



**UNIVERSIDADE FEDERAL DO CEARÁ**  
**CENTRO DE CIÊNCIAS**  
**DEPARTAMENTO DE ESTATÍSTICA E MATEMÁTICA APLICADA**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM E MÉTODOS**  
**QUANTITATIVOS**

**MASSIMO PINTO ANTONIOLI**

**MODELOS E ALGORITMOS PARA O PROBLEMA DE SEQUENCIAMENTO DE**  
**PEDIDOS DE CLIENTES PARA MINIMIZAÇÃO DO ATRASO TOTAL**

**FORTALEZA**

**2021**

MASSIMO PINTO ANTONIOLI

MODELOS E ALGORITMOS PARA O PROBLEMA DE SEQUENCIAMENTO DE  
PEDIDOS DE CLIENTES PARA MINIMIZAÇÃO DO ATRASO TOTAL

Dissertação apresentada ao Curso de do Programa de Pós-Graduação em Modelagem e Métodos Quantitativos do Centro de Ciências da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Modelagem e Métodos Quantitativos. Área de Concentração: Modelagem e Métodos Quantitativos.

Orientador: Prof. Dr. Bruno de Athayde Prata

Coorientador: Prof. Dr. Carlos Diego Rodrigues

FORTALEZA

2021

Dados Internacionais de Catalogação na Publicação  
Universidade Federal do Ceará  
Biblioteca Universitária

Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

---

A64m Antonioli, Massimo Pinto.

Modelos e algoritmos para o problema de sequenciamento de pedidos de clientes para minimização do atraso total / Massimo Pinto Antonioli. – 2021.

48 f. : il.

Dissertação (mestrado) – Universidade Federal do Ceará, Centro de Ciências, Programa de Pós-Graduação em Modelagem e Métodos Quantitativos, Fortaleza, 2021.

Orientação: Prof. Dr. Bruno de Athayde Prata.

Coorientação: Prof. Dr. Carlos Diego Rodrigues.

1. Sequenciamento de Produção. 2. Heurística. 3. Otimização Combinatória. 4. Programação Linear Inteira. I. Título.

CDD 510

---

MASSIMO PINTO ANTONIOLI

MODELOS E ALGORITMOS PARA O PROBLEMA DE SEQUENCIAMENTO DE  
PEDIDOS DE CLIENTES PARA MINIMIZAÇÃO DO ATRASO TOTAL

Dissertação apresentada ao Curso de do  
Programa de Pós-Graduação em Modelagem e  
Métodos Quantitativos do Centro de Ciências da  
Universidade Federal do Ceará, como requisito  
parcial à obtenção do título de mestre em  
Modelagem e Métodos Quantitativos. Área  
de Concentração: Modelagem e Métodos  
Quantitativos.

Aprovada em: 09/08/2021

BANCA EXAMINADORA

---

Prof. Dr. Bruno de Athayde Prata (Orientador)  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Carlos Diego Rodrigues (Coorientador)  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. David Álvarez Martínez  
Universidade dos Los Andes (UNIANDES)

---

Prof. Dr. Jesus Ossian da Cunha  
Universidade Federal do Ceará (UFC)

À minha família, por sempre acreditar e investir em mim. Mãe e Pai, seu cuidado e dedicação foi que deram, em alguns momentos, a esperança para seguir. Irmão, por sempre estar me incentivando e me apoiando em todos momentos da minha graduação e mestrado.

## AGRADECIMENTOS

Ao Prof. Dr. Bruno de Athayde de Prata e ao Prof. Dr. Carlos Diego Rodrigues por terem me orientado neste trabalho de dissertação.

Ao Doutorando em Engenharia Elétrica, Ednardo Moreira Rodrigues, e seu assistente, Alan Batista de Oliveira, aluno de graduação em Engenharia Elétrica, pela adequação do *template* utilizado neste trabalho para que o mesmo ficasse de acordo com as normas da biblioteca da Universidade Federal do Ceará (UFC).

Ao meu irmão Dr. Roberto Pinto Antonioli, por sempre ter me apoiado e me aconselhado nos momentos que precisei.

Aos amigos do Mestrado, Alexandre Sucupira, Nilton Filho e Anny Suellen, pelas discussões sobre os algoritmos e pelos conselhos dados.

Aos meus pais, que nos momentos de minha ausência dedicados ao estudo superior, sempre fizeram entender que o futuro é feito a partir da constante dedicação no presente!

Agradeço a todos os professores por me proporcionar o conhecimento não apenas racional, mas a manifestação do caráter e afetividade da educação no processo de formação profissional, por tanto que se dedicaram a mim, não somente por terem me ensinado, mas por terem me feito aprender.

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).

“Ogni vincitore ha cicatrici.”

(CASSON)

## RESUMO

Neste trabalho é abordado o problema de sequenciamento de pedidos (*order scheduling*), no qual os pedidos dos clientes (compostos por diferentes tarefas individuais) são agendados, de forma que a função objetivo a ser minimizada é o atraso total (*total tardiness*) dos pedidos. Na literatura revisada, abordagens que tratem o problema de sequenciamento de pedidos com tempos de preparação das máquinas (*setups*) é bastante limitada. Apresentamos uma nova variante para o problema, na qual são considerados explicitamente os tempos de *setups* dependentes da sequência de produção. Como a variante em estudo é NP-difícil, são propostas uma nova formulação de programação linear inteira, uma adaptação da heurística *Order-Scheduling Modified Due-Date* (OMDD) (denotado adiante como *Order-Scheduling Modified Due-Date Setup* (OMMD-S)), uma adaptação da heurística Framinan e Perez-Gonzalez (FP) (denotada adiante como Framinan e Perez-Gonzalez *Setup* (FP-S)), um modelo com Mesma Permutação em Todas as Máquinas (*Same Permutation in All Machines* - SPAM) e uma *matheuristic* híbrida JPO-SPAM fundamentada no *Job-Position Oscillation* (JPO). A extensiva experiência computacional realizada mostra que para as instâncias de pequeno porte avaliadas o SPAM é o mais eficiente, apresentando a menor média e desvio padrão dos valores do *Relative Deviation Index* - *RDI*, enquanto que para as instâncias de grande porte avaliadas mostra que a *matheuristic* híbrida JPO-SPAM e o *Mixed-integer linear programming* (MILP) são as mais eficientes, com a *matheuristic* híbrida JPO-SPAM apresentando a menor média e desvio padrão dos valores do *RDI*.

**Palavras-chave:** sequenciamento de produção; heurística; otimização combinatória; programação linear inteira.

## ABSTRACT

In this work, the order sequencing problem is addressed, 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 revised literature, approaches that address the order sequencing problem with machine setup times are quite limited. We present a new variant for the problem, in which the setup times depending on the production sequence are explicitly considered. As the variant under study is NP-difficult, a new formulation using linear integer programming, an adaptation of the Order-Scheduling Modified Due-Date heuristic (OMDD) (referred to as Order-Scheduling Modified Due-Date Setup (OMMD-S)), an adaptation of the Framinan and Perez-Gonzalez heuristic (FP) (hereinafter referred to as Framinan and Perez-Gonzalez Setup (FP-S)), a model with Same Permutation in All Machines (SPAM), and a SPAM-JPO matheuristic hybrid based on Job-Position Oscillation (JPO) are proposed. The extensive computational experience carried out shows that for the small-sized evaluated instances the SPAM is the most efficient, presenting the lowest average and standard deviation of the values of Relative Deviation Index (*RDI*), while for the large-sized evaluated instances it shows that the SPAM-JPO matheuristic hybrid and Mixed-integer linear programming (MILP) are the most efficient, with SPAM-JPO matheuristic hybrid showing the lowest average and standard deviation of the *RDI* values.

**Keywords:** production sequencing; heuristic; combinatorial optimization; mixed-integer linear programming.

## LISTA DE FIGURAS

Figura 1 – Solução ótima para a instância ilustrativa apresentada . . . . .	19
Figura 2 – Mesma permutação nas máquinas . . . . .	23
Figura 3 – Diferentes permutações nas máquinas . . . . .	24
Figura 4 – Instâncias de Pequeno Porte: Boxplot em relação a média <i>RDI</i> para <i>TF</i> . . .	39
Figura 5 – Instâncias de Pequeno Porte: Boxplot em relação a média <i>RDI</i> para <i>RDD</i> . .	40
Figura 6 – Instâncias de Grande Porte: Boxplot em relação a média <i>RDI</i> para <i>TF</i> . . .	43
Figura 7 – Instâncias de Grande Porte: Boxplot em relação a média <i>RDI</i> para <i>RDD</i> . .	44

## LISTA DE TABELAS

Tabela 1 – Referências Bibliográficas . . . . .	17
Tabela 2 – Tempo de Processamento . . . . .	18
Tabela 3 – <i>Setup</i> da máquina 1 . . . . .	19
Tabela 4 – <i>Setup</i> da máquina 2 . . . . .	19
Tabela 5 – <i>Due Date</i> . . . . .	19
Tabela 6 – Solução ótima detalhada da instância ilustrativa apresentada . . . . .	20
Tabela 7 – Resultados da Figura 2(a) e Figura 2(b) . . . . .	24
Tabela 8 – Resultados da Figura 3 . . . . .	24
Tabela 9 – Instâncias de Pequeno Porte: Heurísticas e Modelos para $m$ e $n$ . . . . .	36
Tabela 10 – Instâncias de Pequeno Porte: Heurísticas e Modelos para $m$ e $n$ . . . . .	36
Tabela 11 – Instâncias de Pequeno Porte: % de otimalidade para $m$ e $n$ . . . . .	37
Tabela 12 – Instâncias de Pequeno Porte: Heurísticas e Modelos para $TF$ e $RDD$ . . . . .	37
Tabela 13 – Instâncias de Pequeno Porte: Heurísticas e Modelos para $TF$ e $RDD$ . . . . .	38
Tabela 14 – Instâncias de Pequeno Porte: % de otimalidade para $TF$ e $RDD$ . . . . .	38
Tabela 15 – Instâncias de Pequeno Porte: $SR$ ( <i>sucess rate</i> ) . . . . .	39
Tabela 16 – Instâncias de Grande Porte: Heurísticas e Modelos para $m$ e $n$ . . . . .	40
Tabela 17 – Instâncias de Grande Porte: Heurísticas e Modelos para $m$ e $n$ . . . . .	41
Tabela 18 – Instâncias de Grande Porte: % de otimalidade para $m$ e $n$ . . . . .	41
Tabela 19 – Instâncias de Grande Porte: Heurísticas e Modelos para $TF$ e $RDD$ . . . . .	42
Tabela 20 – Instâncias de Grande Porte: Heurísticas e Modelos para $TF$ e $RDD$ . . . . .	42
Tabela 21 – Instâncias de Grande Porte: % de otimalidade para $TF$ e $RDD$ . . . . .	42
Tabela 22 – Instâncias de Grande Porte: $SR$ ( <i>sucess rate</i> ) . . . . .	44

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b> . . . . .	12
<b>2</b>	<b>REVISÃO BIBLIOGRÁFICA</b> . . . . .	14
<b>3</b>	<b>DEFINIÇÃO DO PROBLEMA</b> . . . . .	18
<b>3.1</b>	<b>Descrição do Problema</b> . . . . .	18
<b>3.1.1</b>	<i>Formulação MILP</i> . . . . .	21
<b>3.1.2</b>	<i>Propriedades do Problema</i> . . . . .	23
<b>3.1.2.1</b>	<i>Propriedade sobre a permutação dos pedidos nas máquinas</i> . . . . .	23
<b>3.1.2.2</b>	<i>Complexidade do problema</i> . . . . .	24
<b>4</b>	<b>ALGORITMOS PROPOSTOS</b> . . . . .	26
<b>4.1</b>	<b>Heurística OMDD-S</b> . . . . .	26
<b>4.2</b>	<b>Heurística FP-S</b> . . . . .	26
<b>4.3</b>	<b>Modelo com Mesma Permutação em Todas as Máquinas (SPAM)</b> . . . . .	28
<b>4.4</b>	<b>Matheuristic Híbrida</b> . . . . .	30
<b>5</b>	<b>EXPERIMENTOS COMPUTACIONAIS</b> . . . . .	33
<b>5.1</b>	<b>Conjunto de Instâncias</b> . . . . .	33
<b>5.2</b>	<b>Algoritmos em Comparação e indicadores para avaliação dos resultados</b> . . . . .	34
<b>5.3</b>	<b>Resultados para o conjunto de instâncias de pequeno porte</b> . . . . .	36
<b>5.4</b>	<b>Resultados para o conjunto de instâncias de grande porte</b> . . . . .	39
<b>6</b>	<b>CONCLUSÃO</b> . . . . .	45
	<b>REFERÊNCIAS</b> . . . . .	47

## 1 INTRODUÇÃO

O problema tratado neste trabalho é, na literatura, um problema clássico de sequenciamento (*scheduling*) de pedidos, no qual 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 necessário 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, Bagchi e Roemer (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 preparação (*setup*), ou seja, um tempo de adequação 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 o tempo de conclusão de um pedido e sua data de vencimento, se essa diferença for positiva. Ao considerar explicitamente os tempos de preparação (*setups*) das máquinas, é apresentada uma variante não relatada na literatura revisada. Tal problema é classificado como NP-difícil (ver Wagneur e Sriskandarajah (1993), para a variante sem tempos de setup dependentes da sequência), fato que justifica a proposição de métodos exatos ou aproximados para a obtenção de soluções de alta qualidade em tempo computacional admissível.

Com base na revisão bibliográfica realizada, é possível constatar que a consideração explícita de tempo de preparação dependentes da sequência no problema de sequenciamento de pedidos não foi reportada. Assim, é introduzida uma nova variante, a qual é classificada como NP-difícil (ver a Subseção 3.1.2.2).

