, o problema de sequenciamento de pedidos de clientes, encontramos menos referência que em comparação com outros ambientes de produção, como, por exemplo, máquinas paralelas ou *flow shop*. Neste trabalho, apresentamos uma nova variante para o problema, na qual são considerados explicitamente os tempos de preparação das máquinas dependentes da sequência de produção, sendo pertencente à classe NP-difícil. Propomos uma nova formulação de programação linear inteira, uma heurística OMDD-S e uma *matheuristic* denominada Modelo com Permutação Fixa (*Model with fixed permutaion* - MFP). A extensiva experiência computacional realizada mostra que a *matheuristic* MFP apresentou os melhores resultados, pois obteve o menor *RDI* médio (5,69%), dentre todos os métodos, para as instâncias avaliadas.
PALAVRAS CHAVE. Sequenciamento de Produção. Heurística. Tempo de *setup*.
Tópicos: Otimização Combinatória, Programação Linear Inteira.

ABSTRACT

In this work, we address the order scheduling problem, in which customer orders (composed of different individual tasks) are scheduled, where the objective function is the minimization of the total tardiness of completed orders. In the reviewed literature, less studies regarding the problem of sequencing of customer orders were found compared with other production environments, such as, parallel machines or flow shop. In this work, we present a new variant for that problem, where we explicitly consider the setup times of machines dependent on the production sequence, which belongs to the NP-hard class. Then, we propose a new formulation using linear integer programming, an adaptation of the OMDD heuristic (referred to as OMMD-S), and a heuristic called Model with Fixed Permutaion (MFP). The extensive computational simulations carried out show that the MFP matheuristic yields the best results, since it achieved the lowest average RDI (5,69%) considering the analyzed methods and the evaluated instances.
KEYWORDS. Production Sequencing. Heuristic. Setup Times.
Paper topics: Combinatorial Optimization, Integer Linear Programming.

Galoá { Este trabalho foi publicado utilizando o **Galoá** proceedings

1. Introdução

O problema tratado neste trabalho é, na literatura, um clássico sequenciamento (*scheduling*), os trabalhos (*jobs*) a serem processados são tratados como entidades individuais, pertencentes a diferentes clientes, e, portanto, os objetivos buscados estão relacionados aos tempos de conclusão (*completion times*) dos trabalhos individuais ou às diferenças entre os tempos de conclusão e suas datas de vencimento (*due dates*) ou prazos. Entretanto, em muitas situações da vida real, um pedido de um cliente é composto por diferentes produtos que precisam ser processados na loja/fábrica, e pode ser sensato propor objetivos relacionados à conclusão do pedido como um todo, e não aos trabalhos individuais dentro do pedido. Isso é causado pelo fato de muitos clientes exigirem o recebimento do pedido completo, logo não podendo ser enviado antes que todos os processos no pedido sejam concluídos. Portanto, do ponto de vista de Ahmadi et al. [2005], apenas o tempo de conclusão do pedido completo é relevante.

Neste trabalho é considerado um caso de sequenciamento de pedidos no qual temos m máquinas dedicadas em paralelo. Cada máquina só pode processar um (e apenas um) tipo de produto específico, ou seja, são máquinas dedicadas. O problema abordado consiste em agendar n pedidos de clientes compostos por produtos, tal que existe um tempo de *setup*, ou seja, um tempo de preparação da máquina entre os pedidos, dependente da sequência na qual são processados.

A função objetivo considerada é a minimização do atraso total (*total tardiness*), ou seja, a soma das diferenças entre os tempos de conclusão de um pedido e sua data de vencimento, se essa diferença for positiva. Portando, é uma nova configuração para o problema sequenciamento de pedidos, tal problema é classificado como NP-difícil.

Lee [2013] demonstrou que, para o problema sem tempo de preparação entre as tarefas, existe uma solução ótima onde uma mesma permutação de tarefas pode ser aplicada para todas as máquinas. Duas direções são abordadas neste trabalho. Primeiro, é proposta uma nova formulação de programação linear inteira mista (*Mixed integer linear programming* - MILP) baseada na independência das máquinas, ou seja, são permitidas diferentes permutações em cada uma das máquinas. A formulação MILP modela corretamente o problema com tempos de *setups* e pode-se obter valores mais próximos do ótimo, dependendo do tempo configurado para rodar cada instância, conforme a seção 4. Segundo, é proposto um novo Modelo baseado na formulação MILP apresentada por Framinan e Perez-Gonzalez [2018] e a heurística OMDD proposta por Lee [2013], nas duas abordagens são acrescentadas as restrições de tempo de *setup*. Para testar a eficiência desses métodos, é realizado uma ampla experiência computacional que mostra que o MFP dá soluções de maior qualidade, seguido pela heurística OMDD-S (variante da heurística OMDD).

Com base na revisão bibliográfica realizada, foi possível constatar que o problema de sequenciamento de pedidos com o objetivo de minimização do atraso total com restrições de tempo de *setup* não foi estudado anteriormente na literatura. Em Lee [2013], Wagneur e Sriskandarajah [1993] e Framinan e Perez-Gonzalez [2018] é abordado o problema de sequenciamento de pedidos com o objetivo de minimização do atraso total, mas todos sem a consideração dos tempos de *setup*. Observou-se que em Framinan et al. [2019] que nenhum dos artigos estudam o problema de sequenciamento de pedidos com a consideração explícita dos tempos de *setups*.

