



UNIVERSIDADE ESTADUAL PAULISTA
“JÚLIO DE MESQUITA FILHO”
Câmpus de São José do Rio Preto

Nícolas Samuel Assis

Um estudo da natureza biobjetiva do problema de
sequenciamento *just-in-time flow shop*

São José do Rio Preto

2024

Nícolas Samuel Assis

Um estudo da natureza biobjetiva do problema de
sequenciamento *just-in-time flow shop*

Tese apresentada à Universidade Estadual Paulista
(UNESP), Instituto de Biociências, Letras e
Ciências Exatas, São José do Rio Preto, para
obtenção do título de Doutor em Matemática.

Orientadora: Profa. Dra. Socorro Rangel

Coorientador: Prof. Dr. Hélio Yochihiro Fuchigami

São José do Rio Preto

2024

A848e

Assis, Nicolas Samuel

Um estudo da natureza biobjetiva do problema de sequenciamento
just-in-time flow shop / Nicolas Samuel Assis. -- São José do Rio Preto, 2025
121 p. : il., tabs.

Tese (doutorado) - Universidade Estadual Paulista (UNESP), Instituto de
Biociências Letras e Ciências Exatas, São José do Rio Preto

Orientadora: Maria do Socorro Nogueira Rangel

Coorientador: Hélio Yochihiro Fuchigami

1. Otimização multiobjetivo. 2. Planejamento da produção. 3.
Adiantamento total e atraso total. 4. Métodos exatos. 5. Decomposição de
Benders. I. Título.

NÍCOLAS SAMUEL ASSIS

**UM ESTUDO DA NATUREZA BIOBJETIVA DO PROBLEMA DE
SEQUENCIAMENTO *JUST-IN-TIME FLOW SHOP***

Tese apresentada à Universidade Estadual Paulista (UNESP), Instituto de Biociências, Letras e Ciências Exatas, São José do Rio Preto, para obtenção do título de Doutor em Matemática.

Área de Concentração: Matemática

Data da defesa: 12 / 12 / 2024

Banca Examinadora:

Profa. Dra. Socorro Rangel
UNESP – Instituto de Biociências, Letras e Ciências Exatas - Campus de São José do Rio Preto

Prof. Dr. Reinaldo Morabito
UFSCar – São Carlos

Prof. Dr. Bruno de Athayde Prata
UFC - Fortaleza

Profa. Dra. Daniela Renata Cantane
UNESP - Botucatu

Prof. Dr. Thiago Alves de Queiroz
UFCat - Catalão

Agradecimentos

Agradeço, primeiramente, aos meus pais e familiares por todo o carinho, apoio, compreensão e orgulho que sempre demonstraram.

Agradeço imensamente à minha esposa, Drielly, por todo o companheirismo, paciência e carinho nos momentos mais difíceis. Sou grato por toda a sua contribuição e ajuda no desenvolvimento da pesquisa, pelas discussões sempre tão valiosas e por compartilhar seus conhecimentos comigo.

Agradeço profundamente à minha orientadora, Profa. Dra. Socorro Rangel, pelos grandes ensinamentos que vão além da universidade. Agradeço por ter aceitado e continuado a trabalhar comigo, mesmo em momentos difíceis para nós dois. Sou imensamente grato por sempre manter o alto nível de orientação.

Agradeço muito ao meu coorientador, Prof. Dr. Hélio Yochihiro Fuchigami, por fazer parte dessa jornada. Sou grato pelos ensinamentos, pela parceria e por não perder o foco nos objetivos da pesquisa. Agradeço também por evidenciar todas as contribuições do trabalho, mesmo quando eu não conseguia enxergá-las.

Agradeço ao Prof. Dr. Reinaldo Morabito Neto, ao Prof. Dr. Bruno de Athayde Prata, à Profa. Dra. Daniela Renata Cantane e ao Prof. Dr. Thiago Alves de Queiroz por aceitarem o convite para compor a banca de defesa e pelos comentários que contribuíram para a redação da tese e para as propostas de trabalhos futuros.

Agradeço aos meus colegas de trabalho do SESI 298 e da Miró por me incentivarem a não desistir e a seguir firme nesta empreitada. Sou também grato aos meus amigos pelo companheirismo e pelos momentos de lazer, tão importantes para que eu concluísse esta jornada.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001, à qual agradeço.

RESUMO

Nessa tese é tratado o Problema de Sequenciamento em *Just-In-Time Flow shop* (JIT-FSP) com as medidas adiantamento total e atraso total. A natureza biobjetiva dessas medidas é deixada em segundo plano pela literatura ao tratar o problema de forma monoobjetiva, mostrando uma lacuna ainda inexplorada, sendo contribuir para fechar essa lacuna o principal objetivo dessa tese. Conceitos básicos sobre otimização multiobjetivo são apresentados, assim como, os métodos de solução exatos: ε -restrito, ε -restrito aumentando, caixa balanceada e *branch-and-bound*. Ferramentas matemáticas são propostas para analisar a solução da metodologia monoobjetivo dentro de uma perspectiva biobjetivo. Os métodos exatos são pouco explorados para resolver o JIT-FSP devido ao porte de problemas reais. Para contornar a dificuldade com o porte dos problemas reais, aplica-se o método de decomposição de Benders e são propostas 5 variações do método. Resultados do estudo computacional, 1100 execuções com tempos máximos de 3600 segundos em cada execução, mostraram que a metodologia utilizada pela literatura para resolver o JIT-FSP pode obter soluções indesejáveis para alguns cenários industriais. Além disso, evidencia-se as vantagens de utilizar a abordagem biobjetivo, chamando a atenção para a necessidade de utilizar métodos capazes de encontrar toda a fronteira de Pareto principalmente com o aumento do porte das instâncias. As variações do método de Benders mostraram um aumento no desempenho em relação ao método clássico, principalmente quando é utilizado uma busca em árvore única.

Palavras-chave: otimização multiobjetivo; planejamento da produção; adiantamento total e atraso total; métodos exatos; decomposição de Benders.

ABSTRACT

This thesis deals with the Just-In-Time Flow Shop Scheduling Problem (JIT-FSP) with the measures total earliness and total tardiness. The bi-objective nature of these measures is left in the background by the literature when treating the problem in a mono-objective way, showing an gap, and contributing to closing this gap is the main objective of this thesis. Basic concepts of multi-objective optimization are presented, as well as exact solution methods: ε -constrained, augmented ε -constrained, balanced box and branch-and-bound. Mathematical tools are proposed to analyze the solution of the mono-objective methodology within a bi-objective perspective. Exact methods have been little explored to solve the JIT-FSP due to the size of real problems. To overcome the difficulty with the size of real problems, the Benders decomposition method is applied and 5 variations of the method are proposed. The results of the computational study, 1100 runs with a maximum time of 3600 seconds for each run, showed that the methodology used in the literature to solve the FSP-JIT can obtain undesirable solutions for some industrial scenarios. It also shows the advantages of using the bi-objective approach, drawing attention to the need to use methods capable of finding the full Pareto frontier, especially as the size of the instances increases. Variations of the Benders method showed an increase in performance compared to the classic method, especially when a single tree search is used.

Keywords: multiobjective optimization; production planning; total earliness and total tardiness; exact methods; benders decomposition.

Lista de figuras

Figura 1 – Regiões de dominância no espaço critério	18
Figura 2 – Fronteiras de Pareto de um problema biobjetivo	19
Figura 3 – Legenda dos pontos no espaço critério do Problema 1	20
Figura 4 – Espaço critério do Problema 1	20
Figura 5 – Região dominada pelo conjunto limitante inferior $\mathcal{L}$	27
Figura 6 – Conjunto limitante superior $\mathcal{U}$	27
Figura 7 – Dominância parcial de conjuntos limitantes (condição 3 para sondagem biobjetivo)	28
Figura 8 – Espaço critério associado à resolução do nó raiz do Problema 1	30
Figura 9 – Espaço critério associado à resolução do segundo nó da fila <i>branch-and-bound</i>	31
Figura 10 – Espaço critério associado à resolução do oitavo nó da fila <i>branch-and-bound</i>	32
Figura 11 – Ramificação do espaço critério	33
Figura 12 – Hipervolume e quantidade de pontos da AFP_1 e AFP_2	34
Figura 13 – Hipervolume e diversidade da AFP_3 e AFP_4	35
Figura 14 – Hipervolume da fronteira de Pareto	36
Figura 15 – Regiões de equilíbrio baseado em distância e μ como referência	38
Figura 16 – Aumento percentual da soma ponderada em relação à solução monoobjetivo	38
Figura 17 – Sequenciamento de 5 tarefas em 3 máquinas	41
Figura 18 – Sequenciamento e valor das medidas E , T , F e C_{max}	42
Figura 19 – Sequenciamento com tempos ociosos	49
Figura 20 – Diagrama da decomposição de Benders aplicado ao ε -restrito aumentado	61
Figura 21 – Diagrama da variação B_{seq}	66
Figura 22 – AFP's de configurações do BIOBAB da instância $(20, 10)_1$	71
Figura 23 – AFP's dos métodos do sistema BIOBAB e dos métodos $GRBmult_p$ e $GRBmult_l$ na instância $(20, 10)_1$	75
Figura 24 – Métrica hipervolume da instância 1 para as classes (n, m)	76
Figura 25 – Métrica hipervolume das instâncias 2, 3 e 4 para as classes $(20, m)$	79
Figura 26 – Métrica cardinalidade das instâncias 2, 3 e 4 para as classes $(20, m)$	80
Figura 27 – AFP's dos métodos do sistema BIOBAB e dos métodos $GRBmult_p$ e $GRBmult_l$ na instância $(20, 5)_2$	81
Figura 28 – Métrica hipervolume das instâncias 2, 3 e 4 para as classes $(50, m)$ e $(100, m)$	83
Figura 29 – Métrica cardinalidade das instâncias 2, 3 e 4 para as classes $(50, m)$ e $(100, m)$	84

Figura 30 – AFP's do sistema BIOBAB e dos métodos $GRBmult_p$ e $GRBmult_l$ na instância $(50, 10)_4$	85
Figura 31 – Gráficos de caixa do número de pontos na fronteira de Pareto por porte do problema	87
Figura 32 – Gráficos de colunas dos valores das ferramentas de localização por porte do problema	89
Figura 33 – Gráficos de colunas dos aumentos mínimo, médios e máximos do valor objetivo por porte do problema	90
Figura 34 – Gráficos de caixas do percentual de pontos não suportados na fronteira de Pareto por cenário	91
Figura 35 – Gráficos de caixas da cardinalidade das AFP's dos métodos BIOBAB, caixa balanceada, ε -restrito aumentado e GRBmult nas classes $0,2_0,6_5_3$ e $0,4_1,2_5_3$	94
Figura 36 – Gráficos de caixas do <i>hiper ratio</i> das AFP's dos métodos BIOBAB, caixa balanceada, ε -restrito aumentado e GRBmult nas classes $0,2_0,6_5_3$ e $0,4_1,2_5_3$	94
Figura 37 – Gráficos de <i>Gantt</i> de 4 soluções da instância $0,2_1,2_10_3_1$	99
Figura 38 – Gráficos de caixas da cardinalidade das AFP's para cada variação de ϵ	100
Figura 39 – <i>Trade-off</i> entre cardinalidade e <i>hiper ratio</i> para cada classe de instâncias	102
Figura 40 – Gráficos de caixas do tempo de resolução para cada variação de ϵ	103
Figura 41 – Gráficos de caixas para o tempo de execução por porte das instâncias	104
Figura 42 – Gráfico de colunas da cardinalidade das AFP's para instâncias com 10 tarefas e 7 máquinas	105
Figura 43 – Gráficos de caixas do tempo de execução do problema mestre para cada porte de instância	106
Figura 44 – Gráficos de caixas do número de subproblemas resolvidos por porte de instância	107
Figura 45 – Diagrama de módulos do sistema BIOBAB	120

Lista de tabelas

Tabela 1 – Características desejáveis nos métodos de escalarização	21
Tabela 2 – Autores que resolveram o FSP com objetivo JIT	45
Tabela 3 – Dimensão dos modelos (3.10)-(3.18) e (3.28)-(3.36)	51
Tabela 4 – Autores que aplicam o método de decomposição de Benders nos problemas de sequenciamento	59
Tabela 5 – Variações do método de decomposição de Benders	67
Tabela 6 – Objetivos dos estudos computacionais	69
Tabela 7 – Valores das métricas de cardinalidade e de hipervolume de configurações do BIOBAB	70
Tabela 8 – Efeito do tipo de busca na árvore de busca do BIOBAB	72
Tabela 9 – Valores das métricas de cardinalidade e de hipervolume dos métodos do sistema BIOBAB e dos métodos $GRBmult_p$ e $GRBmult_l$	74
Tabela 10 – Valores das métricas de cardinalidade e de hipervolume das instâncias 2, 3 e 4 para as classes $(20, m)$	78
Tabela 11 – Valores das métricas de cardinalidade e de hipervolume das instâncias 2, 3 e 4 para as classes $(50, m)$ e $(100, m)$	82
Tabela 12 – Esforço computacional das abordagens multiobjetivo e monoobjetivo (média de 10 instâncias)	86
Tabela 13 – Valores das ferramentas de localização e aumento de custo do ponto monoobjetivo (média de 10 instâncias)	88
Tabela 14 – Número de pontos suportados e não suportados na fronteira de Pareto (média de cada classe de instâncias)	91
Tabela 15 – Valores do tempo de execução e das métricas de cardinalidade e de hipervolume dos métodos BIOBAB, caixa balanceada, ε -restrito aumentado e GRBmult	93
Tabela 16 – Métricas de cardinalidade e de <i>hiper ratio</i> para variações de ϵ (média das classes)	97

Lista de siglas

FSP	<i>Flowshop Scheduling Problem</i>
JIT	<i>Just-In-Time</i>
JIT-FSP	<i>Just-In-Time Flowshop Scheduling Problem</i>
BIOBAB	<i>Bi-objective Branch-and-Bound</i>
POM	Problema de Otimização Multiobjetivo
POIB	Problema de Otimização Inteiro Biobjetivo
LB	<i>Lower Bound</i>
AFP	Aproximação de Fronteira de Pareto
MIP	<i>Mixed Integer optimization Problem</i>
B&B	<i>Branch-and-Bound</i>
PMB	Problema Mestre de Benders
PMBB	Problema Mestre de Benders Biobjetivo
LBBD	<i>Logic-Based Benders Decomposition</i>
PM_B	Problema Mestre de Benders (dentro do algoritmo do método ε -restrito aumentado)
SP_B	Subproblema de Benders
SP_ε	Subproblema do método ε -restrito aumentado
B_{cl}	Decomposição de Benders Clássico
B_{seq}	Decomposição de Benders com cortes pré-gerados
B_{RL}	Decomposição de Benders com Relaxação Linear
B_{seq}^{RL}	Decomposição de Benders com cortes pré-gerados e Relaxação Linear
BBC_{cl}	<i>Branch-and-Benders-cut</i> Clássico
BBC_{seq}	<i>Branch-and-Benders-cut</i> com cortes pré-gerados
GRB_{mult}	GUROBI Multiobjetivo
GRB_{mono}	GUROBI Monoobjetivo

Sumário

1	INTRODUÇÃO	13
2	O PROBLEMA DE OTIMIZAÇÃO LINEAR INTEIRO MULTIOBJETIVO	15
2.1	Métodos de solução	20
2.1.1	Métodos de escalarização	21
2.1.2	<i>Branch-and-bound</i> multiobjetivo	25
2.2	Métricas de desempenho	33
2.3	Ferramentas matemáticas propostas para análise de pontos na fronteira de Pareto	37
3	O PROBLEMA DE SEQUENCIAMENTO EM <i>FLOW SHOP</i>	40
3.1	Revisão de literatura sobre os problemas de <i>flow shop</i> com objetivos <i>just-in-time</i>	42
3.2	Modelagem matemática do problema de <i>flow shop</i>	45
4	MÉTODO DE DECOMPOSIÇÃO DE BENDERS APLICADO NO PROBLEMA DE SEQUENCIAMENTO	52
4.1	Aplicações de Benders nos problemas de sequenciamento	57
4.2	Variações do método de decomposição de Benders para o JIT-FSP	60
5	RESULTADOS COMPUTACIONAIS	68
5.1	Estudo 1 - O sistema BIOBAB	70
5.2	Estudo 2 - Vantagens do tratamento biobjetivo comparado ao monoobjetivo do JIT-FSP	85
5.3	Estudo 3 - Comparação dos métodos biobjetivos	91
5.4	Estudo 4 - Variações do método de decomposição de Benders	103
6	CONSIDERAÇÕES FINAIS	108
	REFERÊNCIAS	111
	APÊNDICE A - Decomposição de Benders	118
	APÊNDICE B - O sistema BIOBAB	120

1 Introdução

Em algumas indústrias, para fazer o planejamento da produção, é necessário determinar uma sequência de tarefas que precisam ser executadas, na mesma ordem, em diferentes máquinas para atender objetivos distintos. Esse sequenciamento de tarefas é uma solução de um problema conhecido na literatura como Problema de Sequenciamento em *Flow shop* (FSP, sigla em inglês). Um dos objetivos a serem otimizados no FSP é o de concluir a tarefa no prazo (*just-in-time*), que pode ser modelado através dos objetivos de pontualidade: adiantamento total e atraso total. Para diminuir o adiantamento total, uma possibilidade seria concluir as tarefas no prazo, o que é difícil ser feito para todas as tarefas. Outra possibilidade seria atrasá-las, aumentando o atraso total. Para diminuir o atraso total, de forma análoga, ou conclui-se as tarefas no prazo ou adianta-se a conclusão delas. Essa dinâmica de melhorar o adiantamento total resultar, quase inevitavelmente, em prejudicar o atraso total, e vice-versa, é constantemente deixada em segundo plano pela literatura ao tratar ambos os objetivos simultaneamente em um único objetivo.

A globalização econômica e a intensa competição de mercado têm incentivado as empresas a aprimorarem suas estruturas de produção para diminuir os custos, aumentar a qualidade dos processos e aperfeiçoar a satisfação de seus clientes ([1]). No FSP, os objetivos de adiantamento total e atraso total estão, por vezes, relacionados à satisfação do cliente. Esse fato dificulta a atribuição de valores monetários a esses objetivos, fazendo com que a soma deles não reflita as necessidades práticas do problema, resultando em soluções pouco realistas ([2, 3]). Outra dificuldade em combinar objetivos conflitantes em um único critério está na atribuição de pesos aos objetivos, o que pode exigir diversas iterações com o tomador de decisão até se alcançar os resultados desejados ([4]). Uma forma de abordar esse problema, evidenciando o trade-off entre o adiantamento total e o atraso total, é por meio da otimização multiobjetivo, que fornece um leque de soluções para que o tomador de decisão escolha a que mais se adeque ao seu cenário industrial.

Na resolução de um problema de otimização multiobjetivo alguns dos principais métodos utilizados são os métodos de escalarização que transformam uma função multiobjetivo em monoobjetivo e encontram um conjunto de soluções através de uma sequência de subproblemas monoobjetivos. A escolha de métodos multiobjetivos adequados é de grande importância, principalmente para problemas de otimização linear inteira, nesse caso, o uso de métodos alternativos como os de busca em árvore tornam-se atrativos. No entanto, métodos de busca em árvore por vezes possuem limitação para resolver problemas realísticos de grande porte. Um recurso utilizado na literatura para resolver problemas maiores são métodos baseados na decomposição de Benders.

Assim, essa tese tem como objetivo explorar a natureza biobjetivo do problema de

sequenciamento *just-in-time flow shop* (JIT-FSP). Além disso, é proposto o uso de métodos exatos para resolver o problema, metodologia pouco explorada na literatura, e a aplicação da decomposição de Benders. As principais contribuições são:

- Formulação do modelo de otimização linear biobjetivo para o JIT-FSP (Seção 3.2);
- Proposição das ferramentas matemáticas para analisar a solução obtida pela abordagem monoobjetivo em uma perspectiva biobjetivo (Seção 2.3)
- Uso do sistema (BIOBAB) proposto em [5] para resolver o JIT-FSP (Seções 5.1, 5.1 e 5.3);
- Implementação de uma relação de dominância da literatura para auxiliar o tomador de decisão na filtragem da fronteira de Pareto (Seção 5.3);
- Aplicação da decomposição de Benders em um método de escalarização para resolver o JIT-FSP, bem como cinco modificações de Benders (Seção 4.2);
- Estudo computacional e estatístico do impacto no desempenho de cinco modificações no método de Benders clássico (Seção 5.4);

Os capítulos dessa tese possuem a seguinte organização, no Capítulo 2 apresentam-se os conceitos básicos da otimização linear inteira multiobjetivo, métodos de solução e as ferramentas matemáticas propostas. No Capítulo 3 é realizado uma revisão do JIT-FSP e se propõe uma modelagem do problema. No Capítulo 4 aplica-se o método de decomposição de Benders no JIT-FSP, bem como, cinco variações no método. No Capítulo 5 realiza-se quatro estudos computacionais com os objetivos de: comparar métodos do sistema BIOBAB, demonstrar as vantagens de fazer uma abordagem biobjetiva do JIT-FSP e analisar os impactos no desempenho das variações do método de decomposição de Benders.

2 O problema de otimização linear inteiro multiobjetivo

A otimização multiobjetivo é uma área do conhecimento que tem crescido cada vez mais ([6]). Diferente da otimização monoobjetivo, que considera apenas um objetivo, a otimização multiobjetivo leva em consideração vários objetivos conflitantes para o problema. Um problema de otimização multiobjetivo (POM) pode ser representado conforme (2.1)-(2.2). Nesse problema, x são as variáveis de decisão e $\mathcal{X} \subset \mathbb{R}^n$ representa o conjunto factível do problema. A função objetivo f é uma função vetorial com imagem no espaço $\mathbb{R}^p$, com $p > 1$, no qual cada entrada dessa imagem representa o valor de um objetivo específico.

$$\text{Min } f(x) = (f_1(x), f_2(x), \dots, f_p(x)) \quad (2.1)$$

$$\text{S.a : } x \in \mathcal{X} \subset \mathbb{R}^n \quad (2.2)$$

A função objetivo f em (2.1)-(2.2) é o que caracteriza o problema como multiobjetivo. Na literatura esse problema multiobjetivo está inserido no contexto da análise de decisão com múltiplos critérios ([7, 8, 6]) pelo fato do decisor possuir diferentes critérios conflitantes para analisar o problema. Também apresenta algumas similaridades com a otimização vetorial ([9, 10]) pois a imagem de cada solução representa um vetor no espaço $\mathbb{R}^p$. Nessa tese utiliza-se as terminologias presentes na otimização multiobjetivo e a principal base teórica é o livro do Greco, Ehrgott e Figueira [6].

Considerando a função $f : \mathbb{R}^n \rightarrow \mathbb{R}^p$, se define os espaços do problema (2.1)-(2.2), Definições 1 e 2. Com essas definições, conclui-se que o conjunto factível do problema é subconjunto do espaço decisão e $\mathcal{Y} = f(\mathcal{X})$ é a imagem do conjunto factível, um subconjunto do espaço critério.

Definição 1 (Espaço decisão [6]). $\mathbb{R}^n$ representa o espaço decisão.

Definição 2 (Espaço critério [6]). $\mathbb{R}^p$ é chamado de espaço critério ou espaço objetivo.

Em um problema de minimização monoobjetivo, para saber se uma solução é melhor do que outra, basta verificar qual imagem possui o menor valor. Sendo o espaço critério um espaço p -dimensional, existem diferentes relações de ordem para comparar pontos. Na Definição 3 apresentam-se três notações necessárias para estabelecer as relações de ordem para comparar pontos multidimensionais.

Definição 3 ([6]). Sejam $y^1, y^2 \in \mathbb{R}^p$, assim:

1. $y^1 \leq y^2$ se $y_k^1 \leq y_k^2, k = 1, \dots, p$;

2. $y^1 \leq y^2$ se $y_k^1 \leq y_k^2, k = 1, \dots, p$ e $y^1 \neq y^2$;
3. $y^1 < y^2$ se $y_k^1 < y_k^2, k = 1, \dots, p$.

Através dessas notações define-se a relação de Pareto (e.g. [11, 12, 6]).

Definição 4 (Relação de Pareto [6]). *Considere que y^1 é preferível (ou indiferente) do que y^2 segundo a relação de Pareto se, e somente se, $y^1 \leq y^2$.*

A relação de Pareto (Definição 4) é uma relação de ordem parcial. Essa característica possibilita que dois pontos no espaço critério não se relacionem, ou seja, sendo $y^1, y^2 \in \mathbb{R}^p$ não ocorre $y^1 \leq y^2$ nem $y^2 \leq y^1$. Esse fato permite que um POM tenha um conjunto de soluções chamadas de Pareto ótimas ou eficientes (Definição 5) com diferentes imagens. Mais duas variações de soluções eficientes são definidas a seguir.

Definição 5 (Solução eficiente [6]). *Uma solução $x^* \in \mathcal{X}$ é eficiente se não existir $x \in \mathcal{X}$ tal que $f(x) \leq f(x^*)$. O conjunto de todas as soluções eficientes é representado por $\mathcal{X}_E$.*

Definição 6 (Fracamente eficiente e estritamente eficiente [6]). *Considere $\mathcal{X} \subset \mathbb{R}^n$.*

1. $\hat{x} \in \mathcal{X}$ é fracamente eficiente se não existir $x \in \mathcal{X}$ tal que $f(x) < f(\hat{x})$;
2. $\hat{x} \in \mathcal{X}$ é estritamente eficiente se não existir $x \in \mathcal{X}, \hat{x} \neq x$, tal que $f(x) \leq f(\hat{x})$.

Utilizando a relação de Pareto, é dito que se $x^1, x^2 \in \mathcal{X}$ e $f(x^1) \leq f(x^2)$, então x^1 domina x^2 e $f(x^1)$ domina $f(x^2)$. Com isso, a imagem das soluções eficientes são chamadas de pontos não dominados (Definição 7) e seu conjunto é chamado de fronteira de Pareto (Definição 8).

Definição 7 (Ponto não dominado [12]). *Um ponto $f(x^*) \in \mathcal{Y}$ é não dominado se x^* é uma solução eficiente.*

Definição 8 (Fronteira de Pareto). *O conjunto de pontos não dominados $f(\mathcal{X}_E) = \mathcal{Y}_N$ é chamado de fronteira de Pareto.*

Apesar do problema (2.1)-(2.2) possuir p objetivos, isso não garante que ele seja verdadeiramente multiobjetivo. Essa verificação é feita através do ponto ideal (Definição 9). Se o ponto ideal for imagem de alguma solução factível, então não existe conflito entre os objetivos, ou seja, a fronteira de Pareto é composta por apenas um ponto.

Definição 9. *O ponto $y^I = (y_1^I, y_2^I, \dots, y_p^I)$ dado por*

$$y_k^I = \text{Min}\{f_k(x) \mid x \in \mathcal{X} \subset \mathbb{R}^n\}, \quad k = 1, \dots, p$$

é chamado de ponto ideal.

Determinar toda a fronteira de Pareto não é uma tarefa simples, e para o tomador de decisão, analisar a fronteira completa e decidir qual ponto implementar pode ser tão difícil quanto determiná-la ([13]). Além disso, principalmente nas pesquisas que utilizam algoritmos baseados em população (*e.g.* [14]), existe uma dificuldade em garantir a diversidade da fronteira de Pareto e a convergência dos algoritmos. Para lidar com essas problemáticas, foram desenvolvidas novas relações de dominância, uma delas é a ϵ -dominância proposta por Laumanns et al. [15] (Definição 10).

Definição 10 (ϵ -dominância [15]). *Sejam $y^1, y^2 \in \mathbb{R}^p$ e $\epsilon > 0$. É dito que y^1 ϵ -domina y^2 se 1 ou 2 ocorrem.*

1. *Aditiva:* $y_k^1 \leq y_k^2 + \epsilon$, $k = 1, \dots, p$;
2. *Multiplicativa:* $y_k^1 \leq (1 + \epsilon) \cdot y_k^2$, $k = 1, \dots, p$.

A ϵ -dominância permite uma tolerância $\epsilon > 0$ para que um ponto domine outro, por exemplo, sejam $y^1 = (2, 3)$ e $y^2 = (5, 2)$. Esses pontos não se relacionam segundo Pareto, mas pela ϵ -dominância aditiva com $\epsilon = 1$, y^1 ϵ -domina y^2 pois $y_1^1 = 2 \leq 5 + 1 = y_1^2 + \epsilon$ e $y_2^1 = 3 \leq 2 + 1 = y_2^2 + \epsilon$.

Nessa tese será utilizado apenas a ϵ -dominância aditiva, que é referente ao item 1 da Definição 10, pela facilidade de implementação. Por meio da relação de dominância segundo Pareto e da ϵ -dominância é estabelecida uma região de dominância para cada ponto no espaço critério (Definição 11). Na Figura 1 ilustra-se a região $\mathcal{R}$ (Figura 1a) e a região $\mathcal{R}_\epsilon$ (Figura 1b) para um problema biobjetivo. É notório que $\mathcal{R}_\epsilon \supset \mathcal{R}$ por permitir uma tolerância ϵ .

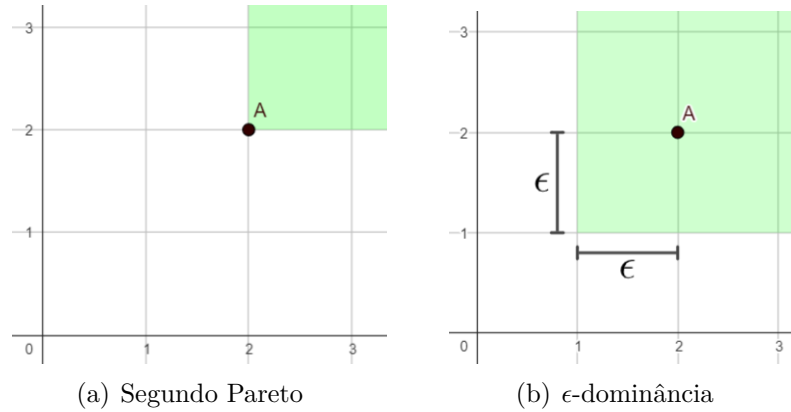
Definição 11 (Região de dominância). *Seja $y^1 = f(x^1)$. A região de dominância do ponto y^1 é dada por:*

1. *(segundo Pareto)* $\mathcal{R} = \{y^2 \in \mathbb{R}^p \mid y^1 \leq y^2\}$;
2. *(ϵ -dominância)* $\mathcal{R}_\epsilon = \{y^2 \in \mathbb{R}^p \mid y^1 \epsilon\text{-domina } y^2\}$.

Através da ϵ -dominância define-se uma fronteira de Pareto ϵ -aproximada (Definição 12) e uma fronteira ϵ -Pareto (Definição 13). Na Figura 2 estão ilustradas um exemplo das três fronteiras definidas, a fronteira de Pareto (Figura 2a) possui uma quantidade maior de pontos, a fronteira de Pareto ϵ -aproximada (Figura 2b) possui menos pontos e pode conter pontos que não pertencem a fronteira de Pareto e a fronteira ϵ -Pareto (Figura 2c) também possui menos pontos, mas somente pontos da fronteira de Pareto.

Definição 12 (Fronteira de Pareto ϵ -aproximada). *Sejam $\mathcal{F}_\epsilon \subset \mathbb{R}^p$, $f(\mathcal{X}) = \mathcal{Y}$ e $\epsilon > 0$. É dito que $\mathcal{F}_\epsilon$ é uma fronteira de Pareto ϵ -aproximada, se para qualquer $y \in \mathcal{Y}$ existe $y' \in \mathcal{F}_\epsilon$ tal que y' ϵ -domina y .*

Figura 1 – Regiões de dominância no espaço critério



Fonte: Elaborado pelo autor.

Definição 13 (Fronteira ϵ -Pareto). $\mathcal{F}_\epsilon^* \subset \mathbb{R}^p$ é uma fronteira ϵ -Pareto se ele é uma fronteira de Pareto ϵ -aproximada e contém apenas pontos da fronteira de Pareto.

Outra relação de ordem importante é a relação lexicográfica. Considerando uma ordem de prioridade $Ord = \{o_1, o_2, \dots, o_p\}$, diz-se que y^1 é preferível (ou indiferente) do que y^2 segundo a relação lexicográfica se, e somente se, $y^1 = y^2$ ou para algum $k, 1 \leq k \leq p$, $y_i^1 = y_i^2, i = o_1, o_2, \dots, o_{k-1}$ e $y_{o_k}^1 < y_{o_k}^2$. Diferente da relação de Pareto, a relação lexicográfica estabelece uma relação de ordem total.

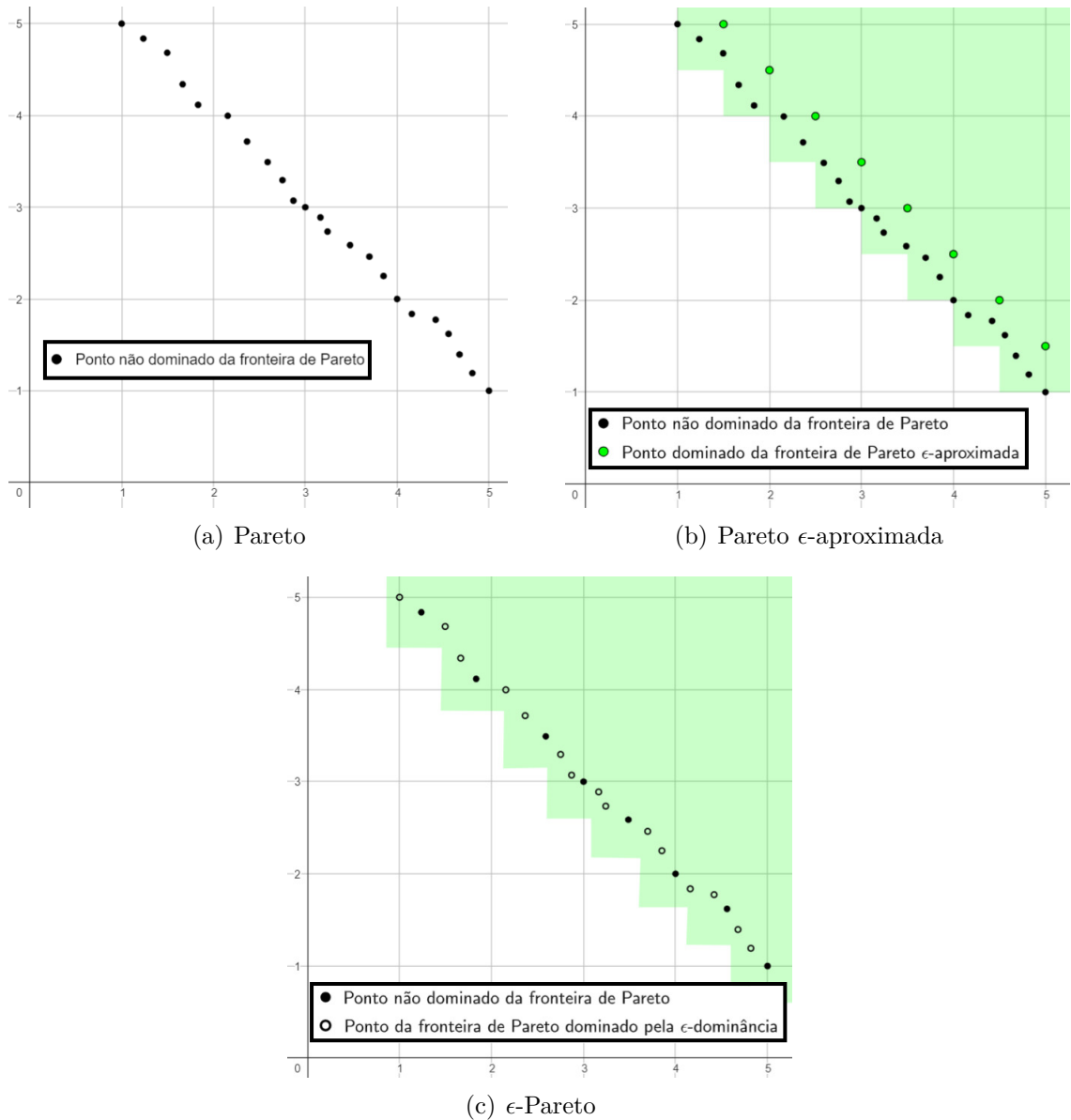
Uma maneira de resolver um POM utilizando a otimização lexicográfica é através de uma sequência de resoluções de subproblemas monoobjetivos dada pelo Algoritmo 1. Para cada $i = 1, \dots, p$ o subproblema monoobjetivo $P_i = \text{Min}\{f_{o_i}(x) \mid x \in \mathcal{X}_i\}$ é resolvido. Se existir apenas uma solução ótima (x^*) para o subproblema P_i , então o método encerra e a solução ótima (x^*) será a saída do algoritmo, pois qualquer outro $x \in \mathcal{X}$ provocaria $f_{o_i}(x) > f_{o_i}^* = f_{o_i}(x^*)$. Caso contrário, o objetivo $f_{o_{i+1}}(x)$ é minimizado sem aumentar o valor do objetivo $f_{o_i}(x)$, ou seja, restrito a $f_{o_i}(x) \leq f_{o_i}^*$.