O problema estudado neste trabalho possui um grande número de aplicações práticas na vida real, como na indústria farmacêutica, na oficina mecânica, na indústria de papel e em uma

fabricante de periféricos de computador (ex. terminais, teclados e unidades de disco), bem como em manutenções de aviões e reparo de navios. Em Leung, Li e Pinedo (2006) e em Magazine e Julien (1990) explicam mais detalhadamente as aplicações citadas.

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.

Dois direções são abordadas neste trabalho. Primeiro, é proposto uma formulação de programação linear inteira mista (MILP) que se baseia em máquinas independentes, ou seja, são permitidas diferentes permutações em cada uma das máquinas. A formulação MILP modela o problema com tempos de *setups* e pode-se obter valores mais próximos do ótimo, dependendo do tempo configurado para rodar cada instância. Segundo, são propostos dois métodos heurísticos para a resolução do problema, duas extensões de heurísticas construtivas existentes na literatura, bem como um modelo e uma *matheuristic híbrida*. Para testar a eficiência desses métodos, é realizada uma ampla experiência computacional que mostra que o modelo SPAM obtém soluções de maior qualidade para as instâncias de pequeno porte, a *matheuristic* híbrida JPO-SPAM e o MILP apresentam soluções de maior qualidade para as instâncias de grandes porte, no qual são avaliadas usando o indicador RDI.

A estrutura deste trabalho é a seguinte. No Capítulo 2, é apresentada a revisão da literatura do sequenciamento de pedidos da forma  $DPm \rightarrow 0 || C_{max}$ ,  $DPm \rightarrow 0 || \sum C_j$  e  $DPm \rightarrow 0 || \sum T_j$ . No Capítulo 3, 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. No Capítulo 4, é proposto uma adaptação de duas heurísticas já descritas na literatura, assim como um modelo e uma *matheuristic*. No Capítulo 5, são avaliados os experimentos computacionais em um conjunto de instâncias geradas aleatoriamente, em seguida são discutidos os resultados. No Capítulo 6, é apresentada a conclusão do trabalho.

## 2 REVISÃO BIBLIOGRÁFICA

Para facilitar o estudo das diferentes contribuições, abaixo são fornecidas como resumos dos diferentes *layouts*, observando o foco principal das referências. As abreviações usadas na Tabela 1 são:

- $DPm \rightarrow 0$ :  $m$  máquinas dedicadas em paralelo com *assembly* igual a zero;
- $ST_{sd}$ : *Setup Times* dependentes da sequência;
- $C_{max}$ : *Makespan*;
- $\sum C_j$ : Tempo de Conclusão (*Completion Time*);
- $\sum T_j$ : Atraso Total (*Total Tardiness*);
- $L_{max}$ : Atraso máximo (*Maximum Lateness*);
- $\sum U_j$ : Número de pedidos atrasados (*Number of Late Orders*);
- CA: Análise da complexidade do problema;
- PP: Propriedades do problema (exceto regras de dominância / *lower bounds*);
- PA: Proposta de Algoritmos de Tempo Polinomial;
- BB: Desenvolvimento de um procedimento *Branch-and-Bound*;
- CH: Proposta de Heurísticas Construtivas;
- MH: Proposta de Meta-Heurística.
- MILP: Desenvolvimento de um modelo MILP

Foram estudados nesta pesquisa problemas de sequenciamento de pedidos do tipo:

- $DPm \rightarrow 0 || C_{max}$ ;
- $DPm \rightarrow 0 | ST_{sd} | C_{max}$ ;
- $DPm \rightarrow 0 || \sum C_j$ ;
- $DPm \rightarrow 0 || \sum T_j$ ;
- $DPm \rightarrow 0 || L_{max}$ ;
- $DPm \rightarrow 0 || \sum U_j$ .

Os problemas acima citados foram escolhidos a partir da Tabela 1 apresentada em Framinan, Perez-Gonzalez e Fernandez-Viagas (2019), considerando apenas as variantes relacionadas ao ambiente de produção em estudo neste trabalho. Assim por estes problemas pesquisados foi feita a revisão bibliográfica adiante.

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. Foi provado que as variantes com as funções objetivo de máximo tempo de conclusão

(*maximum completion time*) e o máximo atraso (*maximum tardiness*) são problemas polinomiais. São apresentados algoritmos de tempo polinomial, ou algoritmos heurísticos para todos os problemas considerados.

Lee (2013) explora o problema de sequenciamento de pedidos para minimizar o atraso total. O sistema é composto de múltiplas máquinas e cada ordem consiste de múltiplos componentes, com cada componente sendo fabricado em uma máquina dedicada. São apresentados quatro algoritmos heurísticos e um algoritmo *branch-and-bound* (B&B).

Framinan e Perez-Gonzalez (2018) apresentam o problema de sequenciamento de pedidos com o objetivo do atraso total, este é um problema considerado NP-difícil. Algumas heurísticas construtivas foram apresentadas, a mais importante é a heurística FP sendo uma nova heurística construtiva para o problema de sequenciamento de pedidos. A *matheuristic Job-Position Oscillation* (JPO) é nova estratégia que foi descrita neste artigo para obter melhores resultados para cenários onde os intervalos de decisão mais longos são permitidos.

Em Leung, Li e Pinedo (2005a) é estudado uma notação para a classe de modelos de sequenciamento de pedidos. Ademais, são estudadas diferentes condições para o ambiente com várias máquinas não idênticas em paralelo podem produzir uma variedade fixa de produtos e várias funções objetivo. Os autores apresentam um algoritmo de tempo polinomial para alguns problemas como a minimização do Tempo de Conclusão e minimização Número de pedidos atrasados, provas de complexidades para problemas NP-difícil e uma análise de desempenho para os piores casos.

Roemer (2006) aborda o problema de *concurrent open shop* que é um relaxamento do problema de *open shop*. Tal problema chamou a atenção de Leung *et al.* (2005) que contrariaram a afirmação de Wagneur e Sriskandarajah (1993), as descobertas obtidas deixaram em aberto a questão sobre a complexidade do problema de minimizar o tempo médio de conclusão do trabalho (*average job completion time*). Entretanto, sugerem que dada a complexidade das medidas internas, há uma necessidade de heurísticas mais adequadas.

Leung, Li e Pinedo (2005b) mostra que a minimização tempo de conclusão quando  $m \geq 3$  que é fortemente NP-difícil. As duas novas heurísticas apresentadas são aplicáveis quando o objetivo do problema se refere ao tempo de conclusão. O modelo estudado possui aplicações práticas. Uma das heurísticas pode ser referida como a primeira heurística *Weighted Earliest Completion Time*.

Leung, Li e Pinedo (2006) aborda dois objetivos, o atraso máximo (*maximum*

*lateness*) e o número total de pedidos atrasados (*late orders*) que são relacionados à data de vencimento. Apresentam algoritmos de tempo polinomial para os casos fáceis e heurísticas para os casos NP-difíceis, foi também proposto um algoritmo exato baseado em propagação de restrições e estratégia de limitação. Relatam a existência de muitos problemas mais gerais de sequenciamento com objetivos das datas de vencimento.

Em Framinan e Perez-Gonzalez (2017) é estudado o problema de sequenciamento de pedidos do cliente para a minimização dos tempos de conclusão, 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 tempo total de conclusão (*total completion time*). Na literatura este problema é NP-difícil ver Wagneur e Sriskandarajah (1993)).

Wu *et al.* (2021) abordou o problema de sequenciamento de pedidos do cliente com tempos de processamento e datas de vencimento dependentes do cenário. A função objetivo é minimizar o atraso total máximo em todos os cenários. Quatro limites inferiores, bem como quatro relações de dominância, foram propostos. Além disso, alguns algoritmos de pesquisa local e baseados em população de alguns algoritmos de busca local baseados na metaheurística *Iterated Greedy* (IG) foram apresentados. As abordagens IG superaram as heurísticas baseadas na *Order-Scheduling Modified Due-Date* (OMDD), embora aumentassem o tempo necessário para a resolução do problema.

Prata, Rodrigues e Framinan (2021) introduziu o problema de sequenciamento de pedido com tempos de preparação dependentes da sequência. Dois modelos de programação linear inteira mista foram apresentados para esta variante. Além disso, foram desenvolvidas duas *metaheuristics*, que reduzem o número de variáveis de decisão. O algoritmo *Fixed Variable List* (FVLA) proposto supera todas as outras abordagens.

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*. Além destes, nenhum outro problema de sequenciamento de pedidos com  $m$  máquinas dedicadas em paralelo com *assembly* igual a zero apresenta restrições de tempos de *setup*. Apenas em Prata, Rodrigues e Framinan (2021) foi reportado uma variante que

considera os tempos de *setup* dependentes da sequência, contudo, a função objetivo abordada é a minimização do *makespan*.

Tabela 1: Referências Bibliográficas

Problema	Referência	Análise	Exato(a)	Heurística
$DPm \rightarrow 0    C_{max}$	Wagneur e Sriskandarajah (1993)	CA		
	Leung, Li e Pinedo (2006)	CA		CH
$DPm \rightarrow 0   ST_{sd}   C_{max}$	Prata, Rodrigues e Framinan (2021)		MILP	CH, MH
$DPm \rightarrow 0    \sum C_j$	Wagneur e Sriskandarajah (1993)	CA		
	Roemer (2006)	CA		
	Leung, Li e Pinedo (2005b)	PP		
	Framinan e Perez-Gonzalez (2017)			CH, MH
$DPm \rightarrow 0    \sum T_j$	Wagneur e Sriskandarajah (1993)	CA		
	Lee (2013)	PP	BB	CH
	Framinan e Perez-Gonzalez (2018)			CH, MH
	Wu <i>et al.</i> (2021)	PA		CH, MH
$DPm \rightarrow 0    L_{max}$	Wagneur e Sriskandarajah (1993)	CA		
$DPm \rightarrow 0    \sum U_j$	Wagneur e Sriskandarajah (1993)	CA		
	Leung, Li e Pinedo (2006)	PP		

Fonte: elaborada pelo autor

Com relação a Tabela 1, nosso trabalho é da seguinte categoria  $DPm \rightarrow 0 | ST_{sd} | \sum T_j$ .

### 3 DEFINIÇÃO DO PROBLEMA

#### 3.1 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  $k$  ( $k = 1, \dots, n$ ) tem seu tempo de *setup* em uma máquina e sua data de vencimento (*due date*). Cada pedido é composto por trabalhos (*jobs*) que precisam ser processados em cada uma das  $m$  máquinas sem preferência. A quantidade total de processamento exigida pelo pedido  $k$  na máquina  $i$  ( $i = 1, \dots, m$ ) 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  $i$ . Considere o tempo de *setup* e a data de vencimento denotados por,  $S_{ikl}$  e  $d_k$ , respectivamente, onde  $l$  ( $l = 1, \dots, n$ ) 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. Visto que cada pedido apresenta um tempo específico que é dependente da máquina onde é processado, 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 seja minimizado (O atraso total pode ser definido como a soma, para todos os pedidos, das diferenças entre os tempos de conclusão dos pedidos e suas datas de vencimento sempre que essa diferença for positiva).

Nosso problema de sequenciamento de pedidos com o objetivo de minimizar o atraso total, trata-se de máquinas dedicadas em paralelo onde depois de todos os pedidos serem processados com tempos de *setups* dependentes das máquinas, o pedido sofre uma operação de *assembly*, porém com tempo igual a zero. Onde  $O_1, O_2$  e  $O_3$  são os pedidos e  $M_1$  e  $M_2$  são as máquinas. Um exemplo para o problema é mostrado nas Tabelas 2, 3, 4 e 5:

Tabela 2: Tempo de Processamento

	$O_1$	$O_2$	$O_3$
$M_1$	4	2	3
$M_2$	3	1	2

Fonte: elaborada pelo autor

As Tabelas 2, 3, 4 e 5 contém todas as informações necessárias para a resolução

Tabela 3: *Setup* da máquina 1

	$O_1$	$O_2$	$O_3$
$O_1$	-	6	4
$O_2$	2	-	3
$O_3$	3	5	-

Fonte: elaborada pelo autor

Tabela 4: *Setup* da máquina 2

	$O_1$	$O_2$	$O_3$
$O_1$	-	2	3
$O_2$	1	-	2
$O_3$	4	4	-

Fonte: elaborada pelo autor

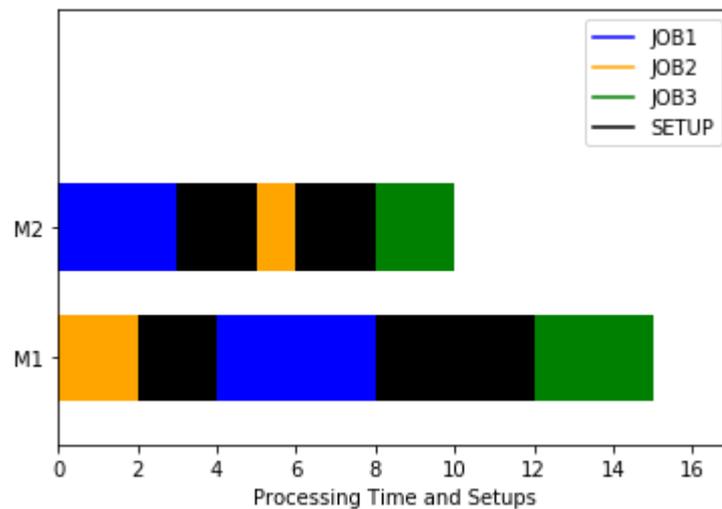
Tabela 5: *Due Date*

$O_1$	4
$O_2$	6
$O_3$	3

Fonte: elaborada pelo autor

do problema, no qual consta o tempo de processamento, tempos de *setup* da máquina 1 e 2, e as datas de vencimentos (*due date*). Na Figura 1 é mostrada a solução ótima para a instância apresentada, a qual apresenta  $\Pi_1 = \{2, 1, 3\}$  que é a permutação apresentada na máquina 1, onde o pedido 2 está na posição 1, pedido 1 está na posição 2 e o pedido 3 está na posição 3. Já na máquina 2, a permutação é  $\Pi_2 = \{1, 2, 3\}$ , assim o pedido 1 está na posição 1, pedido 2 está na posição 2 e pedido 3 está na posição 3. Estas duas permutações implicam em solução com valor de 16 unidades de tempo de atraso total.