Em Wagneur e Sriskandarajah [1993] é abordado o status da complexidade em programar n trabalhos em um *openshop* com m máquinas, quando a sobreposição de trabalhos é permitida. Lee [2013] explora o problema de sequenciamento de pedidos para minimizar o atraso total. É apresentado quatro algoritmos heurísticos e um algoritmo *branch-and-bound* (B&B). Em Framinan e Perez-Gonzalez [2018] é estudado o problema de sequenciamento de pedidos com o objetivo do atraso total, este é um problema considerado NP-difícil e algumas heurísticas construtivas foram

Galoá { Este trabalho foi publicado utilizando o **Galoá** proceedings

propostas. É apresentado uma nova heurística construtiva e uma nova estratégia *matheuristic*.

Em Leung et al. [2005a] é estudado uma notação para a classe de modelos de sequenciamento de pedidos e é focado em diferentes condições para o ambiente da máquina e vários tipos de funções objetivos. Roemer [2006] aborda o problema de *concurrent open shop* que é um relaxamento do problema de *open job shop*. Tal problema chamou a atenção de Leung et al. [2005] que contrariaram à afirmação de Wagneur e Sriskandarajah [1993].

Em Leung et al. [2005b] é abordado principalmente em minimizar tempo de conclusão. As heurísticas que foram introduzidas são aplicáveis ao tempo de conclusão. Uma das heurísticas do artigo pode ser referida como a primeira heurística *Weighted Earliest Completion Time*. Leung et al. [2006] aborda principalmente em dois objetivos, o *maximum lateness* e o número total de *late orders*. É considerado neste artigo apenas os problemas mais básicos de sequenciamento. Em Framinan e Perez-Gonzalez [2017] é estudado o problema de sequenciamento de pedidos do cliente, no qual precisam ser agendados em um conjunto de máquinas paralelas, cada uma só podendo processar um único tipo de produto. O objetivo é minimizar a soma do tempo de conclusão dos pedidos e também a minimização do *work-in-process*.

A estrutura deste trabalho é a seguinte. Na seção 2, é apresentado formalmente o problema de sequenciamento de pedidos, propondo uma nova formulação MILP com a adesão de um conjunto de restrições de tempos de *setups* e propriedades do problema. Na seção 3, é proposto uma adaptação de uma heurísticas já descrita na literatura e um Modelo com Permutação Fixa. Na seção 4, são avaliados os experimentos da formulação MILP e os métodos em um conjunto de instâncias geradas aleatoriamente. Na seção 5, são discutidos os resultados. Na seção 6, são apresentadas as conclusões do trabalho e as sugestões para trabalhos futuros.

2. Definição do Problema

Nesta seção, é apresentada a descrição da nova variante em proposição, uma formulação de programação linear inteira, uma propriedade do problema, assim como a complexidade do mesmo.

2.1. Descrição do Problema

O problema considerado pode ser formalmente descrito da seguinte forma. Existe um ambiente de produção com m máquinas dedicadas em paralelo e n pedidos de clientes, cada pedido k $(k = 1, \ldots, n)$ com seu tempo de *setup* e com sua data de vencimento (*due date*). Cada pedido é composto por trabalho (*jobs*) que precisam ser processados em cada uma das m máquinas sem preferência. A quantidade total de processamento exigida pelo pedido k na máquina i $(i = 1, \ldots, m)$ é indicado por p_{ik}, ou seja, o pedido k contém um número de unidades do tipo do produto fabricado na máquina i, portanto exigindo um total de p_{ik} unidades de tempo na máquina i. É denotado o tempo de *setup* e a data de vencimento da seguinte forma, S_{ikl} e d_k, respectivamente, onde l $(l = 1, \ldots, n)$ também é um pedido. O S_{ikl} é o tempo de preparação da máquina i para executar o pedido l imediatamente depois de ter executado o pedido k e o d_k é o prazo de entrega dos pedidos k. Isso equivale a afirmar que o número de unidades de um tipo de produto solicitado é diferente para cada cliente, pode-se afirmar que o tempo de *setup* é diferente para cada cliente em cada máquina, o que reflete a situação usual da vida real.

O objetivo do problema é sequenciar os pedidos para que o atraso total (definido como a soma de todos os pedidos da diferença entre os tempos de conclusão dos pedidos e suas datas de vencimento sempre que essa diferença for positiva) seja minimizado.

Segundo Lee [2013], no caso do problema sem os tempos de preparação, a solução ótima do problema pode ser codificada através de uma única permutação ($\Pi := (\pi_1, \pi_2, \ldots, \pi_n)$) na ordem das tarefas, que será aplicada a cada máquina no processamento das tarefas. Entretanto, conforme

resultado na Seção 2.1.2, é constatado que a aplicação de diferentes permutações em diferentes máquinas pode levar a um resultado melhor, no caso do problema com tempos de preparação.