Algoritmo 1: Algoritmo $lexmin(f_{o_1}, f_{o_2}, \dots, f_{o_p})$

Entrada: $f, \mathcal{X}, Ord = \{o_1, o_2, \dots, o_p\}$ **Saída:** x^* solução ótima lexicográfica

- 1 $\mathcal{X}_1 = \mathcal{X}$;
 - 2 **para** $i = 1, \dots, p$ **faça**
 - 3 Resolva $P_i = \text{Min}\{f_{o_i}(x) \mid x \in \mathcal{X}_i\}$ e obtenha $f_{o_i}^*$ (valor ótimo de P_i) e $\bar{x}_i$ (solução de P_i) ;
 - 4 **se** $\bar{x}_i$ *é única* **então**
 - 5 **PARE**, $x^* = \bar{x}_i$ (solução ótima lexicográfica é a solução atual)
 - 6 **senão**
 - 7 $\mathcal{X}_{i+1} = \mathcal{X}_i \cup \{f_{o_i}(x) \leq f_{o_i}^*\}$
 - 8 **fim**
 - 9 **fim**
-

Figura 2 – Fronteiras de Pareto de um problema biobjetivo



Fonte: Elaborado pelo autor.

A otimização lexicográfica é muito utilizada em etapas de métodos de solução como a caixa balanceada, ϵ -restrito, soma ponderada e *branch-and-bound* biobjetivo (e.g. [16, 5]). A solução do problema lexicográfico também é uma solução eficiente segundo Pareto que possui uma característica da fronteira (Definição 14). Na Seção 2.1 são apresentados métodos de solução para a relação de Pareto, que é o foco dessa tese.

Definição 14 (Pontos extremos). *A solução do Algoritmo 1 é uma solução eficiente e sua imagem é chamada de ponto extremo ou ponto lexicográfico. Os pontos extremos de um problema definem a região que contém a fronteira de Pareto.*

2.1 Métodos de solução

Para facilitar a compreensão de conceitos e dos métodos de solução discutidos nessa seção, considere o

$$\text{Problema 1} = \text{Min}\{(f_1(x), f_2(x)) \mid x \in \mathcal{X} \subset \mathbb{Z}^n\}$$

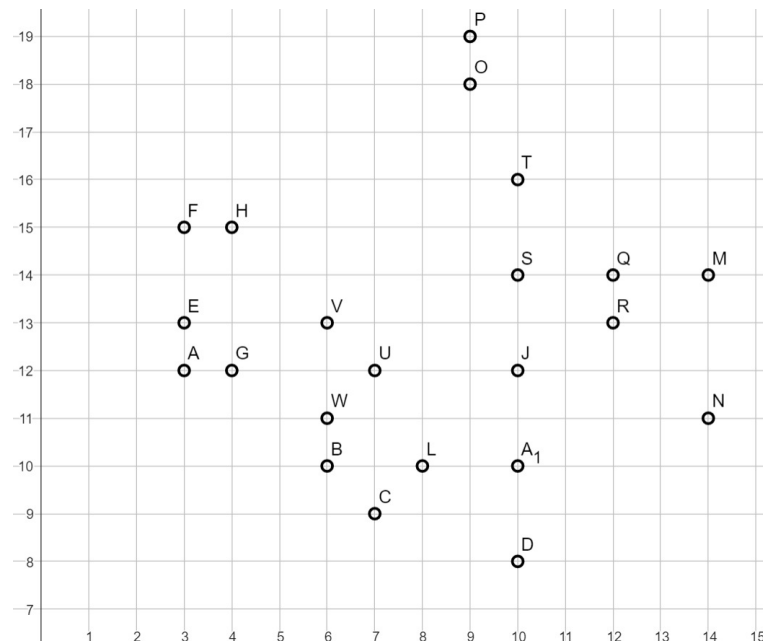
um problema de otimização inteira biobjetivo (POIB), com espaço critério $\mathcal{Y} \subset \mathbb{Z}^2$ e fronteira de Pareto $\mathcal{Y}_N = \{A, B, C, D\}$ exibido na Figura 4. Na Figura 3 apresenta-se uma legenda que será utilizada em figuras seguintes.

Figura 3 – Legenda dos pontos no espaço critério do Problema 1

- Ponto não encontrado
- Ponto encontrado
- Limitante superior
- Limitante inferior

Fonte: Elaborado pelo autor.

Figura 4 – Espaço critério do Problema 1



Fonte: Elaborado pelo autor.

Existem diferentes métodos para encontrar a fronteira de Pareto de um POIB: escalarização, *branch-and-bound*, heurísticos e metaheurísticos. No contexto de métodos de solução para um POIB, a Definição 15 auxilia na escolha do método a ser utilizado.

Definição 15 (Pontos suportados e não suportados). *Sejam $x \in \mathcal{X}_E$ e $y = f(x)$, assim:*

1. *x é uma solução eficiente suportada se y estiver na parte convexa da fronteira de Pareto. Nesse caso, y é um ponto não dominado suportado;*
2. *Caso y não esteja na parte convexa da fronteira, então ele é um ponto não dominado não suportado e x será uma solução eficiente não suportada;*
3. *Os conjuntos das soluções eficientes suportadas e não suportadas são denotados por $\mathcal{X}_{sE}$ e $\mathcal{X}_{nE}$. O conjunto das suas imagens são $\mathcal{Y}_{sN}$ e $\mathcal{Y}_{nN}$.*

Na Figura 4 o ponto B é um ponto não dominado não suportado. Ao traçar o segmento de reta $\overline{AC}$ nota-se que o ponto B fica acima do segmento. Nos problemas de otimização inteira biobjetivos o espaço decisão não é convexo e por isso, em geral, o espaço critério e a fronteira de Pareto não serão convexas, evidenciando a importância de distinguir esses pontos.

2.1.1 Métodos de escalarização

Alguns dos métodos mais populares para encontrar a fronteira de Pareto são os métodos de escalarização. Esses métodos consistem em transformar uma função objetivo vetorial em uma função objetivo escalar, e assim, encontrar os pontos da fronteira através de uma sequência de solução de subproblemas monoobjetivos. Existem diversas formas de se fazer esta transformação: considerar apenas um objetivo e adicionar os outros em restrições, ponderar os objetivos em uma soma, etc. Com essas possibilidades, existem algumas características desejáveis nos métodos de escalarização que são apresentadas na Tabela 1.

Tabela 1 – Características desejáveis nos métodos de escalarização

Característica 1	Se (2.1)-(2.2) com apenas um objetivo não é NP-difícil, então o problema escalar também não é.
Característica 2	Toda solução ótima do problema escalar é uma solução eficiente de (2.1)-(2.2).
Característica 3	Todas as soluções eficientes são obtidas pelo problema escalar com parâmetros adequados.
Característica 4	Se o problema (2.1)-(2.2) é linear, então o problema escalar é linear.

Introduzido por Haimes et al. [17] em 1971, o método ε -restrito é um dos métodos de escalarização mais conhecidos ([12]). Nele, apenas um dos p objetivos em (2.1)-(2.2) é minimizado enquanto os outros são adicionados ao conjunto de restrições do problema. A forma geral do problema escalar é exibido em (2.3)-(2.5).

$$\text{Min } f_k(x) \quad (2.3)$$

$$\text{S.a : } f_j(x) \leq \varepsilon_j, \quad j = 1, \dots, p, j \neq k \quad (2.4)$$

$$x \in \mathcal{X} \quad (2.5)$$

Dentre as características desejadas nos métodos escalares, o ε -restrito satisfaz 2, 3 e 4.

Para problemas biobjetivos, o método ε -restrito é de fácil implementação. Uma maneira de se fazer isso é apresentado no Algoritmo 2. No algoritmo, ρ é um valor estritamente positivo e que varia para cada problema. Em geral, para um problema de otimização inteira em que os parâmetros e variáveis são todos inteiros, $\rho = 1$ é sempre válido. Vale reforçar que as soluções dos problemas lexicográficos (linhas 1 e 2) fazem parte da saída do algoritmo (são soluções eficientes).

Algoritmo 2: Algoritmo ε -restrito biobjetivo

Entrada: $f, \mathcal{X}, \rho$

Saída: $\mathcal{X}_E$

```

1 Resolva  $\text{lexmin}(f_2, f_1)$ , obtenha  $f_2^*$  e  $x^*$  e adicione  $x^*$  a  $\mathcal{X}_E$ ;
2 Resolva  $\text{lexmin}(f_1, f_2)$ , obtenha  $f_1^*$  e  $\bar{x}$  e adicione  $\bar{x}$  a  $\mathcal{X}_E$ ;
3  $\varepsilon = f_2^* + \rho$ ;
4 enquanto  $f_1(x^*) > f_1^*$  faça
5   | Resolva  $\text{Min}\{f_1(x) \mid f_2(x) \leq \varepsilon, x \in \mathcal{X}\} = \mathcal{P}$  e obtenha  $x^*$ ;
6   | se  $x^*$  for único então
7   |   | adicione  $x^*$  a  $\mathcal{X}_E$ 
8   | senão
9   |   | escolha  $x^*$  tal que  $f_2(x^*) \leq f_2(x), \forall x$  solução de  $\mathcal{P}$ ;
10  |   | adicione  $x^*$  a  $\mathcal{X}_E$ 
11  | fim
12  |  $\varepsilon := \varepsilon + \rho$ 
13 fim
```

É importante dizer que esse método pode obter soluções fracamente eficientes e que para apenas uma delas a característica 2 da Tabela 1 é satisfeita. Por exemplo, ao utilizar o Algoritmo 2 no Problema 1, nas linhas 1 e 2 são obtidos os pontos D e A respectivamente. Na linha 3 defini-se $\varepsilon = 8 + 1 = 9$. Dentro do laço de repetição, na primeira iteração é obtido o ponto C, na segunda iteração o ponto B e na terceira iteração, com $\rho = 11$, os pontos W e B são valores ótimos para $\mathcal{P}$. Para identificar qual ponto é não dominado, ou seja, qual das soluções é a eficiente (característica 2) é necessário realizar um teste comparativo de dominância (linha 9).

O método ε -restrito é exato e é capaz de determinar toda a fronteira de Pareto de um problema de otimização inteira com parâmetros inteiros e objetivos calculados por variáveis inteiras. Em problemas da literatura, principalmente com instâncias desafiadoras, a resolução dos subproblemas (linha 5 do Algoritmo 2) até a otimalidade pode não ser

possível. Quando isso ocorre, o método é utilizado de forma heurística limitando o tempo de solução dos subproblemas para um tempo inferior ao máximo. Dessa forma não há garantia que será encontrada a fronteira de Pareto, mas uma aproximação da fronteira, similar a fronteira de Pareto ϵ -aproximada mas sem saber o valor de ϵ .

Uma variação do método ε -restrito é o método ε -restrito aumentado. Em sua versão biobjetivo, um dos objetivos é adicionado ao conjunto de restrições, como no clássico, mas a função objetivo é modificada para uma soma ponderada dos objetivos. Essa ponderação pode ser pré-determinada ou variar conforme mudam os subproblemas (*e.g.* [18]), mas em geral, é um peso pequeno na casa de milésimos. A forma geral do problema escalar é exibido em (2.6)-(2.8). A proposta dessa variação está em impedir que o método encontre soluções fracamente eficientes ao fornecer informação do segundo objetivo na função objetivo, pois para uma solução eficiente x^* e uma fracamente eficiente $\bar{x}$, tal que $f_1(x^*) = f_1(\bar{x})$ é válido $f_2(x^*) < f_2(\bar{x})$, logo $f_1(x^*) + \mu f_2(x^*) < f_1(\bar{x}) + \mu f_2(\bar{x})$ com $\mu > 0$.

$$\text{Min } f_1(x) + \mu f_2(x) \quad (2.6)$$

$$\text{S.a : } f_2(x) \leq \varepsilon \quad (2.7)$$

$$x \in \mathcal{X} \quad (2.8)$$

Outro método de escalarização que é muito conhecido é o método da soma ponderada. Ele possui características diferentes do método ε -restrito. Na Tabela 1, o método da soma ponderada satisfaz as características 1, 2 e 4. Problemas que são resolvidos em tempo polinomial mantêm essa característica ao utilizar a soma ponderada. Porém, isso tem um custo, que é não encontrar toda fronteira de Pareto quando a fronteira é não convexa. A forma geral do problema escalar é exibido em (2.9)-(2.10).

$$\text{Min } \sum_{j=1}^p w_j f_j(x) \quad (2.9)$$

$$\text{S.a : } x \in \mathcal{X} \quad (2.10)$$

Sendo $w \in \mathbb{R}_{\geq}^p$, a função objetivo de (2.9)-(2.10) é uma combinação linear não negativa dos objetivos de (2.1)-(2.2). Além disso, se os pesos forem normalizados a função objetivo torna-se uma combinação convexa. Dessa forma, é mais fácil entender quais pontos da fronteira de Pareto o método não consegue encontrar, que são as soluções eficientes não suportadas. Como a imagem de uma solução eficiente não suportada está na parte não convexa da fronteira de Pareto ($\mathcal{Y}_{nN}$), as curvas de níveis da função objetivo sempre terão um ponto não dominado suportado como mínimo.

Em geral, utilizar apenas soma ponderada ou ε -restrito apresenta algumas limitações. Um caminho que as pesquisas começaram a seguir foi combinar diferentes métodos de escalarização para obter um novo método, como exemplo o método ε -restrito aumentado

que faz uma ponderação dos objetivos, similar à soma ponderada, e adiciona objetivos nas restrições, como no ε -restrito. Em Ehrgott [12], esses métodos são chamados de híbridos. Outros exemplos deles são o algoritmo de particionamento da fronteira ([19]) e o método da caixa balanceada ([16]).

O método da caixa balanceada proposto em [16] é uma combinação do Algoritmo 1 e do Algoritmo 2. Para utilizar o método é preciso ter os pontos extremos da fronteira de Pareto. Resumidamente no caso biobjetivo, com os pontos extremos determina-se um retângulo no espaço critério que será a região de busca. Restrito a essa região, encontra-se um novo ponto utilizando o Algoritmo 1. Com esse novo ponto, determina-se um novo retângulo que será a nova região de busca. O método encerra quando não é mais encontrado um ponto na região de busca.

Extensão do modelo escalar geral da literatura

Em [10] é apresentado um modelo escalar geral que conforme são fixados os parâmetros ele se equivale aos métodos escalares apresentados na Seção 2.1.1. Aqui é proposta uma extensão desse modelo geral para que ele seja capaz de abranger mais métodos de escalarização ao adicionar $\sum_{k=1}^p \mu_k s_k$ na função objetivo e as restrições (2.12) como, por exemplo, Método ε -restrito com restrições elásticas (*e.g.* [18]). O modelo é exibido em (2.11)-(2.13).

$$\text{Min } \max_{k=1}^p [\nu_k(f_k(x) - \tau_k)] + \sum_{k=1}^p \lambda_k(f_k(x) - \tau_k) + \sum_{k=1}^p \mu_k s_k \quad (2.11)$$

$$S.a : f_k(x) + \gamma_k s_k \leq \varepsilon_k, \quad k = 1, \dots, p \quad (2.12)$$

$$x \in \mathcal{X}, \quad s \in \mathbb{R}^p \quad (2.13)$$

A linearização do modelo (2.11)-(2.13) é exibida em (2.14)-(2.17).

$$\text{Min } z + \sum_{k=1}^p \lambda_k(f_k(x) - \tau_k) + \sum_{k=1}^p \mu_k s_k \quad (2.14)$$

$$S.a : f_k(x) + \gamma_k s_k \leq \varepsilon_k, \quad k = 1, \dots, p \quad (2.15)$$

$$\nu_k(f_k(x) - \tau_k) \leq z, \quad k = 1, \dots, p \quad (2.16)$$

$$x \in \mathcal{X}, \quad s \in \mathbb{R}^p, \quad z \in \mathbb{R} \quad (2.17)$$

Uma prática muito comum na otimização é utilizar a relaxação langrangiana para obter limitantes para o problema. Assim como feito em [12], a relaxação lagrangiana do modelo (2.14)-(2.17) é exibida em (2.18)-(2.19), onde π e π' são multiplicadores referentes aos conjuntos de restrições (2.15) e (2.16) respectivamente.

$$\begin{aligned} \text{Min} \quad & z(1 - \sum_{k=1}^p \pi'_k) - \sum_{k=1}^p (\lambda_k \tau_k + \pi_k \varepsilon_k + \pi'_k \nu_k \tau_k) \\ & + \sum_{k=1}^p (\lambda_k + \pi_k + \pi'_k \nu_k) f_k(x) + \sum_{k=1}^p (\mu_k + \pi_k \gamma_k) s_k \end{aligned} \quad (2.18)$$

$$x \in \mathcal{X}, s \in \mathbb{R}^p, z \in \mathbb{R}, \pi_k, \pi'_k \in \mathbb{R}_+ \quad (2.19)$$

Sendo τ um ponto de referência, ν , λ e μ pesos positivos, ε_k parâmetros positivos para todo k e π e π' multiplicadores não negativos, ao fixar $\tau, \nu, \lambda, \mu, \varepsilon, \pi$ e π' o problema (2.18)-(2.19) pode ser analisado, de tal forma que $\alpha = \sum_{k=1}^p \pi'_k$, $\beta = \sum_{k=1}^p (\lambda_k \tau_k + \pi_k \varepsilon_k + \pi'_k \nu_k \tau_k)$, $w_k = \sum_{k=1}^p (\lambda_k + \pi_k + \pi'_k \nu_k)$ e $w'_k = \sum_{k=1}^p (\mu_k + \pi_k \gamma_k)$. Assim, se obtem o problema (2.20).

$$\min_{x \in \mathcal{X}, s \in \mathbb{R}^p, z \in \mathbb{R}} z(1 - \alpha) - \beta + \sum_{k=1}^p w_k f_k(x) + \sum_{k=1}^p w'_k s_k \quad (2.20)$$

Analisando o modelo (2.20) percebe-se que se $\alpha > 1$ o problema é ilimitado. Caso $\alpha \leq 1$ o problema se resume a uma soma de somas ponderadas. Isso mostra que a relaxação lagrangiana, que é um recurso utilizado na otimização monoobjetivo, se resume a uma soma ponderada que pode não encontrar todos os pontos da fronteira de Pareto, principalmente na otimização combinatória. Esse fato reforça a importância de explorar métodos diferentes dos métodos de escalarização, como o *branch-and-bound* multiobjetivo (e.g. [5]) (Ehrgott, [12]).

2.1.2 *Branch-and-bound* multiobjetivo

Os métodos de escalarização apresentados na Seção 2.1.1 são classificados como métodos de busca no espaço critério. Como discutido em [20], esses métodos tendem a ter vantagens sobre os métodos de busca no espaço decisão, pelo fato do espaço critério em geral ter dimensão menor do que o espaço de decisão. Porém, como visto na Seção 2.1.1, os métodos de escalarização possuem suas limitações: não há garantias de encontrar toda a fronteira de Pareto e utilizar a relaxação lagrangiana é equivalente a utilizar a soma ponderada. Dessa forma, assim como sugerido em [12] torna-se interessante investigar outros métodos de solução.

Os métodos de busca no espaço de decisão como o *branch-and-bound* também apresentam suas limitações. A principal limitação está no fato que as árvores de ramificações de problemas de médio e grande porte podem ser muito grandes. Para melhorar seu desempenho, são utilizadas técnicas de métodos de busca no espaço critério para melhorar a ramificação e sondagem dos nós (e.g. [5]).

Utilizar o método *branch-and-bound* para resolver um problema multiobjetivo possui algumas diferenças em relação a um problema monoobjetivo (e.g. [21, 22, 23]). Algumas características são mantidas, como: escolha da variável de ramificação e tipos de árvore (binária, ternária etc.). Outras características mantêm a mesma função, porém são bem diferentes: limitantes e testes de sondagem. Além disso, em cada nó da árvore é gerada

uma aproximação da fronteira de Pareto e existem novos elementos a serem explorados: ramificação no espaço critério.

Na otimização monoobjetivo, utilizar somente dois limitantes (um superior e um inferior) é suficiente para realizar as podas, pois se trata de um problema escalar (dois pontos distintos determinam um intervalo). No problema multiobjetivo, apenas um valor para limitante superior e um para inferior, em geral, é insuficiente ou não gera benefício. Para que se cumpra o objetivo de ter limitantes para um problema será necessário um conjunto de limitantes inferiores e um conjunto de limitantes superiores. Considere $\mathbb{R}_{\geq}^p = \{y \in \mathbb{R}^p \mid y \geq 0\}$ e para $s \in \mathbb{R}^p$, define-se a seguinte adição $s + \mathbb{R}_{\geq}^p = \{s + y \mid y \in \mathbb{R}_{\geq}^p\}$.

Definição 16 ($\mathbb{R}_{\geq}^p$ fechado e limitado [24]). *Considere $S \subset \mathbb{R}^p$. É dito que S é:*

1. $\mathbb{R}_{\geq}^p$ fechado se $S + \mathbb{R}_{\geq}^p = \{s + y \mid s \in S, y \in \mathbb{R}_{\geq}^p\}$ é fechado;
2. $\mathbb{R}_{\geq}^p$ limitado se existe $s^0 \in \mathbb{R}^p$ tal que $S \subset s^0 + \mathbb{R}_{\geq}^p$.

Definição 17 (Conjunto de limitantes [24]). *Seja $\bar{Y} \subset \mathcal{Y}_N$.*

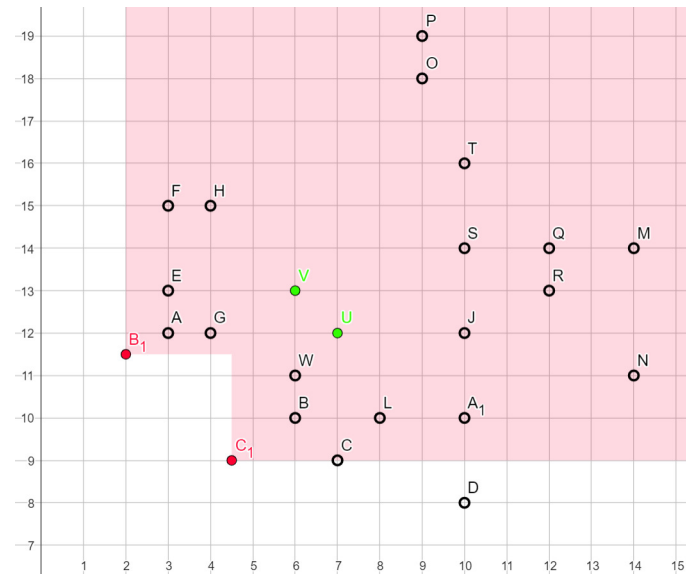
1. $\mathcal{L}$ é um conjunto limitante inferior para $\bar{Y}$ se $\mathcal{L}$ é $\mathbb{R}_{\geq}^p$ fechado e limitado tal que $\bar{Y} \subset \mathcal{L} + \mathbb{R}_{\geq}^p$ e $\mathcal{L} \subset (\mathcal{L} + \mathbb{R}_{\geq}^p)_N$
2. $\mathcal{U}$ é um conjunto limitante superior para $\bar{Y}$ se $\mathcal{U}$ é $\mathbb{R}_{\geq}^p$ fechado e limitado tal que $\bar{Y} \subset cl[(\mathcal{U} + \mathbb{R}_{\geq}^p)^c]$ e $\mathcal{U} \subset (\mathcal{U} + \mathbb{R}_{\geq}^p)_N$, com cl representando o fecho do conjunto.

A relação de dominância segundo Pareto discutida no início deste capítulo (Definição 11) se estende para conjuntos limitantes (Definição 18). É possível que um conjunto limitante $\mathcal{L}^1$ domine outro conjunto limitante $\mathcal{L}^2$. Essa dominância pode ser parcial, quando apenas alguns pontos de $\mathcal{L}^2$ são dominados, ou total, quando todos os pontos de $\mathcal{L}^2$ estão contidos na região de dominância de $\mathcal{L}^1$.

Definição 18 (Região de dominância de um conjunto limitante). *Seja $\mathcal{L}$ um conjunto limitante. A região de dominância de $\mathcal{L}$ é dada pela união das regiões de dominância de todo $y \in \mathcal{L}$.*

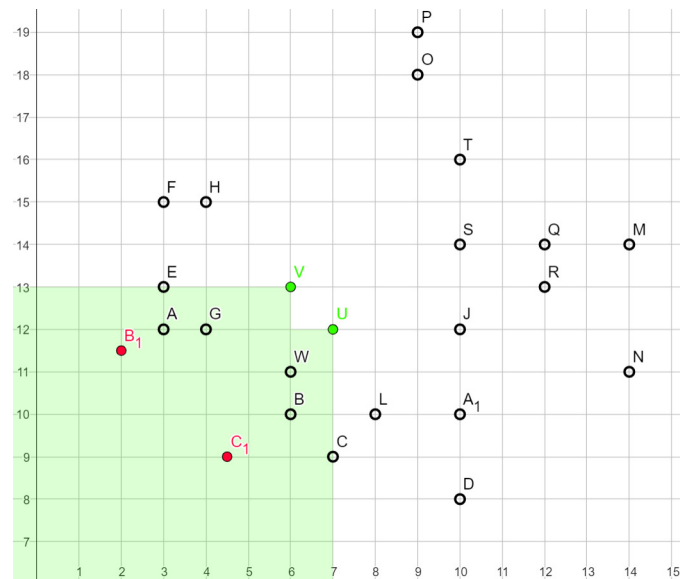
Nas Figuras 5 e 6 estão ilustrados exemplos de conjuntos limitantes inferiores e superiores respectivamente. Na Figura 5 os pontos B_1 e C_1 formam um conjunto limitante inferior de $\bar{Y} = \{A, B, C\}$ pois $\bar{Y}$ está contido na região colorida em vermelho, que é a região dominada por B_1 e C_1 , e $\mathcal{L} = \{B_1, C_1\}$ é $\mathbb{R}_{\geq}^2$ fechado e limitado. Na Figura 6, $\mathcal{U} = \{U, V\}$ é um conjunto limitante superior para $\bar{Y}$, pois $\bar{Y}$ está contido na região colorida em verde e $\mathcal{U}$ é $\mathbb{R}_{\geq}^p$ fechado e limitado. É interessante ver que $\mathcal{L}$ e $\mathcal{U}$ são conjuntos limitantes de apenas uma parte da fronteira de Pareto. O ponto D não está limitado por nenhum dos dois conjuntos. Isso evidencia o aumento de complexidade ao trabalhar com conjuntos limitantes.

Figura 5 – Região dominada pelo conjunto limitante inferior $\mathcal{L}$



Fonte: Elaborado pelo autor.

Figura 6 – Conjunto limitante superior $\mathcal{U}$



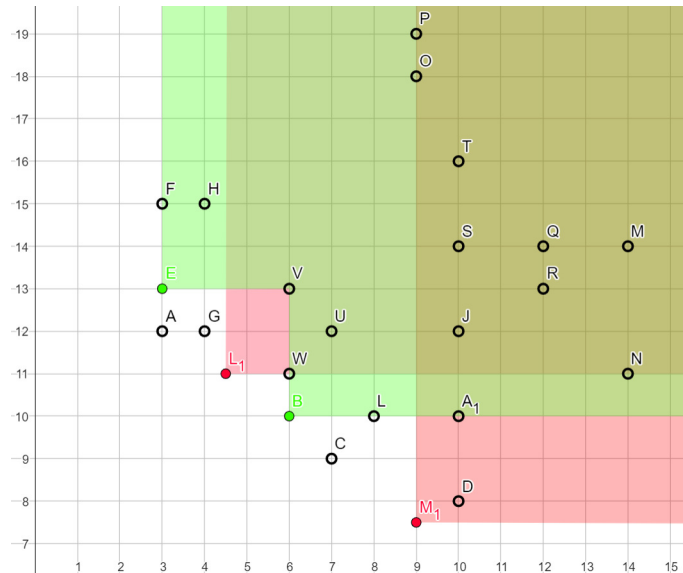
Fonte: Elaborado pelo autor.

Na otimização monoobjetivo, um nó pode ser sondado se satisfizer uma das três condições:

1. o problema relaxado é infactível;
2. solução ótima do problema relaxado é inteira;
3. o limitante inferior obtido é pior que o valor da solução incumbente.

Na otimização multiobjetivo, apenas a condição 1 é exatamente igual. Em geral, para cada nó da árvore que será explorado é gerada uma aproximação da fronteira de Pareto. Esse nó só poderá ser sondado pela condição 2, se cada uma das soluções encontradas pelo problema relaxado for inteira, que em geral é pouco provável. A condição 3 somente é satisfeita se o conjunto limitante superior dominar toda a aproximação da fronteira de Pareto encontrada naquele nó. Na Figura 7, $\mathcal{U} = \{E, B\}$ é conjunto limitante superior e $\mathcal{L} = \{L_1, M_1\}$ conjunto limitante inferior. A região que está colorida de vermelho é toda região de dominância de $\mathcal{L}$ e a região colorida de verde é a região dominada por $\mathcal{U}$. Percebe-se que $\mathcal{U}$ não domina todo $\mathcal{L}$, somente onde as cores se sobrepõem, restando dois retângulos para serem explorados. Note que o ponto D pertence ao retângulo mais a direita e abaixo, ou seja, é importante ramificar esse nó.

Figura 7 – Dominância parcial de conjuntos limitantes (condição 3 para sondagem biobjetivo)



Fonte: Elaborado pelo autor.

Em resumo para sondar um nó, a condição 1 irá depender do problema e do tipo de busca feita, assim como no problema monoobjetivo. A condição 2, em geral, é satisfeita poucas vezes. Por fim, a condição 3 precisa de um conjunto limitante superior de muita qualidade para sondar o nó. Isso evidencia a dificuldade de reduzir o número de nós na árvore *branch-and-bound*. Também, reforça ainda mais a importância de se ter um conjunto limitante superior de qualidade.

Toda essa discussão feita sobre o *branch-and-bound* multiobjetivo mostra que o conjunto limitante superior é de grande importância, porém, obter bons limitantes não é uma tarefa fácil. Nesse sentido, os autores Parragh e Tricoire [5] propuseram outros mecanismos para melhorar a sondagem de nós e para melhorar a exploração de nós ramificados em um

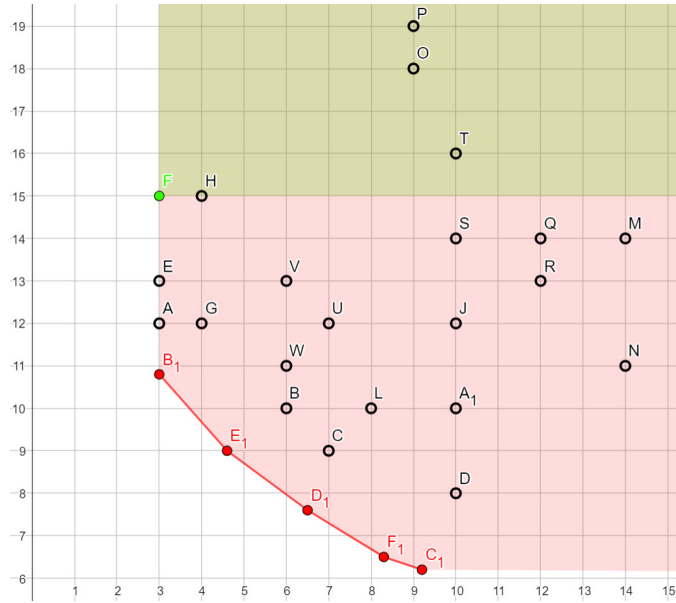
sistema chamado BIOBAB (*biobjective branch-and-bound*). Esse sistema será utilizado no estudo computacional, e por isso, um detalhamento é realizado a seguir.

O sistema BIOBAB

O BIOBAB é um sistema desenvolvido e utilizado em [5], que foi disponibilizado pelos autores no repositório <https://github.com/JKU-PLM/BIOBAB>. No sistema estão implementados os métodos: *branch-and-bound*, ε -restrito e caixa balanceada; para resolver problemas de otimização inteira mista biobjetivos. O pseudo-código do seu método *branch-and-bound* está presente no Algoritmo 3. Os três métodos implementados no sistema são utilizados no estudo computacional apresentado no Capítulo 5. Para detalhar os mecanismos do método *branch-and-bound* do sistema BIOBAB será utilizado o espaço critério associado ao Problema 1 (Figura 4). Na Figura 3 apresenta-se uma legenda para as Figuras 8, 9, 10 e 11.

Para resolver cada nó o BIOBAB inicia utilizando otimização lexicográfica para encontrar os pontos extremos (linhas 1 e 2). Com os pontos extremos, na linha 6 define-se pesos para os objetivos e utiliza-se a soma ponderada (Seção 2.1.1) para encontrar os pontos não dominados suportados. Quando não são mais encontrados pontos pela soma ponderada, a resolução do nó encerra. É importante dizer que os pontos encontrados são do problema inteiro ou relaxado (opções do sistema no Apêndice 6), que em geral são limitantes inferiores. O conjunto limitante inferior é dado pelos segmentos que ligam os pontos de *LB* (linha 7). Isso pode ser observado na Figura 8, os pontos B_1, E_1, D_1, F_1 e C_1 foram os pontos ótimos obtidos na resolução do nó raiz associado à resolução do Problema 1 e os segmentos em vermelho compõem o conjunto limitante inferior. É interessante dizer que durante a otimização lexicográfica foi encontrada uma solução inteira (possui imagem no ponto F), mesmo não sendo o ótimo da otimização lexicográfica ele foi armazenado pelo BIOBAB e já será utilizado como limitante superior. Nas linhas 1, 2 e 6 o asterisco (*) indica que toda solução factível encontrada durante a resolução é testada para possivelmente atualizar o conjunto limitante superior.

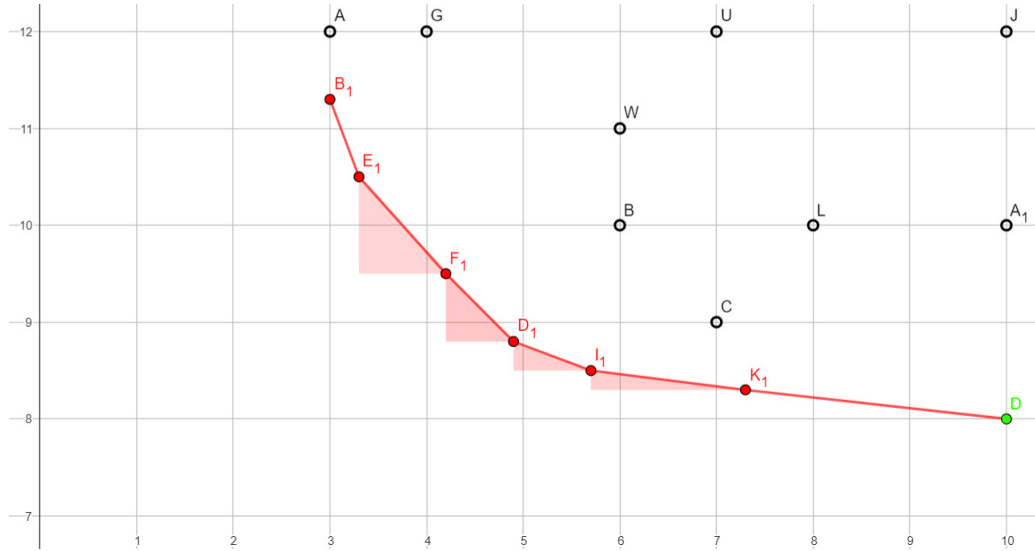
Figura 8 – Espaço critério associado à resolução do nó raiz do Problema 1



Fonte: Elaborado pelo autor.