Figura 1: Solução ótima para a instância ilustrativa apresentada



Fonte: elaborada pelo autor

Ao analisar a Figura 1 é possível gerar a solução mostrada na Tabela 6, onde é

calculado o tempo de conclusão (*completion times*) e o atraso (*tardiness*). O procedimento usado para calcular o atraso é da seguinte forma, tempo de conclusão do trabalho  $x$  menos a data de vencimento (*due date*) do trabalho  $x$  é igual ao atraso do trabalho  $x$ , com esse procedimento é possível encontrar o valor do atraso do trabalho desejado. Em seguida é usado o somatório do atraso total (consiste em somar o valor de todos os atrasos encontrados) para assim obter o resultado do nosso exemplo. Resolvendo:

Trabalho 1:

- Tempo de conclusão – data de vencimento = Atraso  $\Rightarrow 8 - 4 = 4$

Trabalho 2:

- Tempo de conclusão – data de vencimento = Atraso  $\Rightarrow 6 - 6 = 0$

Trabalho 3:

- Tempo de conclusão – data de vencimento = Atraso  $\Rightarrow 15 - 3 = 12$

Assim para o somatório do atraso total:

- $\sum T = 16$ .

Tabela 6: Solução ótima detalhada da instância ilustrativa apresentada

		Figura 1		
		$J_1$	$J_2$	$J_3$
<i>Completion Times</i>		8	6	15
<i>Due Date</i>		4	6	3
<i>Tardiness</i>		4	0	12
	$\sum T_k$	16		

Fonte: elaborada pelo autor

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 ( $\Pi := (\pi_1, \pi_2, \dots, \pi_n)$ ) na ordem das tarefas, que será aplicada a cada máquina no processamento das tarefas. Entretanto, conforme resultado na Seção 3.1.2, 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  $\Pi$ ,  $C_{i,\pi_j}(\Pi)$ , pode ser calculado usando a seguinte equação recursiva:

$$C_{i,\pi_j}(\Pi) = C_{i,\pi_{j-1}}(\Pi) + p_{i,\pi_j} + S_{i,\pi_{j-1},\pi_j}, \quad i = 1, \dots, m, \quad j = 1, \dots, n, \quad (3.1)$$

quando  $C_{i,\pi_0}(\Pi) := 0, \forall i$ . Assim,  $C_{\pi_j}(\Pi)$  o tempo de conclusão do pedido programado na

$j$ -ésima posição é então

$$C_{\pi_j}(\Pi) = \max_{1 \leq i \leq m} \{C_{i,\pi_j}(\Pi)\}. \quad (3.2)$$

Analogamente, o atraso do pedido programado na  $j$ -ésima posição

$$T_{\pi_j}(\Pi) = \max\{C_{\pi_j}(\Pi) - d_{\pi_j}; 0\}. \quad (3.3)$$

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 dos pedidos programados de acordo com o  $\Pi$  pode ser calculado como  $C(\Pi) = \sum_{j=1}^n C_{\pi_j}(\Pi)$ . O problema de sequenciamento de pedidos com tempo de conclusão como objetivo é NP-difícil para  $m \geq 2$  (ver Roemer (2006)), além disso, o problema considera o tempo de *setup* entre as máquinas. Nestes casos, é frequente que procedimentos heurísticos sejam 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*.

### 3.1.1 Formulação MILP

Nesta seção é apresentada a formulação MILP que foi baseada na mostrada em 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 sequência. Definimos o problema da seguinte forma:

Conjuntos:

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

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  $i$ ;

- $D_{ij}$  : tempo de *setup* da máquina  $i$  na posição  $j$ ;
- $x_{ijk}$  : variável booleana que informa se na posição  $j$  o pedido  $k$  é processado na máquina  $i$ .

A formulação MILP resultante é a seguinte:

$$\min \sum_{k=1}^n T_k \quad (3.4)$$

$$\text{sujeito a: } \sum_{k=1}^n x_{ijk} = 1, \quad \forall i \in I, \quad \forall j \in J \quad (3.5)$$

$$\sum_{j=1}^n x_{ijk} = 1, \quad \forall i \in I, \quad \forall k \in K \quad (3.6)$$

$$S_{ikl}(x_{ij-1k} + x_{ijl} - 1) \leq D_{ij}, \quad \forall i \in I, \quad \forall j > 1 \in J, \quad \forall k, l \in K \quad (3.7)$$

$$\sum_{r=1}^j D_{ir} + \sum_{k=1}^n \sum_{r=1}^j p_{ik} \cdot x_{irk} \leq C_{ij}, \quad \forall i \in I, \quad \forall j \in J \quad (3.8)$$

$$C_{ij} - d_k + x_{ijk} \cdot M \leq M + T_k \quad \forall i \in I, \forall j \in J, \forall k \in K \quad (3.9)$$

$$x_{ijk} \in \{0, 1\}, \quad T_k \geq 0, \quad C_{ij} \geq 0, \quad D_{ij} \geq 0, \quad \forall i \in I, \quad \forall k \in K, \quad \forall j \in J \quad (3.10)$$

A Equação (3.4) expressa a função objetivo a ser minimizada. O conjunto de restrições do tipo (3.5) impõe que um (e apenas um) pedido é sequenciado na  $j$ -ésima posição da máquina  $i$ , enquanto o conjunto de restrições do tipo (3.6) impõe que cada pedido  $k$  seja sequenciado em uma (e apenas uma) posição da máquina  $i$ . O conjunto de restrições do tipo (3.7) define a ordem do processamento e é computado o tempo de *setup* da máquina  $i$  na posição  $j$  em  $D_{ij}$ , o conjunto de restrições do tipo (3.8) 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 (3.9), enquanto os conjuntos de restrições do tipo (3.10) 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  $M$  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  $M$  como a soma dos tempos de processamentos dos pedidos mais a soma dos tempos de *setups*. Definido abaixo:

$$M = \sum_{i=1}^m \sum_{k=1}^n p_{ik} + \sum_{i=1}^m \sum_{k=1}^n \sum_{l=1}^n S_{ikl} \quad (3.11)$$

### 3.1.2 Propriedades do Problema

Nesta subsecção são apresentadas a complexidade do problema e a uma propriedade, tal que propriedade comprova que ao permutarmos os pedidos nas máquinas obtém-se uma solução melhor que a solução obtida com permutações iguais.

#### 3.1.2.1 Propriedade sobre a permutação dos pedidos nas máquinas

**Propriedade 3.1.** Na variante  $DPm \rightarrow 0|ST_{sd}|\sum T_j$  a solução ótima global não necessariamente apresenta a mesma permutação de pedido em todas as máquinas.

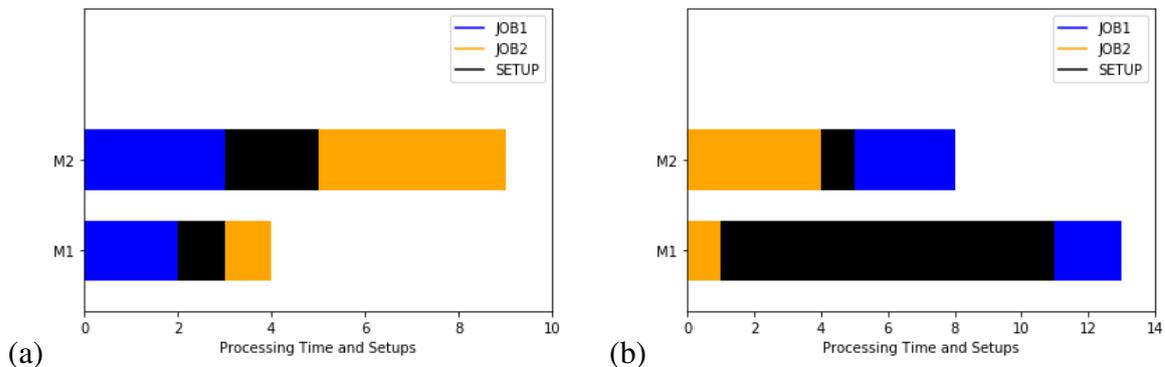
*Demonstração.* Considere o seguinte exemplo:

Tempo de processamento		Setup da máquina 1		Setup da máquina 2		Due Date		
	$O_1$	$O_2$	$O_1$	$O_2$	$O_1$	$O_2$		
$M_1$	2	1	$O_1$	-	$O_1$	-	$O_1$	5
$M_2$	3	4	$O_2$	10	$O_2$	1	$O_2$	5

Fonte: elaborada pelo autor

Ao se utilizar a mesma permutação em todas as máquinas obtém-se os seguintes resultados apresentados na Figura 2. Em Lee (2013) sempre é considerado a mesma permutação em todas as máquinas quando o problema é sem os tempos de *setups* com valor não-negativo, porém ao inserimos os tempos de *setups*, pode-se observar que o tempo de término de processamento na Figura 2(a) é inferior ao tempo apresentado na Figura 2(b), observando a Tabela 7.

Figura 2: Mesma permutação nas máquinas



Fonte: elaborada pelo autor

Ao se utilizar diferentes permutações nas máquinas, obtém-se os resultados apresentados na Figura 3. Pode-se observar que tempo de término de processamento é inferior a todos

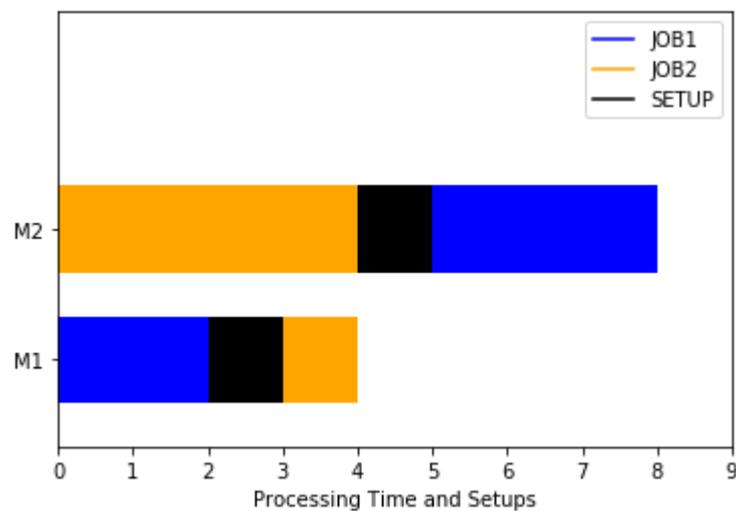
Tabela 7: Resultados da Figura 2(a) e Figura 2(b)

	Figura 2(a)		Figura 2(b)	
	$O_1$	$O_2$	$O_1$	$O_2$
Completion Times	3	9	13	4
Due date	5	5	5	5
Tardiness	0	4	8	0
$\Sigma T$	4		8	

Fonte: elaborada pelo autor

os outros obtidos, como observado na Tabela 8.

Figura 3: Diferentes permutações nas máquinas



Fonte: elaborada pelo autor

Tabela 8: Resultados da Figura 3

	Figura 3	
	$O_1$	$O_2$
Completion Times	8	4
Due Date	5	5
Tardiness	3	0
$\Sigma T$	3	

Fonte: elaborada pelo autor

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.  $\square$

### 3.1.2.2 Complexidade do problema

Em Wagneur e Sriskandarajah (1993) o problema de sequenciamento de pedidos apresentado é da seguinte forma  $DPM \rightarrow 0 || \Sigma T_j$  sendo da classe NP-difícil, este problema foi

considerado que o tempo de *setup* é nulo. Entretanto, o problema que é abordado neste trabalho considera o tempo de *setup*, então o problema é denotado da seguinte forma  $D P m \rightarrow 0 | S T_{sd} | \sum T_j$ . Portanto, como o problema de Wagneur e Sriskandarajah (1993) é da classe NP-difícil, então sendo em nosso caso uma generalização, temos que o problema abordado também irá pertencer a classe NP-difícil.

## 4 ALGORITMOS PROPOSTOS

Neste Capítulo, são apresentados duas heurísticas, um modelo e uma *matheuristic*. As primeiras são a OMDD-S e a FP-S que são uma variação das heurísticas OMDD (ver Lee (2013)) e FP (ver Framinan e Perez-Gonzalez (2018)), respectivamente, mas agora considerando os tempos de *setups*. A segunda é a SPAM, um modelo relaxado cujo índice referente às máquinas na variável de decisão é suprimido. Assim, as permutações serão as mesmas nas máquinas. Por fim, a última é uma *matheuristic* híbrida JPO-SPAM, a qual consiste em usar o SPAM e a heurística OMDD-S (variação da heurística OMDD) para iniciar o algoritmo.