O tempo de conclusão na máquina i de um tipo de produto na posição j da sequência Π, $C_{i,\pi_j}(\Pi)$, pode ser calculado usando a seguinte equação recursiva:

$$C_{i,\pi_j}(\Pi) = C_{i,\pi_{j-1}}(\Pi) + p_{i,\pi_j} + S_{i,\pi_{j-1},\pi_j}, \quad i = 1,\ldots,m, \quad j = 1,\ldots,n, \tag{1}$$

quando $C_{i,\pi_0}(\Pi) := 0, \forall i$. Assim, $C_{\pi_j}(\Pi)$ o tempo de conclusão da ordem programada na j-ésima posição é então

$$C_{\pi_j}(\Pi) = \max_{1\leq i\leq m}\{C_{i,\pi_j}(\Pi)\}. \tag{2}$$

Analogamente, o atraso da ordem programada na j-ésima posição

$$T_{\pi_j}(\Pi) = \max\{C_{\pi_j}(\Pi) - d_{\pi_j}; 0\}. \tag{3}$$

O atraso total é calculado como $T(\Pi) = \sum_{j=1}^{n} T_{\pi_j}(\Pi)$. Da mesma forma, $C(\Pi)$ a soma dos tempos de conclusão das ordens programadas de acordo com o Π pode ser calculado como $C(\Pi) = \sum_{j=1}^{n} C_{\pi_j}(\Pi)$. Segundo Roemer [2006], o problema de sequenciamento de pedidos com atraso total como objetivo é NP-difícil para $m \geq 2$, além disso, o problema considera o tempo de *setup* entre as máquinas. Consequentemente, procedimentos aproximados são a melhor opção para garantir soluções de boa qualidade (porém sem garantia de ótimo) em problemas de tamanho realista. Tais conceitos definidos nessa seção foram apresentados em Framinan e Perez-Gonzalez [2018], mas com alterações para o acréscimo dos tempos de *setup*.

2.1.1. Formulação MILP

A formulação MILP proposta nesta seção é baseada na formulação de Framinan e Perez-Gonzalez [2018], porém na nova formulação as permutações nas máquinas não são fixas, além disso, é adicionado as restrições de tempos de *setup* que são dependentes da sequencia. Definimos o problema da seguinte forma:

Conjuntos:

- Conjunto de máquinas $I = \{1,\ldots,m\}$;
- Conjunto de pedidos $K = \{1,\ldots,n\}$;
- Conjunto de posições $J = \{1,\ldots,n\}$.

Parâmetros:

- p_{ik} : tempo de processamento do pedido k na máquina i;
- S_{ikl} : tempo de *setup* do pedido k antes do pedido l na máquina i;
- d_k : prazo das datas de entrega dos pedidos k;
- M : é valor tão grande quanto queira, denominado *Big-M*.

Variáveis de decisão:

- T_k : atraso do pedido k;
- C_{ij} : o tempo de conclusão da posição j na máquina i;
- x_{ijk} : variável booleana que informa se na posição j o pedido k é processado na máquina i.

A formulação MILP resultante é a seguinte:

$$\min \sum_{k=1}^{n} T_k \tag{4}$$

$$\text{sujeito a:} \quad \sum_{k=1}^{n} x_{ijk} = 1, \quad \forall i \in I, \quad \forall j \in J \tag{5}$$

$$\sum_{j=1}^{n} x_{ijk} = 1, \quad \forall i \in I, \quad \forall k \in K \tag{6}$$

$$S_{ikl}(x_{ij-1k} + x_{ijl} - 1) \leq D_{ij}, \quad \forall i \in I, \quad \forall j > 1, \quad \forall k, l \in K \tag{7}$$

$$\sum_{r=1}^{j} D_{ir} + \sum_{k=1}^{n} \sum_{r=1}^{j} p_{ik} \cdot x_{irk} \leq C_{ij}, \quad \forall i \in I, \quad \forall j \in J \tag{8}$$

$$C_{ij} - d_k + x_{ijk} \cdot M \leq M + T_k \quad \forall i \in I, \quad \forall j \in J, \quad \forall k \in K \tag{9}$$

$$x_{ijk} \in \{0, 1\}, \quad T_k \geq 0, \quad C_{ij} \geq 0, \quad D_{ij} \geq 0, \quad \forall i \in I, \quad \forall k \in K, \quad \forall j \in J \tag{10}$$

Equação (4) expressa a função objetivo a ser minimizada. O conjunto de restrições do tipo (5) impõe que um (e apenas um) pedido é sequenciada na *j-ésima* posição da máquina i, enquanto o conjunto de restrições do tipo (6) impõe que cada pedido *k* seja sequenciada em uma (e apenas uma) posição da máquina i. O conjunto de restrições do tipo (7) impõe o tempo de *setup* na máquina *i* do pedido *k* antes do pedido *l*, o conjunto de restrições do tipo (8) impõe que o tempo de conclusão de um pedido é dado pelo tempo máximo de processamento dos componentes do pedido nas máquinas com a soma dos tempos de *setups*. A definição de atraso é assegurada pelo conjunto de restrições do tipo (9), enquanto os conjuntos de restrições do tipo (10) expressam o domínio das variáveis do modelo. A formulação MILP resultante possui $m \cdot n \cdot (n^2 + n + 3)$ restrições e $n \cdot (m \cdot n + 2m + 1)$ variáveis.