O processo realizado pela soma ponderada é pegar os pontos extremos e encontrar um ponto não dominado suportado entre eles. Com esse novo ponto, o mesmo processo se repete alterando esses pontos até que não sejam mais encontrados pontos não dominados suportados. Esse processo é realizado na expectativa de encontrar uma solução factível ou um limitante inferior. Na Figura 9 mostra-se como o segundo nó da fila *branch-and-bound* é explorado. As regiões em vermelho, que são os triângulos abaixo dos segmentos $\overline{E_1F_1}$, $\overline{F_1D_1}$, $\overline{D_1I_1}$ e $\overline{I_1K_1}$ não possuem pontos inteiros. Como a soma ponderada busca pontos nessas regiões, não há necessidade de resolver esses problemas ponderados. Além disso, esses segmentos já definem limitantes inferiores, então eles podem compor o conjunto limitante inferior para esse nó. É importante dizer que isso só vale para problemas em que os coeficientes e variáveis da função objetivo são inteiros, que é o caso do Problema 1. Essa é uma melhoria presente no BIOBAB, com dois pontos é realizado um teste para verificar se há pontos inteiros na região de busca da soma ponderada. Se não houver pontos inteiros, esses dois pontos são considerados limitantes inferiores e o método segue pulando a resolução desse problema ponderado. Caso contrário, o método segue normalmente. Essa melhoria é chamada de *lower bound lifting* e é realizada na linha 3 e durante a soma ponderada na linha 6 do Algoritmo 3.

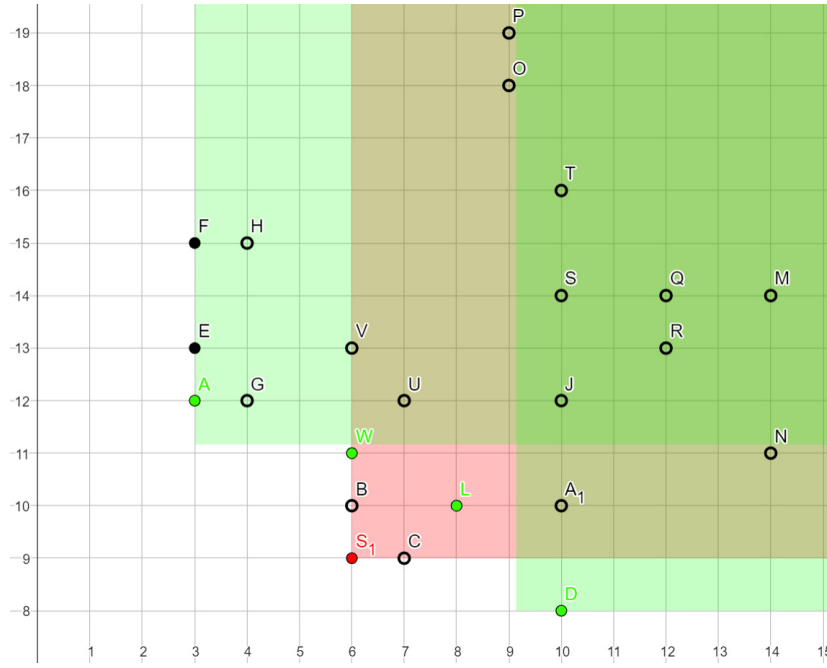
Figura 9 – Espaço critério associado à resolução do segundo nó da fila *branch-and-bound*



Fonte: Elaborado pelo autor.

Na Figura 10 explora-se o oitavo nó da fila *branch-and-bound*. Nesse momento $\mathcal{U} = \{A, W, L, D\}$ e $\mathcal{L} = \{S_1\}$. Após definir $\mathcal{U}$, o BIOBAB faz uma filtragem (Linha 9) para ver se esse nó pode ser sondado (condição 3). Então, três artifícios do BIOBAB são ativados. O primeiro artifício é verificar se há pontos inteiros na região não filtrada (região somente em vermelho), caso não tenha, o nó pode ser sondado (Linha 10). Como há pontos inteiros dentro da região é necessário fazer a ramificação. O segundo artifício é válido para problemas em que os coeficientes das variáveis na função objetivo são inteiros. É possível reduzir mais a região colorida de vermelho garantindo que se um ponto for encontrado nessa região ele não será dominado por algum limitante superior (linha 11). Isso não ocorre na Figura 7, o ponto V poderia ser encontrado e seria eliminado por ser dominado. O terceiro artifício é definir cortes no espaço objetivo para restringir a busca somente na região vermelha que tenha pontos inteiros. Esse artifício é chamado de ramificação do espaço objetivo, que é uma técnica utilizada pelos métodos de escalarização. A ramificação do espaço objetivo é representado pelo duplo asterisco (**) na linha 12 do Algoritmo 3.

Figura 10 – Espaço critério associado à resolução do oitavo nó da fila *branch-and-bound*



Fonte: Elaborado pelo autor.

Algoritmo 3: Algoritmo BIOBAB [5]

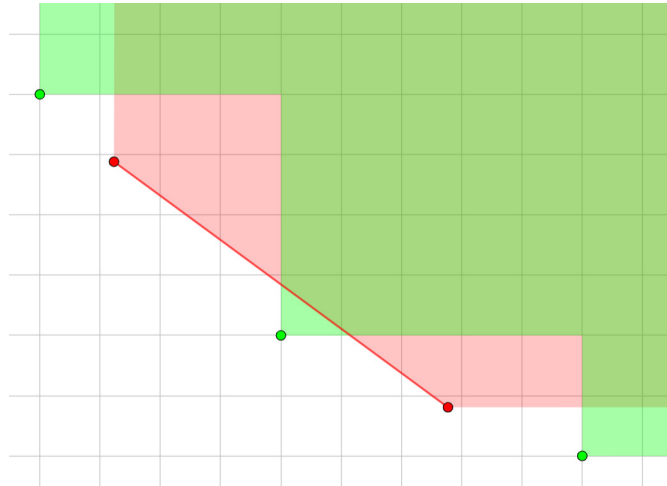
Entrada: $f, \mathcal{X}, UB, LB, L$
Saída: $\mathcal{X}_E, \mathcal{Y}_N$

- 1 $e_1 = \text{lexmin}(f_1, f_2) * ;$
- 2 $e_2 = \text{lexmin}(f_2, f_1) * ;$
- 3 **se** *LowerBoundLifting* é aplicável **então**
- 4 $\overline{LB} = \text{segment}(\{e_1, e_2\})$
- 5 **senão**
- 6 $LB = SP_{bl}(e_1, e_2) * ;$
- 7 $\overline{LB} = \text{segment}(LB)$
- 8 **fim**
- 9 $\overline{LB} = \text{Filtering}(\overline{LB}) ;$
- 10 $\overline{LB} = \text{SegmentTightening}(\overline{LB}) ;$
- 11 $\overline{LB} = \text{IntegerDominance}(\overline{LB}) ;$
- 12 Ramifica $\overline{LB}$ e adiciona nós a $L * * ;$
- 13 **enquanto** $L \neq \emptyset$ **faça**
- 14 Retire um nó de $L ;$
- 15 Repita as linhas de 1 a 12
- 16 **fim**

As melhorias feitas no BIOBAB foram para aprimorar a sondagem multiobjetivo, evitar resolver problemas desnecessários e para reduzir o espaço de busca em cada nó, que são os maiores problemas no *branch-and-bound* multiobjetivo. É importante dizer que um dos principais usos da ramificação do espaço objetivo pode ser observado na Figura 11. Nela

percebe-se que após a filtragem são obtidas duas regiões em vermelho desconexas. Nesse caso a ramificação do espaço objetivo será feita junto com a ramificação das variáveis em cada uma dessas regiões, de forma independente, resultando em 4 nós na árvore.

Figura 11 – Ramificação do espaço critério



Fonte: Elaborado pelo autor.

2.2 Métricas de desempenho

Com tantos métodos de solução diferentes, em que cada método possui vantagens e desvantagens, saber qual método tem o melhor desempenho não é tão simples. Como saber se a aproximação da fronteira de Pareto (AFP) do método A é melhor do que a aproximação da fronteira do método B? Para realizar essa tarefa, na literatura surgiram diversas métricas de desempenho.

Como há diferentes fatores a serem analisados em uma AFP, as métricas de desempenho não conseguem avaliar a AFP por completo. Por esse motivo é muito comum nas pesquisas os autores utilizarem mais de uma métrica de desempenho. Em [25] são discutidos três fatores de interesse para serem avaliados em uma AFP. São eles:

1. Convergência: o quanto a AFP está próxima da fronteira de Pareto;
2. Diversidade: extensão e disposição dos pontos da AFP;
3. Cardinalidade: quantos pontos a AFP possui.

Em trabalhos mais recentes, como visto em [26], os autores se referem às métricas de desempenho como indicadores de performance, mas não há distinção dos significados.

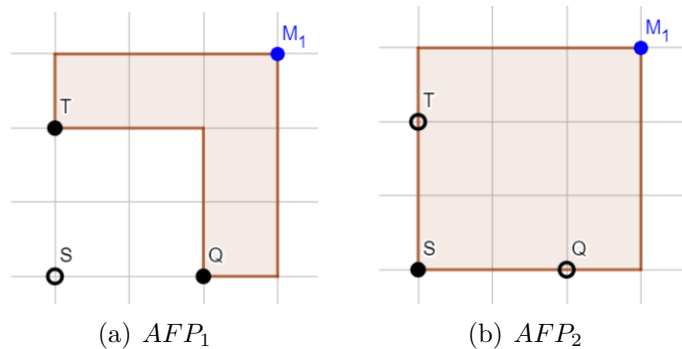
Nessa tese, para analisar o desempenho dos métodos de solução serão utilizados duas métricas: cardinalidade ($I_{\#}$) e Hipervolume (I_H). A cardinalidade é uma métrica unária e avalia o número de pontos da AFP, então quanto maior, melhor o método. O Hipervolume

[27] é uma métrica unária que consegue avaliar a convergência e diversidade da AFP. Essa métrica calcula o hipervolume da região dominada pela AFP e delimitada por um ponto de referência, em que esse ponto de referência é dominado por todos os pontos da AFP. Assim, quanto maior o hipervolume, melhor o desempenho do método. Para problemas biobjetivos o hipervolume é a área e pode ser calculado pela equação (2.21), no qual $A[y_N^i, r]$ é a área formada pelo ponto não dominado y_N^i da AFP e o ponto de referência r . Note que $\cap_{k=1}^1 [y_N^k, r] = \emptyset$, ou seja, para o primeiro ponto não é descontado nenhuma área de interseção.

$$I_H(\mathcal{Y}_N) = \sum_{i=1}^{|\mathcal{Y}_N|} \left(A[y_N^i, r] - A \cap_{k=1}^i [y_N^k, r] \right) \quad (2.21)$$

Nas Figuras 12, 13 e 14 mostra-se o hipervolume de algumas AFPs. A AFP observada estará sendo representada por pontos pretos preenchidos ($\bullet$) e os outros pontos no espaço serão representados por pontos pretos não preenchidos ($\circ$). Na Figura 12 estão o hipervolume (área dominada) da $AFP_1 = \{T, Q\}$ e da $AFP_2 = \{S\}$ em relação ao ponto de referência M_1 (ponto em azul). Nela a métrica hipervolume não reflete a característica cardinalidade das AFPs, pois a AFP_1 com mais pontos (Figura 12(a)) tem hipervolume igual a 5 *u.a.*, que é menor do que o hipervolume da AFP_2 (Figura 12(b)) com 9 *u.a.*. O fato do ponto S dominar os pontos T e Q faz com que o hipervolume da AFP_2 , que contém o ponto S , seja maior.

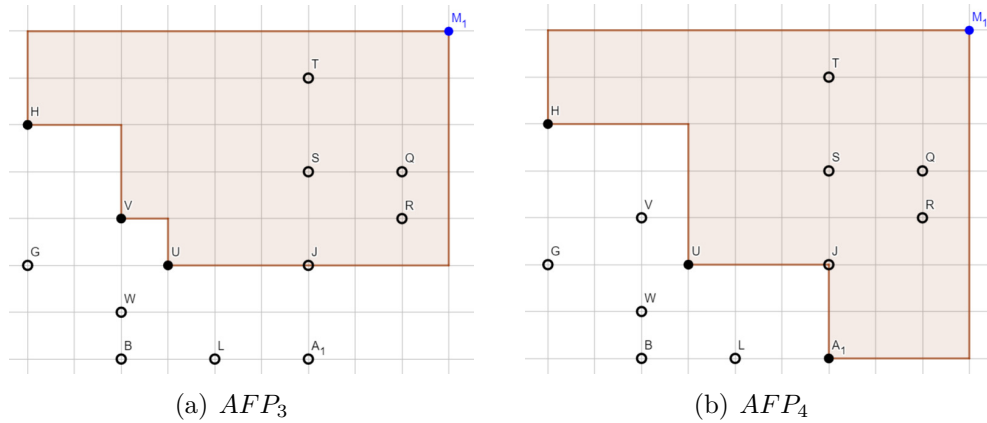
Figura 12 – Hipervolume e quantidade de pontos da AFP_1 e AFP_2



Fonte: Elaborado pelo autor.

Na Figura 13 estão o hipervolume da $AFP_3 = \{H, V, U\}$ e da $AFP_4 = \{H, U, A_1\}$ em relação ao ponto de referência M_1 e a característica diversidade é analisada. É importante dizer que para isso, as duas AFPs possuem a mesma quantidade de pontos e esses pontos não são comparáveis, dessa forma as AFPs se diferenciam apenas em diversidade. Na Figura 13(a) os pontos V e U estão relativamente próximos, o que reflete em uma área menor de dominância ($I_H = 38$ *u.a.*), enquanto na Figura 13(b) os três pontos estão melhor distribuídos, provocando uma área de dominância maior ($I_H = 42$ *u.a.*). É importante lembrar que a diversidade reflete a extensão e a distribuição dos pontos na AFP. Na Figura 13(b) a AFP_4 é mais extensa e está melhor distribuída.

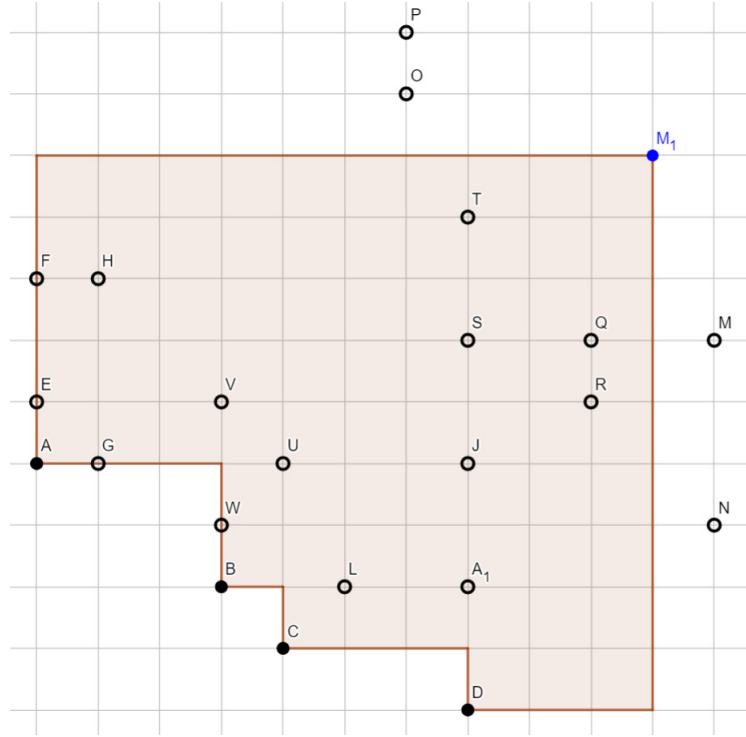
Figura 13 – Hipervolume e diversidade da AFP_3 e AFP_4



Fonte: Elaborado pelo autor.

Na Figura 14 estão o hipervolume da fronteira de Pareto do Problema 1 ($I_H = 73 u.a.$). É notório que o hipervolume da fronteira de Pareto é maior do que as AFPs das Figuras 12 e 13. Além disso, quanto mais próximo a AFP está da fronteira (convergência) mais seus hipervolumes se aproximam. Isso mostra a capacidade do hipervolume em analisar a convergência das AFPs. É claro que para ter a noção de quão próximo da fronteira a AFP está, é necessário ter a fronteira de Pareto. Porém, como em geral a fronteira não é conhecida, é dito que quanto maior o valor do hipervolume da AFP, mais próximo ela está da fronteira de Pareto. É dessa forma que o hipervolume é utilizado para comparar métodos de solução.

Figura 14 – Hipervolume da fronteira de Pareto



Fonte: Elaborado pelo autor.

Duas razões justificam a escolha da métrica cardinalidade e hipervolume para serem utilizadas no estudo apresentado no Capítulo 5. A primeira, é porque o hipervolume junto com o cardinalidade conseguem avaliar todos os critérios de interesse de uma AFP de cada método. A segunda razão é por essas métricas serem uma das mais utilizadas segundo o levantamento feito em [25].

Uma forma alternativa de utilizar informações do hipervolume é através do *hyper ratio*, que é a razão entre o hipervolume da AFP e da fronteira de Pareto (Equação 2.22) (*e.g.* [26]). Dessa forma, o valor máximo que o *hyper ratio* assume é 1, quando a AFP é exatamente a fronteira de Pareto, facilitando a interpretação da métrica. Para contornar a necessidade de ter o hipervolume da fronteira de Pareto, ao avaliar o desempenho de mais de um método, se propõe utilizar o maior hipervolume conhecido para substituir o hipervolume da fronteira de Pareto. Essa prática é utilizada na Seção 5.3

$$HR(AFP) = \frac{I_H(AFP)}{I_H(\mathcal{Y}_N)} \quad (2.22)$$

2.3 Ferramentas matemáticas propostas para análise de pontos na fronteira de Pareto

Para alcançar os objetivos dessa tese foram desenvolvidas ferramentas matemáticas capazes de localizar pontos na fronteira de Pareto. Essa necessidade surgiu porque, até onde sabemos, essa pesquisa é a primeira a explorar a natureza biobjetivo do JIT-FSP. Apesar das ferramentas serem propostas para explorar a fronteira de Pareto do JIT-FSP, elas podem ser aplicadas na fronteira de Pareto de qualquer problema de otimização biobjetivo.

As três primeiras ferramentas foram desenvolvidas para localizar a imagem da solução monoobjetivo na fronteira de Pareto. O intuito delas é informar, algebricamente, o quão próximo esse ponto está de algum ponto extremo. Para os cálculos não sofrerem distorções, a fronteira de Pareto é normalizada. A primeira ferramenta (D_I) é baseada em distância e utiliza do ponto ideal (Definição 9) como referência. A segunda (D_μ) também é baseada em distância com a diferença do ponto de referência ser o ponto médio entre os pontos extremos da fronteira de Pareto. A terceira (D_Δ) é dada pelo módulo da diferença entre os objetivos.

Para as três ferramentas, quanto maior o valor retornado, mais próximo de algum ponto extremo esse ponto está, sendo 1 o valor máximo e 0 o mínimo. Será dito que o ponto está equilibrado quando os objetivos não são priorizados (entre 0 e 0.1), moderado (entre 0.1 e 0.5) e desequilibrado quando prioriza algum dos objetivos (entre 0.5 e 1). A D_μ será utilizada para explicar como é determinada a posição de um ponto na fronteira de Pareto. Primeiro calcula-se a distância de cada ponto até μ . Em seguida identifica-se o máximo (Max) e o mínimo (Min) dessas distâncias. Finalmente, uma função suave (2.23) é usada para criar uma função teste em (2.24), e através dessa, define-se a D_μ em (2.25). A D_μ é uma função teste que leva o valor da distância entre um ponto não dominado e o ponto de referência até o intervalo entre zero e um. A D_I e D_Δ são calculadas da mesma forma e também são funções testes.

$$f(x) = \begin{cases} 0, & \text{se } x \leq 0 \\ e^{-\frac{1}{x}}, & \text{se } x > 0 \end{cases} \quad (2.23)$$

$$g(x) = \frac{f(x)}{f(x) + f(1-x)} \quad (2.24)$$

$$D_\mu(x) = g\left(\frac{x - Min}{Max - Min}\right) = \begin{cases} 0, & \text{se } x \leq Min \\ e^{-\frac{1}{x}}, & \text{se } Min < x < Max \\ 1, & \text{se } x \geq Max \end{cases} \quad (2.25)$$

Graficamente, a medida D_μ pode ser representada pela Figura 15. No arco circular verde estão representadas as menores distâncias até o ponto de referência μ ilustrado pelo

“x” de cor preta. A seta em verde indica a região dos pontos equilibrados (3 pontos dessa fronteira de Pareto). O arco circular laranja indica o início da região dos pontos moderados (nenhum ponto nessa fronteira) e sua seta exibe a extensão dessa região. Por fim, o arco circular vermelho indica o início da região dos pontos desequilibrados (3 pontos dessa fronteira de Pareto).

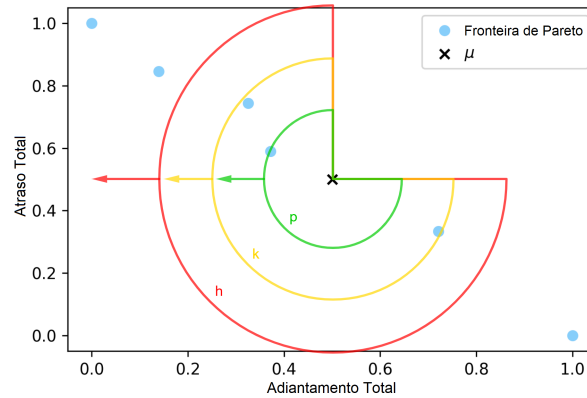


Figura 15 – Regiões de equilíbrio baseado em distância e μ como referência

Além de analisar o desequilíbrio da solução monoobjetivo, será analisado o aumento percentual máximo (SP_{max}), médio (SP_{μ}) e mínimo (SP_{min}) da soma ponderada dos pontos da fronteira de Pareto em relação à solução monoobjetivo. Essa medida irá informar o *trade-off* entre priorizar o valor monoobjetivo e adequar a solução ao cenário industrial, ou seja, essa medida permite conhecer o aumento na solução monoobjetivo para cada ponto na fronteira de Pareto. Na Figura 16 demonstra-se que para a fronteira de Pareto ilustrada na figura, o aumento percentual máximo não chega a 1.5%, em média esse aumento é de 0.61% e existem outras soluções que não apresentam aumento no objetivo. O uso dessa ferramenta, assim como é exibido na Figura 16, pode ser útil para o tomador de decisão, pois o permite visualizar a variação da função objetivo ao mesmo tempo que escolhe um ponto na fronteira de Pareto.

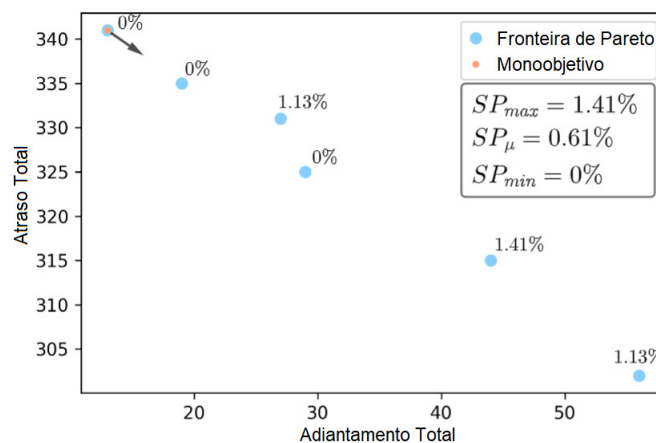


Figura 16 – Aumento percentual da soma ponderada em relação à solução monoobjetivo

A quarta ferramenta proposta consiste em identificar os pontos não dominados não suportados presentes na fronteira de Pareto. O pseudocódigo utilizado está apresentado no Algoritmo 4. Este algoritmo verifica, para cada par de pontos (A, B) na fronteira de Pareto, se um determinado ponto P está acima do segmento $\overline{AB}$, ou seja, se é não suportado. A identificação de pontos não dominados não suportados visa demonstrar a importância da utilização de métodos multiobjetivos adequados. Se os pontos não suportados constituírem a maioria da fronteira de Pareto e for utilizado o método da soma ponderada, a abordagem multiobjetivo pode não fornecer efetivamente uma gama de escolhas para os decisores. Isto poderia também levar à conclusão incorreta de que existe pouco conflito entre os objetivos. Por conseguinte, é importante utilizar um método que determine todos os pontos não dominados, suportados e não suportados.

Algoritmo 4: Pseudo-código para identificar pontos não suportados

Entrada: $\mathcal{Y}_N$
Saída: $\mathcal{Y}_{nN}, \mathcal{Y}_{sN}$

```

1 para  $(A, B) \in C_2^{|\mathcal{Y}_N|}$  faça
2   para  $P$  entre  $A$  e  $B$  faça
3     se  $y_P > \frac{y_B - y_A}{x_B - x_A}(x_P - x_A) + y_A$  então
4       adicione  $P$  a  $\mathcal{Y}_{nN}$  (é não dominado não suportado)
5     senão
6       continue
7   fim
8 fim
9 fim
10  $\mathcal{Y}_{sN} = \mathcal{Y}_N - \mathcal{Y}_{nN}$ 

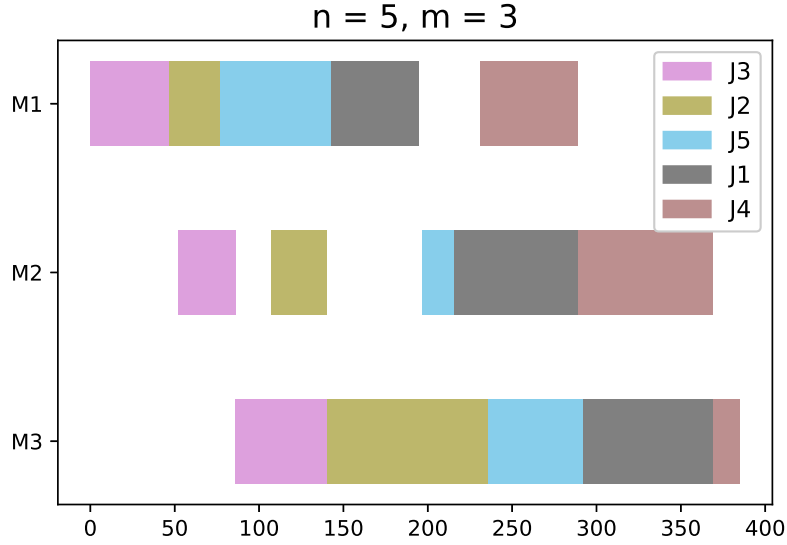
```

3 O problema de sequenciamento em *flow shop*

O Problema de sequenciamento em *flow shop* (FSP) é um importante problema de planejamento da produção. A solução do problema consiste em encontrar uma sequência de n tarefas que devem ser processadas por todas as m máquinas, respeitando um fluxo unidirecional, ou seja, sempre da primeira à última máquina. Cada tarefa j possui um tempo p_{jk} para ser processada em cada máquina k . Será considerado que não há tempo de preparo das máquinas e as tarefas em execução não podem ser interrompidas. Além disso, as sequências de execução das tarefas devem ser iguais para todas as máquinas, configurando um problema de sequenciamento em *flow shop* permutacional ([28]).

Um recurso utilizado para visualizar a solução de um problema de sequenciamento é o gráfico de Gantt. Um exemplo de gráfico de Gantt está presente na Figura 17. Nele estão ilustrados o sequenciamento de 5 tarefas (barras coloridas) em 3 máquinas (eixo vertical). Cada barra colorida mostra o instante de início e de término do processamento da tarefa em determinada máquina, sendo o seu comprimento exatamente o tempo de processamento da tarefa naquela máquina. Por exemplo, a tarefa $J3$ iniciou seu processamento na máquina 1 no instante 0, terminou no instante 50 e já iniciou seu processamento na máquina 2. Essa forma de visualizar a solução também pode servir como recurso para o tomador de decisão, pois facilita a validação e/ou escolha da solução, quando existem diversas soluções eficientes.

Figura 17 – Sequenciamento de 5 tarefas em 3 máquinas

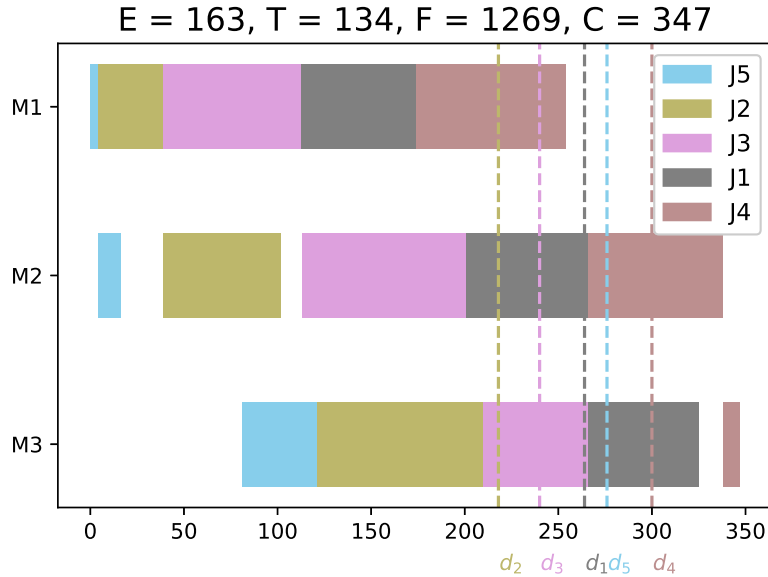


Fonte: Elaborado pelo autor.

O problema de sequenciamento pode ser resolvido considerando diversos objetivos. Sejam C_{jk} o instante de conclusão da tarefa j na máquina k e d_j a data de vencimento da tarefa j , se define alguns dos principais objetivos tratados na literatura ([29]). O *makespan* é definido como sendo o instante de conclusão da última tarefa da sequência na última máquina, ou seja, a duração total da programação. Matematicamente $C_{max} = \max\{C_{1m}, C_{2m}, \dots, C_{nm}\}$. Minimizar C_{max} é um objetivo que visa maximizar a utilização das máquinas ([28]). Outra medida que se relaciona com o aproveitamento das máquinas, é o tempo total de fluxo, ou *flowtime* (F), calculado como $F = \sum_j C_{jm}$. Outros dois objetivos são minimizar o adiantamento total ($\min E$) e minimizar o atraso total ($\min T$). Considerando $E_j = \max\{0, d_j - C_{jm}\}$ como o adiantamento da tarefa j e $T_j = \max\{0, C_{jm} - d_j\}$ o atraso da tarefa j , são definidos $E = \sum_j E_j$ e $T = \sum_j T_j$. Essas duas medidas são de pontualidade da entrega das tarefas e, quando minimizada a soma delas, é obtido o objetivo *just-in-time*, que busca concluir a tarefa no prazo. O objetivo *just-in-time*, assim definido, será o principal objetivo abordado nessa tese. Para saber de outros objetivos do sequenciamento veja, por exemplo, [28, 29, 30].

É natural que dado um sequenciamento de tarefas, este pode ser benéfico de acordo com alguns objetivos, mas ruim para outros. Na Figura 18 está exibido um gráfico de Gantt com o valor das medidas E , T , F e C_{max} . Nesse sequenciamento o objetivo considerado durante a otimização foi a minimização de uma ponderação entre adiantamento total e atraso total. Perceba que para a medida F , este mesmo sequenciamento poderia ser reduzido apenas com o ajuste das conclusões das tarefas $J5$ (barra azul) e $J2$ (barra cáqui). Isso não ocorreu porque seria prejudicial para a minimização do adiantamento total.

Figura 18 – Sequenciamento e valor das medidas E , T , F e C_{max}



Fonte: Elaborado pelo autor.

Extensas revisões da literatura sobre o problema de sequenciamento em *flow shop* podem ser encontradas em [31, 32, 33]. Além disso, revisões de *flow shop* multiobjetivo foram publicadas em [34, 3, 35, 29]. Há na literatura uma predominância de abordagens heurísticas e metaheurísticas para resolver o problema de sequenciamento em *flow shop* (e.g. [34, 36, 37, 38]), o que possivelmente explica-se por razões práticas, devido ao grande porte dos problemas reais ([3]). Os métodos exatos, como o *branch-and-bound*, são menos explorados. Geralmente eles estão acompanhados de heurísticas para determinar soluções iniciais ou limitantes para o problema (e.g. [30]). Isto evidencia o potencial de pesquisa considerando modelos matemáticos, métodos de solução exata e métodos de decomposição, como é o caso do presente trabalho.

Nas próximas seções, apresenta-se um recorte da literatura dos problemas de sequenciamento *just-in-time flow shop* e modelos de otimização matemática para representá-lo.

3.1 Revisão de literatura sobre os problemas de *flow shop* com objetivos *just-in-time*

Existem mais de uma forma de trabalhar com o objetivo *just-in-time* no problema de sequenciamento em *flow shop*. Uma delas é utilizar o adiantamento total e o atraso total. Outras formas podem ser encontradas em [39, 40]. Esse recorte foi definido utilizando a notação de três campos presente em [41] com $\alpha = F_m$, β qualquer e γ contendo pelo menos um dos seguintes objetivos: ET , wET , (E, T) ou (wE, wT) .

Nesse recorte da literatura foi possível notar que os objetivos adiantamento total e atraso total sempre são tratados simultaneamente em um único objetivo através de uma soma ponderada. Em sua maioria, são considerados pesos unitários para os objetivos (*e.g.* [42, 43, 44, 45]). Foi visto na Seção 2.1 que essa abordagem corresponde a uma iteração do método da soma ponderada.

Dessas pesquisas, poucas trabalham apenas com métodos exatos. Em [46], os autores fazem um estudo comparativo entre modelos posicionais e sequenciais apenas utilizando solvers comerciais. Em [47] também é realizado um estudo computacional para avaliar um modelo posicional para uma outra variante do problema. Por trabalharem com métodos exatos, no estudo computacional de ambas as pesquisas são utilizados instâncias de pequeno porte, máximo de 20 tarefas e máximo de 10 máquinas. Para melhorar o desempenho dos solvers, em [47], são propostos limitantes inferiores e superiores para o problema.

Investir em métodos exatos tem grande importância até mesmo para desenvolver e/ou validar novas heurísticas. Em [43], os autores transformaram o problema de m máquinas em máquina única e propuseram um método *branch-and-bound* para poder avaliar o desempenho de heurísticas. Para instâncias de pequeno porte constatou-se que há um gap significativo entre o valor obtido pelas heurísticas quando comparado ao do método exato (valor ótimo). Isso mostrou a necessidade de aprimorar as heurísticas analisadas pelos autores.

Para que esse tipo de comparação seja possível em instâncias de grande porte, é necessário melhorar os métodos exatos de alguma forma. Em [48] o autor propõe um método híbrido combinando a resolução de um problema inteiro misto com limitantes obtidos por uma heurística. Apesar dessa prática ser comum na literatura, nesse recorte somente essa pesquisa utiliza dessa metodologia.

Os métodos mais utilizados pela literatura são os heurísticos e metaheurísticos. Em [49, 50, 51, 42] são propostos diferentes heurísticas construtivas e de composição. Em [42] os autores propuseram um método *beam search* com uma heurística construtiva capaz de aprimorar resultados obtidos em [49]. Essa técnica também é utilizada com métodos exatos, como o *branch-and-bound*. É uma forma de utilizar métodos exatos de forma heurística. Em [50] é proposta uma heurística construtiva ACH1 e através dela, são propostas 3 heurísticas compostas baseadas na ACH1 com diferentes estratégias de busca. Em seus testes computacionais, a heurística ACH4 que utiliza busca local iterativa obteve melhores resultados. Essa heurística foi utilizada posteriormente em [51] no problema de flow shop permitindo tempos ociosos para diminuir o adiantamento total.