### 4.1 Heurística OMDD-S

Nesta Seção é apresentada a heurística OMDD-S que é uma adaptação da heurística OMDD proposta por Lee (2013), tal que pode ser implementada de uma maneira bastante simples para ser executada no tempo  $O(mn^2)$ . 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  $\Omega := (1, 2, \dots, n)$  (pedidos não agendados),  $\Pi := \emptyset$  (pedidos agendados) e  $t_i = 0$  ( $1 \leq i \leq m$ ).
- Passo 1: Selecione o pedido  $r$  em  $\Omega$  com o valor mínimo do índice  $\alpha_j$ :

$$\alpha_j = \max \left\{ \max_{1 \leq i \leq m} \{t_i + S_{irj} + p_{ij}\} - \max_{1 \leq i \leq m} \{t_i\}; d_j - \max_{1 \leq i \leq m} \{t_i\} \right\}. \quad (4.1)$$

- Passo 2: Remova o pedido  $r$  de  $\Omega$ , adicione-o no final do  $\Pi$ .
- 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  $\Pi$ .
- Passo 4: Se  $\Omega$  não estiver vazio, retorne ao Passo 1.

### 4.2 Heurística FP-S

A heurística proposta constrói uma solução  $\Pi := (\pi_1, \dots, \pi_n)$  a partir de um conjunto  $\mathcal{U}$ , o qual representa o conjunto de ordens não escalonadas. Inicialmente, nenhuma ordem foi escalonada e, portanto,  $\mathcal{U}$  contém todos os pedidos. Então, a heurística começa um processo iterativo de  $n$  passos: no passo  $k$ , cada pedido em  $\mathcal{U}$  é selecionado como candidato para ser acrescentado no final de  $\Pi$ . Para selecionar uma ordem entre as candidatas, para cada pedido

$\omega_l \in \mathcal{U}$  com data de vencimento  $d_l$ , uma sequência (parcial)  $S_l$  é formada adicionando  $\omega_l$  no final de  $\Pi$ , isto é,  $\Pi := (\pi_1, \dots, \pi_{k-1}, \omega_l)$ .

Então, o próximo indicador de atraso  $\eta_l$  para a  $l$ -ésimo candidato (pedido  $w_l$ ) é computada como:

$$\eta_l = \max\{C(S_l) - d_l; 0\} + T^*(S_l) \quad (4.2)$$

em que o primeiro termo representa a contribuição do atraso total caso  $\omega_l$  for selecionada, com  $C(S_l)$  sendo o tempo de conclusão da sequência (parcial)  $S_l$  e  $d_l$  a sua data de vencimento correspondente.  $T^*(S_l)$  é uma estimacão da contribuição do atraso total das sequências não escalonadas restantes que foi mencionado em Framinan e Perez-Gonzalez (2018).

**Algoritmo 1: FP-S**


---

**Entrada :** Número de máquinas  $m$ , número de pedidos  $n$ , tempo de processamento  $p$ ,  
datas de vencimento  $d$ , tempo de *setup*  $s$ .

**Saída :** Permutação  $\Pi$

$\Pi := \emptyset$ ;

Obtenha a sequência  $\Omega := (\omega_1, \dots, \omega_n)$  classificando pedidos de acordo com a regra EDD;

Seja  $ct_i = 0$  ( $1 \leq i \leq m$ );

//Anexe pedidos um por um

**Para**  $j$  em  $n$  **faça**

//Calcule o indicador para cada pedido não programado

**Para** cada  $\omega_l \in \Omega$  **faça**

//Calcule  $ect_i$  dos tempos de conclusão estimados de  $\omega_l$  para cada máquina  $i$ ;

**Para**  $i = 1$  em  $m$  **faça**

$ect_i = ct_i + p_{i\omega_l} + s_{i\omega_l}$

**Fim**

//Onde  $l$  é o último elemento de  $\Pi$

//Calcule o primeiro termo do indicador  $\eta_l$ , consulte a equação 4.2

$\eta_l = \max\{\max_{1 \leq i \leq m}\{ect_i\} - d_{\omega_l}; 0\}$

//Calcule o segundo termo do indicador  $\eta_l$

**Para** cada  $w_k \in \Omega - \{\omega_l\}$  **faça**

**Para**  $i = 1$  em  $m$  **faça**

$ect_i = ect_i + p_{i w_k} + s_{i w_k}$

**Fim**

$\eta_l = \eta_l + \max\{\max_{1 \leq i \leq m}\{ect_i\} - d_{w_k}; 0\}$

**Fim**

**Fim**

//Selecione o pedido com o valor mínimo de  $\eta_l$

$r := \operatorname{argmin}_{1 \leq l \leq n-j+1} \eta_l$

Anexe  $\omega_r$  no final de  $\Pi$ , isto é,  $\Pi := (\pi_1, \dots, \pi_{j-1}, \omega_r)$

Extraia  $\omega_r$  de  $\Omega$ , isto é,  $\Omega := (\omega_1, \dots, \omega_{r-1}, \omega_{r+1}, \dots, \omega_{n-j+1})$

//Atualize os tempos de conclusão

**Para**  $i = 1$  em  $m$  **faça**

$ct_i = ct_i + p_{i\omega_r} + s_{i\omega_r}$

**Fim**

**Fim**

---

**4.3 Modelo com Mesma Permutação em Todas as Máquinas (SPAM)**

Visto que, no problema em estudo, cada máquina pode receber uma sequência distinta, enfrentamos um problema com uma codificação não permutacional. Podemos observar que essa codificação não leva necessariamente a melhores soluções, pois o espaço de busca pode aumentar substancialmente. Fernandez-Viagas, Perez-Gonzalez e Framinan (2019) demonstram que as codificações permutacionais podem fornecer melhores resultados do que as codificações não permutacionais em outros problemas de programação de produção. Isso se deve ao fato de

que um procedimento de solução pode avaliar um número maior de soluções com codificação permutacional, para um determinado limite de tempo.

O modelo proposta aqui teve um índice da variável de decisão removido, assim a permutação será a mesma em todas as máquinas e com restrições de tempo de *setup*. Logo, o modelo apresentado é nova na literatura de sequenciamento de pedidos. Definimos o problema da seguinte forma:

Conjuntos:

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

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  $j$ ;
- $D_{ij}$  : tempo de *setup* da máquina  $i$  na posição  $j$ ;
- $x_{kj}$  : variável booleana que informa se o pedido  $k$  é processado na posição  $j$ .

$$\min \sum_{j=1}^n T_j \quad (4.3)$$

$$\text{sujeito a: } \sum_{k=1}^n x_{kj} = 1, \quad \forall j \in J \quad (4.4)$$

$$\sum_{j=1}^n x_{kj} = 1, \quad \forall k \in K \quad (4.5)$$

$$S_{ikl}(x_{kj-1} + x_{lj} - 1) \leq D_{ij}, \quad \forall i \in I, \quad \forall j > 1 \in J, \quad \forall k, l \in K \quad (4.6)$$

$$\sum_{r=1}^j D_{ir} + \sum_{k=1}^n \sum_{r=1}^j p_{ik} \cdot x_{kr} \leq C_j, \quad \forall i \in I, \quad \forall j \in J \quad (4.7)$$

$$C_j - \sum_{k=1}^n d_k \cdot x_{kj} \leq T_j, \quad \forall j \in J \quad (4.8)$$

$$x_{kj} \in \{0, 1\}, \quad T_j \geq 0, \quad C_j \geq 0, \quad D_{ij} \geq 0, \quad \forall i \in I, \quad \forall k \in K, \quad \forall j \in J \quad (4.9)$$

A Equação (4.3) expressa a função objetivo a ser minimizada. O conjunto de restrições do tipo (4.4) impõe que um (e apenas um) pedido é sequenciado na  $j$ -ésima posição, enquanto o conjunto de restrições do tipo (4.5) impõe que cada pedido  $k$  seja sequenciado em uma (e apenas uma) posição. O conjunto de restrições do tipo (4.6) define a ordem do processamento e é computado o tempo de *setup* da máquina  $i$  na posição  $j$  em  $D_{ij}$ , o conjunto de restrições do tipo (4.7) 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 (4.8), enquanto o conjunto de restrições do tipo (4.9) expressam o domínio das variáveis do modelo. A *matheuristic* SPAM resultante possui  $n \cdot (m \cdot n^2 + m + 3)$  restrições e  $n \cdot (n + m + 2)$  variáveis.

#### 4.4 Matheuristic Híbrida

Nossa estratégia de proposta para a *matheuristic* híbrida é a seguinte, será usado o algoritmo *Job-Position Oscillation* (JPO) junto com o Modelo com Mesma Permutação em Todas as Máquinas, sendo denominado como JPO-SPAM.

Começa com uma determinada solução, porém, no nosso caso, ao buscar uma solução usando o SPAM, restringimos a ordem na posição  $j$  em ser deslocado para frente e para trás de no máximo  $\delta$  posições.

Mais especificamente, dada uma sequência, para todas as ordens  $k$  ( $k = 1, \dots, K$ ) encontramos  $w_k(\Pi)$  a posição do pedido  $k$  em  $\Pi$ . Então, um conjunto de posições permitidas para  $k$  é encontrado de forma que o pedido  $k$  só pode se mover no máximo  $\delta$  posições para trás e  $\delta$  posições para frente ( $\delta$  é, portanto, um parâmetro *matheuristic* que seria chamado de oscilação). O conjunto de posições permitidas para o pedido  $k$  é  $\mathcal{P}_k(\Pi, \delta) = \{\underline{w}_k(\Pi, \delta), \underline{w}_k(\Pi, \delta) + 1, \dots, \bar{w}_k(\Pi, \delta)\}$ , onde  $\underline{w}_k(\Pi, \delta) = \max\{0; w_k(\Pi) - \delta\}$  e  $\bar{w}_k(\Pi, \delta) = \min\{n; w_k(\Pi) + \delta\}$ . Desta forma, para cada pedido  $k$  ( $k = 1, \dots, K$ ), as variáveis posicionais  $x_{kj}$  não são significativas para  $j \in \mathcal{P}_k(\Pi, \delta)$ , e o tamanho resultante do SPAM é reduzido em relação ao número de variáveis. Analogamente, o conjunto de pedidos que podem ocupar a posição  $j$  pode ser definido como  $\mathcal{O}_j(\Pi, \delta) = \{k : j \in \mathcal{P}_k(\Pi, \delta)\}$ .

O modelo SPAM é expressada da seguinte forma:

$$\min \sum_{j=1}^n T_j \tag{4.10}$$

$$\text{sujeito a: } \sum_{k \in \mathcal{O}_j(\Pi, \delta)} x_{kj} = 1, \quad \forall j \in J \quad (4.11)$$

$$\sum_{j \in \mathcal{P}_k(\Pi, \delta)} x_{kj} = 1, \quad \forall k \in K \quad (4.12)$$

$$S_{ikl}(x_{kj-1} + x_{lj} - 1) \leq D_{ij}, \quad \forall i \in I, \quad \forall j > 1 \in J, \quad \forall k, l \in K \quad (4.13)$$

$$\sum_{r=\underline{w}_k(\Pi, \delta)}^{\min\{j: \bar{w}_k(\Pi, \delta)\}} D_{ir} + \sum_{k \in \mathcal{O}_j(\Pi, \delta)} \sum_{r=\underline{w}_k(\Pi, \delta)}^{\min\{j: \bar{w}_k(\Pi, \delta)\}} p_{ik} \cdot x_{kr} \leq C_j, \quad \forall i \in I, \quad \forall j \in J \quad (4.14)$$

$$C_j - \sum_{k \in \mathcal{O}_j(\Pi, \delta)} d_k \cdot x_{kj} \leq T_j, \quad \forall j \in J \quad (4.15)$$

$$x_{kj} \in \{0, 1\}, \quad T_j \geq 0, \quad C_j \geq 0, \quad D_{ij} \geq 0, \quad \forall i \in I, \quad \forall k \in K, \quad \forall j \in J \quad (4.16)$$

$$j \in \mathcal{P}_k(\Pi, j) \quad (4.17)$$

Equação (4.10) expressa a função objetivo a ser minimizada. O conjunto de restrições do tipo (4.11) impõe que um (e apenas um) pedido é sequenciado na  $j$ -ésima posição, enquanto o conjunto de restrições do tipo (4.12) impõe que cada pedido  $k$  seja sequenciado em uma (e apenas uma) posição. O conjunto de restrições do tipo (4.13) define a ordem do processamento e é computado o tempo de *setup* da máquina  $i$  na posição  $j$  em  $D_{ij}$ , o conjunto de restrições do tipo (4.14) 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 (4.15), enquanto o conjunto de restrições do tipo (4.16) e (4.17) expressam o domínio das variáveis do modelo.

---

**Algoritmo 2: JPO-SPAM( $\delta, t_w, t_{limit}$ )**


---

Seja  $\bar{x}$  as variáveis binárias posicionais correspondentes que contém uma solução heurística  $\Pi$ ;

$\Pi_{best} := \Pi$

$\delta_{curr} := \delta$

**Enquanto**  $t_{limit}$  não é ultrapassado **faça**

**Para**  $k = 1, \dots, n$  **faça**

    | Calcule  $\mathcal{P}_k(\Pi, \delta_{curr})$

**Fim**

  Resolva a minimização de (4.10) sujeito a (4.11), (4.12), (4.13), (4.14), (4.15), (4.16), (4.17) usando  $\mathcal{P}_k(\Pi, \delta_{curr})$  ( $k = 1, \dots, n$ ) durante  $t_w$  segundos. Armazene em  $\bar{x}$  a solução assim obtida;

  Armazene em  $\Pi$  a sequência correspondente à solução  $\bar{x}$

**Se**  $f(x_{best}) > f(\bar{x})$  **então**

    | //Atualiza a melhor sequência

    |  $\Pi_{best} := \Pi$

    | //Atualiza o melhor resultado

    |  $x_{best} := \bar{x}$

    | //Atualiza o valor de delta

    |  $\delta_{curr} := \delta$

**Senão**

    |  $\delta_{curr} := \delta + 10$

**Fim**

**Fim**

---

## 5 EXPERIMENTOS COMPUTACIONAIS

Neste Capítulo, são apresentados os experimentos para avaliar o desempenho do MILP, do Modelo SPAM, da *matheuristic* híbrida JPO-SPAM, do JPO e das heurísticas que são propostas nos Capítulos 3 e 4. Na Seção 5.1, 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 5.2. Finalmente, nas Seções 5.3 e 5.4, são descritos os resultados dos experimentos computacionais.