O *Big-M* nesta formulação é aplicado de tal forma que possa obter-se a melhor solução possível, visto que ao colocar um valor muito grande as soluções pioraram. Logo, é definido o *Big-M* como a soma dos tempos de processamentos dos pedidos mais a soma dos tempos de *setups*. Definido abaixo:

$$\textit{Big-M} = \sum_{i=1}^{m} \sum_{k=1}^{n} p_{ik} + \sum_{i=1}^{m} \sum_{k=1}^{n} \sum_{l=1}^{n} S_{ikl} \tag{11}$$

2.1.2. Propriedade sobre a permutação dos pedidos nas máquinas

Propriedade 2.1. *Na variante $DPm \rightarrow 0|S_{ikl}| \sum T_j$ a solução ótima global não necessariamente apresenta a mesma permutação de pedido em todas as máquinas.*

Demonstração. Considere o seguinte exemplo:

Tempo de processamento

	O_1	O_2
M_1	2	1
M_2	3	4

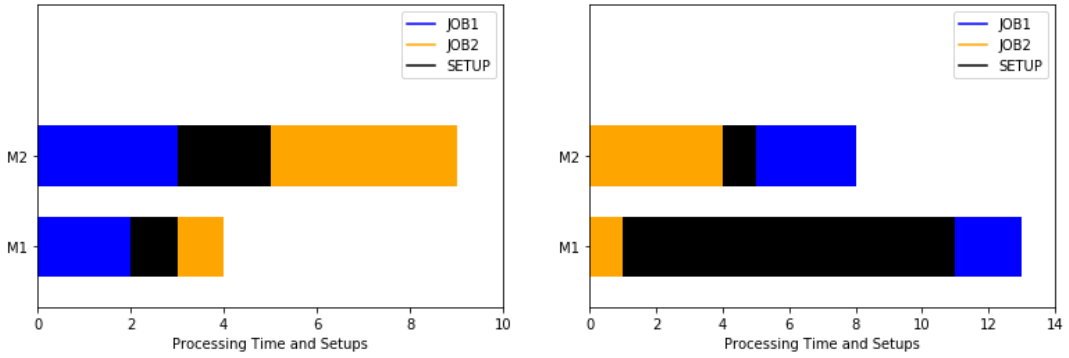
Setup da máquina 1

	O_1	O_2
O_1	-	1
O_2	10	-

Setup da máquina 2

	O_1	O_2
O_1	-	2
O_2	1	-

Ao se utilizar a mesma permutação em todas as máquinas obtém-se os seguintes resultados apresentados na Figura 1. Em Lee [2013] sempre é considerado a mesma permutação em todas as máquinas quando o problema é sem os tempos de *setups*, porém ao inserimos os tempos de *setups*, pode-se observar que o tempo de término de processamento na Figura 1(a) é inferior ao tempo apresentado na Figura 1(b).



(a) $\Pi = \{1, 2\}$ (b) $\Pi = \{2, 1\}$

Figura 1: Mesma permutação nas máquinas.

Ao utilizarmos diferentes permutações nas máquinas, obtém-se os resultados apresentados na Figura 2. Pode-se observar que tempo de término de processamento é inferior a todos os outros obtidos.

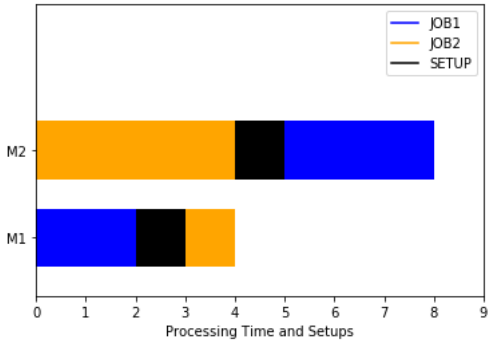


Figura 2: Diferentes permutações nas máquinas.

Portanto, ao aplicarmos diferentes permutações aos pedidos nas máquinas, pode-se obter melhores resultados do que se aplicar a mesma permutação aos pedidos nas máquinas. □

2.1.3. NP-complexidade do problema

Em Wagneur e Sriskandarajah [1993] o problema de sequenciamento de pedidos apresentado é da forma $DPm \rightarrow 0||\sum T_j$ que é da classe NP-difícil, este problema considera que o tempo

de *setup* é nulo. Entretanto, o problema que é citado neste trabalho considera o tempo de *setup*, logo é da forma $DPm \rightarrow 0|S_{ikl}|\sum T_j$. Portanto, como o problema da forma $DPm \rightarrow 0||\sum T_j$ é da classe NP-difícil, então se for considerado o tempo de *setup* o problema da forma $DPm \rightarrow 0|S_{ikl}|\sum T_j$ também vai pertencer a classe NP-difícil.

3. Algoritmos Propostos

Nesta seção, é proposta a *matheuristic* MFP, bem como a heurística OMDD-S (*Order-Scheculing Modified Due-Date with Setups)*, a qual é uma variação da heurística OMDD (*Order-Scheculing Modified Due-Date*) proposta por Lee [2013].

3.1. Modelo com Permutação Fixa (MFP)

O modelo proposto nesta seção considera permutação fixa com restrições de tempo de *setup*. Logo, o modelo apresentado é novo na literatura de sequenciamento de pedidos. Definimos o problema da seguinte forma:

Conjuntos:

- Conjunto de máquinas $I = \{1, \ldots, m\}$;
- Conjunto de pedidos $K = \{1, \ldots, n\}$;
- Conjunto de posições $J = \{1, \ldots, n\}$.

Parâmetros:

- p_{ik} : tempo de processamento do pedido k na máquina i;
- S_{ikl} : tempo de *setup* do pedido k antes do pedido l na máquina i;
- d_k : prazo das datas de entrega dos pedidos k.