Os métodos metaheurísticos em sua maioria são algoritmos genéticos. Na pesquisa [52] são comparados diferentes métodos metaheurísticos e apresentadas vantagens e limitações para diferentes cenários operacionais. Em [44] é abordado o problema sem tempo de espera e com tempos de *setup* dependendo do sequenciamento. É utilizado um modelo inteiro misto para instâncias pequenas e um algoritmo genético para instâncias de médio e grande

porte. Prata e Fuchigami [53] exploram o problema considerando o bloqueio de tarefas em sua posição na máquina. O problema é resolvido utilizando um algoritmo genético guloso iterativo. Outros métodos também são explorados. Em [54] aborda-se o problema sem tempo de espera e utiliza um *parallel simulated annealing* para resolver o problema. Behnamian [55] faz uma abordagem biobjetiva sendo o primeiro objetivo o makespan e o segundo a soma do adiantamento total e atraso total. Para resolver o problema faz a ponderação dos objetivos e utiliza um algoritmo genético baseado em colônia de formigas. Tavana et al. [45] utiliza dos mesmos objetivos para explorar a variante de bloqueio de tarefas. O problema é resolvido combinando o método ε -restrito e métodos metaheurísticos multiobjetivos. Para todas essas pesquisas o adiantamento total e o atraso total são tratados simultaneamente em um único objetivo. Mesmo em [55, 45] que utilizam métodos multiobjetivos, não há uma separação.

Mesmo quando o objetivo *just-in-time* é proposto de forma mais genérica, utilizando pesos não unitários, as pesquisas não variam esses pesos para implementar o método da soma ponderada, apenas fixam um par de pesos e resolvem o problema de forma monoobjetivo uma única vez, ou seja, uma iteração do método da soma ponderada. Isso pode ser visto em [56] em seu método *branch-and-bound* e em [57] transformando o problema em máquina única e utilizando programação dinâmica para resolvê-lo.

Para essa forma de abordar o problema, as metaheurísticas continuam sendo as metodologias mais utilizadas, predominando os algoritmos genéticos, [58] propõem um modelo matemático que é resolvido pelo algoritmo proposto. Em [59] considera-se três objetivos que são ponderados em uma função objetivo escalar e resolve-se esse problema de forma monoobjetivo utilizando um algoritmo robusto. Um modelo com restrições não lineares é proposto em [60]. Em [61] aborda-se o problema integrado com manutenção das máquinas. Também são encontrados métodos baseados em *simulated annealing* em [62] para resolver um problema com parâmetros incertos, métodos de busca local iterativo em [63] para resolver uma variação com bloqueio de tarefas e em [64] são propostos quatro modelos matemáticos e quatro métodos baseados no coronavírus. Métodos baseados em *particle swarm optimization* são vistos em [65] para resolver o problema com *setup* dependendo do sequenciamento e em [66] o problema com *setup* dependendo do grupo de tarefas.

A Tabela 2 apresenta a lista das pesquisas dentro do recorte feito da literatura. Até o presente momento e conhecimento, essa tese é a primeira a abordar o problema de sequenciamento permutacional em *flow shop* biobjetivo com o adiantamento total e o atraso total sendo tratados em objetivos distintos. Na Tabela 2, a primeira coluna lista o nome dos autores e na segunda os problemas resolvidos. Na terceira coluna apresentam-se o tipo de modelo, sendo posicional (P), sequencial (S) e temporal (T). A quarta coluna, estão descritos os métodos de solução utilizados: heurístico (H), metaheurístico (MH), *branch-and-bound* (B&B), otimização inteira mista (MIP) e transformação de problema

(Tr). Na última coluna, são exibidas as faixas de variação do porte das instâncias utilizadas nos testes computacionais. Em geral, os modelos matemáticos propostos na literatura são apenas descritos e, em alguns casos, apresentam pequenos comentários sobre sua implementação e desempenho computacional.

Tabela 2 – Autores que resolveram o FSP com objetivo JIT

Autoria	Problema	MIP	Método	Porte (n, m)
Chandra et al. (2009) [49]	$F_m prmu, d_j = d ET$	S	H	$([5, 100], [5, 20])$
Ramezani et al. (2010) [58]	$F_m prmu, r_j, \delta_{ij} wET$	S	MH	$([3, 30], [3, 30])$
Madhushini e Rajendran (2011) [56]	$F_m prmu wET + wF$		$B\&B$	$([9, 16], [5, 20])$
Akhshabi et al. (2012) [62]	$F_m d_j fuzzy wET$	P	MH	$([3, 50], [3, 45])$
Roconi e Birgin (2012) [46]	$F_m block ET$	P, S	MIP	$([5, 20], [3, 10])$
Arabameri e Salmasi (2013) [65]	$F_m nwt, s_{ijk} wET$	S	MH	$([5, 100], [2, 6])$
Schaller e Valente (2013) [52]	$F_m prmu ET$		MH	$([8, 100], [5, 20])$
Behnamian (2014) [55]	$F_m s_{ijk} C_{max}, ET$		MH	$([6, 100], [2, 10])$
Mhallah (2014) [48]	$F_m ET$	P	H, MIP	$([10, 100], [2, 12])$
Mohammadi (2015) [59]	$F_m wET + C_{max} + N_j$		MH	$([20, 200], [5, 20])$
Fernandez-Viagas et al. (2016) [50]	$F_m prmu ET$		H	$([50, 350], [10, 50])$
Hamdi e Loukil (2017) [47]	$prmu, \theta_{jk} ET$	P	MIP	$([5, 20], [3, 10])$
Keshavarz et al. (2019) [66]	$F_m prmu, s_{hgi}, fmls wET$	S	MH	$([4, 100], [2, 6])$
Rezaeian et al. (2019) [60]	$F_m prmp wET$	P	MH	$([2, 40], [2, 5])$
Schaller e Valente (2019) [51]	$F_m ET$		H	$([15, 100], [5, 20])$
Birgin et al. (2020) [42]	$F_m prmu, d_j = d ET$	S	H	$([5, 100], [5, 20])$
Schaller e Valente (2020) [43]	$F_m nwt ET$	P	Tr, H, $B\&B$	$([8, 100], [5, 20])$
Branda et al. (2021) [61]	$F_m maintenance wET$		MH	$([20, 40], [5, 5])$
Guevara-Guevara et al. (2022) [44]	$F_m nwt, s_{ijk} ET$	S	MH	$([4, 100], [3, 20])$
Missaoui e Boujelbene (2022) [63]	$F_m block, d_j = [d_j^-, d_j^+] wET$		MH	$([20, 200], [5, 20])$
Fuchigami e Prata (2023) [64]	$F_m prmu wET + wd$	P, S	MH	$([20, 500], [5, 20])$
Geng et al. (2023) [57]	$F_m prmu, p_{ij} = p_j, d_j = d, rej wET + wd + e_j$		Tr, PD	$(7, 3)$
Karacan et al. (2023) [54]	$F_m nwt ET$	S	MH	$([20, 500], [5, 20])$
Prata e Fuchigami (2023) [53]	$F_m block ET$	P	MH	$([20, 100], [5, 20])$
Tavana et al. (2023) [45]	$F_m block C_{max}, ET$	P	MH	$([50, 500], [5, 40])$
Nessa tese	$F_m prmu E, T$	P	$B\&B$	$([5, 100], [3, 20])$

3.2 Modelagem matemática do problema de *flow shop*

Modelagens matemáticas para o problema de sequenciamento em *flow shop* têm sido pouco exploradas na literatura ([35]). Nos trabalhos que apresentam modelos matemáticos de otimização para o problema, definem-se três tipos: sequencial, posicional e temporal. Em [29], os autores observam que para variantes do FSP, os tipos de modelos mais utilizados são os sequenciais e posicionais. Os modelos temporais tornam-se restritivos pelo grande aumento no número de variáveis. Um modelo da literatura que tem um bom controle no número de variáveis é o proposto em [67].

Em Wilson [67] é proposto um modelo posicional para o FSP e são definidos dois objetivos possíveis para o problema, minimizar C_{max} e minimizar F . Nessa modelagem o autor trabalha com a posição h da tarefa j na sequência (x_{jh}) e com os instantes de início das tarefas, em determinada posição h , na máquina k (S_{hk}). Todos os parâmetros e variáveis do modelo são apresentados a seguir.

Índices

- j = índice da tarefa, com $j = 1 \dots n$
- h = índice da posição em que a tarefa será processada, com $h = 1 \dots n$

- k = índice da máquina, com $k = 1 \dots m$

Parâmetros

- p_{jk} = tempo de processamento da tarefa j na máquina k

Variáveis

- $x_{jh} = 1$ se a tarefa j está na posição h para ser processada e 0 caso contrário
- S_{hk} = instante de início do processamento da tarefa da posição h na máquina k

O modelo de Wilson [67] é apresentado em (3.1)-(3.9).

$$\text{Min } C_{max} = S_{nm} + \sum_{j=1}^n p_{jm} x_{jn} \quad \text{ou} \quad \text{Min } F = \sum_{h=1}^n S_{hm} \quad (3.1)$$

$$S.a : \sum_{j=1}^n x_{jh} = 1, \quad h = 1 \dots n \quad (3.2)$$

$$\sum_{h=1}^n x_{jh} = 1, \quad j = 1 \dots n \quad (3.3)$$

$$S_{11} = 0 \quad (3.4)$$

$$S_{1,k+1} = S_{1k} + \sum_{j=1}^n p_{jk} x_{j1}, \quad k = 1 \dots m-1 \quad (3.5)$$

$$S_{h+1,1} = S_{h1} + \sum_{j=1}^n p_{j1} x_{jh}, \quad h = 1 \dots n-1 \quad (3.6)$$

$$S_{h,k+1} \geq S_{h,k} + \sum_{j=1}^n p_{jk} x_{jh}, \quad h = 2 \dots n, \quad k = 1 \dots m-1 \quad (3.7)$$

$$S_{h+1,k} \geq S_{hk} + \sum_{j=1}^n p_{jk} x_{jh}, \quad h = 1 \dots n-1, \quad k = 2 \dots m \quad (3.8)$$

$$S_{hk} \geq 0, \quad x_{jh} \in \mathbb{B} \quad (3.9)$$

Em (3.1) é definido o critério de otimização, minimizar o *makespan* ou minimizar o *flowtime*, respectivamente, como um dos objetivos do problema. As restrições (3.2) e (3.3) são restrições de designação. Em (3.2) determina-se que para cada posição apenas uma tarefa será processada, e em (3.3), determina-se que cada tarefa ocupará apenas uma posição. A restrição (3.4) garante que o início do processamento da tarefa na posição 1 na máquina 1 comece no instante 0. As restrições (3.5) calculam os instantes de início da tarefa na posição 1 em todas as máquinas. As restrições (3.6) determinam os instantes de início de todas as tarefas na máquina 1. As restrições (3.7) e (3.8) garantem os instantes de início de cada tarefa nas máquinas seguintes e os instantes de início das tarefas seguintes em cada máquina. Em (3.9) é dado o domínio das variáveis.

No início do capítulo discutiu-se sobre o C_{max} e o F serem medidas que são calculadas a partir do instante de conclusão das tarefas. No modelo (3.1)-(3.9) utilizam-se esses mesmos

objetivos, porém as variáveis utilizadas para cálculos são os instantes de início. Com isso, é notório que existe uma relação entre as variáveis de início e as de conclusão que é dada por $C_{hk} = S_{hk} + \sum_{j=1}^n p_{jk}x_{jh}$. Com essa relação propõe-se o modelo (3.10)-(3.18) que é adaptado do modelo (3.1)-(3.9).

$$\text{Min } C_{max} = C_{nm} \text{ ou } \text{Min } F = \sum_{h=1}^n C_{hm} \quad (3.10)$$

$$S.a : \sum_{j=1}^n x_{jh} = 1, \quad h = 1 \dots n \quad (3.11)$$

$$\sum_{h=1}^n x_{jh} = 1, \quad j = 1 \dots n \quad (3.12)$$

$$C_{11} = \sum_{j=1}^n p_{j1}x_{j1} \quad (3.13)$$

$$C_{1,k+1} = C_{1k} + \sum_{j=1}^n p_{j,k+1}x_{j1}, \quad k = 1 \dots m-1 \quad (3.14)$$

$$C_{h+1,1} = C_{h1} + \sum_{j=1}^n p_{j1}x_{j,h+1}, \quad h = 1 \dots n-1 \quad (3.15)$$

$$C_{h,k+1} \geq C_{h,k} + \sum_{j=1}^n p_{j,k+1}x_{j,h}, \quad h = 2 \dots n, \quad k = 1 \dots m-1 \quad (3.16)$$

$$C_{h+1,k} \geq C_{hk} + \sum_{j=1}^n p_{jk}x_{j,h+1}, \quad h = 1 \dots n-1, \quad k = 2 \dots m \quad (3.17)$$

$$C_{hk} \geq 0, \quad x_{jh} \in \mathbb{B} \quad (3.18)$$

Em (3.10) define-se um dos objetivos para o problema, minimizar o *makespan* ou minimizar o *flowtime*, respectivamente. As restrições (3.11)-(3.18) são análogas às restrições (3.2)-(3.9).

Em Abreu e Fuchigami [30], os autores propuseram quatro modelos para o FSP com o objetivo de minimizar tempos ociosos e de espera. Dentre os quatro modelos, o que teve melhor desempenho foi uma adaptação do modelo (3.1)-(3.9) (modelo 4). A seguir apresentam-se as variáveis de decisão que se diferem do modelo (3.10)-(3.9).

Variáveis

- I_{hk} = tempo ocioso da máquina k antes da tarefa da posição h
- W_{hk} = tempo de espera da tarefa na posição h até iniciar na máquina k

O modelo 4 proposto em [30] é exibido em (3.19)-(3.27).

$$\text{Min} \sum_{h=1}^n \sum_{k=1}^m (I_{hk} + W_{hk}) \quad (3.19)$$

$$S.a : \sum_{h=1}^n x_{jh} = 1, \quad j = 1 \dots n \quad (3.20)$$

$$\sum_{j=1}^n x_{jh} = 1, \quad h = 1 \dots n \quad (3.21)$$

$$\sum_{j=1}^n (p_{jk} x_{j1}) + I_{1k} + W_{1k} = I_{1,k+1}, \quad k = 1 \dots m-1 \quad (3.22)$$

$$C_{11} = \sum_{j=1}^n p_{j1} x_{j1} \quad (3.23)$$

$$C_{h,k+1} - C_{hk} - W_{hk} = \sum_{j=1}^n p_{j,k+1} x_{jh}, \quad h = 1 \dots n, \quad k = 1 \dots m-1 \quad (3.24)$$

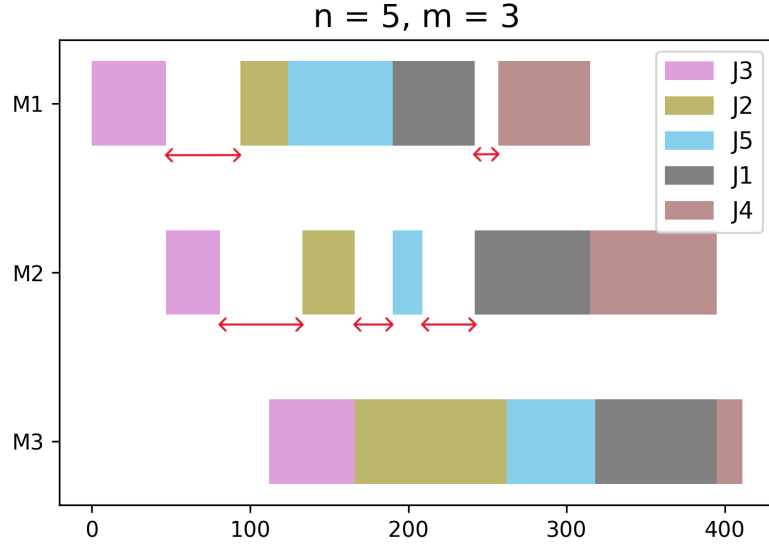
$$C_{h+1,k} - C_{hk} - I_{h+1,k} = \sum_{j=1}^n p_{j,k} x_{j,h+1}, \quad h = 1 \dots n-1, \quad k = 1 \dots m \quad (3.25)$$

$$\sum_{k=1}^m W_{1k} = 0 \quad (3.26)$$

$$C_{hk}, I_{hk}, W_{hk} \geq 0, \quad x_{jh} \in \mathbb{B} \quad (3.27)$$

O objetivo desse modelo (3.19)-(3.27) é minimizar os tempos ociosos das máquinas e de espera das tarefas. Uma forma de visualizar a minimização dessa medida é reduzir as lacunas no gráfico de Gantt, como mostra a Figura 19. As restrições (3.20) e (3.21) são restrições de designação, garantindo que cada tarefa ocupe apenas uma posição e que cada posição tenha exatamente uma tarefa. As restrições (3.22) calculam o tempo ocioso das máquinas até iniciar o processamento da tarefa de posição 1. A restrição (3.23) calcula o instante de conclusão da tarefa de posição 1 na máquina 1. As restrições (3.24) e (3.25) calculam os instantes de conclusão e, respectivamente, o tempo de espera das tarefas e o tempo ocioso das máquinas. A restrição (3.26) garante que a tarefa de posição 1 não terá tempo de espera e em (3.27) é dado o domínio das variáveis.

Figura 19 – Sequenciamento com tempos ociosos



Fonte: Elaborado pelo autor.

Com os modelos (3.10)-(3.18) e (3.19)-(3.27) é interessante notar que ao adicionar variáveis de folga em (3.16) e (3.17) são obtidas as mesmas restrições de (3.24) e (3.25).

Para abordar o objetivo *just-in-time* é proposto uma adaptação do modelo (3.10)-(3.18). Como os objetivos E e T são definidos pela função máximo, é necessários adicionar as variáveis E_h e T_h para $h = 1, \dots, n$ e as restrições (3.34) e (3.35) para linearizar, respectivamente, os objetivos. O modelo biobjetivo proposto é apresentado em (3.28)-(3.36).

Índices:

- j = índice da tarefa, com $j = 1 \dots n$;
- h = índice da posição em que a tarefa será processada, com $h = 1 \dots n$;
- k = índice da máquina, com $k = 1 \dots m$.

Parâmetros:

- p_{jk} = tempo de processamento da tarefa j na máquina k ;
- d_j = data de vencimento da tarefa j .

Variáveis:

- $x_{jh} = 1$ se a tarefa j está na posição h para ser processada e 0 caso contrário;
- C_{hk} = instante de conclusão da tarefa da posição h na máquina k ;

- E_h = adiantamento da tarefa da posição h ;
- T_h = atraso da tarefa da posição h .

$$\text{Min } \left(\sum_{h=1}^n E_h, \sum_{h=1}^n T_h \right) \quad (3.28)$$

$$\text{s.t. } \sum_{j=1}^n x_{jh} = 1, \quad h = 1, \dots, n \quad (3.29)$$

$$\sum_{h=1}^n x_{jh} = 1, \quad j = 1, \dots, n \quad (3.30)$$

$$C_{11} \geq \sum_{j=1}^n p_{j1} x_{j1} \quad (3.31)$$

$$C_{h,k+1} \geq C_{hk} + \sum_{j=1}^n p_{j,k+1} x_{jh}, \quad h = 1, \dots, n, k = 1, \dots, m-1 \quad (3.32)$$

$$C_{h+1,k} \geq C_{hk} + \sum_{j=1}^n p_{jk} x_{j,h+1}, \quad h = 1, \dots, n-1, k = 1, \dots, m \quad (3.33)$$

$$\sum_{j=1}^n (d_j x_{jh}) - C_{hm} \leq E_h, \quad h = 1, \dots, n \quad (3.34)$$

$$C_{hm} - \sum_{j=1}^n (d_j x_{jh}) \leq T_h, \quad h = 1, \dots, n \quad (3.35)$$

$$x_{jh} \in \{0, 1\}, C_{hk} \geq 0, E_h \geq 0, T_h \geq 0, \quad j = 1, \dots, n, h = 1, \dots, n, k = 1, \dots, m. \quad (3.36)$$

Em (3.28) é definida a função biobjetivo contendo o adiantamento total como primeira coordenada do espaço objetivo e o atraso total como segunda coordenada do espaço objetivo. As restrições (3.29) e (3.30) são de designação. A restrição (3.31) permite que a tarefa de posição 1 inicie na máquina 1 no instante 0 ou após. As restrições (3.32) calculam os instantes de conclusão das tarefas nas máquinas seguintes e as restrições (3.33) calculam os instantes de conclusão das tarefas seguintes em cada máquina. As restrições (3.34) e (3.35) são usadas para linearizar a função objetivo e calculam, respectivamente, os objetivos adiantamento total e atraso total. Em (3.36) é dado o domínio das variáveis.

Ao comparar os modelos (3.10)-(3.18) e (3.28)-(3.36), além da mudança dos objetivos, houve uma alteração nas restrições que calculam os instantes de conclusão. No modelo proposto por Wilson [67], como os objetivos tratados eram o $\text{Min } C_{\max}$ ou $\text{Min } F$, exigir que a conclusão da primeira tarefa nas máquinas seguintes e a conclusão das tarefas seguintes na máquina 1 sejam satisfeitas com igualdade, ajuda restringindo mais o modelo e beneficia o objetivo escolhido por ser de aproveitamento das máquinas. Ao tratar a minimização do adiantamento total, que é um objetivo de pontualidade, manter a igualdade nas restrições faria com que, em geral, a primeira tarefa processada sempre fosse concluída com adiantamento, prejudicando a minimização desse objetivo. Portanto, relaxar as restrições (3.13)-(3.15) é uma forma de atender os objetivos de pontualidade.

Um aspecto importante na representação de problemas está no número de variáveis utilizadas para modelá-lo. O número de variáveis de um modelo não é o fator decisivo da qualidade do modelo, porém, ter menos variáveis pode contribuir para isso. Em [67] é apresentado a dimensão do modelo (3.10)-(3.18) como $2mn + n - m$ restrições, n^2 variáveis binárias e mn variáveis contínuas. O modelo (3.28)-(3.36) possui um aumento de $2n$ restrições e variáveis contínuas. Na Tabela, 3 apresenta-se a dimensão do modelo (3.10)-(3.18) com o objetivo $Min C_{max}$ ou $Min F$ e do modelo (3.28)-(3.36). Também, apresentam-se os valores numéricos para instâncias de pequeno e de grande porte utilizadas no estudo computacional do Capítulo 5.

Tabela 3 – Dimensão dos modelos (3.10)-(3.18) e (3.28)-(3.36)

	Modelo (3.10)-(3.18)			Modelo (3.28)-(3.36)		
	Geral	(5, 3)	(100, 10)	Geral	(5, 3)	(100, 10)
Restrições	$2mn + n - m$	32	2090	$2mn + 3n - m$	42	2290
Var. Binárias	n^2	25	10000	n^2	25	10000
Var. Contínuas	mn	15	1000	$2n + mn$	25	1200

4 Método de decomposição de Benders aplicado no problema de sequenciamento

No Apêndice 6 é discutida a teoria geral do método de decomposição de Benders, para o leitor não familiarizado com essa teoria. Um questionamento possível de ser feito a respeito da decomposição de Benders é: só é benéfica sua utilização quando há uma estrutura conveniente? Possivelmente, não. Também pode ser levado em conta, se há um conjunto de restrições consideradas complicadas e outros conjuntos de restrições que são mais fáceis ao fixar algumas variáveis. Essa será a ideia utilizada para decompor os modelos. Primeiro será aplicado a decomposição de Benders no modelo (3.10)-(3.18), com o objetivo de minimizar o *makespan*, por ser mais simples e para auxiliar na decomposição do modelo (3.28)-(3.36), com o objetivo JIT.

Ao analisar o FSP, o fator que mais interfere na qualidade da solução é a ordem que as tarefas serão processadas. Ao fixar uma determinada ordem, as outras restrições vão basicamente calcular os instantes de conclusão e o valor dos objetivos. Então, no modelo (3.10)-(3.18) as restrições consideradas complicadas serão as restrições (3.11) e (3.12) de designação (*e.g.* [68]). Todo o restante será o subproblema $\Phi(x)$ cujo dual será usado para gerar os cortes de Benders. Considerando o *makespan*, $\Phi(x)$ é dado por $\Phi(x) = \text{Min}\{C_{nm} \mid (3.13) - (3.17), C_{hk} \geq 0\}$. O dual do problema $\Phi(x)$, para $\bar{x}$ satisfazendo (3.11) e (3.12), é apresentado em (4.1)-(4.11).

$$\text{Max} \quad (\sum_{j=1}^n p_{j1} \bar{x}_{j1}) \lambda^1 + \sum_{k=1}^{m-1} (\sum_{j=1}^n p_{j,k+1} \bar{x}_{j1}) \lambda_k^2 + \sum_{k=1}^{n-1} (\sum_{j=1}^n p_{j1} \bar{x}_{j,h+1}) \lambda_h^3 + \sum_{h=2}^n \sum_{k=1}^{m-1} (\sum_{j=1}^n p_{j,k+1} \bar{x}_{jh}) \lambda_{hk}^4 + \sum_{h=1}^{n-1} \sum_{k=2}^m (\sum_{j=1}^n p_{jk} \bar{x}_{jh}) \lambda_{hk}^5 \quad (4.1)$$

$$\text{S.a : } \lambda^1 - \lambda_1^2 - \lambda_1^3 \leq 0 \quad (4.2)$$

$$\lambda_{n-1}^3 - \lambda_{n1}^4 \leq 0 \quad (4.3)$$

$$\lambda_{m-1}^2 - \lambda_{1m}^5 \leq 0 \quad (4.4)$$

$$\lambda_{n,m-1}^4 + \lambda_{n-1,m}^5 \leq 1 \quad (4.5)$$

$$\lambda_h^3 - \lambda_{h+1}^3 - \lambda_{h+1,1}^4 \leq 0, \quad h = 1 \dots n-2 \quad (4.6)$$

$$\lambda_k^2 - \lambda_{k+1}^2 - \lambda_{1,k+1}^5 \leq 0, \quad k = 1 \dots m-2 \quad (4.7)$$

$$\lambda_{nk}^4 - \lambda_{n,k+1}^4 + \lambda_{n-1,k+1}^5 \leq 0, \quad k = 1 \dots m-2 \quad (4.8)$$

$$\lambda_{h+1,m-1}^4 + \lambda_{hm}^5 - \lambda_{h+1,m}^5 \leq 0, \quad h = 1 \dots n-2 \quad (4.9)$$

$$\lambda_{h+1,k}^4 - \lambda_{h+1,k+1}^4 + \lambda_{h,k+1}^5 - \lambda_{h+1,k+1}^5 \leq 0, \quad h = 1 \dots n-2, \quad k = 1 \dots m-2 \quad (4.10)$$

$$\lambda^1, \lambda_k^2, \lambda_h^3, \lambda_{hk}^4, \lambda_{hk}^5 \geq 0 \quad (4.11)$$

Um ponto interessante para o problema (3.10)-(3.18) é que dado $\bar{x}$ satisfazendo (3.11) e (3.12), $\Phi(\bar{x})$ sempre terá solução factível. Como o primal de $\Phi(\bar{x})$ sempre possui solução

factível, seu dual (D_Φ) também sempre terá soluções factíveis e não haverá raios extremos (Definição 19). Dessa forma, o Problema Mestre de Benders (PMB) para o FSP é dado por (4.12)-(4.16), sendo (4.15) os cortes de otimalidade.

$$\text{Min } v \quad (4.12)$$

$$S.a : \sum_{j=1}^n x_{jh} = 1, \quad h = 1 \dots n \quad (4.13)$$

$$\sum_{h=1}^n x_{jh} = 1, \quad j = 1 \dots n \quad (4.14)$$

$$\begin{aligned} & (\sum_{j=1}^n p_{j1}x_{j1})\bar{\lambda}^1 + \sum_{k=1}^{m-1} (\sum_{j=1}^n p_{j,k+1}x_{j1})\bar{\lambda}_k^2 \\ & + \sum_{k=1}^{n-1} (\sum_{j=1}^n p_{j1}x_{j,h+1})\bar{\lambda}_h^3 + \sum_{h=2}^n \sum_{k=1}^{m-1} (\sum_{j=1}^n p_{j,k+1}x_{jh})\bar{\lambda}_{hk}^4 \\ & + \sum_{h=1}^{n-1} \sum_{k=2}^m (\sum_{j=1}^n p_{jk}x_{jh})\bar{\lambda}_{hk}^5 \leq v, \quad \forall \lambda \in D_\Phi \end{aligned} \quad (4.15)$$

$$x_{jh} \in \mathbb{B} \quad (4.16)$$

Como o conjunto de pontos extremos costuma ser muito grande, o PMB é resolvido pelo método de planos de corte, ou seja, os cortes de otimalidade são adicionados iterativamente (*e.g.* [69, 70]). Além disso, sendo o PMB um problema de otimização inteira, é necessário utilizar, por exemplo, o método *branch-and-bound* ou *branch-and-cut* para gerar uma solução factível para o problema dual. Como esse processo é muito custoso computacionalmente, na literatura foi proposto o método *branch-and-Benders-cut* como uma alternativa para acelerar a convergência do método (*e.g.* [71, 72, 73]). Nele os cortes são adicionados cada vez que é encontrada uma solução factível.

A ideia de aplicar a decomposição de Benders no problema (3.28)-(3.36) é a mesma. As restrições (3.29) e (3.30) são as consideradas complicadas e todo o restante irá para o subproblema $\Phi(x)$ cujo dual será usado para gerar os cortes de Benders. Porém, o que é o dual de um problema multiobjetivo? Esse é um tema que apresenta uma literatura extremamente escassa. No limite das buscas da literatura dessa tese, os trabalhos de Isermann [74] e TenHuisen e Wiecek [9] são as principais referências a respeito da dualidade multiobjetivo. Nesses trabalhos, assim como em outros [75, 76], antes de definir o problema dual é utilizado algum método de escalarização transformando o problema multiobjetivo em uma sequência de subproblemas monoobjetivos. Com isso, é possível utilizar todos os resultados já existentes dos métodos de escalarização e de dualidade monoobjetivo.

Aplicando o método ε -restrito aumentado, o modelo escalar resultante que é utilizado para cada iteração está apresentado em (4.17)-(4.26), com $E^{\text{lexmin}21}$ sendo o valor do adiantamento total obtido na resolução do problema lexicográfico com $Ord = \{2, 1\}$.

$$\text{Min } \rho \cdot \sum_{h=1}^n E_h + \sum_{h=1}^n T_h \quad (4.17)$$

$$\text{s.a. } \sum_{j=1}^n x_{jh} = 1, \quad h = 1, \dots, n \quad (4.18)$$

$$\sum_{h=1}^n x_{jh} = 1, \quad j = 1, \dots, n \quad (4.19)$$

$$C_{11} \geq \sum_{j=1}^n p_{j1} x_{j1} \quad (4.20)$$

$$C_{h,k+1} \geq C_{hk} + \sum_{j=1}^n p_{j,k+1} x_{jh}, \quad h = 1, \dots, n, k = 1, \dots, m-1 \quad (4.21)$$

$$C_{h+1,k} \geq C_{hk} + \sum_{j=1}^n p_{jk} x_{jh}, \quad h = 1, \dots, n-1, k = 1, \dots, m \quad (4.22)$$

$$\sum_{j=1}^n (d_j x_{jh}) - C_{hm} \leq E_h, \quad h = 1, \dots, n \quad (4.23)$$

$$C_{hm} - \sum_{j=1}^n (d_j x_{jh}) \leq T_h, \quad h = 1, \dots, n \quad (4.24)$$

$$\sum_{h=1}^n E_h \leq E^{\text{lexmin}21} - \varepsilon \quad (4.25)$$

$$x_{jh} \in \{0, 1\}, C_{hk} \geq 0, E_h \geq 0, T_h \geq 0, \quad j = 1, \dots, n, h = 1, \dots, n, k = 1, \dots, m. \quad (4.26)$$

Para um $\bar{x}$ que satisfaça (4.18) e (4.19), o dual do subproblema $\Phi(\bar{x})$ é apresentado em (4.27)-(4.39).

$$\begin{aligned}
& (\sum_{j=1}^n p_{j1} \bar{x}_{j1}) \lambda^1 + \sum_{h=1}^n \sum_{k=1}^{m-1} (\sum_{j=1}^n p_{j,k+1} \bar{x}_{jh}) \lambda_{hk}^2 + \\
Max \quad & \sum_{h=1}^{n-1} \sum_{k=1}^m (\sum_{j=1}^n p_{jk} \bar{x}_{jh}) \lambda_{hk}^3 + \sum_{h=1}^n (\sum_{j=1}^n d_j \bar{x}_{jh}) (\lambda_h^4 - \lambda_h^5) + \\
& (E^{lexmin21} - \varepsilon) \lambda^6
\end{aligned} \tag{4.27}$$

$$S.a : \lambda^1 - \lambda_{11}^2 - \lambda_{11}^3 \leq 0 \tag{4.28}$$

$$\lambda_{n1}^2 - \lambda_{n-1,1}^3 \leq 0 \tag{4.29}$$

$$\lambda_{1,m-1}^2 - \lambda_{1m}^3 + \lambda_1^4 - \lambda_1^5 \leq 0 \tag{4.30}$$

$$\lambda_{n,m-1}^2 + \lambda_{n-1,m}^3 + \lambda_n^4 - \lambda_n^5 \leq 0 \tag{4.31}$$

$$\lambda_{1k}^2 - \lambda_{1,k+1}^2 - \lambda_{1,k+1}^3 \leq 0, \quad k = 1 \dots m-2 \tag{4.32}$$

$$-\lambda_{h1}^2 + \lambda_{h-1,1}^3 - \lambda_{h1}^3 \leq 0, \quad h = 2 \dots n-1 \tag{4.33}$$

$$\lambda_{nk}^2 - \lambda_{n,k+1}^2 + \lambda_{n-1,k+1}^3 \leq 0, \quad k = 1 \dots m-2 \tag{4.34}$$

$$\lambda_{h,m-1}^2 + \lambda_{h-1,m}^3 - \lambda_{hm}^3 + \lambda_h^4 - \lambda_h^5 \leq 0, \quad h = 2 \dots n-1 \tag{4.35}$$

$$\lambda_{hk}^2 - \lambda_{h,k+1}^2 + \lambda_{h-1,k+1}^3 - \lambda_{h,k+1}^3 \leq 0, \quad h = 2 \dots n-1, \quad k = 1 \dots m-2 \tag{4.36}$$

$$\lambda_h^4 - \lambda_h^6 \leq \rho, \quad h = 1 \dots n \tag{4.37}$$

$$\lambda_h^5 \leq 1, \quad h = 1 \dots n \tag{4.38}$$

$$\lambda^1, \lambda_{hk}^2, \lambda_{hk}^3, \lambda_h^4, \lambda_h^5, \lambda^6 \geq 0, \quad h = 1, \dots, n, \quad k = 1, \dots, m. \tag{4.39}$$

As diferenças desse dual para o visto em (4.1)-(4.11) estão, principalmente, na adição das variáveis duais referentes às restrições de cálculo de atraso total e adiantamento total, que aparecem na função objetivo e em restrições referentes às conclusões na última máquina. Além disso, as restrições (4.37), que são referentes às variáveis de adiantamento, nem sempre serão satisfeitas, exatamente por conta do método ε -restrito aumentado cortar o espaço objetivo exigindo valores menores para o adiantamento total, uma informação que o problema mestre não possui. Dessa forma, além dos cortes de otimalidades em (4.43), também haverá cortes de factibilidade em (4.44). O Problema Mestre de Benders biobjetivo (PMBB) é apresentado em (4.40)-(4.45).

$$PMBB^\varepsilon = \text{Min } v \quad (4.40)$$

$$S.a : \sum_{j=1}^n x_{jh} = 1, \quad h = 1 \dots n \quad (4.41)$$

$$\sum_{h=1}^n x_{jh} = 1, \quad j = 1 \dots n \quad (4.42)$$

$$\begin{aligned} & (\sum_{j=1}^n p_{j1}x_{j1})\bar{\lambda}^1 + \\ & + \sum_{h=1}^n \sum_{k=1}^{m-1} (\sum_{j=1}^n p_{j,k+1}x_{jh})\bar{\lambda}_{hk}^2 + \sum_{h=1}^{n-1} \sum_{k=1}^m (\sum_{j=1}^n p_{jk}x_{jh})\bar{\lambda}_{hk}^3 + \\ & + \sum_{h=1}^n (\sum_{j=1}^n d_jx_{jh})(\bar{\lambda}_h^4 - \bar{\lambda}_h^5) + (E^{lexmin21} - \varepsilon)\bar{\lambda}^6 \end{aligned} \quad (4.43)$$

$$\leq v, \quad \forall \lambda \in D_\Phi^\varepsilon$$

$$\begin{aligned} & (\sum_{j=1}^n p_{j1}x_{j1})\bar{\lambda}^1 + \\ & + \sum_{h=1}^n \sum_{k=1}^{m-1} (\sum_{j=1}^n p_{j,k+1}x_{jh})\bar{\lambda}_{hk}^2 + \sum_{h=1}^{n-1} \sum_{k=1}^m (\sum_{j=1}^n p_{jk}x_{jh})\bar{\lambda}_{hk}^3 + \\ & + \sum_{h=1}^n (\sum_{j=1}^n d_jx_{jh})(\bar{\lambda}_h^4 - \bar{\lambda}_h^5) + (E^{lexmin21} - \varepsilon)\bar{\lambda}^6 \end{aligned} \quad (4.44)$$

$$\leq 0, \quad \forall \lambda \in D_\Phi^\varepsilon$$

$$x_{jh} \in \mathbb{B} \quad (4.45)$$

A implementação do método *branch-and-Benders-cut* também pode ser realizada para cada iteração do método multiobjetivo (*e.g.* [77]).