### 5.1 Conjunto de Instâncias

Os experimentos computacionais propõe-se avaliar dois objetivos:

1. Avaliar a eficiência das heurísticas construtivas, do Modelo SPAM e do MILP, propostos nos Capítulos 3 e 4, respectivamente, e também avaliar as heurísticas construtivas EDD, OMDD e FP já presentes na literatura.
2. Avaliar a eficiência da *matheuristic* híbrida JPO-SPAM proposta na Seção 4.4 e do JPO proposto por Framinan e Perez-Gonzalez (2018).

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 é estudado não têm essas restrições, porém nesta abordagem são consideradas tais restrições, logo, são geradas instâncias relacionadas. É importante destacar que os conjuntos de instâncias gerados neste trabalho é uma das contribuições que inserimos na literatura.

Foram criados dois conjuntos de instâncias o de pequeno porte e o de grande porte, no qual o pequeno possui  $n \in \{8, 12, 16\}$  e  $m \in \{3, 5, 9\}$  e o grande  $n \in \{10, 20, 30, 40\}$  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  $\mu$ , isto é,

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

e, portanto, pode ser calculada para cada instância do problema. O parâmetro  $\mu$  é utilizado para tornar as datas de vencimento mais apertadas. Após experimentos computacionais preliminares, foi aditado o valor  $\mu = 0,1$ ;

- *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:  $TF : \{0.2, 0.5, 0.8\}$ .

Com relação ao primeiro objetivo, será usado o conjunto de instâncias de pequeno porte para avaliar a eficiências dos métodos propostos. Em relação ao segundo objetivo, foi criado o conjunto de instâncias de grande porte contendo instâncias maiores, pois o de pequeno porte não é suficiente. Além disso, pode-se avaliar o desempenho das heurísticas construtivas para que seja mais completa a análise.

Para cada combinação de parâmetros do conjunto de instâncias pequeno, 10 instâncias foram geradas, resultando num total de 2430 instâncias. Agora para cada combinação de parâmetros do conjunto de instâncias grande, 10 instâncias foram geradas, resultando num total de 3240 instâncias. As instâncias que foram usadas neste trabalho podem ser baixadas no link a seguir: [https://www.researchgate.net/publication/352289324\\_Evaluated\\_Instances](https://www.researchgate.net/publication/352289324_Evaluated_Instances).

## 5.2 Algoritmos em Comparação e indicadores para avaliação dos resultados

Em relação às heurísticas, a *matheuristic* híbrida, o Modelo SPAM e o modelo MILP, testadas no banco de teste, os seguintes algoritmos foram comparados:

- Heurística EDD (*Earliest Due Date*);
- Heurística OMDD por Lee (2013);
- Heurística FP por Framinan e Perez-Gonzalez (2018);
- JPO(20, 60, 600) denotado por JPO e proposto por Framinan e Perez-Gonzalez (2018);
- Heurística OMDD-S (*Order-Scheduling Modified Due-Date Setup*);
- Heurística FP-S (Framinan e Perez-Gonzalez Setup);
- O Modelo SPAM proposto no Seção 4.3 e indicado como SPAM nos resultados;
- JPO-SPAM(20, 60, 600) denotado por JPO-SPAM e proposto na Seção 4.4;
- Modelo MILP proposto na Seção 3.1.1, indicado como MILP nos resultados.

É importante destacar que a heurística que iniciou o JPO e a *matheuristic* híbrida JPO-SPAM foi a OMDD-S, diferente do que em Framinan e Perez-Gonzalez (2018) que iniciou com a FP. Essa decisão foi embasada em experimentos computacionais preliminares.

Os resultados obtidos pelas heurísticas, pelo JPO, pelo Modelo SPAM, pela *matheuristic* híbrida JPO-SPAM e pelo MILP em cada instância foram registrados usando os seguintes quatro indicadores:

- *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:

$$RDI_{st} = \begin{cases} 0, & \text{se } \min_{h \in H} T_{ht} = \max_{h \in H} T_{ht}, \\ \frac{T_{st} - \min_{h \in H} T_{ht}}{\max_{h \in H} T_{ht} - \min_{h \in H} T_{ht}} \cdot 100, & \text{caso contrário.} \end{cases} \quad (5.1)$$

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 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)).

- *SR* (*success rate*). É calculado como o número de vezes que um determinado método resulta na melhor solução (com ou sem empate) dividido pelo número de instâncias de teste em uma determinada classe de instância, é expresso como na Eq. (5.2):

$$SR = \frac{n_{BEST}}{n_{INST}} \times 100 \quad (5.2)$$

em que  $n_{BEST}$  é o número de instâncias nas quais um determinado método alcançou a melhor solução e  $n_{INST}$  é o número de instâncias no conjunto de instâncias fornecido.

- Otimalidade,  $Opt \in \{0, 1\}$ , sendo 1 se a solução obtida for igual à fornecida pelo MILP, se o MILP tiver atingido o ótimo; e 0 caso contrário.
- Tempo de CPU, expresso em segundos.

As heurísticas, as *matheuristics* e o modelo MILP foram codificados na linguagem *Python 3.7* usando o resolvidor CPLEX 12.10.0 na configuração padrão, 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.

### 5.3 Resultados para o conjunto de instâncias de pequeno porte

Nas Tabelas 9 e 10 são apresentados os resultados para os diferentes valores de  $m$  (máquinas) e  $n$  (pedidos), incluindo a média e o desvio padrão dos valores de  $RDI$ , também estão incluídos a porcentagem de ótimos encontrados e o tempo médio para cada conjunto de instâncias. Observando o primeiro conjunto de instâncias ( $m = 3$  e  $n = 8$ ) temos que o MILP é o melhor entre todos os métodos estudados e obteve a menor média e o menor desvio padrão, porém quando o conjunto de instâncias aumenta os dois melhores métodos são o Modelo SPAM, tendo as menores médias e os menores desvios padrões.

Tabela 9: Instâncias de Pequeno Porte: Heurísticas e Modelos para  $m$  e  $n$

$m$	$n$	EDD			FP			FP-S			OMDD		
		Média	Desvio Pd.	Tempo									
3	8	95,5924	10,7901	0,0000	72,7618	26,1200	0,0010	45,0979	24,0381	0,0014	69,9282	25,4258	0,0010
	12	97,9772	6,9836	0,0001	75,9761	25,5007	0,0030	48,9008	22,8246	0,0045	72,9033	27,9774	0,0030
	16	92,1670	24,7728	0,0001	73,2740	30,1649	0,0066	47,3200	26,5247	0,0103	70,0654	31,4608	0,0069
5	8	93,2917	14,4990	0,0000	68,9482	25,3962	0,0013	42,6921	23,2525	0,0020	63,9066	27,7710	0,0013
	12	95,1717	12,1162	0,0001	72,4786	24,0260	0,0038	46,8317	23,7087	0,0062	70,3361	27,2906	0,0039
	16	88,7469	26,0195	0,0001	69,2486	28,9762	0,0087	48,4049	27,7139	0,0144	67,2913	30,6391	0,0090
9	8	91,3742	16,2296	0,0000	68,3929	25,7720	0,0019	42,1489	22,6573	0,0031	60,7880	26,8431	0,0019
	12	90,7482	15,9556	0,0001	68,6689	23,9219	0,0056	44,3806	22,2601	0,0098	64,7505	25,2717	0,0058
	16	80,2203	29,7194	0,0001	60,9753	29,9828	0,0127	40,2145	25,8252	0,0226	57,2266	31,6399	0,0131
Total		91,6988	19,4780	0,0001	70,0805	27,0107	0,0050	45,1101	24,5031	0,0083	66,3551	28,7118	0,0051

Fonte: elaborada pelo autor

Tabela 10: Instâncias de Pequeno Porte: Heurísticas e Modelos para  $m$  e  $n$

$m$	$n$	OMDD-S			SPAM			MILP		
		Média	Desvio Pd.	Tempo	Média	Desvio Pd.	Tempo	Média	Desvio Pd.	Tempo
3	8	39,6805	22,7597	0,0003	5,9073	8,6510	5,2375	<b>2,8183</b>	7,1515	334,8832
	12	39,6883	23,0271	0,0007	<b>2,6774</b>	6,7317	438,3857	18,1819	20,9791	601,7716
	16	41,2252	26,7577	0,0011	<b>2,2289</b>	6,2924	542,1021	23,2568	25,6060	540,4455
5	8	34,9788	21,1950	0,0005	<b>2,2768</b>	4,9861	9,7212	23,7074	28,8696	550,4171
	12	38,3124	23,1388	0,0009	<b>1,0847</b>	5,5263	479,4585	46,7985	30,8030	600,1471
	16	36,8398	25,4770	0,0015	<b>1,3108</b>	5,0498	555,0010	49,9690	33,9772	549,4129
9	8	32,1479	20,2378	0,0006	<b>0,0859</b>	0,7310	19,4298	56,5587	30,3852	600,2293
	12	35,4173	21,4594	0,0014	<b>0,5604</b>	4,0074	555,9576	68,1536	29,6034	608,7139
	16	29,3475	24,8520	0,0022	<b>1,5982</b>	6,1035	565,6225	67,5235	35,8523	582,6817
Total		36,4042	23,5445	0,0010	<b>1,9700</b>	5,9310	352,3240	39,6631	35,7772	552,0780

Fonte: elaborada pelo autor

Como podemos observar na Tabela 11, o MILP é o que obtém a maior quantidade de valores ótimos com relação as heurísticas e o Modelo SPAM, no qual o conjunto de instâncias com  $m = 3$  e  $n = 8$  o MILP obteve 61,11%, sendo a maior porcentagem de otimalidade.

Tabela 11: Instâncias de Pequeno Porte: % de otimalidade para  $m$  e  $n$

$m$	$n$	EDD	FP	FP-S	OMDD	OMDD-S	SPAM	MILP
3	8	0,00%	0,00%	1,11%	0,00%	1,11%	17,78%	61,11%
	12	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	1,11%
	16	6,30%	6,30%	6,30%	6,30%	6,30%	10,37%	12,22%
5	8	0,00%	0,37%	0,00%	0,00%	0,00%	4,07%	17,04%
	12	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,74%
	16	5,93%	5,93%	5,93%	5,93%	5,93%	9,26%	11,48%
9	8	0,00%	0,00%	0,00%	0,00%	0,00%	0,37%	1,11%
	12	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
	16	5,56%	5,56%	5,56%	5,56%	5,56%	7,04%	7,78%
Total		1,98%	2,02%	2,10%	1,98%	2,10%	5,43%	12,51%

Fonte: elaborada pelo autor

Nas Tabelas 12 e 13 são apresentados os mesmos resultados agregados em relação ao  $TF$  e  $RDD$ , incluindo a média e o desvio padrão dos valores de  $RDI$ , também estão incluídos a porcentagem de ótimos encontrados e o tempo médio para cada conjunto de instâncias. Observando a tabela o melhor método é o Modelo SPAM, tal método obteve os melhores resultados em média e desvio padrão.

Tabela 12: Instâncias de Pequeno Porte: Heurísticas e Modelos para  $TF$  e  $RDD$

$TF$	$RDD$	EDD			FP			FP-S			OMDD		
		Média	Desvio Pd.	Tempo									
0.2	0.2	97,2370	8,8203	0,0001	78,5159	26,0079	0,0049	55,4418	27,1227	0,0083	77,8198	27,5458	0,0051
	0.5	88,6677	27,4747	0,0001	70,8692	31,8983	0,0050	49,3033	28,6754	0,0082	68,6830	32,7437	0,0051
	0.8	85,1766	31,4354	0,0001	69,8468	33,3590	0,0049	49,5453	30,5102	0,0082	65,3195	34,7565	0,0051
0.5	0.2	94,2726	12,7031	0,0001	72,1748	26,3138	0,0050	46,3215	22,3224	0,0082	72,3025	25,9000	0,0051
	0.5	92,1407	15,6173	0,0000	72,5721	25,2546	0,0049	44,8446	21,2566	0,0083	67,2162	28,1337	0,0051
	0.8	89,5994	20,0316	0,0001	70,2081	26,2634	0,0050	42,7151	24,0551	0,0083	65,9920	29,0104	0,0051
0.8	0.2	94,3325	13,0639	0,0001	66,6091	23,0282	0,0050	42,7166	20,5596	0,0082	58,6990	24,3036	0,0051
	0.5	92,2061	16,5154	0,0001	63,6130	23,5619	0,0050	37,7572	18,4806	0,0082	59,3584	24,4476	0,0051
	0.8	91,6570	15,9317	0,0001	66,3153	22,9051	0,0050	37,3459	19,1649	0,0083	61,8058	24,6797	0,0051
Total		91,6988	19,4780	0,0001	70,0805	27,0107	0,0050	45,1101	24,5031	0,0083	66,3551	28,7118	0,0051

Fonte: elaborada pelo autor

Tabela 13: Instâncias de Pequeno Porte: Heurísticas e Modelos para  $TF$  e  $RDD$ 

$TF$	$RDD$	OMDD-S			SPAM			MILP		
		Média	Desvio Pd.	Tempo	Média	Desvio Pd.	Tempo	Média	Desvio Pd.	Tempo
0.2	0.2	51,5566	24,1698	0,0010	<b>4,3980</b>	8,6981	373,9367	22,5721	29,3454	541,7492
	0.5	44,0141	27,3227	0,0010	<b>3,3462</b>	7,2883	306,9237	27,1313	34,3152	494,8373
	0.8	42,4777	30,1257	0,0010	<b>4,0201</b>	9,2227	297,3631	27,2049	34,7209	482,5090
0.5	0.2	37,8550	19,4474	0,0010	<b>0,9346</b>	3,6179	375,1919	39,0991	32,4801	566,5620
	0.5	36,4583	20,5938	0,0010	<b>1,6021</b>	5,0088	64,7990	44,6720	37,0097	573,6156
	0.8	32,7976	22,2552	0,0010	<b>1,2377</b>	4,3834	360,7954	46,0753	37,4934	575,5708
0.8	0.2	29,2587	17,6455	0,0010	<b>0,6277</b>	2,8225	359,8273	48,1889	31,7879	582,6589
	0.5	27,2532	17,4737	0,0010	<b>0,8362</b>	3,1991	367,9543	52,0129	35,1051	577,6879
	0.8	25,9665	16,3588	0,0010	<b>0,7279</b>	3,1273	364,1244	50,0111	35,1252	573,5116
Total		36,4042	23,5445	0,0010	<b>1,9700</b>	5,9310	352,3240	39,6631	35,7772	552,0780

Fonte: elaborada pelo autor

Da mesma forma que observamos na Tabela 11, na Tabela 14, o MILP é o que obteve os maiores percentuais de valores ótimos com relação as heurísticas e o Modelo SPAM, onde no conjunto com  $TF = 0.2$  e  $RDD = 0.8$  o MILP obteve o maior percentual de otimalidade com 27,41%.