Variáveis de decisão:

- T_j : o atraso do pedido sequenciado na posição j;
- C_j : tempo de conclusão na posição j;
- x_{kj} : variável booleana que informa se o pedido k é processado na posição j.

$$\min \sum_{j=1}^{n} T_j \tag{12}$$

$$\text{sujeito a:} \quad \sum_{k=1}^{n} x_{kj} = 1, \quad \forall j \in J \tag{13}$$

$$\sum_{j=1}^{n} x_{kj} = 1, \quad \forall k \in K \tag{14}$$

$$S_{ikl}(x_{kj-1} + x_{lj} - 1) \leq D_{ij}, \quad \forall i \in I, \quad \forall j > 1, \quad \forall k, l \in K \tag{15}$$

$$\sum_{r=1}^{j} D_{ir} + \sum_{k=1}^{n}\sum_{r=1}^{j} p_{ik} \cdot x_{kr} \leq C_j, \quad \forall i \in I, \quad \forall j \in J \tag{16}$$

$$C_j - \sum_{k=1}^{n} d_k \cdot x_{kj} \leq T_j, \quad \forall j \in J \tag{17}$$

$$x_{kj} \in \{0,1\}, \quad T_j \geq 0, \quad C_j \geq 0, \quad D_{ij} \geq 0, \quad \forall i \in I, \quad \forall k \in K, \quad \forall j \in J \tag{18}$$

Equação (12) expressa a função objetivo a ser minimizada. O conjunto de restrições do tipo (13) impõe que um (e apenas um) pedido é sequenciada na *j-ésima* posição, enquanto o conjunto de restrições do tipo (14) impõe que cada pedido *k* seja sequenciada em uma (e apenas uma) posição. O conjunto de restrições do tipo (15) impõe o tempo de *setup* na máquina *i* do pedido *k* antes do pedido *l*, o conjunto de restrições do tipo (16) impõe que o tempo de conclusão de um pedido é dado pelo tempo máximo de processamento dos componentes do pedido nas máquinas com a soma dos tempos de *setups*. A definição de atraso é assegurada pelo conjunto de restrições do tipo (17), enquanto o conjunto de restrições do tipo (18) expressam o domínio das variáveis do modelo. O modelo MFP resultante possui $n \cdot (m \cdot n^2 + m + 3)$ restrições e $n \cdot (n + m + 2)$ variáveis.

3.2. Heurística OMDD-S

A heurística OMDD-S é uma adaptação da heurística OMDD proposta por Lee [2013]. Portanto, a heurística OMDD-S considera o tempo de *setup* que é dependente da sequência de processamentos dos pedidos nas máquinas. Essa heurística tem os seguintes passos:

- Passo 0: Seja $\Omega := (1, 2, \ldots, n)$ (pedidos não agendados), $\Pi := \emptyset$ (pedidos agendados) e $t_i = 0$ $(1 \leq i \leq m)$.
- Passo 1: Selecione o pedido r em Ω com o valor mínimo do índice α_j:

$$\alpha_j = \max \left\{ \max_{1 \leq i \leq m} \{t_i + S_{irj} + p_{ij}\} - \max_{1 \leq i \leq m} \{t_i\}; d_j - \max_{1 \leq i \leq m} \{t_i\} \right\}. \tag{19}$$

- Passo 2: Remova o pedido r de Ω, adicione-o no final do Π.
- Passo 3: Atualiza t_i. Se $|\Omega| = n - 1$, $t_i := t_i + p_{ir}$, senão, $t_i := t_i + S_{iwr} + p_{ir}$, onde w é o penúltimo elemento de Π.
- Passo 4: Se Ω não estiver vazio, retorne ao Passo 1.

4. Experimentos Computacionais

Nesta seção, são apresentados os experimentos para avaliar o desempenho do MILP e das heurísticas propostas nas Seções 2 e 3, respectivamente. Na Seção 4.1, são apresentadas as instâncias de teste empregadas para conduzir os experimentos. Os algoritmos em comparação e os critérios empregados para avaliar seu desempenho são apresentados na Seção 4.2. Finalmente, na Seção 5, são descritos os resultados dos experimentos computacionais.

4.1. Conjunto de Instâncias

O conjunto de instâncias tem por objetivo avaliar a eficiência das heurísticas propostas na Seção 3 e avaliar a eficiência do modelo MILP proposto na Seção 2, em comparação com as heurísticas EDD, OMDD e FP. Com relação aos dois objetivos, a fim de realizar uma comparação justa, desenvolvemos um banco de testes, seguindo um padrão similar ao de Lee [2013], mas com uma diferença, Lee [2013] não gera instâncias para tempos de *setups*, pois o problema que estuda não tem essas restrições, porém nesta abordagem são consideradas tais restrições, logo, são geradas instâncias relacionadas. Para tal, utilizamos $n \in \{10, 20, 30, 40\}$ e $m \in \{2, 5, 8\}$. Os tempos de processamento na máquina i do pedido k são gerados seguindo uma distribuição uniforme $U[1, 100]$. Os tempos de *setups* na máquina i do pedido k antes do pedido l são gerados seguindo um conjunto de distribuição uniforme $\{U[1, 25], U[26, 75], U[76, 125]\}$. A data de vencimento d_k para cada pedido é gerada aleatoriamente a partir de uma distribuição uniforme $U[P(1 - TF - RDD/2), P(1 - TF + RDD/2)]$, onde:

- P é a soma dos tempos de processamentos mais tempos de *setups* dividido pelo número de máquinas tudo multiplicado por um fator μ, isto é,

$$P = \left\lceil \frac{\sum_{i=1}^{m}\sum_{k=1}^{n} p_{ik} + \sum_{i=1}^{m}\sum_{k=1}^{n}\sum_{l=1}^{n} S_{ikl}}{m} \right\rceil \cdot \mu,$$

  e, portanto, pode ser calculada para cada instância do problema;

- *RDD* é o intervalo das datas de vencimento: $RDD : \{0.2, 0.5, 0.8\}$;

- *TF* é o fator de atraso das datas de vencimento: $TF : \{0.2, 0.5, 0.8\}$

Para cada combinação de parâmetros, 10 instâncias foram geradas, resultando num total de 3240 instâncias.