O método de decomposição de Benders clássico proposto em [78] possui limitações numéricas e teóricas: dificuldade para o método convergir e necessidade de uma decomposição do problema bem definida (*e.g.* [68, 79, 80]). Para aprimorar o método, diversas estratégias são utilizadas em diferentes etapas. Em [79] é feita uma revisão sobre a decomposição de Benders com foco nos problemas de otimização combinatória. Os autores dividem o método em quatro partes: como decompor o problema, como resolver o problema mestre e o subproblema, como gerar ou aprimorar soluções para o problema mestre e como gerar os cortes. Para cada uma dessas partes, são apresentados diferentes trabalhos da literatura que fizeram modificações na decomposição de Benders junto com comentários do desempenho e das possíveis barreiras dessas modificações. Para resumir essa revisão, são apresentados breves comentários:

- Estratégia de decomposição: afirma-se que a forma de decompor o problema interfere diretamente no desempenho do método. Deixar o problema mestre sem informações do objetivo pode ser prejudicial, assim como, relaxar pouco o problema dificultando sua resolução. O mesmo vale para o subproblema, resumi-lo a um problema de factibilização é ruim por não fornecer limitantes primais;
- Resolução dos problemas: aborda-se a importância de controlar a quantidade de cortes que são adicionados ao problema mestre e possíveis estratégias para isso. Além de métodos para resolver os problemas, como exemplo o *branch-and-Benders-cut* que resolve o problema mestre com uma árvore de busca única;
- Gerar e/ou aprimorar soluções: uma das modificações com maiores possibilidades através do uso de heurísticas, metaheurísticas, *warm start*, técnicas de estabilização,

inequações válidas, múltiplos cortes, etc. Uma delas é a resolução em duas etapas proposta em [81]. Na primeira etapa o problema mestre é relaxado, gerando cortes mais rapidamente, e na segunda etapa o domínio das variáveis é retomado;

- Gerar cortes: discute-se sobre a força dos cortes e das dificuldades quando o dual do subproblema possui múltiplas soluções. Também, sobre gerar múltiplos cortes com poucas variáveis em comparação a cortes com muitas variáveis.

Além dessas discussões, são feitos breves comentários sobre extensões do método de Benders como *Logic-based Benders*, *Generalized Benders* e *Nested Benders*. Na Seção 4.1 apresenta-se um pequeno recorte mais recente sobre o uso de Benders nos problemas de sequenciamento.

4.1 Aplicações de Benders nos problemas de sequenciamento

Esse recorte sobre Benders no problema de sequenciamento não se restringe ao *flow shop* com objetivo *just-in-time*, como na Seção 3.1, por representar uma lacuna na literatura. Não se pretende fazer uma ampla revisão, apenas destacar alguns trabalhos mais recentes da literatura para verificar se as modificações propostas se diferem das discutidas em [79].

Em [82] os autores tratam do problema de sequenciamento em *flow shop* permutacional com duas máquinas minimizando o custo total de energia. Nesse problema, são tratados restrições de flutuação dos custos energéticos e de custo para máquinas paradas. Para resolver o problema, foram aplicadas três versões de *Logic-based Benders* (*LBBD*) que se diferenciam na forma de resolver o subproblema: programação dinâmica, sequências aleatórias mais programação dinâmica e busca local iterativa mais programação dinâmica; além de utilizar um corte baseado na literatura. Dessa forma, a modificação proposta pelos autores está na geração e/ou aprimoramento das soluções e na geração dos cortes. Os testes realizados mostraram que o método baseado em decomposição de Benders obteve solução ótima para instâncias com até 50 tarefas com tempo de execução máximo de 30 minutos.

O problema de sequenciamento em *flow shop* distribuído, também chamado de *parallel flowline*, em que o processamento das tarefas pode ser realizado em diferentes fábricas foi abordado em [83]. Em seus testes computacionais o autor compara a aplicação direta da decomposição de Benders com uma modificação que adiciona cortes extras através de um método de busca local para agilizar a convergência. Como resultados, o autor mostra que utilizar a decomposição de Benders junto com uma busca local obtém-se os melhores resultados para instâncias com até 2 fábricas, 20 tarefas e 10 máquinas com tempo máximo de execução de 30 minutos.

Um sistema de produção japonês chamado *seru* é abordado em [84]. Basicamente, as esteiras que deslocam as tarefas possuem uma organização semicircular, diferente das

esteiras lineares, gerando maior flexibilidade na produção. Nos testes realizados, os autores comparam a decomposição de Benders com a resolução direta do modelo pelo solver. Em seus resultados, a implementação da decomposição de Benders sempre encontrou solução ótima para instâncias de até 50 tarefas e 8 máquinas com tempo máximo de execução de 4 horas. A fim de estudar instâncias maiores, até 1000 tarefas e 20 máquinas, os autores realizaram testes comparando o método proposto com um algoritmo genético e constataram que o método obteve soluções de maior qualidade. Novamente, a modificação proposta está no aprimoramento de soluções.

Outra modificação no aprimoramento da solução é apresentada em [68]. Os autores tratam do problema de sequenciamento em *flow shop* permutacional com restrição de não permitir tempo ocioso de máquina para um subconjunto de máquinas com o objetivo de minimizar o *makespan*. Os autores fazem a decomposição de Benders e, assim como em [83], utilizam uma heurística de busca local para aprimorar a solução fornecida pelo problema mestre, e com essa solução aprimorada, adiciona-se um corte extra para agilizar a convergência do método. Testes preliminares são realizados para verificar a quantidade de cortes extras que são adicionados, ou seja, quantas vezes a solução fornecida pelo problema mestre é aprimorada sucessivamente antes dele ser resolvido novamente. Em seus testes mostra-se que a implementação da decomposição de Benders aliada com heurísticas, gerando um corte extra, é capaz de obter soluções de qualidade superior ao do solver para instâncias com até 500 tarefas e 15 máquinas em um tempo de execução de 2 horas.

Uma aplicação do *LBBD* em uma variação do problema de sequenciamento (*quay crane scheduling problem*), que faz a programação de carga e descarga de navios com o uso de guindastes é tratada em [85]. A modificação proposta pelos autores está na definição do subproblema. Devido a estrutura do problema não permitir o cruzamento de guindastes, os autores desenvolveram um algoritmo que divide o subproblema em problemas menores e independentes. A composição da solução desses problemas compõem uma solução para o subproblema e é fornecido um corte para o problema mestre.

O uso da decomposição de Benders com simulação de eventos discretos pode ser vista em [86]. Os autores resolvem o problema de *flow shop* minimizando o *makespan* através da decomposição de Benders. Eles mostram um diferencial na resolução do subproblema, que é realizada através de simulação de eventos discretos fornecendo os instantes de conclusão, valor objetivo e variáveis duais. Os cortes de otimalidade de cada iteração são adicionados apenas se o limitante superior da iteração for melhor do que o atual, caso contrário é adicionado um corte combinatório para eliminar aquele sequenciamento.

Em [77] é abordado o problema de seleção de máquinas paralelas e sequenciamento de tarefas em uma abordagem biobjetivo, minimizando o *makespan* e o custo total do uso de máquinas e recursos. O método biobjetivo utilizado foi o ε -restrito aliado à ϵ -dominância, em que a tolerância ϵ foi calculada através do número de pontos não dominados esperado, que foi fixado em 20. Para resolver cada subproblema do ε -restrito, os autores aplicaram

LBBD, em que o problema mestre foi resolvido em árvore única, assim como no *branch-and-Benders-cut*, e o subproblema de Benders foi decomposto em diferentes problemas de máquina única com tempos de início para as tarefas. Para eliminar soluções que violavam a restrição adicionada pelo ε -restrito, foram utilizados cortes combinatórios para a designação de tarefas em cada máquina.

O método de Benders em duas etapas é proposto em [87] para resolver três problemas de sequenciamento com incerteza e com objetivos baseados em custo de energia. A decomposição do problema manteve a designação das tarefas no problema mestre e o cálculos dos custos no subproblema. Pelo problema ser estocástico, o subproblema é particionado. Assim os autores propõem duas versões de cortes, o primeiro sendo um corte único de otimalidade e o segundo múltiplos cortes de otimalidade. Similar ao proposto em [81], numa primeira etapa o método de Benders é resolvido com o problema mestre relaxado, com a diferença do critério de parada ser o método convergir, e na segunda etapa o domínio das variáveis é retomado e o método de Benders inicia com os cortes da primeira etapa.

A Tabela 4 apresenta a lista das pesquisas mais recentes que aplicam o método de decomposição de Benders e suas variações para resolver problemas de sequenciamento. Na primeira coluna da Tabela 4 estão listados o nome dos autores, na segunda coluna apresentam-se o método de Benders, sendo clássico (CI) e *logic-based* (LB). Na terceira coluna estão descritos as modificações utilizadas por cada autor.

Tabela 4 – Autores que aplicam o método de decomposição de Benders nos problemas de sequenciamento

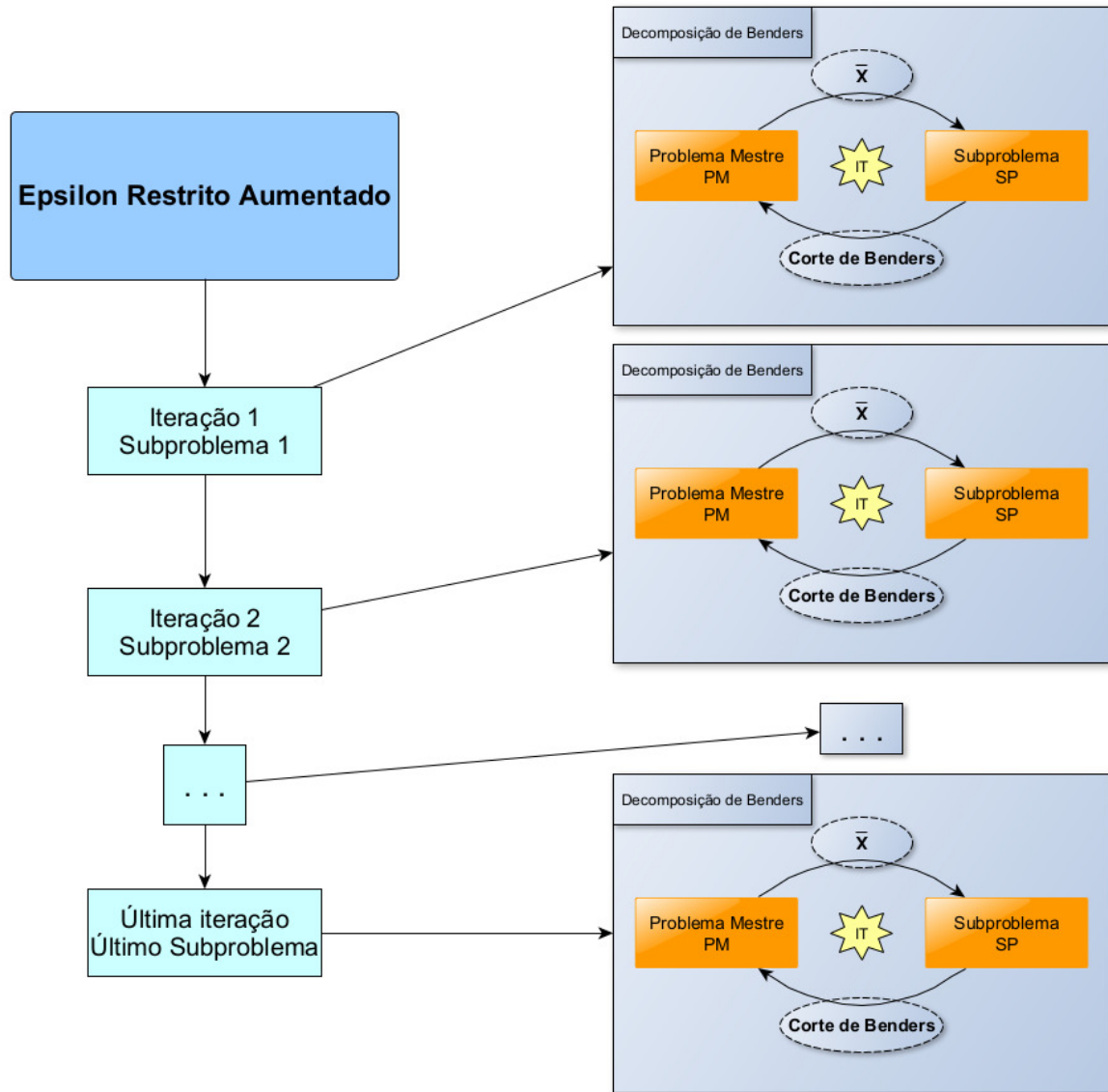
Autoria	Extensão	Modificação
Sun et al. (2019) [85]	LB	Estratégia de decomposição
Hamzadayi et al. (2020) [83]	CI	Geração de cortes
Bektacs (2020) [68]	CI	Aprimoração das soluções e geração de cortes
Ho et al. (2022) [82]	LB	Geração das soluções e geração de cortes
Zhang et al. (2022) [84]	CI	Aprimoração das soluções
Wallrath e Franke (2024) [86]	CI	Resolução dos problemas e geração de cortes
Wu et al. (2024) [77]	LB	Estratégia de decomposição, resolução dos problemas e geração de cortes
Ghorbanzadeh et al. (2024) [87]	CI	Estratégia de decomposição e geração de soluções

4.2 Variações do método de decomposição de Benders para o JIT-FSP

A decomposição de Benders e o método ε -restrito aumentado são ambos métodos iterativos e possuem subproblemas. Para distingui-los, quando houver necessidade, o problema mestre e subproblema de Benders serão referidos respectivamente como PM_B e SP_B , e suas iterações como IT_B . De forma análoga, o subproblema e as iterações do método ε -restrito aumentado são citados como SP_ε e IT_ε , respectivamente.

O método de decomposição de Benders clássico aplicado ao método ε -restrito aumentado (B_d) está representado no diagrama da Figura 20. Para resolver cada subproblema do ε -restrito aumentado é aplicado a decomposição de Benders. No Algoritmo 5 está detalhado o método ε -restrito aumentado e no Algoritmo 6 apresenta-se o método de decomposição de Benders aplicado ao problema (4.17)-(4.26). Vale ressaltar que as linhas 1 e 2 (problema $\text{lexmin}(f_1, f_2)$) e as linhas 3 e 4 (problema $\text{lexmin}(f_2, f_1)$) do Algoritmo 5 representam respectivamente as iterações 1 e 2 ilustradas na Figura 20. Além disso, não há nenhum tipo de comunicação entre as aplicações de Benders em cada iteração do ε -restrito aumentado.

Figura 20 – Diagrama da decomposição de Benders aplicado ao ε -restrito aumentado



Fonte: Elaborado pelo autor.

Dada as similaridades da decomposição de Benders com outros métodos de decomposição e com a geração de coluna, em [18] é abordado o problema de corte de estoque multiobjetivo e utilizou-se o método de geração de colunas para resolver o problema. Uma forma de melhorar o desempenho do método de geração de colunas foi armazenar as colunas geradas em cada iteração do método multiobjetivo utilizado. Trazendo isso para a decomposição de Benders, seria o mesmo que armazenar os cortes gerados de uma IT_ε para a outra. O problema desse processo está no subproblema em cada iteração do método ε -restrito aumentado se alterar, o que provoca alterações no: SP_B , função objetivo do dual do SP_B e corte. Dessa forma, não há garantias de que os cortes gerados em uma iteração do ε -restrito aumentado sejam efetivos para iterações seguintes. Uma forma de contornar isso é armazenar um número limitado de sequenciamentos ao longo da resolução de cada

Algoritmo 5: Algoritmo do método ε -restrito aumentado

Entrada: (4.17) – (4.26) e ρ
Saída: $\mathcal{X}_E$ e $\mathcal{Y}_N$

- 1 Resolva $\text{Min}\{E \mid (4.18) - (4.24), (4.26)\} = E^*$ por Benders;
- 2 Resolva $\text{Min}\{T \mid (4.18) - (4.24), (4.26), E \leq E^*\} = T^*$ por Benders, obtenha $(x_{jh}^*, C_{hk}^*, E_h^*, T_h^*)$ e adicione $(x_{jh}^*, C_{hk}^*, E_h^*, T_h^*)$ a $\mathcal{X}_E$ e (E^*, T^*) a $\mathcal{Y}_N$;
- 3 Resolva $\text{Min}\{T \mid (4.18) - (4.24), (4.26)\} = T^*$ por Benders;
- 4 Resolva $\text{Min}\{E \mid (4.18) - (4.24), (4.26), E \leq E^*\} = \bar{E}$ por Benders, obtenha $(x_{jh}^*, C_{hk}^*, E_h^*, T_h^*)$ e adicione $(x_{jh}^*, C_{hk}^*, E_h^*, T_h^*)$ a $\mathcal{X}_E$ e $(\bar{E}, T^*)$ a $\mathcal{Y}_N$;
- 5 $\varepsilon = T^* + 1$;
- 6 **enquanto** $\bar{E} < E^*$ **faça**
 - 7 Resolva $\text{Min}\{\rho \cdot E + T \mid (4.18) - (4.26)\} = \rho \cdot \bar{E} + \bar{T}$ por Benders, obtenha $(x_{jh}^*, C_{hk}^*, E_h^*, T_h^*)$ e adicione $(x_{jh}^*, C_{hk}^*, E_h^*, T_h^*)$ a $\mathcal{X}_E$ e $(\bar{E}, \bar{T})$ a $\mathcal{Y}_N$;
 - 8 $\varepsilon := \bar{T} + 1$
- 9 **fim**

SP_ε e no início da resolução do próximo SP_ε , pré-gerar cortes para o PM_B através de resoluções sucessivas do SP_B . Nas linhas 1, 2, 3, 4 e 7 do Algoritmo 7 apresentam-se o armazenamento dos últimos sequenciamentos durante a resolução dos SP_ε pelo método de decomposição de Benders. O asterisco (*) presente nas linhas 2, 3, 4 e 7 representa a resolução do SP_ε utilizando o método de decomposição de Benders com cortes pré-gerados descrito no Algoritmo 8. É importante ressaltar que os sequenciamentos são armazenados em fila sem repetição, e se a fila atingir seu limite, para armazenar outro sequenciamento o primeiro da fila é retirado e adiciona-se o novo sequenciamento no final, seguindo a lógica (*FIFO*). Na Figura 21 está exibido o diagrama dessa variação que será nomeada como B_{seq} .

Ao decompor o JIT-FSP, o PM_B é o que exige mais tempo para gerar uma solução, enquanto o SP_B é resolvido rapidamente. Isso ocorre pelo PM_B ser um problema de otimização binário que é resolvido pelo *branch-and-bound* ou *branch-and-cut*. Em [81] os autores propõem uma modificação no algoritmo de Benders utilizando a relaxação linear do PM_B , em um método de duas etapas. Na primeira etapa, por uma quantidade limitada de vezes e/ou até atingir um limite na diferença entre os limitantes, o PM_B é resolvido relaxando o domínio das variáveis, gerando cortes rapidamente. Após atingir o(s) critério(s) de parada estabelecido(s), se inicia a segunda etapa, na qual é retomada a integralidade das variáveis do PM_B e o método de Benders clássico continua mantendo os cortes gerados na primeira etapa. Essa variação será nomeada como B_{RL} e o pseudo-código com as alterações no método de decomposição de Benders está apresentado no Algoritmo 9. O método ε -restrito aumentado utilizado no B_{RL} é igual ao Algoritmo 5 com a diferença de utilizar o Algoritmo 9 para resolver cada SP_ε .

Um ponto negativo da variação B_{RL} está no fato de que os cortes gerados na primeira etapa, possivelmente, são fracos devido a relaxação linear do problema não ser forte.

Algoritmo 6: Algoritmo do método de decomposição de Benders aplicado em (4.17)-(4.26)

Entrada: (4.17)-(4.26) e ρ
Saída: $(x_{jh}^*, C_{hk}^*, E_h^*, T_h^*)$

- 1 Decomponha (4.17)-(4.26) em $PM_B = \{(4.40) - (4.45)\}$ e $SP_B = \{(4.27) - (4.39)\}$;
- 2 $UB = \infty$;
- 3 $LB = -\infty$;
- 4 $it = 0$;
- 5 **enquanto** $UB - LB > 10^{-3}$ **faça**
- 6 Resolva PM_B e obtenha $\bar{x}_{jh}$ e $\bar{v}$ (solução e valor ótimos);
- 7 **se** $\bar{v} > LB$ **então**
- 8 $LB := \bar{v}$
- 9 **senão**
- 10 Continue
- 11 **fim**
- 12 Resolva SP_B com $\bar{x}_{jh}$ e obtenha $\bar{\lambda}^i$, $i = 1, \dots, 6$;
- 13 **se** SP_B *teve solução ótima* **então**
- 14 Adicione um corte de otimalidade em PM_B e obtenha $\rho \cdot \bar{E} + \bar{T}$ (valor ótimo de PM_B) e $(\bar{C}_{hk}, \bar{E}_h, \bar{T}_h)$ (solução ótima de PM_B);
- 15 **se** $UB > \rho \cdot \bar{E} + \bar{T}$ **então**
- 16 $UB := \rho \cdot \bar{E} + \bar{T}$;
- 17 $sol = (\bar{x}_{jh}, \bar{C}_{hk}, \bar{E}_h, \bar{T}_h)$ (solução atual)
- 18 **senão**
- 19 Continue
- 20 **fim**
- 21 **senão**
- 22 Adicione um corte de factibilidade em PM_B
- 23 **fim**
- 24 $it := it + 1$
- 25 **fim**
- 26 $(x_{jh}^*, C_{hk}^*, E_h^*, T_h^*) = sol$

Uma alternativa para fortalecer esses cortes é seguir a ideia do B_{seq} , assim, no início da resolução de cada SP_ε , com excessão do primeiro, são pré-gerados cortes para o PM_B relaxado através da resolução sucessiva do SP_B . Ao atingir o(s) critério(s) de parada da primeira etapa o método continua igual ao B_{RL} . É importante dizer que não são armazenados sequenciamentos do PM relaxado. Essa variação de Benders será referenciada como B_{seq}^{RL} e no Algoritmo 10 apresenta-se o pseudo-código com as alterações no método de decomposição de Benders. O método ε -restrito aumentado utilizado no B_{seq}^{RL} é igual ao Algoritmo 7 com a diferença de utilizar o Algoritmo 10 para resolver cada SP_ε .

Nas variações B_{seq} , B_{RL} e B_{seq}^{RL} , o PM_B é resolvido até a otimalidade em cada IT_B , ou seja, como o PM_B é binário, em cada IT_B é criado uma árvore *branch-and-bound* que será explorada até a prova de otimalidade. Esse processo, em geral, é custoso e há possibilidade de visitar soluções descartadas nas IT_B anteriores. Para contornar isso, na

Algoritmo 7: Algoritmo do método ε -restrito aumentado (B_{seq})

Entrada: (4.17) – (4.26) e ρ
Saída: $\mathcal{X}_E$ e $\mathcal{Y}_N$

- 1 Resolva $\text{Min}\{E \mid (4.18) - (4.24), (4.26)\} = E^*$ por Benders e armazene os últimos sequenciamentos;
 - 2 Resolva $\text{Min}\{T \mid (4.18) - (4.24), (4.26), E \leq E^*\}$ por Benders*, obtenha $(x_{jh}^*, C_{hk}^*, E_h^*, T_h^*)$, adicione $(x_{jh}^*, C_{hk}^*, E_h^*, T_h^*)$ a $\mathcal{X}_E$ e (E^*, T^*) a $\mathcal{Y}_N$ e armazene os últimos sequenciamentos;
 - 3 Resolva $\text{Min}\{T \mid (4.18) - (4.24), (4.26)\} = T^*$ por Benders* e armazene os últimos sequenciamentos;
 - 4 Resolva $\text{Min}\{E \mid (4.18) - (4.24), (4.26), E \leq E^*\} = \bar{E}$ por Benders*, obtenha $(x_{jh}^*, C_{hk}^*, E_h^*, T_h^*)$, adicione $(x_{jh}^*, C_{hk}^*, E_h^*, T_h^*)$ a $\mathcal{X}_E$ e $(\bar{E}, T^*)$ a $\mathcal{Y}_N$ e armazene os últimos sequenciamentos;
 - 5 $\varepsilon = T^* + 1$;
 - 6 **enquanto** $\bar{E} < E^*$ **faça**
 - 7 Resolva $\text{Min}\{\rho \cdot E + T \mid (4.18) - (4.26)\} = \rho \cdot \bar{E} + \bar{T}$ por Benders*, obtenha $(x_{jh}^*, C_{hk}^*, E_h^*, T_h^*)$, adicione $(x_{jh}^*, C_{hk}^*, E_h^*, T_h^*)$ a $\mathcal{X}_E$ e $(\bar{E}, \bar{T})$ a $\mathcal{Y}_N$ e armazene os últimos sequenciamentos;
 - 8 $\varepsilon := \bar{T} + 1$
 - 9 **fim**
-

literatura utiliza-se o método *branch-and-Benders-cut* que constrói uma árvore única no qual os cortes de Benders são gerados para cada solução factível encontrada. Dessa forma, as soluções descartadas e os nós explorados não serão revisitados, além disso, quando for provado a otimalidade pela primeira vez, já será a solução ótima para o SP_ε . A implementação dessa variação é feita através de funções do tipo *call back* presentes em muitos solvers comerciais e será chamada de BBC_d . A geração de cortes iniciais para o PM_B também foi implementada nesse algoritmo, e é referida como BBC_{seq} . Implementar a variação proposta em [81] no BBC_d tem suas limitações, devido a função *call back* não ser ativada ao encontrar soluções factíveis durante o método simplex utilizando o solver Gurobi [88]. Dessa forma, durante a resolução da relaxação linear não há possibilidade de chamar o SP_B para cada solução encontrada e gerar cortes para o algoritmo convergir, inviabilizando o método.

Na Tabela 5 lista-se o nome das Variações, a Autoria: própria ou da literatura; e as Modificações que diferem cada variação apresentada nessa seção, do método de decomposição de Benders clássico.

No Capítulo 5 realiza-se um estudo computacional mostrando o desempenho de cada variação descrita nessa seção.

Algoritmo 8: Algoritmo do método de decomposição de Benders com cortes pré-gerados

Entrada: (4.17)-(4.26) e ρ
Saída: $(x_{jh}^*, C_{hk}^*, E_h^*, T_h^*)$

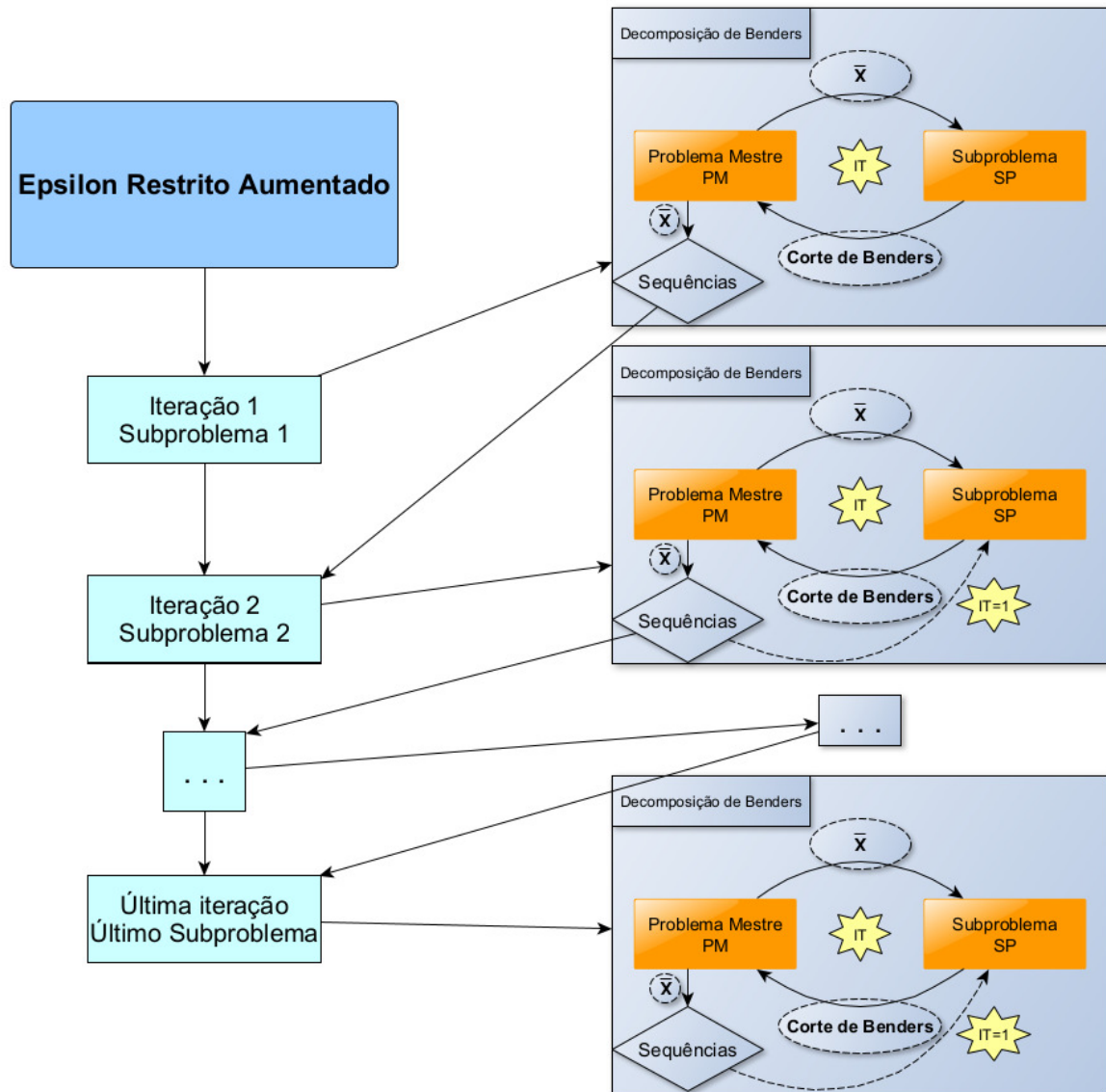
- 1 Decomponha (4.17)-(4.26) em $PM_B = \{(4.40) - (4.45)\}$ e $SP_B = \{(4.27) - (4.39)\}$;
- 2 $UB = \infty$;
- 3 $LB = -\infty$;
- 4 $it = 0$;
- 5 **para** cada sequenciamento **faça**
- 6 Determine $\bar{x}_{jh}$;
- 7 Resolva SP_B com $\bar{x}_{jh}$ e obtenha $\bar{\lambda}^i$, $i = 1, \dots, 6$;
- 8 **se** SP_B teve solução ótima **então**
- 9 Adicione um corte de otimalidade em PM_B e obtenha $\rho \cdot \bar{E} + \bar{T}$ (valor ótimo de PM_B) e $(\bar{C}_{hk}, \bar{E}_h, \bar{T}_h)$ (solução ótima de PM_B);
- 10 **se** $UB > \rho \cdot \bar{E} + \bar{T}$ **então**
- 11 $UB := \rho \cdot \bar{E} + \bar{T}$;
- 12 $sol = (\bar{x}_{jh}, \bar{C}_{hk}, \bar{E}_h, \bar{T}_h)$ (solução atual)
- 13 **senão**
- 14 Continue
- 15 **fim**
- 16 **senão**
- 17 Adicione um corte de factibilidade em PM_B
- 18 **fim**
- 19 **fim**
- 20 **enquanto** $UB - LB > 10^{-3}$ **faça**
- 21 Repita as linhas de 7 à 18;
- 22 $it := it + 1$
- 23 **fim**
- 24 $(x_{jh}^*, C_{hk}^*, E_h^*, T_h^*) = sol$

Algoritmo 9: Pseudo-código do método de decomposição de Benders em duas etapas

Entrada: (4.17) – (4.26), ρ e Algoritmo 6
Saída: $(x_{jh}^*, C_{hk}^*, E_h^*, T_h^*)$

- 1 Etapa 1;
- 2 **enquanto** não atingir um critério de parada **faça**
- 3 | Aplique o Algoritmo 6 com o domínio das variáveis do PM_B relaxado
- 4 **fim**
- 5 Etapa 2;
- 6 Retome o domínio das variáveis do PM_B da Fase 1 (mantendo os cortes de Benders);
- 7 Execute o Algoritmo 6 pulando a linha 1

Figura 21 – Diagrama da variação B_{seq}



Fonte: Elaborado pelo autor.

Algoritmo 10: Pseudo-código do método de decomposição de Benders em duas etapas com cortes pré-gerados

Entrada: (4.17) – (4.26), ρ e Algoritmo 6

Saída: $(x_{jh}^*, C_{hk}^*, E_h^*, T_h^*)$

- 1 Etapa 1;
 - 2 Adicione cortes de Benders para cada sequenciamento armazenado;
 - 3 **enquanto** não atingir um critério de parada **faça**
 - 4 | Aplique o Algoritmo 6 com o domínio das variáveis do PM_B relaxado
 - 5 **fim**
 - 6 Etapa 2;
 - 7 Retome o domínio das variáveis do PM_B da Fase 1 (mantendo os cortes de Benders);
 - 8 Execute o Algoritmo 6 pulando a linha 1
-

Tabela 5 – Variações do método de decomposição de Benders

Método	Variação
B_{cl}	Método clássico
B_{seq}	Pré-geração de cortes para o PM_B
B_{RL}	Etapa 1: Geração de cortes com o PM_B relaxado; Etapa 2: Método clássico mantendo os cortes da Etapa 1
B_{seq}^{RL}	Etapa1: Pré-geração de cortes para o PM_B relaxado e geração de cortes com o PM_B relaxado; Etapa 2: Método clássico mantendo os cortes da Etapa 1
BBC_{cl}	Resolução do PM_B com árvore única e geração de cortes pra cada solução factível da árvore utilizando <i>call back</i>
BBC_{seq}	Pré-geração de cortes para o PM_B e resolução do PM_B com árvore única adicionando cortes para cada solução factível da árvore utilizando <i>call back</i>

5 Resultados computacionais

Nesse capítulo apresentam-se os resultados computacionais obtidos na realização de quatro estudos computacionais. No primeiro estudo compara-se os métodos presentes no sistema BIOBAB: caixa balanceada, ε -restrito e *branch-and-bound*; e dois tipos de busca do GRB_{mult} (resolução direta do problema pelo resolvidor multiobjetivo disponível no Gurobi [88]). É importante dizer que apesar do método ε -restrito obter soluções fracamente eficientes, a implementação disponível no sistema BIOBAB determina apenas soluções eficientes, pois em cada iteração do método é utilizada otimização lexicográfica. Assim, todos os métodos disponíveis no sistema são comparáveis. No segundo estudo, demonstram-se as vantagens de fazer uma abordagem biobjetiva do problema JIT-FSP. No terceiro estudo é feita uma comparação entre os métodos biobjetivos: BIOBAB, caixa balanceada, ε -restrito aumentado e GRB_{mult} . No quarto e último estudo, analisa-se o esforço computacional de cada uma das variações do método de decomposição de Benders apresentadas na Seção 4.2. Os objetivos desses estudos estão descritos na Tabela 6.