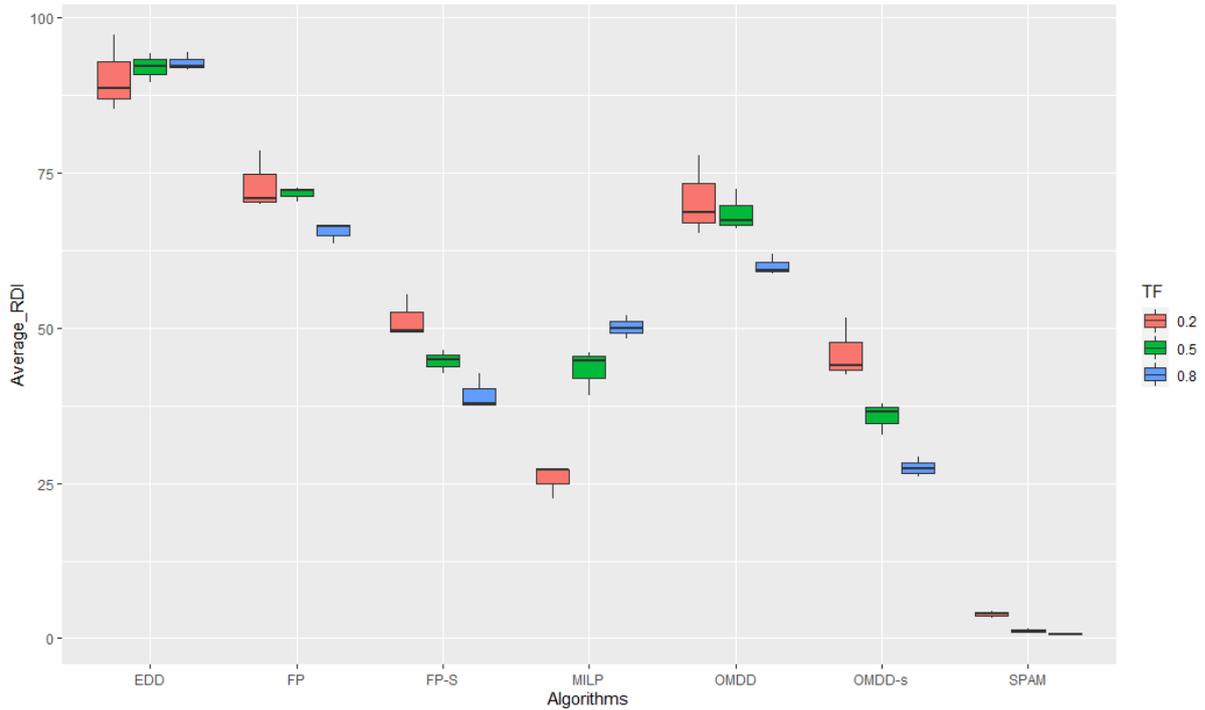
Tabela 14: Instâncias de Pequeno Porte: % de otimalidade para  $TF$  e  $RDD$ 

$TF$	$RDD$	EDD	FP	FP-S	OMDD	OMDD-S	SPAM	MILP
0.2	0.2	0,00%	0,00%	0,00%	0,00%	0,00%	4,44%	13,70%
	0.5	7,41%	7,41%	7,41%	7,41%	7,41%	12,22%	21,85%
	0.8	10,37%	10,37%	10,37%	10,37%	11,11%	14,81%	27,41%
0.5	0.2	0,00%	0,00%	0,00%	0,00%	0,00%	2,96%	10,37%
	0.5	0,00%	0,00%	0,00%	0,00%	0,00%	2,96%	9,63%
	0.8	0,00%	0,37%	0,37%	0,00%	0,00%	4,07%	8,15%
0.8	0.2	0,00%	0,00%	0,37%	0,00%	0,37%	3,33%	6,30%
	0.5	0,00%	0,00%	0,00%	0,00%	0,00%	1,48%	7,78%
	0.8	0,00%	0,00%	0,37%	0,00%	0,00%	2,59%	7,41%
Total		1,98%	2,02%	2,10%	1,98%	2,10%	5,43%	12,51%

Fonte: elaborada pelo autor

Na Figura 4 são apresentados os resultados em *Boxplot* agregados em relação ao  $TF$  das Tabelas 12 e 13. A partir desta figura é constatado que o melhor método é o Modelo SPAM em todos os parâmetros de  $TF$ . O método MILP obteve resultados bons para o  $TF = 0.2$ , porém quando o  $TF = 0.5$  e  $TF = 0.8$  se torna o segundo e terceiro melhor método estudado, respectivamente.

Figura 4: Instâncias de Pequeno Porte: Boxplot em relação a média *RDI* para *TF*



Fonte: elaborada pelo autor

Na Figura 5 são apresentados os resultados em *Boxplot* agregados em relação ao *RDD* das Tabelas 12 e 13. A partir desta figura é constatado que o Modelo SPAM é o melhor entre todos os métodos estudados.

Na Tabela 15 são mostrados os resultados para o *SR* (*sucess rate*), onde é constatado que o Modelo SPAM obteve o maior percentual de *SR* entre todos os métodos avaliados, assim pode-se concluir novamente que é o mais eficiente quando se é avaliado em um conjunto de instâncias de pequeno porte.

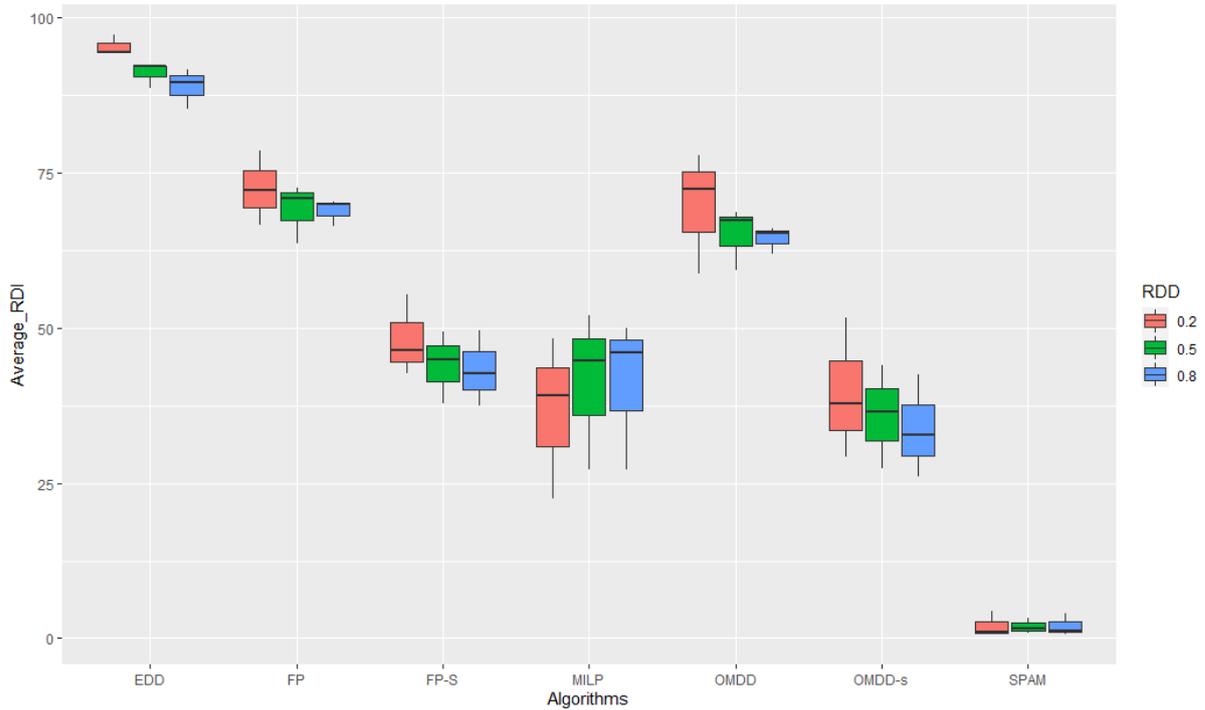
Tabela 15: Instâncias de Pequeno Porte: *SR* (*sucess rate*)

EDD	FP	FP-S	OMDD	OMDD-S	SPAM	MILP
1,98%	0,65%	0,93%	0,82%	1,37%	30,75%	8,93%

Fonte: elaborada pelo autor

#### 5.4 Resultados para o conjunto de instâncias de grande porte

Nas Tabelas 16 e 17 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*. Avaliando o primeiro conjunto de instâncias ( $m = 2$  e  $n = 10$ ) o SPAM é o melhor entre todos os métodos estudados e obteve a menor média e o menor desvio padrão, mas quando o conjunto de instâncias

Figura 5: Instâncias de Pequeno Porte: Boxplot em relação a média *RDI* para *RDD*

Fonte: elaborada pelo autor

começa a aumentar o MILP e a *matheuristic* híbrida JPO-SPAM se tornam os dois melhores métodos avaliados, apresentando os menores valores da média e do desvio padrão.

Tabela 16: Instâncias de Grande Porte: Heurísticas e Modelos para  $m$  e  $n$ 

$m$	$n$	EDD			FP			FP-S			OMDD			OMDD-S		
		Média	Desvio Pd.	Tempo												
2	10	96,0493	9,3616	0,0001	74,0344	25,0666	0,0032	56,2164	27,9392	0,0035	71,7100	26,1257	0,0034	38,9925	21,3457	0,0009
	20	78,0808	39,7039	0,0002	59,7296	36,7141	0,0243	50,7682	35,8774	0,0346	60,5183	37,2953	0,0253	34,4581	28,5890	0,0026
	30	51,2699	47,9732	0,0002	38,7929	38,8785	0,0820	32,3783	34,8791	0,0759	37,3421	38,5414	0,0504	14,9562	21,4639	0,0052
5	10	30,6115	41,6665	0,0003	20,7499	29,2799	0,2033	16,2446	24,2014	0,2487	20,6954	29,5093	0,0896	5,5452	12,3045	0,0099
	20	68,5931	38,5886	0,0001	51,8989	33,9248	0,0406	47,7679	33,6858	0,0562	49,4103	34,2563	0,0274	26,2327	27,6293	0,0047
	30	24,8256	30,0024	0,0002	15,7638	19,9920	0,1469	14,0551	19,2174	0,1121	14,7223	19,7352	0,0740	5,0594	13,3444	0,0101
8	10	12,5403	18,9989	0,0003	7,7343	11,3168	0,3218	6,2689	9,0028	0,4354	7,2335	12,0397	0,1340	0,8205	3,8954	0,0162
	20	91,5210	15,5189	0,0001	68,1319	24,8727	0,0075	56,2488	25,4111	0,0103	64,4638	26,7540	0,0058	34,8510	20,2135	0,0017
	30	47,8643	34,6722	0,0002	35,4441	30,6042	0,0545	32,0611	30,9270	0,0866	34,0785	30,4616	0,0355	16,3654	24,0684	0,0062
Total	10	13,8092	17,0370	0,0002	8,3775	9,3839	0,1907	7,6681	9,6344	0,3239	7,2338	9,0156	0,0918	1,2870	3,4212	0,0135
	20	7,0822	10,0866	0,0003	4,0645	5,4368	0,4241	4,1008	5,6089	0,7314	3,7166	5,1589	0,1812	0,0762	0,4868	0,0231
	30	51,3363	43,6517	0,0002	38,1630	36,6161	0,1254	31,8851	32,8421	0,1771	36,5168	36,2386	0,0602	18,0044	24,0698	0,0080