4.2. Algoritmos em Comparação e indicador para avaliação dos resultados

Em relação às heurísticas e o modelo MILP, testadas no banco de teste, os seguintes algoritmos foram comparados:

- Heurística EDD (*Earliest Due Date*);
- Heurística OMDD (*Order-Scheculing Modified Due-Date*) por Lee [2013];
- Heurística FP por Framinan e Perez-Gonzalez [2018];
- Heurística OMDD-S (*Order-Scheculing Modified Due-Date with Setups*);
- Modelo com Permutação Fixa proposto na seção 3, indicado como MFP nos resultados;
- Modelo MILP proposto na Seção 2.1.1, indicado como MILP nos resultados.

Os resultados obtidos pelas heurísticas e pelo MILP de cada instância foram registrados usando o seguinte indicador:

- *RDI* (*Relative Deviation Index*). Quando são testados um conjunto de heurísticas H, o *RDI* obtido pela heurística $s \in H$ quando aplicado a instância t é definido da seguinte forma:

$$RDI_{st} = \begin{cases} 0, & \text{se } \min_{h\in H} T_{ht} = \max_{h\in H} T_{ht}, \\ \frac{T_{st} - \min_{h\in H} T_{ht}}{\max_{h\in H} T_{ht} - \min_{h\in H} T_{ht}} \cdot 100, & \text{caso contrário.} \end{cases} \quad (20)$$

  sendo T_{st} o valor de atraso obtido pela heurística s na instância t. No nosso caso, $\min_{h\in H} T_{ht}$ é o valor ótimo se o modelo MILP for capaz de encontrar a solução ótima dentro do tempo de computacional designado ou a melhor solução encontrada entre as heurísticas e o MILP. Observe que o *RDI* é um indicador padrão para medir a qualidade de soluções aproximadas em problemas de sequenciamento no qual as datas de vencimento estão envolvidas (ver, por exemplo, [Kim, 1993], [Fernandez-Viagas e Framinan, 2015] ou [Karabulut, 2016]).

As heurísticas e o modelo MILP foram codificados na linguagem *Python* 3.7 usando o resolvedor CPLEX 12.8.0, configurado para rodar cada instância por um tempo de 600 segundos. Os algoritmos foram executados em Intel Core i5-3470 com 3.20 *gigahertz* e 8 *gigabyte* de memória RAM.

5. Análise e discussão dos resultados

Na Tabela 1 são apresentados os resultados para os diferentes valores de m (máquinas) e n (pedidos), incluindo a média e o desvio padrão dos valores de *RDI*. Observando o primeiro conjunto de instâncias ($m = 2$ e $n = 10$) temos que o MILP é o melhor entre todos os métodos estudados, pois obteve o menor *RDI* médio (2,56%) e o menor *RDI* de desvio padrão (5,01%), porém quando o conjunto de instâncias aumenta os dois melhores métodos são o MFP e a heurística OMDD-S, tendo os menores *RDI* médios avaliados e os menores *RDI* de desvio padrão.

Tabela 1: Heurísticas e Modelos para m e n.

m	n	EDD		FP		OMDD		OMDD-S		MFP		MILP	
		Média	Desvio Pd.	Média	Desvio Pd.	Média	Desvio Pd.	Média	Desvio Pd.	Média	Desvio Pd.	Média	Desvio Pd.
2	10	96.0493	9.3616	74.0338	25.0662	71.7094	26.1252	38.9890	21.3437	4.0962	6.0115	**2.5621**	5.0161
	20	78.0802	39.7039	59.6758	36.7352	60.4745	37.3009	34.3478	28.6449	**1.3400**	3.7905	9.3785	12.9342
	30	51.2699	47.9732	38.7799	38.8774	37.3278	38.5403	14.9357	21.4638	**3.2868**	11.3513	29.6013	35.5334
	40	30.6066	41.6651	20.6749	29.2184	20.6244	29.4594	**5.3353**	12.1637	7.2014	16.0259	47.7702	46.3508
5	10	93.7856	12.8498	73.2224	23.6788	67.0564	26.0498	37.3649	21.3112	**0.7320**	3.5071	44.3834	31.3414
	20	68.5931	38.5886	51.8753	33.9332	49.3869	34.2624	26.2016	27.6356	**1.5034**	5.5553	56.6971	40.8459
	30	24.7276	29.9120	15.6652	19.8134	14.6226	19.5513	**4.8897**	13.0445	7.9156	18.7946	63.7574	47.0489
	40	12.5207	18.9695	7.7062	11.2894	7.2035	12.0241	**0.7864**	3.8819	13.7649	23.4343	63.6526	48.1386
8	10	91.4871	15.5632	68.0999	24.8717	64.4250	26.7497	34.7448	20.1781	**0.3887**	2.6978	59.7256	29.8778
	20	47.8521	34.6717	35.4197	30.6066	34.0509	30.4694	16.3356	24.0658	**2.1620**	6.2871	72.9362	42.2317
	30	13.7397	16.9201	8.3019	9.1562	7.1592	8.7618	**1.1804**	3.0884	8.6284	15.6743	70.3357	45.7281
	40	7.0810	10.0842	4.0630	5.4360	3.7153	5.1559	**0.0747**	0.4838	16.1054	24.7717	65.4765	47.5565
Total		51.3161	43.6518	38.1265	36.6097	36.4797	36.2311	17.9322	24.0582	**5.5937**	14.6616	48.8564	44.5224