Os estudos computacionais foram realizados em uma máquina com processador Core i7 de segunda geração e 3.40 GHz de CPU, memória RAM de 16 GB e sistema operacional Windows 10 Home de 64 bits. Todos os códigos utilizados estão escritos na linguagem Python 3.8 [89] e todos os problemas e subproblemas foram resolvidos pelo Gurobi v.10.0.1 [88]. O sistema BIOBAB [5], além de conter uma implementação do método *branch-and-bound* biobjetivo, contém também os métodos caixa balanceada [16] e ε -restrito [17]. Assim os resultados referentes a esses três métodos foram obtidos usando o sistema BIOBAB. Os autores disponibilizaram o sistema no repositório <https://github.com/JKU-PLM/BIOBAB> e para utilizá-lo, foram criadas duas classes em Python: a primeira que implementa o modelo (3.28)-(3.36) apresentado na Seção 3.2 e a segunda implementa a leitura das instâncias. Também foi usado o Gurobi multiobjetivo, que se assemelha à abordagem feita pela literatura utilizando pesos fixos e unitários, nomeada de GRB_{mult} , e duas variações, GRB_{mult}_p com busca em profundidade e GRB_{mult}_l com busca em largura. Além disso, uma versão monoobjetivo do problema utilizando pesos unitários foi nomeada de GRB_{mono} . O método ε -restrito aumentado, a ϵ -dominância e variações do método de Benders foram de implementação própria. Em todo o estudo, os objetivos considerados foram o adiantamento total e o atraso total.

As instâncias utilizadas na Seção 5.1 foram propostas inicialmente por Taillard [90] e depois foram adicionadas datas de vencimento por Minella et al [34]. Com n sendo o número de tarefas e m o número de máquinas, foram utilizadas 8 classes (n, m) : (20, 5), (20, 10), (20, 20), (50, 5), (50, 10), (50, 20), (100, 5) e (100, 10). Para cada um dos 8 tipos (n, m) existem 4 instâncias, totalizando 32 instâncias. Os tempos de processamento

foram gerados segundo uma distribuição normal no intervalo $[1, 99]$. As datas de vencimento foram geradas pela expressão $d_i = P_i \cdot (1 + \text{random} \cdot 3)$ sendo P_i a soma dos tempos de processamento da tarefa i e random um número uniformemente distribuído no intervalo $[0, 1]$. Na Seção 5.1 são analisados os resultados de execuções com tempos máximo de 3600 segundos para cada uma.

Com os resultados e conclusões obtidos na Seção 5.1, houve a necessidade de mudar o porte das instâncias para se adequar à proposta dos próximos estudos. Então nas Seções 5.2, 5.3 e 5.4 as instâncias utilizadas foram propostas em Ronconi e Birgin [46]. Foram definidos 4 pares $(n, m) : (5, 3), (10, 3), (10, 5)$ e $(10, 10)$. Também, consideram-se 4 cenários variando por fator de atraso (TF) e amplitude de dispersão da data de vencimento (DR), sendo $(TF, DR) : (0.2, 0.6), (0.2, 1.2), (0.4, 0.2)$ e $(0.4, 1.2)$. Dessa forma, foram obtidos 16 tipos diferentes de classes e para cada classe foram geradas 10 instâncias, totalizando 160 instâncias. Os tempos de processamento foram gerados uniformemente no intervalo de $[1, 99]$. As datas de entrega foram geradas uniformemente no intervalo $[P(1 - TF - DR/2), P(1 - TF + DR/2)]$ com P sendo um limite inferior para o objetivo *makespan*. Ao todo foram 1100 execuções com tempos máximo de 3600 segundos para cada uma.

Tabela 6 – Objetivos dos estudos computacionais

Estudo	Objetivos
1.	Parte 1: determinar a melhor configuração do BIOBAB e, dada essa configuração, definir qual método possui melhor desempenho segundo as métricas cardinalidade e hipervolume utilizando uma instância de cada classe. Parte 2: verificar se as conclusões obtidas na Parte 1 do estudo são mantidas com mais três instâncias de cada classe. (Seção 5.1)
2.	Mostrar o esforço computacional da abordagem multiobjetivo em comparação à monoobjetivo; analisar geometricamente a fronteira de Pareto do JIT-FSP para diferentes cenários industriais e mostrar a importância de utilizar métodos multiobjetivos que encontram toda a fronteira de Pareto para o problema. (Seção 5.2)
3.	Primeiro determinar o método (BIOBAB, caixa balanceada, ε -restrito aumentado e <i>GRBmult</i>) que possui melhor desempenho segundo as métricas cardinalidade e <i>hiper ratio</i> . Segundo, avaliar o <i>trade-off</i> entre reduzir o número de pontos na AFP e manter a AFP representativa utilizando o método que obteve melhor desempenho na primeira parte do estudo. (Seção 5.3)
4.	Analisar o impacto causado no desempenho do método de decomposição de Benders clássico em cada variação proposta ($B_{seq}, B_{RL}, B_{seq}^{RL}, BBC_{cl}$ e BBC_{seq} resumidas na Tabela 5). (Seção 5.4)

5.1 Estudo 1 - O sistema BIOBAB

Nessa seção é realizado um estudo da melhor configuração do sistema BIOBAB (Configurações do sistema no Apêndice 6) ao utilizar o método *branch-and-bound*. Em seguida, na Parte 1, é feita uma comparação entre os métodos do sistema BIOBAB e os métodos $GRBmult_p$ e $GRBmult_l$ pelas métricas cardinalidade e hipervolume, em uma instância de cada classe. Na Parte 2, é feita uma comparação entre os métodos do sistema BIOBAB e os métodos $GRBmult_p$ e $GRBmult_l$ pelas métricas cardinalidade e hipervolume, em mais três instâncias de cada classe.

Configurando o sistema

Na Tabela 7 são apresentados os valores de desempenho do BIOBAB variando sua configuração entre utilizar ou não a relaxação linear e escolher busca em profundidade ou em largura (quando é utilizada a relaxação linear). Na coluna 1 apresenta-se o porte $(n, m)_1$ da primeira instância (subscrito 1) das classes. Para cada instância, são apresentados nas colunas correspondentes os valores da métrica de cardinalidade ($I_{\#}$) e de hipervolume (I_H) respectivamente relacionados a cada configuração do BIOBAB. É importante dizer que a não utilização da relaxação linear no BIOBAB resume o método a uma soma ponderada, assim como discutido na Seção 2.1.1. Por este motivo não há uma coluna de busca em profundidade e outra de busca em largura, pois não é feita ramificação.

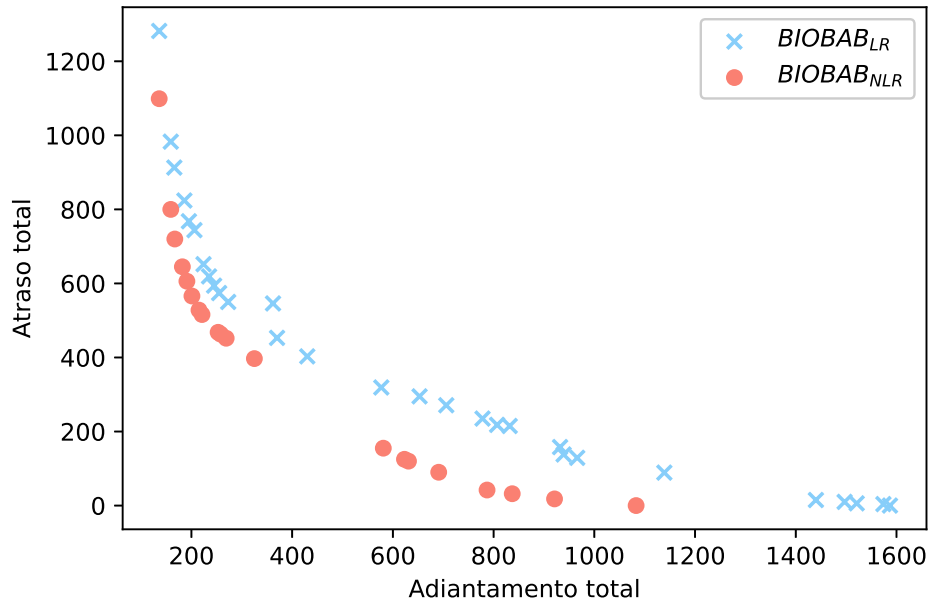
Tabela 7 – Valores das métricas de cardinalidade e de hipervolume de configurações do BIOBAB

	Relaxação linear				Sem relaxação linear	
	Profundidade		Largura			
(n, m)	$I_{\#}$	I_H	$I_{\#}$	I_H	$I_{\#}$	I_H
$(20, 5)_1$	65	$4,58 \cdot 10^6$	1	$8,68 \cdot 10^2$	2	$4,79 \cdot 10^6$
$(20, 10)_1$	29	$1,38 \cdot 10^7$	1	$1,45 \cdot 10^3$	20	$1,39 \cdot 10^7$
$(20, 20)_1$	88	$2,57 \cdot 10^8$	1	$9,06 \cdot 10^3$	2	$2,73 \cdot 10^8$
$(50, 5)_1$	8	$2,31 \cdot 10^7$	1	$2,20 \cdot 10^7$	1	$6,70 \cdot 10^1$
$(50, 10)_1$	71	$3,49 \cdot 10^8$	1	$2,94 \cdot 10^8$	1	$9,67 \cdot 10^2$
$(50, 20)_1$	62	$3,29 \cdot 10^9$	1	$2,68 \cdot 10^9$	1	$6,89 \cdot 10^3$
$(100, 5)_1$	7	$1,09 \cdot 10^{10}$	1	$1,08 \cdot 10^{10}$	1	$4,15 \cdot 10^2$
$(100, 10)_1$	1	$2,94 \cdot 10^7$	1	$2,94 \cdot 10^7$	1	$1,00 \cdot 10^0$

Nas instâncias $(20, m)_1$, não utilizar a relaxação linear gerou resultados melhores do que utilizando a relaxação linear em relação ao hipervolume. Isso ocorreu pelo fato dos pontos encontrados sem a relaxação linear ($BIOBAB_{NLR}$) dominarem parte da AFP quando

utilizado relaxação linear ($BIOBAB_{LR}$) como é exibido na Figura 22, que representa as AFP's dos métodos ao resolver a instância $(20, 10)_1$. Por outro lado, a partir da instância $(50, 5)_1$, sem considerar a relaxação linear, o hipervolume fica menor, mostrando a superioridade da relaxação linear, principalmente com busca em profundidade. Isso é causado pelo sistema BIOBAB não controlar o tempo gasto em cada subproblema. Dessa forma, com o aumento do porte do problema, permanecer com o domínio das variáveis $(x_{jh} \in \{0, 1\}, \forall i, j)$ exige um esforço computacional maior para provar a otimalidade das soluções. Considerando a métrica cardinalidade a configuração relaxação linear com busca em profundidade foi superior em todas as instâncias.

Figura 22 – AFP's de configurações do BIOBAB da instância $(20, 10)_1$



Fonte: Elaborado pelo autor.

Fazendo uma análise dos indicadores da Tabela 7 em conjunto, nas instâncias $(20, m)_1$ percebe-se que muitos pontos da AFP do BIOBAB com relaxação linear e busca em profundidade são dominados pelos pontos da AFP do BIOBAB sem relaxação linear, pois com menos pontos obteve um hipervolume maior. O inverso ocorre nas instâncias $(50, m)_1$ e $(100, m)_1$, a busca em profundidade encontrou mais pontos e obteve hipervolume maior do que não utilizando a relaxação linear. Vale ressaltar que mesmo a busca em largura obtendo hipervolume próximo da busca em profundidade nas instâncias $(50, m)_1$ e $(100, m)_1$, por encontrar apenas um ponto na AFP, a busca em largura não oferece o *trade-off* entre os objetivos, evidenciando que a busca em profundidade, que em média encontrou 41 pontos, é mais recomendada. Obter mais de um ponto na AFP é muito importante, pois é uma das características desejadas na otimização multiobjetivo, exibir o

trade-off entre os objetivos e oferecer diferentes escolhas para o tomador de decisão.

Na Tabela 8 estão exibidos o número de subproblemas e nós resolvidos durante o BIOBAB considerando busca em profundidade e largura. Na coluna 1 apresenta-se o porte $(n, m)_1$ da primeira instância das classes. Para cada instância, são apresentados o número de subproblemas resolvidos (LPs), número de nós resolvidos (Nós) e a razão entre LPs e Nós ($\frac{LPs}{Nós}$) respectivamente relacionados a cada tipo de busca da relaxação linear do BIOBAB.

Tabela 8 – Efeito do tipo de busca na árvore de busca do BIOBAB

	Profundidade			Largura		
(n, m)	LPs	Nós	$\frac{LPs}{Nós}$	LPs	Nós	$\frac{LPs}{Nós}$
$(20, 5)_1$	947404	150160	6,30	533185	6983	76,35
$(20, 10)_1$	758176	117351	6,46	329425	14151	23,27
$(20, 20)_1$	544625	67140	8,11	86725	8013	10,82
$(50, 5)_1$	253933	54346	4,67	17588	1252	14,04
$(50, 10)_1$	114726	17655	6,49	3889	100	38,89
$(50, 20)_1$	25737	2731	9,42	910	79	11,51
$(100, 5)_1$	46967	13589	3,45	1974	87	22,68
$(100, 10)_1$	228	16	14,25	286	10	28,60

Pelo número de nós explorados percebe-se que a busca em profundidade explorou mais nós do que a busca em largura em todas as instâncias. Isso provavelmente foi provocado pela busca em profundidade encontrar rapidamente soluções factíveis, melhorando o conjunto limitante superior, o que colabora para a sondagem de nós. Isso é reforçado observando as colunas 4 e 7 que mostram a relação do número de subproblemas resolvidos por número de nós explorados, na busca em largura esse valor é sempre maior. Nas Tabelas 7 e 8, apesar de haver indícios dos limitantes superiores da busca em profundidade não serem de qualidade, eles ainda contribuem para a sondagem de nós. Isso é um fato que se difere um pouco da otimização monoobjetivo. Como a análise da otimização monoobjetivo é unidimensional, o único fator de interesse é obter o menor limitante superior e o maior limitante inferior, apenas um de cada. Na otimização multiobjetivo, além da qualidade dos limitantes, o número de pontos nos conjuntos limitantes também é importante.

Pela discussão feita, o BIOBAB considerando a relaxação linear com busca em profundidade, no geral, obteve melhores resultados segundo as métricas cardinalidade e hipervolume. Dessa forma, será essa configuração a utilizada para comparar com os outros métodos de solução.

Comparação entre os métodos do sistema - Parte 1

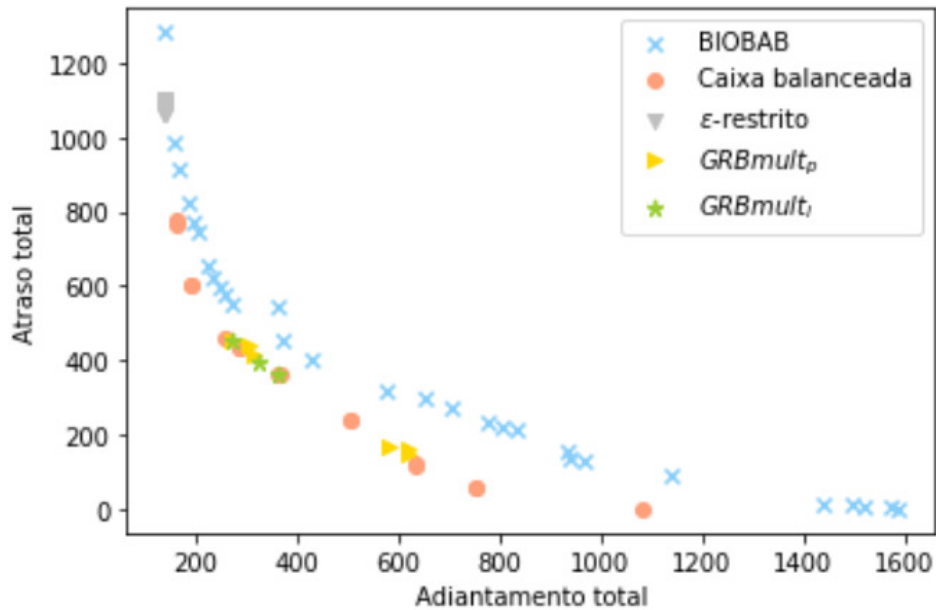
Na Tabela 9 estão exibidos os valores de desempenho, pelas métricas cardinalidade e hipervolume, dos métodos BIOBAB, caixa balanceada, ε -restrito, $GRBmult_p$ (busca em profundidade) e $GRBmult_l$ (busca em largura). Na coluna 1 apresenta-se o porte das instâncias utilizadas. Para cada instância, apresentam-se nas colunas correspondentes os valores da métrica cardinalidade ($I_{\#}$) e da métrica hipervolume (I_H), respectivamente, relacionados a cada um dos métodos de solução.

Tabela 9 – Valores das métricas de cardinalidade e de hipervolume dos métodos do sistema BIOBAB e dos métodos $GRBmult_p$ e $GRBmult_l$

	BIOBAB			Caixa balanceada			ε -restrito			$GRBmult_p$			$GRBmult_l$		
	$I_{\#}$	I_H		$I_{\#}$	I_H		$I_{\#}$	I_H		$I_{\#}$	I_H		$I_{\#}$	I_H	
$(m, n)_1$	65	$4,58 \cdot 10^6$	2	$4,79 \cdot 10^6$	2	$4,79 \cdot 10^6$	7	$3,89 \cdot 10^6$	4	$3,92 \cdot 10^6$					
$(20, 5)_1$	29	$1,46 \cdot 10^6$	17	$1,61 \cdot 10^6$	40	$3,24 \cdot 10^5$	6	$1,41 \cdot 10^6$	3	$1,21 \cdot 10^6$					
$(20, 10)_1$	88	$2,95 \cdot 10^7$	81	$4,56 \cdot 10^7$	95	$4,56 \cdot 10^7$	1	$4,49 \cdot 10^7$	2	$4,54 \cdot 10^7$					
$(50, 5)_1$	8	$6,63 \cdot 10^7$	1	$1,91 \cdot 10^2$	1	$1,91 \cdot 10^2$	9	$4,08 \cdot 10^7$	4	$4,08 \cdot 10^7$					
$(50, 10)_1$	71	$3,49 \cdot 10^8$	1	$9,67 \cdot 10^2$	1	$9,67 \cdot 10^2$	5	$1,38 \cdot 10^8$	5	$1,96 \cdot 10^8$					
$(50, 20)_1$	62	$3,29 \cdot 10^9$	1	$6,89 \cdot 10^3$	1	$6,89 \cdot 10^3$	5	$3,26 \cdot 10^9$	6	$3,14 \cdot 10^9$					
$(100, 5)_1$	7	$1,17 \cdot 10^{10}$	1	$4,46 \cdot 10^2$	1	$4,46 \cdot 10^2$	2	$3,67 \cdot 10^8$	4	$1,26 \cdot 10^9$					
$(100, 10)_1$	1	$3,12 \cdot 10^{10}$	1	$1,06 \cdot 10^3$	1	$1,06 \cdot 10^3$	6	$1,95 \cdot 10^{10}$	9	$1,42 \cdot 10^{10}$					

Nas instâncias $(20, m)_1$, todos os métodos obtiveram valores parecidos para o hipervolume, com exceção do BIOBAB na instância $(20, 20)$ e do ε -restrito na instância $(20, 10)$. Na Figura 24 percebe-se que nas instâncias $(20, m)_1$ o método ε -restrito obteve maior valor para o hipervolume em duas instâncias e a caixa balanceada o maior valor para o hipervolume em uma instância. Além disso, na Figura 23 apresentam-se as AFP's dos métodos na instância $(20, 10)_1$. Nessa figura, é notório que a AFP do BIOBAB é a mais extensa, porém, possui alguns pontos pouco distribuídos e/ou dominados. Isso justifica o método ter um hipervolume inferior à caixa balanceada mesmo tendo mais pontos na AFP. Também a extensão da AFP do BIOBAB é um fator que contribuiu muito para seu hipervolume ser superior ao hipervolume dos métodos ε -restrito, $GRBmult_p$ e $GRBmult_l$, principalmente em relação aos dois últimos, pois a AFP do $GRBmult_p$ e $GRBmult_l$ dominam parte da AFP do BIOBAB.

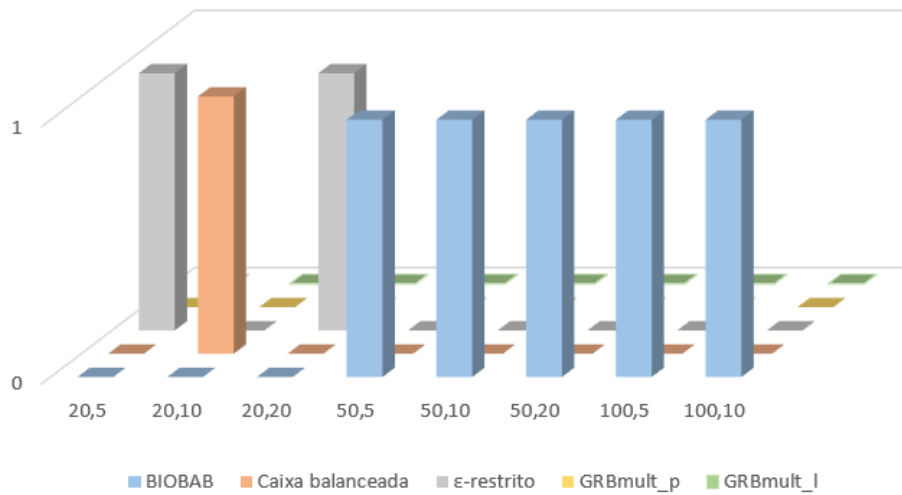
Figura 23 – AFP's dos métodos do sistema BIOBAB e dos métodos $GRBmult_p$ e $GRBmult_l$ na instância $(20, 10)_1$



Fonte: Elaborado pelo autor.

A partir da instância $(50, 5)_1$, os métodos caixa balanceada e ε -restrito encontraram apenas um ponto na fronteira, mesmo fato que aconteceu com o BIOBAB sem considerar a relaxação linear. O motivo disso ocorrer é do sistema BIOBAB não controlar o tempo de resolução dos subproblemas. Nessas instâncias, a Figura 24 evidencia que o BIOBAB teve desempenho superior aos outros métodos considerando o hipervolume. Analisando a métrica cardinalidade, o BIOBAB foi superior em quatro instâncias das oito, o ε -restrito foi melhor em duas instâncias das oito e o $GRBmult_p$ e $GRBmult_l$ foram melhores em uma das oito instâncias.

Figura 24 – Métrica hipervolume da instância 1 para as classes (n, m)



Fonte: Elaborado pelo autor.

Fazendo uma análise geral, o BIOBAB se destacou em comparação aos outros métodos por apresentar em média 41 pontos na AFP e ter o maior hipervolume em algumas instâncias. Também, o método *GRBmult* não mostrou grandes diferenças ao alternar busca em profundida e busca em largura. É importante dizer que a partir da instância $(50, 5)_1$ não há uma comparação justa da caixa balanceada e ε -restrito em relação aos outros métodos. Na literatura, a resolução de cada subproblema durante os métodos de escalarização é limitada em um tempo bem menor do que o tempo máximo de execução, ou seja, são métodos exatos sendo utilizados de forma heurística justamente pela dificuldade de provar a otimalidade de cada subproblema.

Comparação entre os métodos do sistema - Parte 2

Na expectativa do comportamento e conclusões da **Comparação entre os métodos do sistema - parte 1** se manter, a Parte 2 da comparação é dividida em duas (Parte 2a e Parte 2b), sendo a Parte 2a com as classes $(20, m)$ e a Parte 2b com as classes $(50, m)$ e $(100, m)$. Essa divisão é uma forma mais justa para comparar os métodos dado os resultados obtidos na Parte 1 da comparação. A configuração utilizada para o BIOBAB é relaxação linear com busca em profundida.

Comparação Parte 2a - com as classes $(20, m)$

Na Tabela 10 são exibidos os valores de desempenho, pelas métricas cardinalidade e hipervolume, dos métodos BIOBAB, caixa balanceada, ε -restrito, *GRBmult_p* e *GRBmult_l*. Na coluna 1 apresenta-se o porte (m, n) das classes com o subscrito para diferenciar as instâncias. Para cada instância, são apresentados nas colunas correspondentes os valores

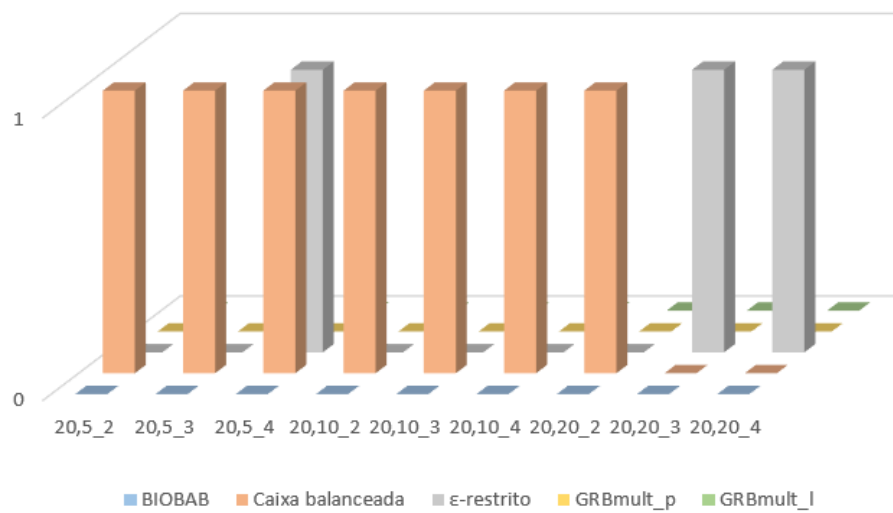
da métrica de cardinalidade ($I_{\#}$) e de hipervolume (I_H) respectivamente relacionados a cada método de solução.

Tabela 10 – Valores das métricas de cardinalidade e de hipervolume das instâncias 2, 3 e 4 para as classes $(20, m)$

	BIOBAB			Caixa balanceada			ε -restrito			$GRBmult_p$			$GRBmult_l$		
(n, m)	$I_{\#}$	I_H		$I_{\#}$	I_H		$I_{\#}$	I_H		$I_{\#}$	I_H		$I_{\#}$	I_H	
$(20, 5)_2$	26	$8,36 \cdot 10^4$	6	6	$1,46 \cdot 10^5$	5	5	$8,60 \cdot 10^4$	3	3	$1,21 \cdot 10^5$	2	2	$1,14 \cdot 10^5$	
$(20, 5)_3$	19	$1,16 \cdot 10^5$	2	2	$2,75 \cdot 10^5$	1	1	$2,31 \cdot 10^5$	1	1	$2,38 \cdot 10^5$	4	4	$2,66 \cdot 10^5$	
$(20, 5)_4$	15	$1,27 \cdot 10^6$	1	1	$5,09 \cdot 10^6$	1	1	$5,09 \cdot 10^6$	5	5	$2,77 \cdot 10^6$	4	4	$4,49 \cdot 10^6$	
$(20, 10)_2$	15	$3,82 \cdot 10^5$	183	7,18	10^5	208	6,76	10^5	3	3	$5,87 \cdot 10^5$	2	2	$5,48 \cdot 10^5$	
$(20, 10)_3$	55	$3,86 \cdot 10^6$	4	4	$6,01 \cdot 10^6$	62	5,48	10^6	2	2	$5,38 \cdot 10^6$	2	2	$5,44 \cdot 10^6$	
$(20, 10)_4$	120	$2,59 \cdot 10^7$	35	3,71	10^7	30	3,64	10^7	4	4	$3,65 \cdot 10^7$	3	3	$3,64 \cdot 10^7$	
$(20, 20)_2$	26	$2,63 \cdot 10^6$	167	4,04	10^6	145	4,04	10^6	2	2	$3,81 \cdot 10^6$	3	3	$3,81 \cdot 10^6$	
$(20, 20)_3$	35	$5,94 \cdot 10^6$	54	8,51	10^6	70	8,51	10^6	3	3	$8,43 \cdot 10^6$	2	2	$8,50 \cdot 10^6$	
$(20, 20)_4$	21	$6,03 \cdot 10^5$	122	9,14	10^5	160	9,14	10^5	3	3	$8,58 \cdot 10^5$	2	2	$8,33 \cdot 10^5$	

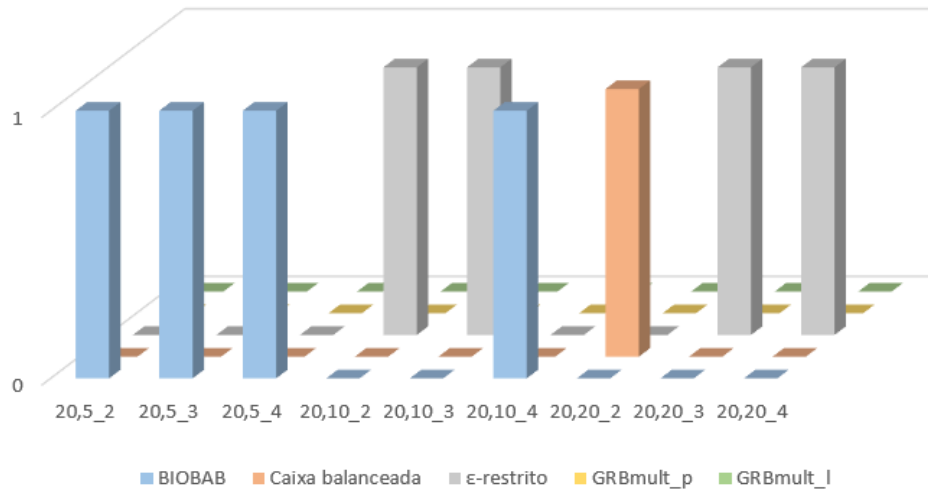
Analisando a Tabela 10 nota-se que considerando a métrica hipervolume, os métodos caixa balanceada, ε -restrito, $GRBmult_p$ e $GRBmult_l$ tiveram desempenhos similares, com um resultado inferior mais evidente para o ε -restrito na instância $(20, 5)_2$ e para o $GRBmult_p$ na instância $(20, 5)_4$. O método BIOBAB também teve desempenho similar aos outros métodos nas instâncias $(20, 5)_3$, $(20, 10)_4$ e $(20, 20)_2$, porém, nas outras instâncias seu desempenho foi inferior. Pela Figura 25 a caixa balanceada teve o melhor desempenho em 7 das 9 instâncias e o ε -restrito teve o melhor desempenho em 3 das 9 instâncias. Pela métrica cardinalidade (Figura 26), o método BIOBAB teve o melhor desempenho em 4 das 9 instâncias, a caixa balanceada em uma das 9 instâncias e o ε -restrito em 4 das 9 instâncias.

Figura 25 – Métrica hipervolume das instâncias 2, 3 e 4 para as classes $(20, m)$



Fonte: Elaborado pelo autor.

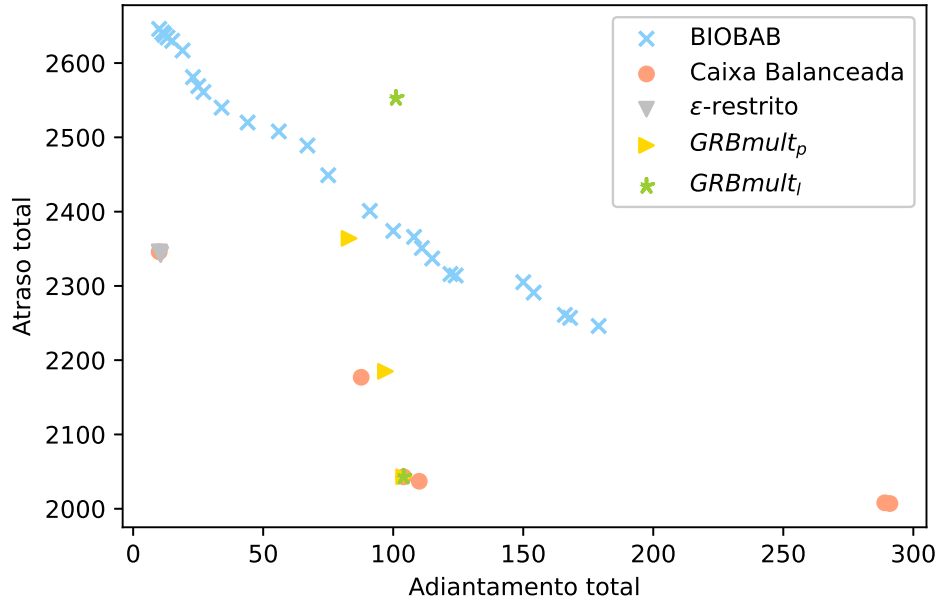
Figura 26 – Métrica cardinalidade das instâncias 2, 3 e 4 para as classes $(20, m)$



Fonte: Elaborado pelo autor.

Nas instâncias que a caixa balanceada obteve melhor hipervolume entre os métodos e comparado ao BIOBAB, nota-se que em apenas duas dessas sete instâncias a caixa balanceada encontrou mais pontos do que o BIOBAB. Isso reforça o que foi discutido na parte 1 da comparação. Os pontos da AFP do BIOBAB são dominados pela AFP da caixa balanceada e para algumas instâncias, os pontos da AFP do BIOBAB também estão mal distribuídos provocando um hipervolume menor. Na Figura 27 apresenta-se a AFP dos métodos de solução na instância $(20, 5)_2$, no qual é possível visualizar os pontos dominados e acumulados da AFP do BIOBAB. A mesma conclusão é obtida ao comparar o BIOBAB e o *GRBmult* (largura ou profundidade), em todas as instâncias o BIOBAB obteve menor hipervolume mesmo encontrando mais pontos.

Figura 27 – AFP's dos métodos do sistema BIOBAB e dos métodos $GRBmult_p$ e $GRBmult_l$ na instância $(20, 5)_2$



Fonte: Elaborado pelo autor.

Em geral, os métodos caixa balanceada e ε -restrito mostraram ser superiores considerando instâncias de pequeno porte. Para a métrica cardinalidade os métodos BIOBAB e ε -restrito tiveram os melhores desempenhos.

Comparação Parte 2b - com as classes $(50, m)$ e $(100, m)$

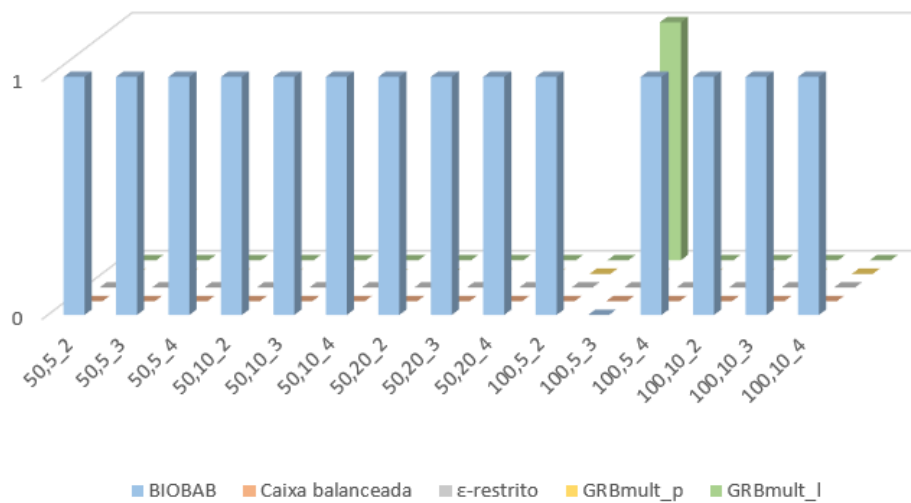
Na Tabela 11, são exibidos os valores de desempenho, pelas métricas cardinalidade e hipervolume, dos métodos BIOBAB, caixa balanceada, ε -restrito, $GRBmult_p$ e $GRBmult_l$ para as instâncias de médio e grande porte. Na coluna 1 apresenta-se o porte (m, n) das classes com o subscrito para diferenciar as instâncias. Para cada instância, são apresentados nas colunas correspondentes os valores da métrica de cardinalidade ($I_{\#}$) e de hipervolume (I_H) respectivamente relacionados a cada método de solução.