Fonte: elaborada pelo autor

Tabela 17: Instâncias de Grande Porte: Heurísticas e Modelos para  $m$  e  $n$ 

$m$	$n$	SPAM			MILP			JPO-SPAM			JPO		
		Média	Desvio Pd.	Tempo	Média	Desvio Pd.	Tempo	Média	Desvio Pd.	Tempo	Média	Desvio Pd.	Tempo
2	10	<b>2,5691</b>	5,0185	565,1671	4,1034	6,0077	151,7989	4,5762	6,4201	600,00	22,7252	18,4103	600,00
	20	9,4939	13,0045	463,8058	<b>1,4788</b>	3,8262	464,7019	10,6221	15,5238	600,00	29,8185	25,6702	600,00
	30	29,6135	35,5353	377,5357	<b>3,3109</b>	11,3456	365,4263	6,2066	9,5153	600,00	14,9408	21,4362	600,00
5	10	47,8850	46,3269	364,4728	7,4285	16,0226	337,6289	<b>3,2077</b>	7,8071	600,00	5,5452	12,3045	600,00
	20	44,4169	31,2991	602,7449	<b>0,8070</b>	3,4905	281,1029	2,6394	5,1434	600,00	35,9667	20,1085	600,00
	30	56,7089	40,8491	498,2330	<b>1,5417</b>	5,5628	494,8419	18,0544	24,0035	600,00	26,0185	27,2406	600,00
8	10	63,7955	47,0304	432,0187	8,1344	18,8562	410,0134	<b>3,5100</b>	10,4422	600,00	5,0594	13,3444	600,00
	20	63,6526	48,1386	397,5462	13,7993	23,4163	367,5082	<b>0,6903</b>	3,8231	600,00	0,8205	3,8954	600,00
	30	59,8336	29,7481	601,5557	<b>0,5792</b>	2,9924	345,8448	2,3233	5,0796	600,00	34,0763	19,5936	600,00
Total	20	72,9362	42,2317	529,1508	<b>2,1991</b>	6,2835	502,8810	10,4192	18,2998	600,00	16,2987	23,8797	600,00
	30	70,3357	45,7281	442,2819	8,7350	15,6913	418,0169	<b>1,0305</b>	2,7315	600,00	1,2870	3,4212	600,00
	40	65,4765	47,5565	409,8493	16,1070	24,7707	388,1069	<b>0,0684</b>	0,4767	600,00	0,0762	0,4868	600,00
Total		48,8931	44,5045	473,6968	5,6854	14,6673	377,3227	<b>5,2790</b>	12,3564	600,00	16,0528	22,1632	600,00

Fonte: elaborada pelo autor

Na Tabela 18 são apresentadas as porcentagens de otimalidade das heurísticas, matheuristics e modelos para o  $m$  e  $n$ . Pode-se observar que o MILP e a *matheuristic* híbrida JPO-SPAM são capazes de encontrar os maiores percentuais de valores ótimos. Analisando os conjuntos separadamente, constata-se que onde  $m = 5$  e  $n = 10$  o MILP obteve o maior percentual de otimalidade com 64,81%.

Tabela 18: Instâncias de Grande Porte: % de otimalidade para  $m$  e  $n$ 

$m$	$n$	EDD	FP	FP-S	OMDD	OMDD-S	SPAM	MILP	JPO-SPAM	JPO
2	10	0,00%	0,00%	0,00%	0,00%	0,37%	56,30%	43,33%	39,26%	4,07%
	20	20,00%	20,00%	20,00%	20,00%	20,00%	25,93%	25,93%	23,33%	20,37%
	30	40,00%	40,00%	40,00%	40,00%	40,00%	41,48%	41,48%	40,37%	40,00%
	40	45,19%	45,19%	45,19%	45,19%	45,19%	42,22%	45,19%	45,19%	45,19%
5	10	0,00%	0,00%	0,00%	0,00%	0,37%	4,44%	64,81%	49,63%	00,37%
	20	17,78%	17,78%	17,78%	17,78%	17,78%	20,74%	21,11%	17,78%	17,78%
	30	33,70%	33,70%	33,70%	33,70%	33,70%	33,33%	34,07%	33,70%	33,70%
	40	40,37%	40,37%	40,37%	40,37%	40,37%	35,93%	40,37%	40,37%	40,37%
8	10	0,00%	0,00%	0,00%	0,00%	0,37%	0,37%	55,19%	40,74%	00,37%
	20	15,56%	15,56%	15,56%	15,56%	15,56%	19,26%	19,26%	16,30%	15,56%
	30	31,85%	31,85%	31,85%	31,85%	31,85%	28,89%	31,85%	31,85%	31,85%
	40	38,15%	38,15%	38,15%	38,15%	38,15%	33,33%	38,15%	38,15%	38,15%
Total		23,55%	23,55%	23,55%	23,55%	23,64%	28,52%	38,40%	34,72%	23,98%

Fonte: elaborada pelo autor

Nas Tabelas 19 e 20 são apresentados os mesmos resultados agregados em relação ao  $TF$  e  $RDD$ , incluindo a média e o desvio padrão dos valores de  $RDI$ . Observando as tabelas os melhores métodos são o MILP e a *matheuristic* híbrida JPO-SPAM, tais métodos obtiveram os melhores resultados em média e desvio padrão.

Tabela 19: Instâncias de Grande Porte: Heurísticas e Modelos para  $TF$  e  $RDD$ 

$TF$	$RDD$	EDD			FP			FP-S			OMDD			OMDD-S		
		Média	Desvio Pd.	Tempo												
0.2	0.2	49,7108	46,3151	0,0002	37,9304	39,5661	0,1253	32,3280	35,8786	0,1682	39,0234	39,8507	0,0596	24,8065	28,6791	0,0082
	0.5	44,1941	45,6207	0,0002	35,7960	39,8416	0,1258	33,3222	37,9785	0,1704	35,1458	39,3492	0,0618	21,5752	29,1973	0,0075
	0.8	35,8728	44,9106	0,0002	27,9817	37,7023	0,1257	24,3531	34,2296	0,1838	26,2024	36,6533	0,0608	14,4111	23,6940	0,0081
0.5	0.2	60,7937	42,2037	0,0002	46,5010	38,2813	0,1250	39,9098	35,5864	0,1684	46,3644	37,7785	0,0591	26,4463	26,4100	0,0078
	0.5	52,1433	44,0704	0,0002	39,5498	37,8962	0,1274	34,9246	35,7333	0,1811	37,6886	37,8441	0,0596	19,9796	26,1782	0,0081
	0.8	45,3101	44,6130	0,0002	32,7084	35,6869	0,1246	26,9918	31,5203	0,1867	32,0222	36,1065	0,0588	15,1196	22,7104	0,0077
0.8	0.2	64,1586	36,8215	0,0002	44,7169	30,9769	0,1246	34,4809	25,7323	0,1705	41,7298	30,6978	0,0647	14,5048	17,2718	0,0079
	0.5	57,2549	38,8929	0,0002	40,3317	31,0763	0,1252	30,9592	25,4458	0,1810	36,5535	30,5876	0,0582	12,9058	16,5442	0,0083
	0.8	52,5883	41,7649	0,0002	37,9509	34,1006	0,1246	29,6960	28,5363	0,1839	33,9210	32,5531	0,0591	12,2908	17,1234	0,0081
Total		51,3363	43,6517	0,0002	38,1630	36,6161	0,1254	31,8851	32,8421	0,1771	36,5168	36,2386	0,0602	18,0044	24,0698	0,0080

Fonte: elaborada pelo autor

Tabela 20: Instâncias de Grande Porte: Heurísticas e Modelos para  $TF$  e  $RDD$ 

$TF$	$RDD$	SPAM			MILP			JPO-SPAM			JPO		
		Média	Desvio Pd.	Tempo	Média	Desvio Pd.	Tempo	Média	Desvio Pd.	Tempo	Média	Desvio Pd.	Tempo
0.2	0.2	22,0448	35,8186	331,6373	<b>2,5654</b>	7,2692	289,6659	6,6008	13,2974	600,00	20,9657	26,1605	600,00
	0.5	25,3158	38,6867	310,3186	<b>3,9393</b>	12,3742	256,3614	5,7427	14,7685	600,00	19,8600	27,4597	600,00
	0.8	26,1552	40,3818	313,6964	5,5456	17,7981	237,3922	<b>3,3797</b>	9,0318	600,00	12,6304	21,5056	600,00
0.5	0.2	39,4726	41,5005	452,8478	<b>1,9476</b>	7,9712	375,2230	9,1069	14,8537	600,00	23,5571	24,2580	600,00
	0.5	45,1672	44,4587	485,0640	<b>2,9747</b>	11,2436	362,8758	7,7670	18,1712	600,00	17,6779	23,9755	600,00
	0.8	58,6733	44,3084	548,3612	5,8597	16,5763	376,1073	<b>4,5871</b>	13,3385	600,00	14,0253	21,9911	600,00
0.8	0.2	71,9319	36,8566	610,3065	7,7371	15,4082	500,0889	<b>3,6934</b>	6,2072	600,00	12,9980	15,8555	600,00
	0.5	75,3769	35,5378	607,4346	9,9924	16,8006	502,3950	<b>3,1289</b>	5,8179	600,00	11,6613	14,7730	600,00
	0.8	75,9003	34,8084	603,6050	10,6064	18,7741	495,7944	<b>3,5047</b>	7,9139	600,00	11,0991	15,7275	600,00
Total		48,8931	44,5045	473,6968	5,6854	14,6673	377,3227	<b>5,2790</b>	12,3564	600,00	16,0528	22,1632	600,00

Fonte: elaborada pelo autor

Na Tabela 21 são apresentadas as porcentagens de otimalidade das heurísticas, matheuristics e modelos para o  $TF$  e  $RDD$ . Ao ser analisado cada conjunto é possível observar que quando o  $TF = 0.2$  e  $RDD = 0.8$  é encontrado o maior percentual de otimalidade, no qual é o MILP com 59,72%. Assim o MILP seguido da *matheuristic* híbrida JPO-SPAM são capazes de encontrar a maior quantidade de de soluções ótimas se comparados com os outros métodos.

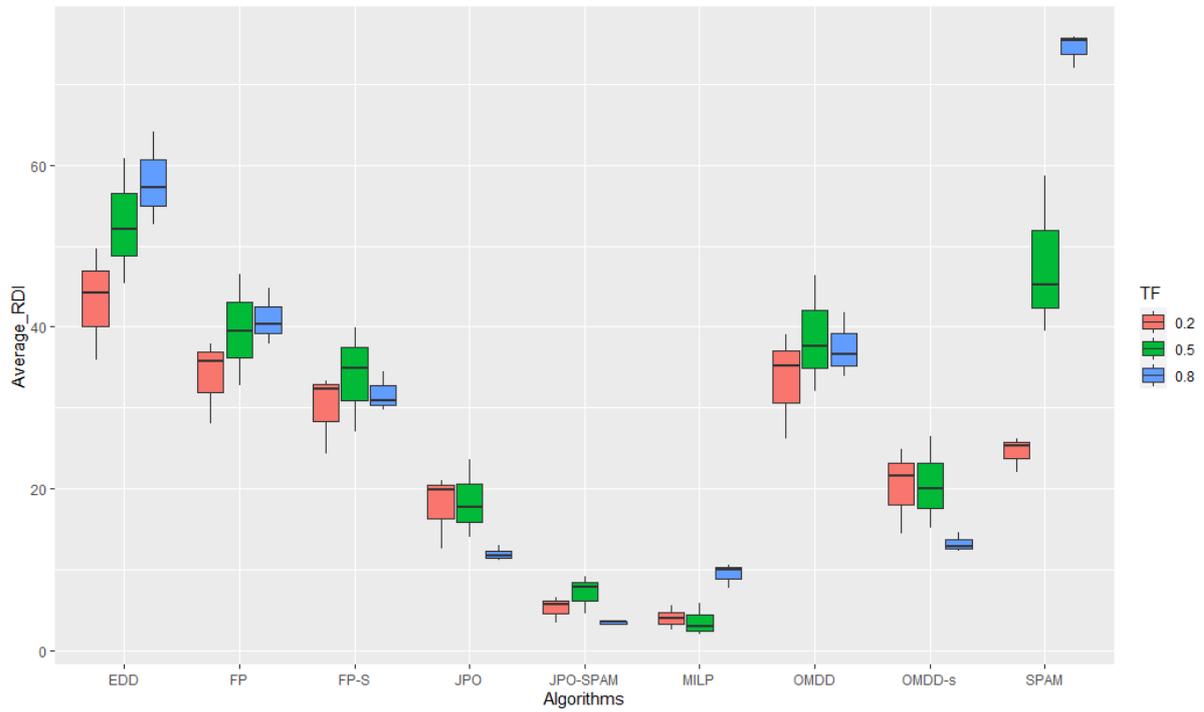
Tabela 21: Instâncias de Grande Porte: % de otimalidade para  $TF$  e  $RDD$ 

$TF$	$RDD$	EDD	FP	FP-S	OMDD	OMDD-S	SPAM	MILP	JPO-SPAM	JPO
0.2	0.2	41,67%	41,67%	41,67%	41,67%	41,67%	46,39%	51,11%	50,28%	41,94%
	0.5	44,44%	44,44%	44,44%	44,44%	44,44%	56,39%	56,39%	53,06%	45,00%
	0.8	49,72%	49,72%	49,72%	49,72%	49,72%	57,78%	59,72%	58,06%	50,28%
0.5	0.2	25,00%	25,00%	25,00%	25,00%	25,00%	30,56%	38,33%	34,44%	25,28%
	0.5	26,11%	26,11%	26,11%	26,11%	26,11%	32,22%	42,78%	36,67%	26,67%
	0.8	24,17%	24,17%	24,17%	24,17%	24,44%	20,56%	42,78%	38,33%	25,00%
0.8	0.2	00,00%	00,00%	00,00%	00,00%	00,00%	4,17%	18,06%	13,89%	00,00%
	0.5	00,00%	00,00%	00,00%	00,00%	00,56%	4,44%	17,78%	13,06%	00,83%
	0.8	00,83%	00,83%	00,83%	00,83%	00,83%	4,17%	18,61%	14,72%	00,83%
Total		23,55%	23,55%	23,55%	23,55%	23,64%	28,52%	38,40%	34,72%	23,98%

Fonte: elaborada pelo autor

Na Figura 6 são apresentados os resultados em *Boxplot* agregados em relação ao  $TF$  das Tabelas 19 e 20. A partir desta figura é constatado que o melhor os métodos são a *matheuristic* híbrida JPO-SPAM e o MILP. O Modelo SPAM obteve bons resultados, porém não supera os dois citados anteriormente.