Na Tabela 2 são apresentados os mesmos resultados agregados em relação ao *TF* e *RDD*, incluindo a média e o desvio padrão dos valores de *RDI*. Observando a tabela os melhores métodos são o MFP e a heurística OMDD-S, tais métodos obtiveram os melhores resultados com relação ao *RDI* em média e desvio padrão.

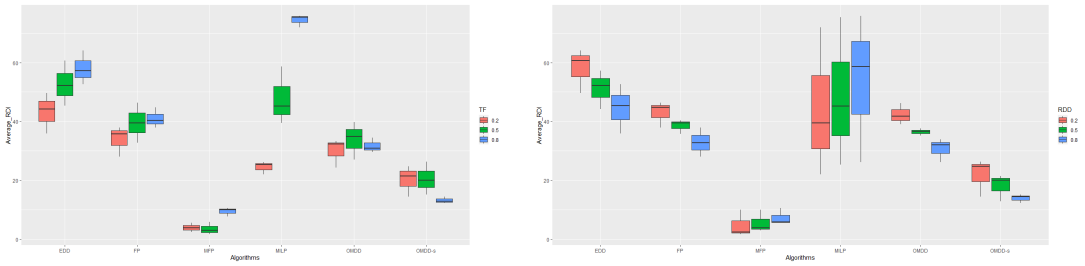
Tabela 2: Heurísticas e Modelos para TF e RDD.

TF	RDD	EDD		FP		OMDD		OMDD-S		MFP		MILP	
		Média	Desvio Pd.	Média	Desvio Pd.	Média	Desvio Pd.	Média	Desvio Pd.	Média	Desvio Pd.	Média	Desvio Pd.
0.2	0.2	49.6773	46.3111	37.8822	39.5605	38.9762	39.8445	24.7103	28.6539	**2.4181**	7.2048	21.9678	35.8238
	0.5	44.1905	45.6202	35.7829	39.8318	35.1393	39.3478	21.4946	29.2083	**3.8455**	12.3581	25.2510	38.6798
	0.8	35.8707	44.9105	27.9736	37.6936	26.1867	36.6468	14.3961	23.6771	**5.5174**	17.8131	26.1323	40.3943
0.5	0.2	60.6846	42.2314	46.3787	38.2754	46.2399	37.7658	26.2242	26.4104	**1.6616**	7.6374	39.4062	41.5167
	0.5	52.1371	44.0711	39.5297	37.8891	37.6649	37.8340	19.9293	26.1946	**2.9192**	11.2144	45.1481	44.4619
	0.8	45.3009	44.6112	32.6849	35.6797	31.9980	36.0993	15.0814	22.7089	**5.8147**	16.5870	58.6621	44.3116
0.8	0.2	64.1548	36.8234	44.6951	30.9881	41.7105	30.7024	14.4790	17.2662	**7.7060**	15.4199	71.9201	36.8675
	0.5	57.2449	38.8950	40.3117	31.0855	36.5306	30.6018	12.8570	16.5625	**9.9410**	16.8242	75.3578	35.5601
	0.8	52.5838	41.7645	37.8996	34.0979	33.8709	32.5434	12.2177	17.1035	**10.5200**	18.8095	75.8623	34.8575
Total		51.3161	43.6518	38.1265	36.6097	36.4797	36.2311	17.9322	24.0582	**5.5937**	14.6616	48.8564	44.5224

Na Figura 3(a) são apresentados os resultados em *Boxplot* agregados em relação ao *TF* da tabela 2. A partir desta figura é constado que o melhor método é o MFP seguido da heurística OMDD-S, sendo o MFP o melhor método em todos os parâmetros de *TF*. O método MILP obteve resultados bons, porém quando o *TF* é 0.8 se torna o segundo pior método estudado. A heurística FP é a pior entre todas as estudadas. Na Figura 3(b) são apresentados os resultados em *Boxplot* agregados em relação ao *RDD* da Tabela 2. A partir desta figura é constado que o método MFP é o melhor entre todos seguido da heurística OMDD-S.

6. Conclusão

Este trabalho aborda o problema de sequenciamento de pedidos, no qual os pedidos dos clientes (compostos por diferentes tarefas individuais) são agendados, que a função objetivo a ser minimizada é o atraso total dos pedidos. Apresentamos uma nova variante para o problema, na qual são considerados explicitamente os tempos de preparação das máquinas, os quais são dependentes



(a) Boxplot em relação a média *RDI* para *TF*.

(b) Boxplot em relação a média *RDI* para *RDD*.