Tabela 11 – Valores das métricas de cardinalidade e de hipervolume das instâncias 2, 3 e 4 para as classes (50, m) e (100, m)

	BIOBAB		Caixa balanceada		ε -restrito		$GRBmult_p$		$GRBmult_t$	
(m, n)	$I_{\#}$	I_H	$I_{\#}$	I_H	$I_{\#}$	I_H	$I_{\#}$	I_H	$I_{\#}$	I_H
$(50, 5)_2$	26	$1,03 \cdot 10^8$	1	$2,65 \cdot 10^2$	1	$2,65 \cdot 10^2$	4	$1,68 \cdot 10^7$	5	$6,90 \cdot 10^7$
$(50, 5)_3$	19	$2,68 \cdot 10^8$	1	$7,50 \cdot 10^2$	1	$7,50 \cdot 10^2$	3	$2,51 \cdot 10^8$	5	$2,61 \cdot 10^8$
$(50, 5)_4$	17	$2,62 \cdot 10^8$	1	$7,14 \cdot 10^2$	1	$7,14 \cdot 10^2$	5	$8,24 \cdot 10^7$	5	$8,20 \cdot 10^7$
$(50, 10)_2$	24	$3,29 \cdot 10^8$	1	$9,11 \cdot 10^2$	1	$9,11 \cdot 10^2$	9	$1,12 \cdot 10^8$	7	$1,12 \cdot 10^8$
$(50, 10)_3$	27	$3,17 \cdot 10^8$	1	$9,76 \cdot 10^2$	1	$9,76 \cdot 10^2$	11	$2,62 \cdot 10^8$	9	$2,55 \cdot 10^8$
$(50, 10)_4$	73	$8,47 \cdot 10^8$	1	$2,39 \cdot 10^3$	1	$2,39 \cdot 10^3$	2	$4,46 \cdot 10^8$	4	$4,88 \cdot 10^8$
$(50, 20)_2$	50	$1,31 \cdot 10^9$	1	$2,91 \cdot 10^3$	1	$2,91 \cdot 10^3$	12	$1,05 \cdot 10^9$	3	$8,66 \cdot 10^8$
$(50, 20)_3$	102	$2,16 \cdot 10^9$	1	$5,14 \cdot 10^3$	1	$5,14 \cdot 10^3$	9	$2,09 \cdot 10^9$	3	$2,06 \cdot 10^9$
$(50, 20)_4$	79	$1,26 \cdot 10^9$	1	$2,92 \cdot 10^3$	1	$2,92 \cdot 10^3$	8	$9,68 \cdot 10^8$	9	$1,07 \cdot 10^9$
$(100, 5)_2$	4	$1,15 \cdot 10^{10}$	1	$4,47 \cdot 10^2$	1	$4,47 \cdot 10^2$	10	$4,05 \cdot 10^9$	4	$4,62 \cdot 10^9$
$(100, 5)_3$	1	$5,98 \cdot 10^9$	1	$2,35 \cdot 10^2$	1	$2,35 \cdot 10^2$	3	$4,07 \cdot 10^8$	3	$5,98 \cdot 10^9$
$(100, 5)_4$	2	$1,28 \cdot 10^{10}$	1	$5,24 \cdot 10^2$	1	$5,24 \cdot 10^2$	2	$5,38 \cdot 10^9$	5	$1,47 \cdot 10^9$
$(100, 10)_2$	1	$3,05 \cdot 10^{10}$	1	$1,11 \cdot 10^3$	1	$1,11 \cdot 10^3$	10	$1,21 \cdot 10^{10}$	2	$2,15 \cdot 10^{10}$
$(100, 10)_3$	1	$2,50 \cdot 10^{10}$	1	$8,51 \cdot 10^2$	1	$8,51 \cdot 10^2$	7	$1,63 \cdot 10^{10}$	4	$1,63 \cdot 10^{10}$
$(100, 10)_4$	1	$3,45 \cdot 10^{10}$	1	$1,18 \cdot 10^3$	1	$1,18 \cdot 10^3$	8	$1,27 \cdot 10^{10}$	8	$1,39 \cdot 10^{10}$

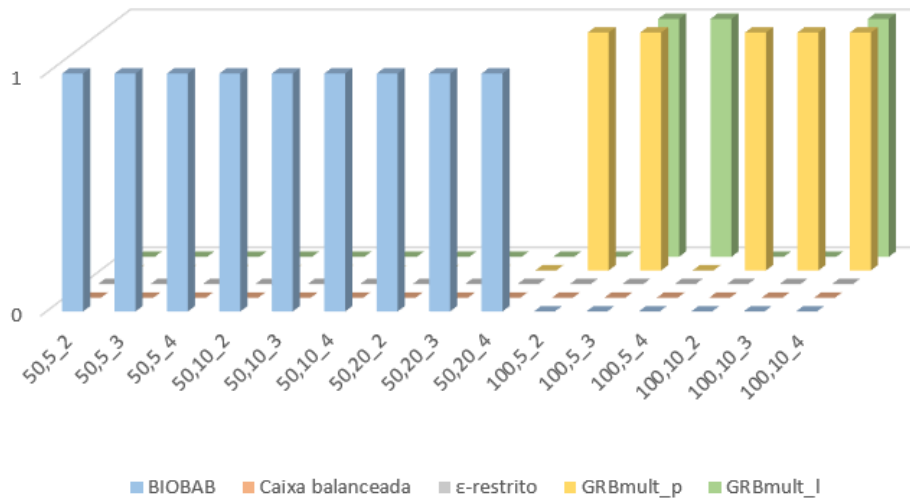
Pela Tabela 11 é notório que os métodos de escalarização caixa balanceada e ε -restrito mantiveram seus comportamentos da Parte 1 da comparação ao encontrar apenas um ponto na fronteira de Pareto. Então daqui em diante a análise se restringirá aos métodos BIOBAB, $GRBmult_p$ e $GRBmult_l$. Pela métrica hipervolume, em 8 das 15 instâncias apresentadas na Tabela 11 os valores obtidos pelos métodos não apresentam muita diferença, mas nas instâncias $(50, 5)_2$, $(50, 5)_4$, $(50, 10)_2$, $(50, 10)_4$, $(100, 5)_2$, $(100, 5)_4$ e $(100, 10)_4$ essa diferença é maior, sendo o BIOBAB o método que obteve melhor desempenho. Na Figura 28 evidencia-se o método que obteve melhor desempenho em relação a métrica hipervolume. O BIOBAB foi superior em todas as instâncias com exceção da instância $(100, 5)_3$, em que o melhor desempenho foi obtido pelo método $GRBmult_l$. Analisando a métrica cardinalidade, através da Tabela 11 e da Figura 29 percebe-se que o método BIOBAB teve melhor desempenho nas 9 primeiras instâncias das 15 instâncias, o método $GRBmult_p$ obteve melhor desempenho em 5 das 15 instâncias e o método $GRBmult_l$ obteve melhor desempenho em 3 das 15 instâncias.

Figura 28 – Métrica hipervolume das instâncias 2, 3 e 4 para as classes $(50, m)$ e $(100, m)$



Fonte: Elaborado pelo autor.

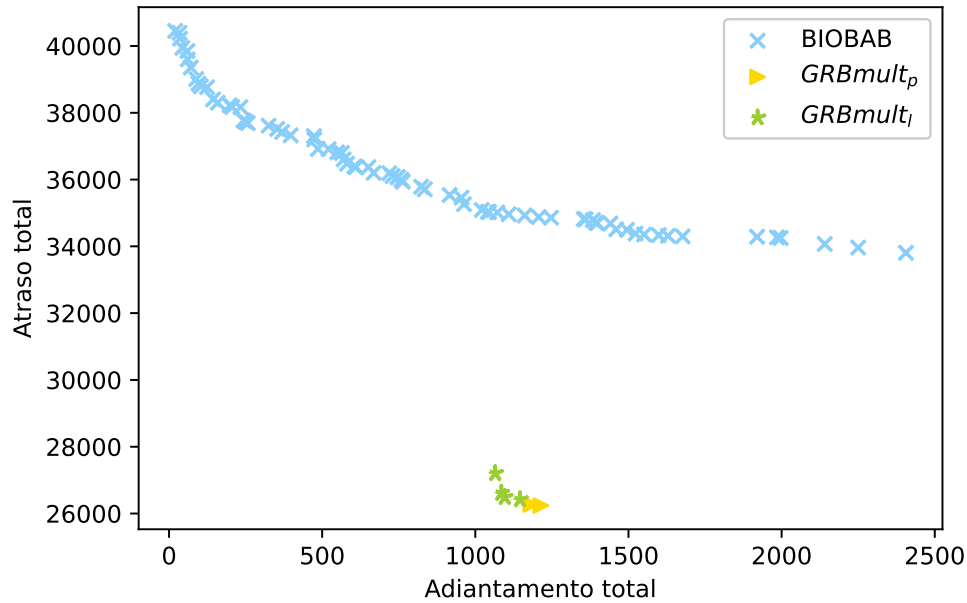
Figura 29 – Métrica cardinalidade das instâncias 2, 3 e 4 para as classes $(50, m)$ e $(100, m)$



Fonte: Elaborado pelo autor.

Em 73,33% das instâncias os valores da métrica de hipervolume obtidos pelo método BIOBAB são praticamente iguais aos valores obtidos pelos métodos $GRBmult_p$ e $GRBmult_l$, porém, para a métrica cardinalidade os valores se diferem chegando a uma diferença máxima de 99 pontos na instância $(50, 20)_3$, com o método BIOBAB apresentando mais pontos. Isso mostra uma característica no método BIOBAB, que também foi notada nas discussões anteriores. O método BIOBAB consegue obter muitos pontos na AFP em instâncias com menos de 100 tarefas, mas muitos desses pontos são dominados e, por vezes, estão acumulados, como pode ser visto na Figura 30 com as AFP's na instância $(50, 10)_4$.

Figura 30 – AFP's do sistema BIOBAB e dos métodos $GRBmult_p$ e $GRBmult_l$ na instância $(50, 10)_4$



Fonte: Elaborado pelo autor.

Em geral, as conclusões da comparação da Parte 2 corroboram com as da Parte 1, ressaltando que o método BIOBAB se destaca em obter muitos pontos nas AFP's dos problemas de pequeno e médio porte, e também, em obter maior hipervolume com 50 e 100 tarefas.

5.2 Estudo 2 - Vantagens do tratamento biobjetivo comparado ao monoobjetivo do JIT-FSP

Das 160 instâncias resolvidas, 4 não foram resolvidas pelo método BIOBAB. A análise dos resultados será baseada nas 156 instâncias resolvidas de forma ótima dentro do tempo limite de 3600 segundos. O uso da palavra “significativa” e suas conjugações sempre será empregada como estatisticamente significativo.

Primeiro, analisa-se o esforço computacional para resolver o problema em sua versão monoobjetivo e utilizando o método *branch-and-bound* biobjetivo. Na Tabela 12, na primeira coluna apresenta-se a classe das instâncias no formato $TF_DR_n_m$ com TF sendo o fator de atraso e DR a dispersão das datas de vencimento. Nas colunas seguintes para o método *branch-and-bound* biobjetivo (BIOBAB) será analisado: o número de soluções (nSol), número de nós (nNós) e tempo de execução em segundo (Tempo(s)); e

para a versão monoobjetivo (GRB_{mono}) será analisado: número de nós (nNós) e tempo de execução em segundos (Tempo(s)).

Tabela 12 – Esforço computacional das abordagens multiobjetivo e monoobjetivo (média de 10 instâncias)

Classe	BIOBAB			GRB_{mono}	
	nSol	nNós	Tempo(s)	nNós	Tempo(s)
0,2_0,6_5_3	8,40	97,40	0,20	1,00	0,01
0,2_0,6_10_3	40,00	56838,60	172,26	37,80	0,06
0,2_0,6_10_7	16,55	95328,11	723,92	1291,80	0,21
0,2_0,6_10_10	9,75	137406,00	1495,84	5627,70	0,64
0,2_1,2_5_3	3,50	94,40	0,15	1,00	0,01
0,2_1,2_10_3	15,50	132058,20	477,28	426,00	0,10
0,2_1,2_10_7	7,30	115600,40	861,74	644,10	0,21
0,2_1,2_10_10	5,60	171497,60	1418,16	3344,70	0,43
0,4_0,6_5_3	4,00	64,80	0,12	0,01	0,01
0,4_0,6_10_3	20,00	32858,80	109,03	281,20	0,08
0,4_0,6_10_7	7,30	109635,20	731,62	1498,80	0,21
0,4_0,6_10_10	5,30	163233,30	600,42	3390,10	0,39
0,4_1,2_5_3	4,00	73,60	0,12	1,00	0,01
0,4_1,2_10_3	9,44	87927,44	302,16	336,70	0,10
0,4_1,2_10_7	5,60	116169,00	649,27	1463,00	0,24
0,4_1,2_10_10	3,70	122221,60	693,14	3174,10	0,37

Os métodos de solução multiobjetivos são propostos para encontrar diferentes soluções exibindo o *trade-off* entre os objetivos do problema. É natural que esses métodos exijam um esforço computacional maior, pois são resolvidos diversos problemas monoobjetivos. Então eram esperados tempos de execução maiores, principalmente utilizando um método exato como o BIOBAB. O maior interesse nessa parte do estudo está na análise do conflito entre o adiantamento total e o atraso total, que pode ser medido através do número de soluções na fronteira de Pareto.

Para compreender os resultados obtidos, é necessário analisar a interferência dos parâmetros DR e TF no conflito entre o adiantamento total e o atraso total. Na medida que DR aumenta, o intervalo de distribuição dos instantes de vencimento das tarefas também aumenta. Além disso, os parâmetros são gerados seguindo uma distribuição uniforme. Esses dois fatos contribuem para que a alocação das tarefas seja favorecida, diminuindo o conflito entre atrasá-las ou adiantá-las. O parâmetro TF também influencia

no conflito do problema, pois com o seu aumento, o intervalo de distribuição das datas de vencimento é deslocado para a esquerda, ou seja, todas as tarefas devem ser concluídas antes. Isso diminui a chance de ter alguma tarefa com adiantamento. Se não há adiantamento de tarefa, o objetivo adiantamento total pode ser ignorado, e o problema deixa de ser multiobjetivo. Em 21 das 156 instâncias (13,46%), a fronteira de Pareto é composta por apenas um ponto, descaracterizando o problema como verdadeiramente multiobjetivo.

Com os gráficos de caixas confirma-se o padrão de comportamento do número de soluções. Conforme DR e TF aumentam, em geral provoca-se uma queda no número de soluções. Essa queda é mais significativa em relação ao parâmetro TF , justamente pela possibilidade de não permitir que as tarefas adiantem.

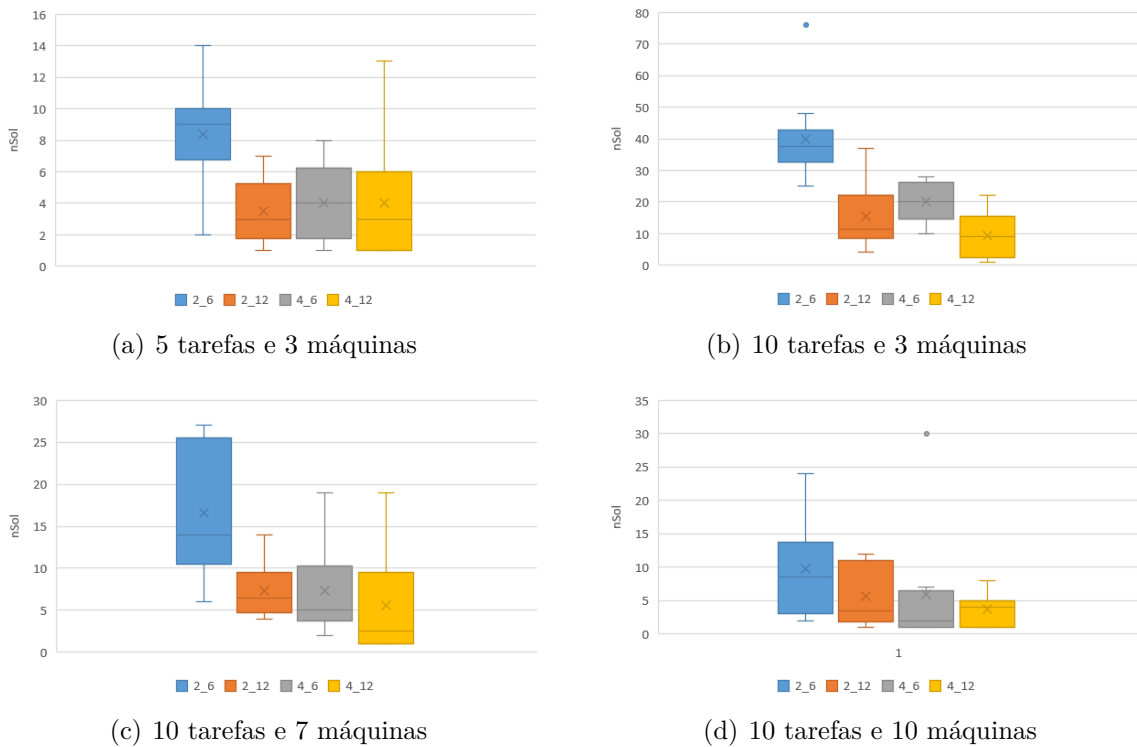


Figura 31 – Gráficos de caixa do número de pontos na fronteira de Pareto por porte do problema

Os resultados exibidos na Tabela 13 são as médias para 10 instâncias. A primeira coluna informa as classes das instâncias, seguindo o formato $TF_DR_n_m$. Nas colunas 2 (D_I), 3 (D_μ) e 4 (D_Δ) exibem-se os valores das ferramentas de localização do ponto na fronteira de Pareto. Nas colunas 5 ($SP_{min}(\%)$), 6 ($SP_\mu(\%)$) e 7 ($SP_{max}(\%)$) são apresentados os aumentos da função objetivo ao selecionar outro ponto na fronteira de Pareto. A coluna 8 ($f_i = f^*(\%)$) exibe o percentual de pontos na fronteira de Pareto que não provocam aumento na função objetivo.

Em 14 classes de instâncias, D_μ , D_I e D_I determinaram a mesma classificação para o ponto do problema monoobjetivo, na classe 0,2_0,6_10_10 D_μ classificou o ponto como desequilibrado enquanto D_I e D_Δ classificou ele como moderado e na classe 0,4_1,2_10_3

Tabela 13 – Valores das ferramentas de localização e aumento de custo do ponto monoobjetivo (média de 10 instâncias)

Classe	D_I	D_μ	D_Δ	$SP_{min}(\%)$	$SP_\mu(\%)$	$SP_{max}(\%)$	$f_i = f^*(\%)$
0,2_0,6_5_3	0,21	0,43	0,45	7,62	22,75	41,74	40,00
0,2_0,6_10_3	0,00	0,03	0,03	0,53	34,40	107,00	60,00
0,2_0,6_10_7	0,14	0,38	0,28	4,02	18,85	44,50	50,00
0,2_0,6_10_10	0,38	0,56	0,49	4,21	10,57	19,98	60,00
0,2_1,2_5_3	0,70	0,70	0,70	3,20	11,33	18,54	40,00
0,2_1,2_10_3	0,21	0,24	0,27	0,00	24,86	71,88	100,00
0,2_1,2_10_7	0,41	0,43	0,49	0,52	10,29	24,07	60,00
0,2_1,2_10_10	0,70	0,86	0,86	0,88	2,94	5,43	30,00
0,4_0,6_5_3	0,60	0,64	0,63	6,14	13,07	20,94	30,00
0,4_0,6_10_3	0,15	0,38	0,38	0,17	9,41	24,93	80,00
0,4_0,6_10_7	0,50	0,67	0,73	0,85	4,00	9,42	30,00
0,4_0,6_10_10	0,70	0,82	0,83	1,76	3,04	4,23	30,00
0,4_1,2_5_3	0,67	0,73	0,72	2,59	6,60	13,10	20,00
0,4_1,2_10_3	0,38	0,49	0,51	1,02	9,77	18,95	60,00
0,4_1,2_10_7	0,65	0,75	0,71	0,15	1,16	2,95	40,00
0,4_1,2_10_10	0,74	0,84	0,87	0,24	2,01	4,02	40,00

D_Δ classificou o ponto como desequilibrado enquanto D_μ e D_I classificaram como moderado. Das 14 classes em que houve uma concordância entre as ferramentas, apenas para uma delas o ponto monoobjetivo foi classificado como equilibrado, enquanto em 5 (35,71%) delas o ponto foi classificado como moderadas e em 8 (57,14%) delas foi classificado como desequilibrado. Isso mostra que na maioria das instâncias o ponto monoobjetivo está mais próximo de um ponto lexicográfico, dando mais prioridade em adiantar ou atrasar a produção, uma característica que pode não ser desejada nos sistemas de produção.

Analisando os gráficos da Figura 32 é notório que ao alterar os valores de TF e DR , o ponto monoobjetivo tende a estar mais próximo de algum ponto extremo da fronteira de Pareto. Esse movimento é uma consequência direta da diminuição de conflito entre os objetivos. Com menos pontos na fronteira de Pareto, a chance do ponto monoobjetivo ser um ponto extremo ou estar mais próximo de um ponto extremo aumenta.

Fixando TF e DR , ao analisar as variações no porte das instâncias, não é possível estabelecer uma relação direta com os valores retornados pelas ferramentas, pois há uma queda com 10 tarefas e 3 máquinas e depois um aumento considerando 10 tarefas e 7 e 10 máquinas.

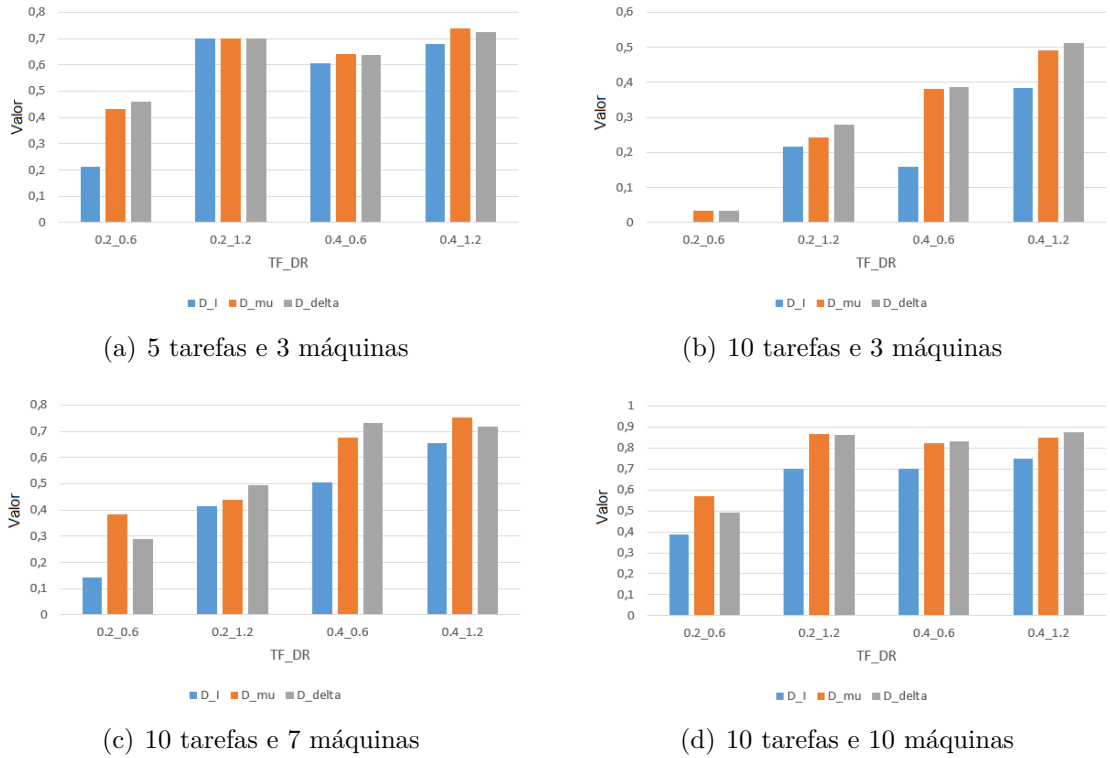


Figura 32 – Gráficos de colunas dos valores das ferramentas de localização por porte do problema

Ao observar os aumentos percentuais na Figura 33, há um movimento inverso ao observado na Figura 32. Ao alterar os valores de TF e DR , os aumentos, principalmente o máximo, diminuem. Apesar desse movimento de queda nos aumentos, os valores são significativamente altos, com uma média de 2,12% para o aumento mínimo, considerando todas as instâncias e desvio padrão de 2,25%. Pelo desvio padrão desse dado, percebe-se que o aumento mínimo pode ser igual a 0. Isso é confirmado ao analisar a coluna 8 da Tabela 13, em que mostra pontos que não provocam aumento no valor objetivo, em média 48,12% das instâncias.

A possibilidade de, em quase 50% das instâncias, terem pontos na fronteira de Pareto que não aumentam o valor da função monoobjetivo e oferecem um sequenciamento diferente das tarefas, reforça o grande potencial da abordagem multiobjetivo para esse problema.

Os resultados apresentados na Tabela 14 consideram o número de soluções suportadas e não suportadas para 121 (77,56%) das 156 instâncias, aquelas que possuem pelo menos 3 pontos na fronteira de Pareto (média de cada classe). A primeira coluna fornece as classes de instâncias (Classe), as colunas 2 (Suportados) e 3 (Não suportados) mostram o número de pontos suportados e não suportados na fronteira de Pareto, respectivamente, a coluna 4 (Não Sup/Total) mostra a percentagem de pontos não suportados na fronteira de Pareto e na coluna 5 ($|\mathcal{N}| \geq 3$), são exibidos o número de instâncias que têm pelo menos 3 pontos na fronteira de Pareto. É perceptível que no problema de sequenciamento *just-in-time flow*

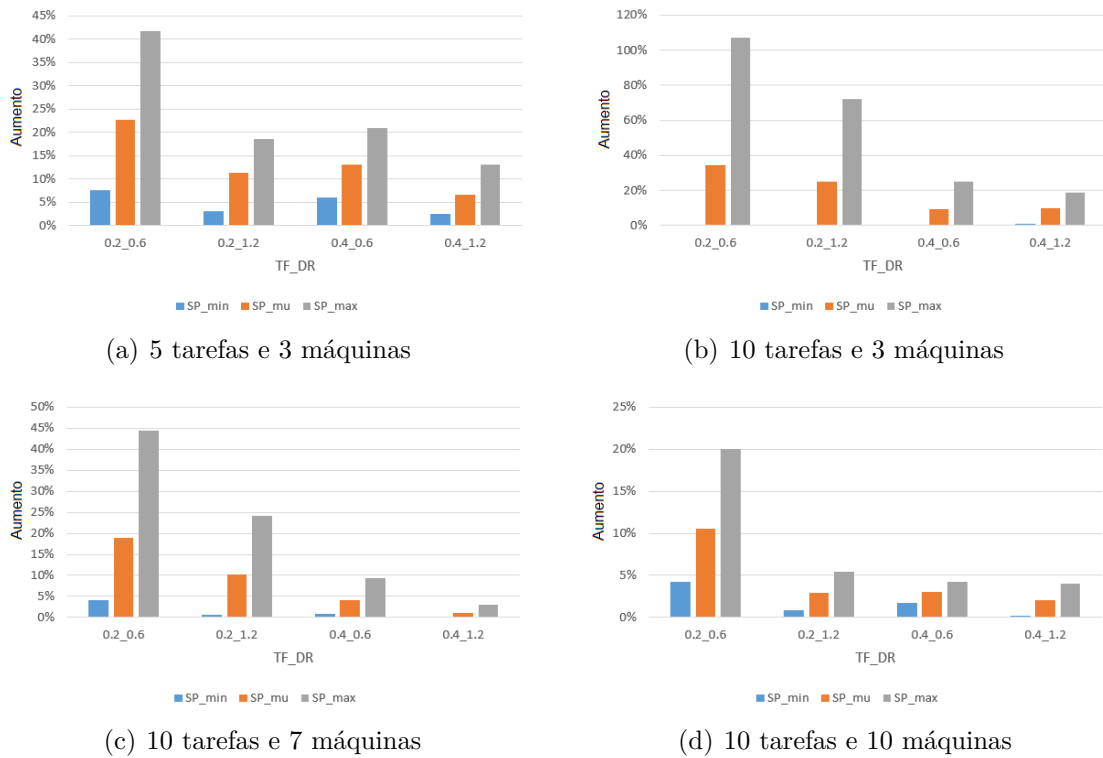


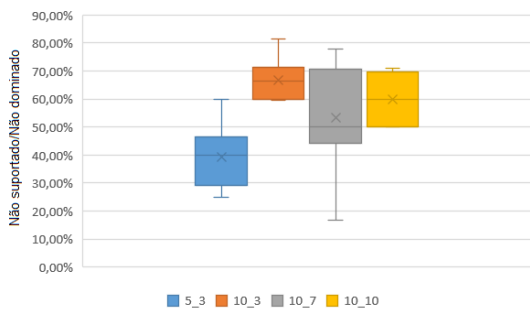
Figura 33 – Gráficos de colunas dos aumentos mínimo, médios e máximos do valor objetivo por porte do problema

shop, os pontos não suportados constituem uma parte significativa da fronteira de Pareto. A média máxima atinge 66,71% de pontos não suportados na fronteira de Pareto, com uma média de 43,90% e um desvio padrão de 13,84%. Esta presença torna-se mais pronunciada à medida que o tamanho da instância aumenta. Isto pode ser melhor observado na Figura 34, que apresenta o percentual de pontos não suportados na fronteira de Pareto por porte n_m considerando os parâmetros $(TF, DR) = \{(0, 2; 0, 6), (0, 2; 1, 2)\}$, no qual o conflito entre os objetivos é maior. Comparando as instâncias com 5 tarefas e 3 máquinas com as instâncias com 10 tarefas e 10 máquinas, o aumento de pontos não suportados na fronteira de Pareto aumenta significativamente, atingindo máximos de 70% e 77,78%, respetivamente.

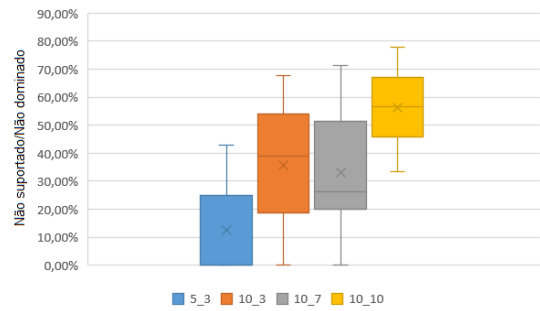
Com estes resultados, é evidente que a utilização de um método multiobjectivo que apenas determina os pontos suportados na fronteira de Pareto pode ocultar as vantagens da abordagem multiobjectivo para o problema JIT-FSP. Esta questão é agravada à medida que a dimensão do problema aumenta, uma vez que uma grande parte da fronteira não será apresentada, reduzindo as opções do tomador de decisão. Além disso, os parâmetros TF e DR também influenciam no conflito entre os objetivos para instâncias de todos os portes, podendo reduzir ou aumentar o número de pontos na fronteira de Pareto.

Tabela 14 – Número de pontos suportados e não suportados na fronteira de Pareto (média de cada classe de instâncias)

Classe	Suportados	Não suportados	Não Sup/Total	$ \mathcal{N} \geq 3$
0,2_0,6_5_3	5,55	3,55	39,13%	9
0,2_0,6_10_3	12,60	27,40	66,71%	10
0,2_0,6_10_7	6,55	10,00	53,38%	9
0,2_0,6_10_10	4,50	7,83	59,98%	6
0,2_1,2_5_3	3,71	0,71	12,55%	7
0,2_1,2_10_3	8,20	7,30	35,69%	10
0,2_1,2_10_7	4,60	2,70	33,18%	10
0,2_1,2_10_10	3,33	5,00	56,27%	6
0,4_0,6_5_3	3,50	2,16	35,48%	6
0,4_0,6_10_3	7,40	12,60	58,56%	10
0,4_0,6_10_7	3,77	4,11	45,39%	9
0,4_0,6_10_10	3,80	9,60	57,14%	5
0,4_1,2_5_3	4,40	2,40	28,77%	5
0,4_1,2_10_3	5,00	6,71	48,14%	7
0,4_1,2_10_7	5,00	5,00	39,06%	5
0,4_1,2_10_10	3,14	1,71	32,98%	7



(a) $TF = 0,2$ e $DR = 0,6$



(b) $TF = 0,2$ e $DR = 1,2$

Figura 34 – Gráficos de caixas do percentual de pontos não suportados na fronteira de Pareto por cenário

5.3 Estudo 3 - Comparação dos métodos biobjetivos

Nessa seção apresenta-se um estudo comparativo entre os métodos: BIOBAB, caixa balanceada, ε -restrito aumentado e GRBmult (com busca em profundidade); utilizando as métricas de cardinalidade e de *hiper ratio*. Vale lembrar que a métrica de *hiper ratio* possui valor máximo igual a 1, quando a AFP obteve maior hipervolume, e valor mínimo igual a

0, quando não há pontos na AFP. Nas análises a seguir, o uso da palavra “significativa” e suas conjugações sempre será empregada como estatisticamente significativo.

Na Tabela 15 apresenta-se na primeira coluna a classe das instâncias (Classe). Para cada classe, são apresentados nas colunas correspondentes os valores da métrica de cardinalidade (#) e de *hiper ratio* (HR) e o tempo de execução (tempo) respectivamente relacionados a cada método de solução. Pela Tabela 15 percebe-se que o ε -restrito aumentado é o método que mais determina pontos na fronteira de Pareto, seguido pela caixa balanceada, BIOBAB e por último, o *GRBmult*. A diferença na cardinalidade do ε -restrito aumentado e da caixa balanceada não é significativa para as 16 classes de instâncias. Para esses dois métodos, existe uma diferença significativa em relação ao BIOBAB e ao *GRBmult* nas mesmas 10 classes de instâncias. O fato interessante é que essa diferença é significativa nas classes em que o conflito entre os objetivos é maior, seguindo os resultados obtidos na Seção 5.2. Na Figura 35(a), mostra-se o gráfico de caixa referente à cardinalidade obtida pelos métodos de solução na classe 0,2_0,6_5_3. Nota-se que o BIOBAB e o GRB obtêm uma AFP significativamente menor do que a caixa balanceada e o ε -restrito aumentado. Esse comportamento se mantém para os 4 portes considerando $(TF, DR) = (0, 2; 0, 6)$, diminui para 3 com $(TF, DR) = (0, 2; 1, 2)$, para 2 com $(TF, DR) = (0, 2; 0, 4)$ e para 1 com $(TF, DR) = (0, 4; 1, 2)$. Esse movimento de queda se mostra mais acentuado com o aumento do TF , também entrando em concordância aos resultados do Estudo 2 (Seção 5.2). Para as outras classes, o comportamento se assemelha ao exibido na Figura 35(b), em que não há diferença significativa na cardinalidade obtida pelos métodos de solução, provocado por instâncias que possuem poucos pontos na fronteira de Pareto.