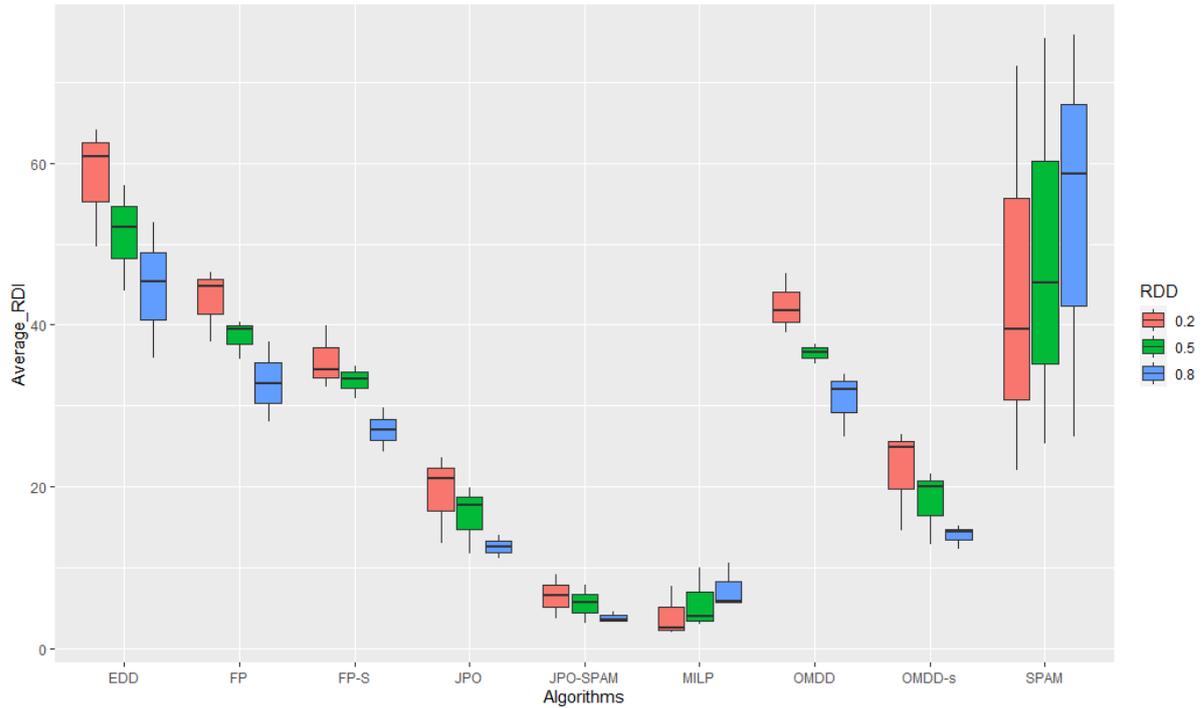
Figura 6: Instâncias de Grande Porte: Boxplot em relação a média *RDI* para *TF*



Fonte: elaborada pelo autor

Na Figura 7 são apresentados os resultados em *Boxplot* agregados em relação ao *RDD* das Tabelas 19 e 20. A partir desta figura é constatado que a *matheuristic* híbrida JPO-SPAM é o melhor entre todos seguido do MILP.

Figura 7: Instâncias de Grande Porte: Boxplot em relação a média *RDI* para *RDD*



Fonte: elaborada pelo autor

Na Tabela 22 são mostrados os resultados para o *SR* (*sucess rate*), onde é constatado que o MILP e a *matheuristic* JPO-SPAM obtiveram os maiores percentuais de *SR* entre todos os métodos avaliados, assim pode-se concluir que o MILP é o mais eficiente seguido da *matheuristic* híbrida JPO-SPAM quando se é avaliado em um conjunto de instâncias de grande porte.

Tabela 22: Instâncias de Grande Porte: *SR* (*sucess rate*)

EDD	FP	FP-S	OMDD	OMDD-S	SPAM	MILP	JPO-SPAM	JPO
28,43%	28,43%	28,52%	28,43%	43,95%	32,47%	69,32%	59,81%	44,29%

Fonte: elaborada pelo autor

## 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 da sequência de produção. Propomos uma nova formulação MILP, uma heurística FP-S, uma heurística OMDD-S, o SPAM e uma *matheuristic* híbrida JPO-SPAM, também foi proposto uma geração de conjunto de dados (instâncias) para o problema abordado neste trabalho. Destacamos que a heurística FP-S, a heurística OMDD-S e o SPAM, são respectivamente adaptações das heurísticas FP (ver em Framinan e Perez-Gonzalez (2018)), da heurística OMDD (ver em Lee (2013)) e do MILP (ver em Framinan e Perez-Gonzalez (2018)).

Experimentos computacionais foram realizados em um conjunto de instâncias geradas aleatoriamente, objetivando avaliar o desempenho dos métodos em estudo. Como indicador de comparação entre os métodos, foi utilizado o *RDI*, os tempos computacionais e otimalidade do problema. São avaliados um total de 5670 instâncias, sendo elas divididas em dois conjuntos de instâncias, o primeiro é o conjunto de instâncias de pequeno porte com 2430 instâncias e o segundo é o conjunto de grande porte com 3240 instâncias.

Com base nos experimentos computacionais realizados, é constatado que no conjunto de instâncias de pequeno porte o SPAM e a heurística OMDD-S obtiveram os menores valores de *RDI* médio, assim são os melhores métodos para esse conjunto de dados (instâncias), porém quando é avaliado o conjunto de instâncias de grandes porte a *matheuristic* híbrida JPO-SPAM e o MILP conseguiram os menores valores de *RDI* médio, sendo os melhores métodos para esse conjunto de dados (instâncias). Portanto, o SPAM e a *matheuristic* híbrida propostos são métodos promissores para resolver o problema em estudo.

Como sugestões para trabalhos futuros, podem ser destacadas a geração de novos conjuntos de dados (instâncias) para o problema em estudo, assim como para outros problemas da literatura que não possuem as restrições de tempos de preparação empregadas, 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 objetivas na literatura de sequenciamento de pedidos com máquinas dedicadas em paralelo com *assembly* igual a zero.

Durante o período de realização do mestrado foram publicados dois artigos em congresso, um em uma conferência nacional e outro em uma conferência internacional, a saber:

1. Antonioli, M. P; Rodrigues, C. D.; Prata, B. de A. Modelos e algoritmos para o problema de sequenciamento de pedidos de clientes para minimização do atraso total. Anais do LII Simpósio Brasileiro de Pesquisa Operacional, João Pessoa: SBPO 2020. Anais.
2. Antonioli, M. P; Rodrigues, C. D.; Prata, B. de A. Customer order scheduling to minimize total tardiness with sequence-depended setup times. International Conference of Production Research, Bahía Blanca: ICPR - Americas 2020.

Nossas maiores contribuições para a literatura neste trabalho foram a inserção das restrições de tempo de preparação dependentes das máquinas e a criação de um novo tipo de conjunto de dados, no qual foram denominados neste trabalho como de pequeno e grande porte.

## REFERÊNCIAS

AHMADI, R.; BAGCHI, U.; ROEMER, T. A. Coordinated scheduling of customer orders for quick response. **Naval Research Logistics (NRL)**, [United States], v. 52, n. 6, p. 493–512, 2005. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1002/nav.20092>. Acesso em: 12 Ago. 2019.

FERNANDEZ-VIAGAS, V.; FRAMINAN, J. M. Neh-based heuristics for the permutation flowshop scheduling problem to minimise total tardiness. **Computers & Operations Research**, [United Kingdom], v. 60, p. 27 – 36, 2015. ISSN 0305-0548. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0305054815000258>. Acesso em: 19 Set. 2019.

FERNANDEZ-VIAGAS, V.; PEREZ-GONZALEZ, P.; FRAMINAN, J. M. Efficiency of the solution representations for the hybrid flow shop scheduling problem with makespan objective. **Computers & Operations Research**, [United Kingdom], v. 109, p. 77–88, 2019. ISSN 0305-0548. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0305054819301157>. Acesso em: 30 Set. 2020.

FRAMINAN, J. M.; PEREZ-GONZALEZ, P. New approximate algorithms for the customer order scheduling problem with total completion time objective. **Computers & Operations Research**, [United Kingdom], v. 78, p. 181 – 192, 2017. ISSN 0305-0548. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0305054816302295>. Acesso em: 27 Maio 2021.

FRAMINAN, J. M.; PEREZ-GONZALEZ, P. Order scheduling with tardiness objective: Improved approximate solutions. **European Journal of Operational Research**, [Netherlands], v. 266, n. 3, p. 840 – 850, 2018. ISSN 0377-2217. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0377221717309931>. Acesso em: 08 Ago. 2019.

FRAMINAN, J. M.; PEREZ-GONZALEZ, P.; FERNANDEZ-VIAGAS, V. Deterministic assembly scheduling problems: A review and classification of concurrent-type scheduling models and solution procedures. **European Journal of Operational Research**, [Netherlands], v. 273, n. 2, p. 401–417, 2019. ISSN 0377-2217. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0377221718303369>. Acesso em: 08 Ago. 2019.

KARABULUT, K. A hybrid iterated greedy algorithm for total tardiness minimization in permutation flowshops. **Computers & Industrial Engineering**, [United States], v. 98, p. 300 – 307, 2016. ISSN 0360-8352. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0360835216302078>. Acesso em: 02 Jun. 2020.

KIM, Y.-D. Heuristics for flowshop scheduling problems minimizing mean tardiness. **Journal of the Operational Research Society**, [United Kingdom], v. 44, n. 1, p. 19–28, 1993. Cited By 88. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-0027240799&doi=10.1057/%2fjors.1993.3&partnerID=40&md5=a74c40f8c6baaac6cfbb5b14b3070e7>. Acesso em: 23 Out. 2019.

LEE, I. S. Minimizing total tardiness for the order scheduling problem. **International Journal of Production Economics**, [Netherlands], v. 144, n. 1, p. 128 – 134, 2013. ISSN 0925-5273.

Disponível em: <http://www.sciencedirect.com/science/article/pii/S0925527313000431>. Acesso em: 08 Ago. 2019.

LEUNG, J. Y.-T.; LI, H.; PINEDO, M. Order scheduling models: An overview. In: KENDALL, G. *et al.* (Ed.). **Multidisciplinary Scheduling: Theory and Applications**. [United States]: Springer US, 2005a. p. 37–53. ISBN 978-0-387-27744-8. Disponível em: [https://link.springer.com/chapter/10.1007/0-387-27744-7\\_3](https://link.springer.com/chapter/10.1007/0-387-27744-7_3). Acesso em: 08 Ago. 2019.

LEUNG, J. Y. T.; LI, H.; PINEDO, M. Order scheduling in an environment with dedicated resources in parallel. **Journal of Scheduling**, [United States], v. 8, n. 5, p. 355–386, Oct 2005b. Disponível em: <https://doi.org/10.1007/s10951-005-2860-x>. Acesso em: 08 Ago. 2019.

LEUNG, J. Y.-T.; LI, H.; PINEDO, M. Scheduling orders for multiple product types with due date related objectives. **European Journal of Operational Research**, [Netherlands], v. 168, n. 2, p. 370 – 389, 2006. ISSN 0377-2217. Feature Cluster on Mathematical Finance and Risk Management. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0377221704003005>. Acesso em: 24 Ago. 2019.

LEUNG, J. Y.-T. *et al.* Open shops with jobs overlap—revisited. **European Journal of Operational Research**, [Netherlands], v. 163, n. 2, p. 569 – 571, 2005. ISSN 0377-2217. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0377221703009214>. Acesso em: 08 Jan. 2020.

MAGAZINE, M. J.; JULIEN, F. M. Scheduling customer orders-an alternative production scheduling approach. **Journal of Manufacturing and Operations Management**, [Netherlands], v. 3, p. 177–199, 1990. Acesso em: 15 Jan. 2021.

PRATA, B. de A.; RODRIGUES, C. D.; FRAMINAN, J. M. Customer order scheduling problem to minimize makespan with sequence-dependent setup times. **Computers & Industrial Engineering**, [United States], v. 151, p. 106962, 2021. ISSN 0360-8352. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0360835220306355>. Acesso em: 27 Maio 2021.

ROEMER, T. A. A note on the complexity of the concurrent open shop problem. **Journal of Scheduling**, [United States], v. 9, n. 4, p. 389–396, Aug 2006. ISSN 1099-1425. Disponível em: <https://doi.org/10.1007/s10951-006-7042-y>. Acesso em: 08 Ago. 2019.

WAGNEUR, E.; SRISKANDARAJAH, C. Openshops with jobs overlap. **European Journal of Operational Research**, [Netherlands], v. 71, n. 3, p. 366 – 378, 1993. ISSN 0377-2217. Disponível em: <http://www.sciencedirect.com/science/article/pii/037722179390347P>. Acesso em: 08 Ago. 2019.

WU, C.-C. *et al.* A robust customer order scheduling problem along with scenario-dependent component processing times and due dates. **Journal of Manufacturing Systems**, [United States], v. 58, p. 291–305, 2021. ISSN 0278-6125. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0278612520302272>. Acesso em: 06 Jan. 2021.