Figura 3: Boxplot em relação ao RDI médio para os métodos em análise.

da sequência de produção. Propomos uma nova formulação MILP, uma heurística OMDD-S e uma heurística MFP.

Experimentos computacionais foram realizados em um conjunto de instâncias geradas aleatoriamente, objetivando avaliar o desempenhos dos métodos em estudo. Como indicador de comparação entre os métodos, foi utilizado o *RDI*. Os tempos computacionais não foram usados para fins de comparação neste trabalho, porém foi avaliado e para continuação do trabalho iremos comparar também os tempos computacionais, assim como a otimalidade do problema.

Com base nos experimentos computacionais realizados, é possível constatar que a heurística MFP consegue os menores valores de *RDI* médio, sendo o melhor método para o conjunto de instâncias avaliado.

Como sugestões para trabalhos futuros (ora em desenvolvimento pelos autores), podem ser destacadas a geração de novos conjuntos de dados (instâncias) para o problema em estudo, bem como a proposição de novos métodos de resolução. Além disso, para mais trabalhos futuros, pode-se acrescentar as restrições de tempos de preparação para outras funções objetivos na literatura de sequenciamento de pedidos.

7. Agradecimentos

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001, e do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

Referências

Ahmadi, R., Bagchi, U., e Roemer, T. A. (2005). Coordinated scheduling of customer orders for quick response. *Naval Research Logistics (NRL)*, 52(6):493–512. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/nav.20092.

Fernandez-Viagas, V. e Framinan, J. M. (2015). Neh-based heuristics for the permutation flowshop scheduling problem to minimise total tardiness. *Computers and Operations Research*, 60:27 – 36. ISSN 0305-0548. URL http://www.sciencedirect.com/science/article/pii/S0305054815000258.

Framinan, J. M. e Perez-Gonzalez, P. (2017). New approximate algorithms for the customer order scheduling problem with total completion time objective. *Computers and Operations Research*, 78:181 – 192. ISSN 0305-0548. URL http://www.sciencedirect.com/science/article/pii/S0305054816302295.

Framinan, J. M. e Perez-Gonzalez, P. (2018). Order scheduling with tardiness objective: Improved approximate solutions. *European Journal of Operational Research*, 266(3):840 – 850.

ISSN 0377-2217. URL http://www.sciencedirect.com/science/article/pii/S0377221717309931.

Framinan, J. M., Perez-Gonzalez, P., e Fernandez-Viagas, V. (2019). Deterministic assembly scheduling problems: A review and classification of concurrent-type scheduling models and solution procedures. *European Journal of Operational Research*, 273(2):401 – 417. ISSN 0377-2217. URL http://www.sciencedirect.com/science/article/pii/S0377221718303369.

Karabulut, K. (2016). A hybrid iterated greedy algorithm for total tardiness minimization in permutation flowshops. *Computers and Industrial Engineering*, 98:300 – 307. ISSN 0360-8352. URL http://www.sciencedirect.com/science/article/pii/S0360835216302078.

Kim, Y.-D. (1993). Heuristics for flowshop scheduling problems minimizing mean tardiness. *Journal of the Operational Research Society*, 44(1):19–28. URL https://www.scopus.com/inward/record.uri?eid=2-s2.0-0027240799&doi=10.1057\%2fjors.1993.3&partnerID=40&md5=a74c40f8c6baaaac6cfbb5b14b3070e7. cited By 88.

Lee, I. S. (2013). Minimizing total tardiness for the order scheduling problem. *International Journal of Production Economics*, 144(1):128 – 134. ISSN 0925-5273. URL http://www.sciencedirect.com/science/article/pii/S0925527313000431.

Leung, J. Y.-T., Li, H., e Pinedo, M. (2005a). Order scheduling models: An overview. In Kendall, G., Burke, E. K., Petrovic, S., e Gendreau, M., editors, *Multidisciplinary Scheduling: Theory and Applications*, p. 37–53, Boston, MA. Springer US. ISBN 978-0-387-27744-8. URL https://link.springer.com/chapter/10.1007/0-387-27744-7_3.

Leung, J. Y. T., Li, H., e Pinedo, M. (2005b). Order scheduling in an environment with dedicated resources in parallel. *Journal of Scheduling*, 8(5):355–386. URL https://doi.org/10.1007/s10951-005-2860-x.

Leung, J. Y.-T., Li, H., e Pinedo, M. (2006). Scheduling orders for multiple product types with due date related objectives. *European Journal of Operational Research*, 168(2):370 – 389. ISSN 0377-2217. URL http://www.sciencedirect.com/science/article/pii/S0377221704003005. Feature Cluster on Mathematical Finance and Risk Management.

Leung, J. Y.-T., Li, H., Pinedo, M., e Sriskandarajah, C. (2005). Open shops with jobs overlap—revisited. *European Journal of Operational Research*, 163(2):569 – 571. ISSN 0377-2217. URL http://www.sciencedirect.com/science/article/pii/S0377221703009214.

Roemer, T. A. (2006). A note on the complexity of the concurrent open shop problem. *Journal of Scheduling*, 9(4):389–396. ISSN 1099-1425. URL https://doi.org/10.1007/s10951-006-7042-y.

Wagneur, E. e Sriskandarajah, C. (1993). Openshops with jobs overlap. *European Journal of Operational Research*, 71(3):366 – 378. ISSN 0377-2217. URL http://www.sciencedirect.com/science/article/pii/037722179390347P.