Tabela 15 – Valores do tempo de execução e das métricas de cardinalidade e de hipervolume dos métodos BIOBAB, caixa balanceada, ε -restrito aumentado e GRBmult

Classe	BIOBAB				Caixa balanceada				ε -restrito aumentado				GRBmult			
	tempo	#	HR	tempo	tempo	#	HR	tempo	tempo	#	HR	tempo	tempo	#	HR	tempo
0,2_0,6_5_3	0,21	8,40	0,8934	0,99	99,90	99,90	0,9987	0,52	117,40	1,0000	0,01	2,30	0,5913			
0,2_0,6_10_3	172,26	40,00	0,9752	29,09	384,20	0,9996	29,57	487,80	1,0000	0,08	2,60	0,7205				
0,2_0,6_10_7	723,93	18,90	0,9562	283,79	134,90	0,9992	32,40	147,00	1,0000	0,21	2,60	0,6904				
0,2_0,6_10_10	1495,85	13,30	0,8801	1041,23	145,00	0,9964	135,81	168,30	1,0000	0,64	3,00	0,6876				
0,2_1,2_5_3	0,15	3,50	0,7811	0,55	51,60	0,9987	0,35	59,30	0,9998	0,01	1,40	0,5774				
0,2_1,2_10_3	477,29	15,50	0,9247	40,86	173,20	0,9991	18,92	215,70	1,0000	0,11	2,20	0,5842				
0,2_1,2_10_7	861,75	7,30	0,8299	107,13	97,50	0,9988	22,30	116,40	1,0000	0,22	1,60	0,5029				
0,2_1,2_10_10	1418,16	5,60	0,7931	214,39	39,00	0,9852	30,97	52,80	1,0000	0,44	2,20	0,5125				
0,4_0,6_5_3	0,13	4,00	0,8701	0,31	36,90	0,9997	0,19	38,50	0,9998	0,01	1,50	0,6182				
0,4_0,6_10_3	109,04	20,00	0,9670	31,75	124,50	0,9993	11,57	137,70	1,0000	0,09	2,40	0,6717				
0,4_0,6_10_7	731,63	7,30	0,8271	228,96	69,30	0,9988	17,88	75,70	0,9999	0,23	1,70	0,3083				
0,4_0,6_10_10	667,14	7,40	0,8897	495,20	77,40	0,9993	39,71	84,10	1,0000	0,42	2,70	0,6071				
0,4_1,2_5_3	0,13	4,00	0,8974	0,61	37,10	0,9989	0,30	40,40	1,0000	0,01	1,60	0,7835				
0,4_1,2_10_3	302,17	10,60	0,8815	40,73	92,60	0,9991	11,42	103,90	1,0000	0,11	2,00	0,5750				
0,4_1,2_10_7	649,27	5,60	0,8910	148,32	58,80	0,9990	16,02	68,50	1,0000	0,25	1,90	0,7153				
0,4_1,2_10_10	693,15	3,70	0,8172	262,65	29,40	0,9973	14,14	33,00	1,0000	0,38	1,80	0,5408				

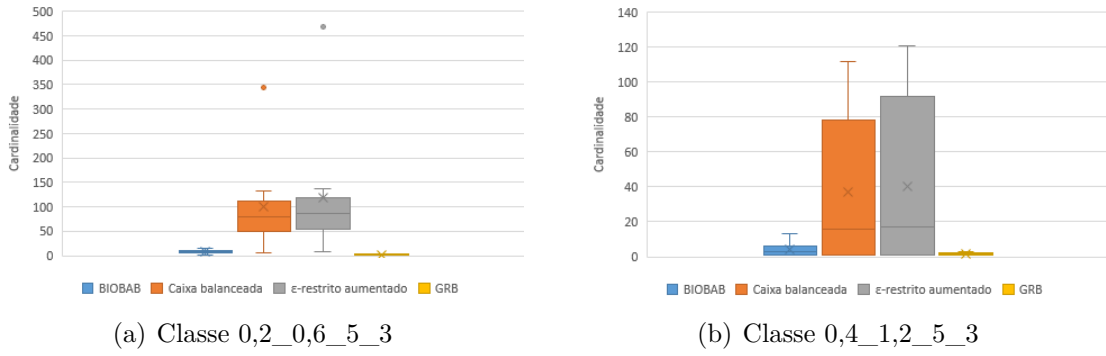


Figura 35 – Gráficos de caixas da cardinalidade das AFP's dos métodos BIOBAB, caixa balanceada, ε -restrito aumentado e GRBmult nas classes 0,2_0,6_5_3 e 0,4_1,2_5_3

As conclusões obtidas ao analisar o *hiper ratio* são iguais quando analisa-se a cardinalidade. O ε -restrito aumentado apresentou o maior valor seguido pela caixa balanceada, mas sem uma diferença significativa e, para essas duas, houve uma diferença significativa no *hiper ratio* do BIOBAB e do *GRBmult*, nas mesmas classes de instâncias analisadas na cardinalidade (Figura 36). Um fato que o diferencia é que, embora o BIOBAB apresente *hiper ratio* significativamente inferior ao *hiper ratio* do ε -restrito aumentado, essa diferença não é tão acentuada quanto na análise da cardinalidade, como representado na Figura 36(a).

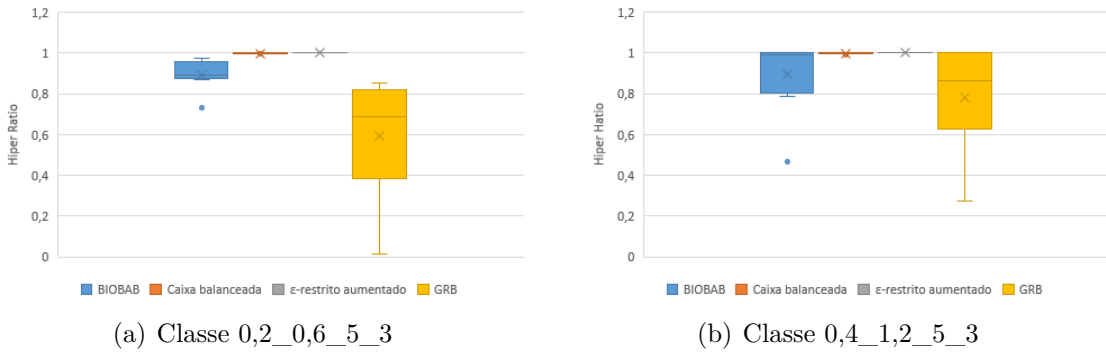


Figura 36 – Gráficos de caixas do *hiper ratio* das AFP's dos métodos BIOBAB, caixa balanceada, ε -restrito aumentado e GRBmult nas classes 0,2_0,6_5_3 e 0,4_1,2_5_3

Pelas métricas de desempenho Cardinalidade e *hiper ratio*, conclui-se que o ε -restrito aumentado e caixa balanceada não possuem diferenças significativas para as fronteiras de Pareto, sendo os dois os que obtiveram melhores resultados. O BIOBAB apresentou significativamente menos pontos na fronteira de Pareto e *hiper ratio* inferior nas mesmas 10 classes de instâncias, com diferença acentuada na cardinalidade. Isso mostra que os pontos nas AFP's obtidas pelo BIOBAB estão bem distribuídos, pois apesar da diferença do *hiper ratio* ser significativa, ela não é tão acentuada. Por último, o *GRBmult* foi o método com menor desempenho nas métricas cardinalidade e *hiper ratio*, com diferenças significativas e

acentuadas nas classes 0,2_0,6_n_m, 0,2_1,2_5_3, 0,2_1,2_10_3, 0,2_1,2_10_7, 0,4_0,6_10_3, 0,4_0,6_10_7 e 0,4_1,2_10_3 (10 classes de instâncias no total).

Na Tabela 15 mostra-se que o método com menor tempo de execução é o *GRBmult*, com tempo inferior a 0,1 segundos para todas as classes de instâncias (diferença significativa aos outros métodos). Desconsiderando o *GRBmult*, o ε -restrito aumentado obteve melhor tempo em 11 classes de instâncias com diferença significativa nas 11 classes em relação ao BIOBAB e em 3 em relação à caixa balanceada. Seguindo, o BIOBAB teve melhor tempo em 4 classes de instâncias, com diferença significativa em uma delas em relação ao ε -restrito aumentado e caixa balanceada. Por fim, a caixa balanceada teve menor tempo em 1 classe de instâncias com diferença significativa apenas em relação ao BIOBAB. Considerando o ε -restrito aumentado, caixa balanceada e BIOBAB, apesar do BIOBAB exigir menor tempo em mais classes do que a caixa balanceada, considerando apenas os dois métodos, a caixa balanceada teve menor tempo em 12 classes de instâncias.

Após analisar todos os parâmetros de desempenho dos métodos, é notório que o alto desempenho do *GRBmult* em relação ao tempo de execução é inversamente proporcional ao seu desempenho em relação às métricas cardinalidade e *hiper ratio*, mostrando não ser uma escolha de método multiobjetivo promissora. O BIOBAB fica no meio em relação aos outros métodos. Ele obtém menos pontos na AFP, porém melhor distribuídos. Em relação ao tempo de execução, conforme o porte das instâncias crescem o tempo cresce rapidamente, característica já relatada no Estudo 2 (Seção 5.2). Por fim, a caixa balanceada e o ε -restrito aumentado se destacaram, principalmente em relação à cardinalidade e ao *hiper ratio*. Em termos de tempo, o ε -restrito aumentado mostrou ser superior em relação à caixa balanceada.

Avaliação do uso da ϵ -dominância no método ε -restrito aumentado

Ter um método de solução capaz de determinar uma AFP representativa em tempo aceitável é muito importante para o tomador de decisão. Retomando os resultados obtidos pelo método ε -restrito aumentado na Seção 5.3, em média o método determinou 121,65 pontos nas suas AFP's. Para o tomador de decisão é inviável analisar todos esses pontos, sendo necessário alguma estratégia para facilitar a análise. Existem algumas estratégias pós otimização, que utilizam de recursos visuais mais elaborados (*e.g.* [91, 92]). Uma forma de facilitar a análise da fronteira de Pareto é utilizar a ϵ -dominância durante a otimização do problema para filtrar parte da fronteira de Pareto mantendo pontos que sejam representativos.

Nesse estudo computacional mostra-se o *trade-off* entre reduzir o número de pontos na AFP através da ϵ -dominância e manter o *hiper ratio* próximo ao obtido pelo ε -restrito aumentado. O método escolhido para esse estudo foi o ε -restrito aumentado por ter sido o destaque da Seção 5.3. Selecionou-se as classes de instâncias 0,2_0,6_10_3, 0,2_1,2_10_3,

0,4_0,6_10_3 e 0,4_1,2_10_3 por serem as classes com mais pontos em média nas AFP's para cada par (TF, DR) . Foram utilizados três valores para a ϵ -dominância, $\epsilon = \{1, 2, 3\}$ e comparados com o método sem utilizar a ϵ -dominância.

Tabela 16 – Métricas de cardinalidade e de *hiper ratio* para variações de ϵ (média das classes)

Sem ϵ -dominância				$\epsilon = 1$			$\epsilon = 2$			$\epsilon = 3$		
Classe	tempo	#	HR	tempo	#	HR	tempo	#	HR	tempo	#	HR
0,2_0,6_10_3	29,57	487,8	1	23,43	303,3	0,9995	12,07	152,6	0,9976	7,38	102,4	0,9959
0,2_1,2_10_3	18,92	215,7	1	15,65	141,2	0,9982	7,79	71,3	0,9923	5,32	48,4	0,9852
0,4_0,6_10_3	11,57	137,7	1	15,84	112,4	0,9997	8,15	57,2	0,9936	5,20	38,9	0,9871
0,4_1,2_10_3	11,42	103,9	1	12,90	82,7	0,9997	6,98	42,1	0,9752	4,48	28,6	0,9686

Pela Tabela 16 nota-se que houve uma diminuição na quantidade de pontos das AFP's, mesmo utilizando $\epsilon = 1$, sendo essa diferença maior conforme ϵ aumenta. Isso mostra que existem pontos nas AFP's com distanciamento inferior a uma unidade, ou seja, uma diferença pequena em um dos objetivos que em questão de análise, talvez não haja uma diferença notável para o tomador de decisão. Observando a Figura 37, estão exibidos gráficos de Gantt referentes à 4 soluções de pontos da AFP obtida quando não utiliza-se a ϵ -dominância. É difícil ver a diferença entre eles, pois entre cada um há um deslocamento de 0,11 unidades à direita na última barra (M3). Como são 10 tarefas, esse deslocamento faz o adiantamento diminuir uma unidade e a tarefa 8 que estava no prazo, passa a atrasar 0,11 unidades.

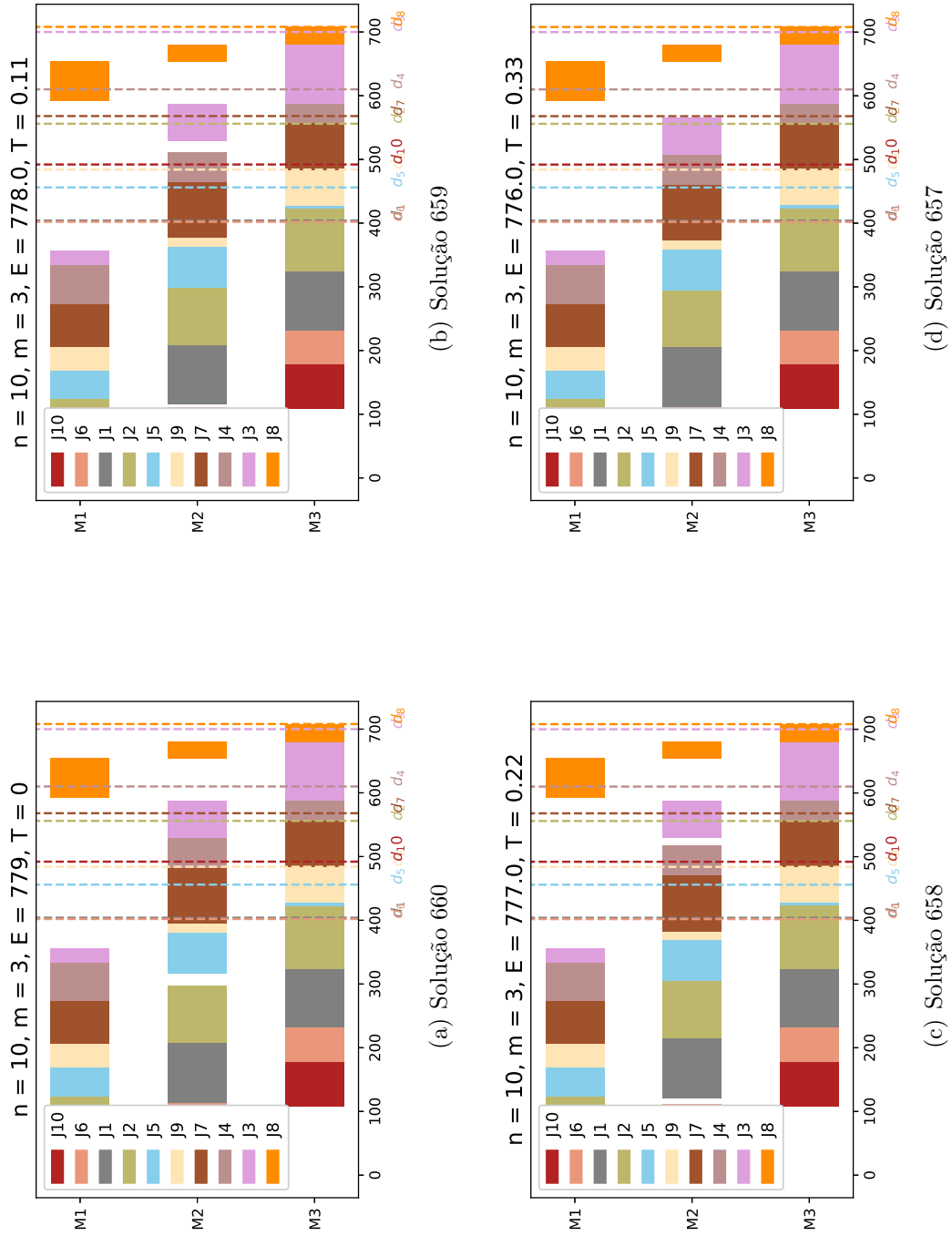


Figura 37 – Gráficos de *Gantt* de 4 soluções da instância 0,2_1,2_10_3_1

Na Figura 38, para a classe 0,2_0,6_10_3 a diferença na cardinalidade é significativa para cada variação do ϵ . Essa diferença diminui nas outras classes de instâncias conforme o conflito entre os objetivos também diminui, porém no teste estatístico ANOVA, que testa se há diferença significativa entre grupos de dados, a hipótese de igualdade das médias nas 4 classes foi rejeitada com um nível de significância de 5%. Com isso, conclui-se que houve uma diferença significativa na cardinalidade ao utilizar a ϵ -dominância para filtrar parte das AFP's.

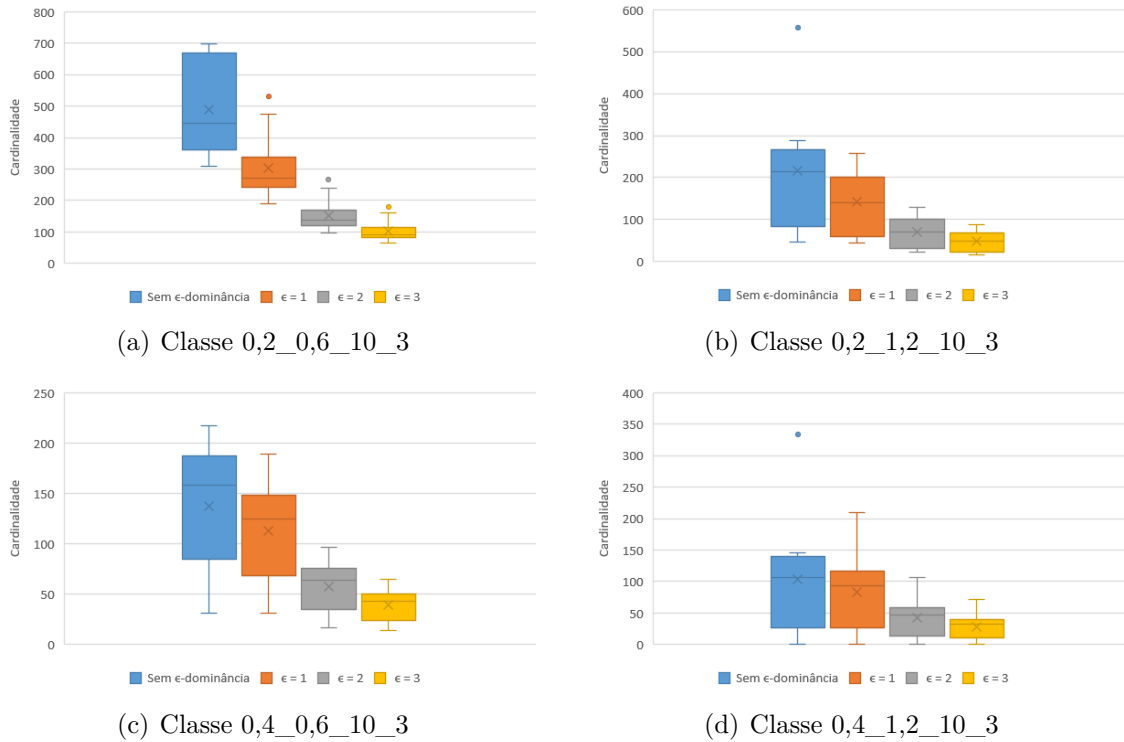


Figura 38 – Gráficos de caixas da cardinalidade das AFP's para cada variação de ϵ

Ao analisar o *hiper ratio* na Tabela 16 existe um decréscimo conforme o valor de ϵ aumenta. Considerando $\epsilon = 1$ a maior diferença, em relação quando não é utilizado a ϵ -dominância, foi de 0,0018 para a classe 0,2_1,2_10_3. Apesar dessa diferença ser pequena nas 4 classes entre esses dois grupos de observações, pelo teste estatístico *T-student*, a hipótese nula de igualdade é rejeitada, evidenciando que não utilizar a ϵ -dominância obtém *hiper ratio* significativamente maior com um nível de significância de 5%. Adicionando os outros valores de ϵ para a análise, o teste ANOVA também rejeita a hipótese de igualdade de médias nas 4 classes de instâncias utilizando o mesmo nível de significância. Com isso, não há dúvidas que as AFP's ficam significativamente menos representativas conforme ϵ aumenta.

Na Figura 39 visualiza-se o *trade-off* entre cardinalidade (eixo das abscissas) e o *hiper ratio* (eixo das ordenadas). É importante lembrar que nessa figura, o interessante para o tomador de decisão seria ter pontos mais a esquerda, representando menos pontos para analisar na AFP, e pontos mais para cima próximos de 1, representando uma AFP

de qualidade. Na classe 0,2_0,6_10_3, os pontos estão mais próximos, se acumulando próximos de 1 ao mesmo tempo que vão para a esquerda, evidenciando os melhores resultados para a ϵ -dominância, pois diminuir a quantidade de pontos penaliza pouco o *hiper ratio*. Para as outras classes, há uma dispersão maior entre os pontos, evidenciando uma penalização maior no *hiper ratio* por diminuir a quantidade de pontos na AFP. Além disso, na classe 0,4_1,2_10_3 existem pontos com penalização acentuada e outros que não possuem penalização. Esse fato mostra a diminuição do conflito entre os objetivos, que geram fronteiras de Pareto com menos pontos. Nesse caso, o uso da ϵ -dominância não é interessante.

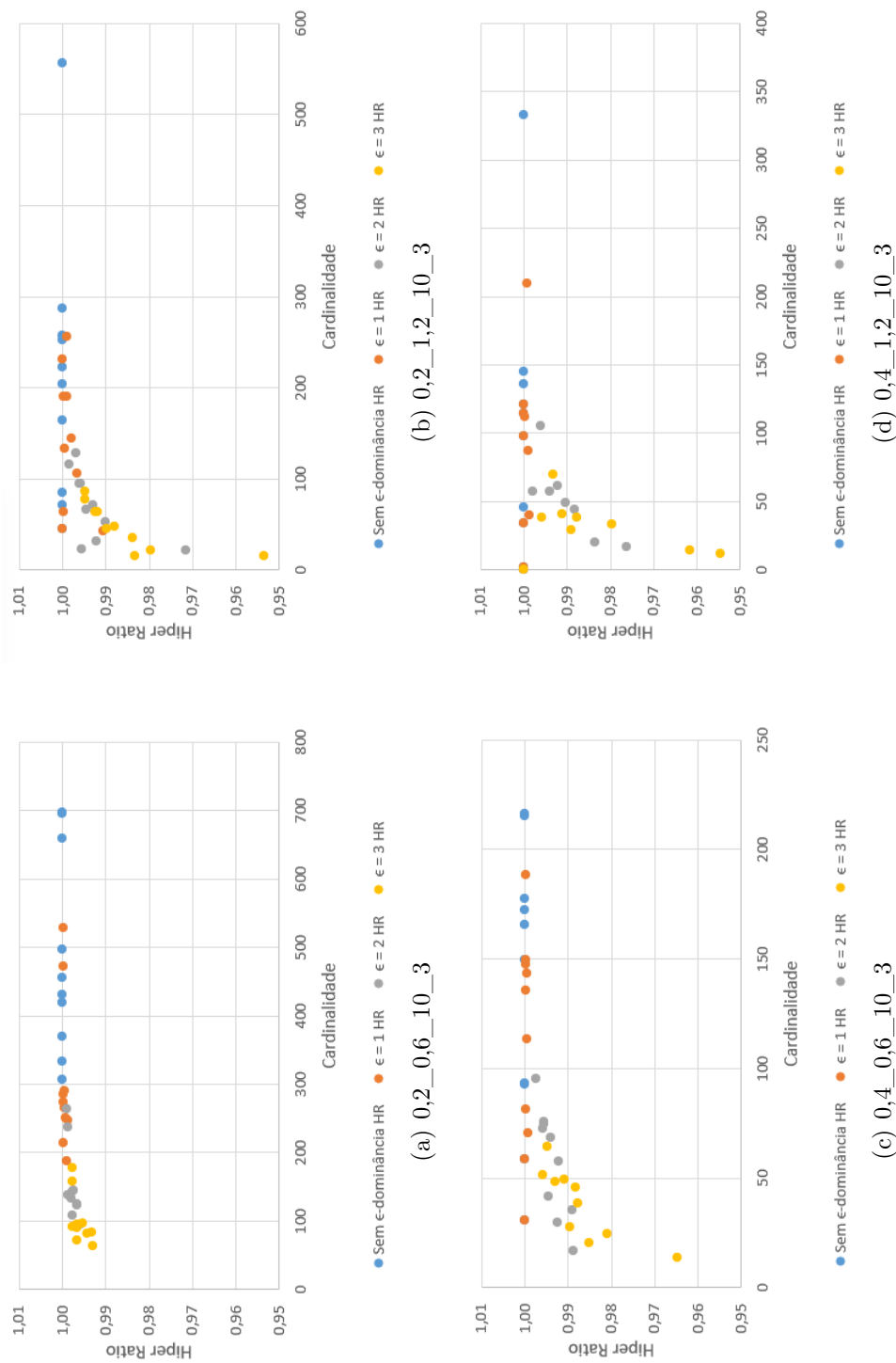


Figura 39 – *Trade-off* entre cardinalidade e *hiper ratio* para cada classe de instâncias

Uma consideração importante a ser feita é em relação ao tempo de solução. Como a ϵ -dominância é aplicada durante a otimização, em geral, existe o decréscimo no tempo computacional, pois menos subproblemas serão resolvidos durante o método multiobjetivo escolhido. Esse também pode ser um fator a ser considerado pelo tomador de decisão. Pela Tabela 16, conforme ϵ aumenta, o tempo de solução diminui. Pela Figura 40, para a classe de maior conflito, a diferença entre os tempos é significativa para $\epsilon = 2$ e $\epsilon = 3$. Conforme o conflito entre os objetivos diminui, a diferença entre os tempos também diminui, comportamento semelhante ao observado na cardinalidade. Uma diferença é que para as classes 0,4_0,6_10_3 e 0,4_1,2_10_3, quando $\epsilon = 1$ houve um aumento no tempo de resolução. Isso é justificável pois a quantidade de pontos é praticamente igual quando não utiliza-se a ϵ -dominância, porém, a implementação da ϵ -dominância é feita adicionando mais restrições aos subproblemas, o que aumentou o tempo de resolução de cada um deles.

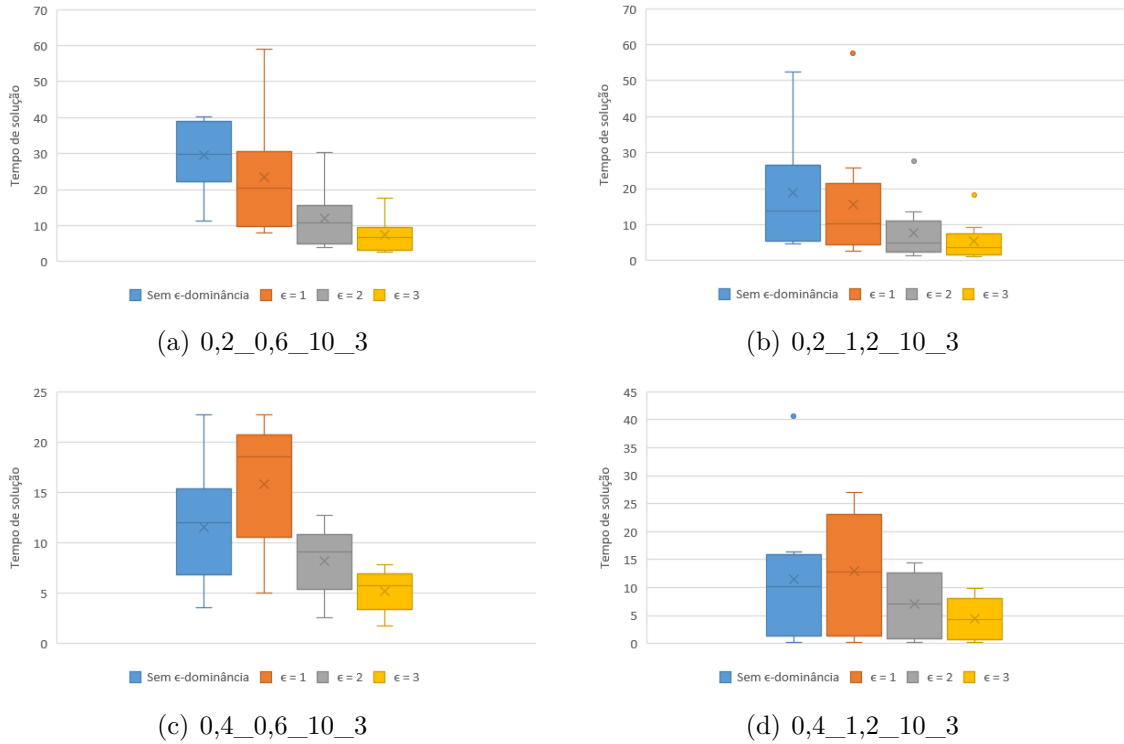


Figura 40 – Gráficos de caixas do tempo de resolução para cada variação de ϵ

5.4 Estudo 4 - Variações do método de decomposição de Benders

As classes de instâncias utilizadas nesse estudo computacional são um recorte das utilizadas nas Seções 5.2 e 5.3. Foi escolhido $TF = 0,2$ e $DR = 0,6$ por serem as instâncias com maior conflito entre os objetivos. São elas: 0,2_0,6_5_3, 0,2_0,6_10_3 e 0,2_0,6_10_3. Para cada uma dessas classes considerou-se 10 instâncias. O uso da

palavra “significativa” e suas conjugações sempre será empregada como estatisticamente significativo.

Na Figura 41, são apresentados os gráficos de caixa referentes aos tempos de execução do método de Benders clássico (B_{cl}) e das 5 variações apresentadas na Seção 4.2. Considerando 5 tarefas e 3 máquinas, a única variação que apresenta tempo significativamente menor ao Benders clássico foi a B_{seq} . Com esse porte de instância as variações geram pouco impacto, justamente porque o B_{cl} resolve todas as instâncias em tempos inferiores a 6 segundos. Para 10 tarefas e 3 máquinas, percebe-se uma queda gradativa nos tempos de execução, no qual torna-se significativamente menor do que o B_{cl} para as variações BBC_{cl} e BBC_{seq} . Já nesse porte de instância, exigir que cada iteração do problema mestre seja resolvida até otimalidade e permitir que soluções sejam revisitadas, como é o caso das variações B_{cl} , B_{seq} , B_{RL} e B_{seq}^{RL} , mostra ser prejudicial no desempenho do método. O uso de árvore única e adicionar cortes conforme são encontradas qualquer solução factível, interfere significativamente no desempenho da resolução de instâncias de maior porte, como o caso de 10 tarefas e 7 máquinas. Nesse porte, as três variações que utilizam a estrutura do método de Benders clássico, incluindo o próprio método, chegaram no tempo de execução limite em 8 das 10 instâncias, enquanto os métodos BBC_{cl} e BBC_{seq} resolveram todas as instâncias com tempos inferiores a 890 segundos.

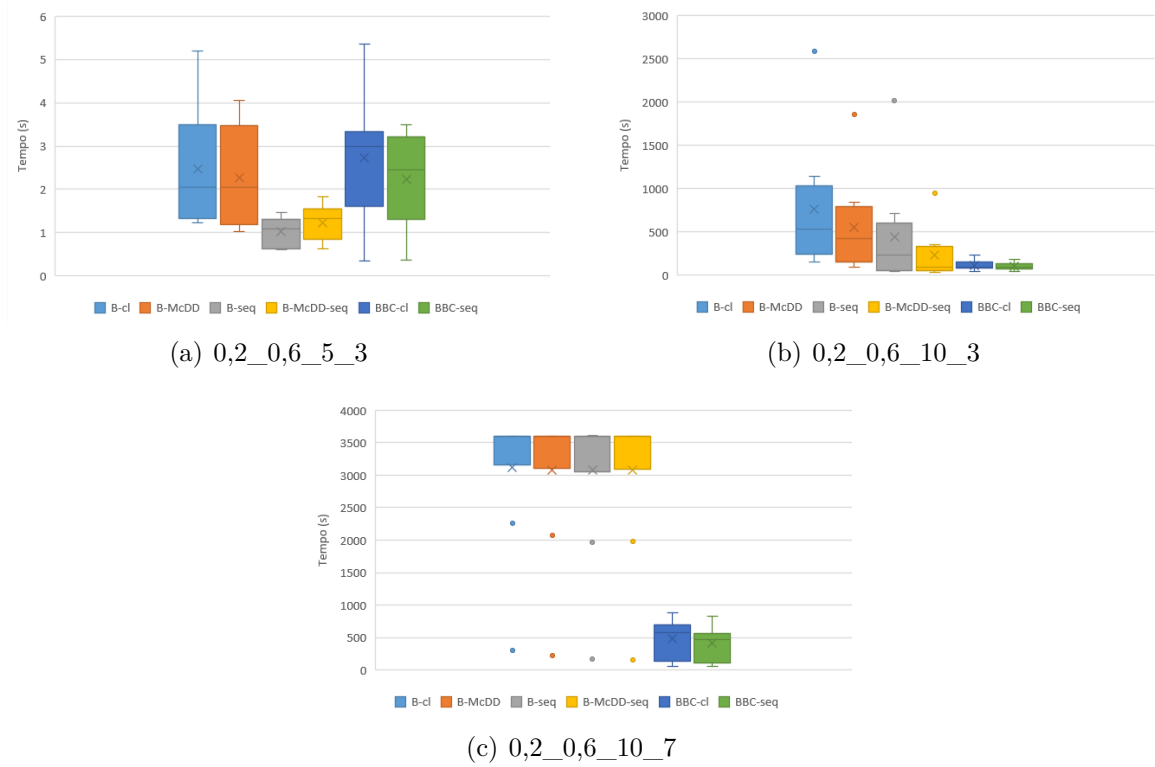


Figura 41 – Gráficos de caixas para o tempo de execução por porte das instâncias

Considerando 10 tarefas e 7 máquinas, com exceção dos métodos BBC , os métodos resolveram apenas 2 instâncias até a otimalidade. Nessas 8 instâncias que não foram

resolvidas até a otimalidade, há uma diferença muito grande na quantidade de pontos encontrados na AFP, conforme mostra a Figura 42. Houve poucas iterações do método ε -restrito aumento, evidenciando que o método de Benders demora para convergir.

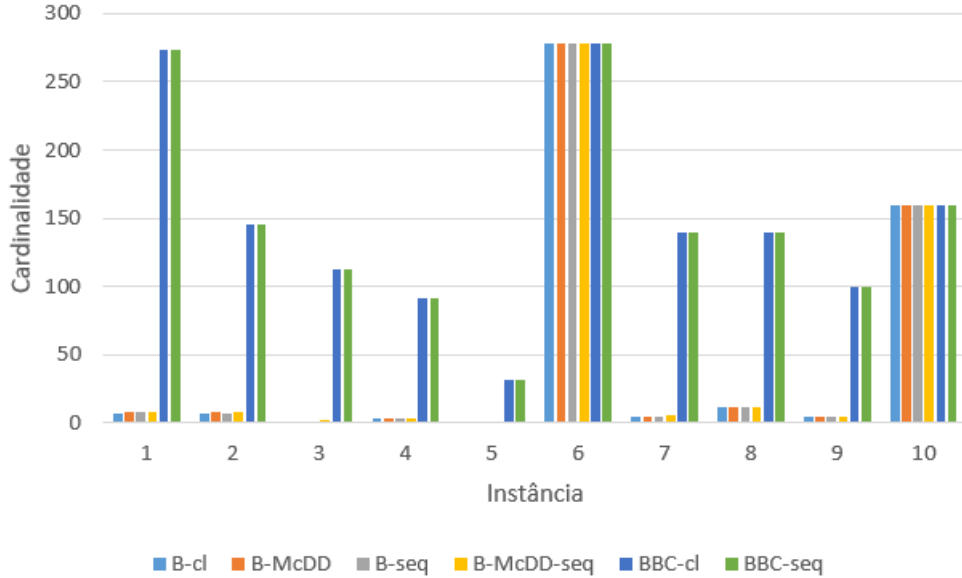


Figura 42 – Gráfico de colunas da cardinalidade das AFP's para instâncias com 10 tarefas e 7 máquinas

Na Figura 43, apresenta-se os gráficos de caixa dos tempos de execução do problema mestre. É notório a semelhança com os gráficos da Figura 41, evidenciando que o problema mestre é o que mais exige esforço computacional. Considerando o método B_{seq}^{RL} , que apresentou melhores resultados comparado aos métodos B_{cl} , B_{seq} e B_{RL} , o problema mestre chega a participar em média de 78,09%, 96,04% e 99,76% do tempo total de execução das classes 0,2_0,6_5_3, 0,2_0,6_10_3 e 0,2_0,6_10_7, respectivamente. Claro que esse comportamento é esperado, pelo fato do problema mestre ser binário. A maior dificuldade é de o problema mestre ter que ser resolvido até a otimalidade para gerar um novo corte e atualizar o problema. Essa dinâmica faz com que, considerando 10 tarefas e 7 máquinas, em média seja utilizado 1,76 segundos para resolver o problema mestre em cada iteração. Comparando com o BBC_{seq} , que foi a variação com melhor desempenho para esse porte, a média é de 0,009 segundos, significativamente menor segundo o teste *T-student*, com nível de significância de 5%.

Outro ponto para ser discutido é a quantidade de vezes que o subproblema é resolvido, gráficos de caixa na Figura 44. Percebe-se que, excluindo o B_{cl} , as variações BBC_{cl} e BBC_{seq} resolvem significativamente mais subproblema do que as outras variações para todos os portes de instâncias. Essa quantidade de iterações mais elevada é causada justamente por ser necessário resolver um subproblema para toda solução factível encontrada, não somente para a solução ótima. Isso é algo que para instâncias de pequeno porte provoca uma queda no desempenho, porém, para instâncias um pouco maiores é o que auxilia na

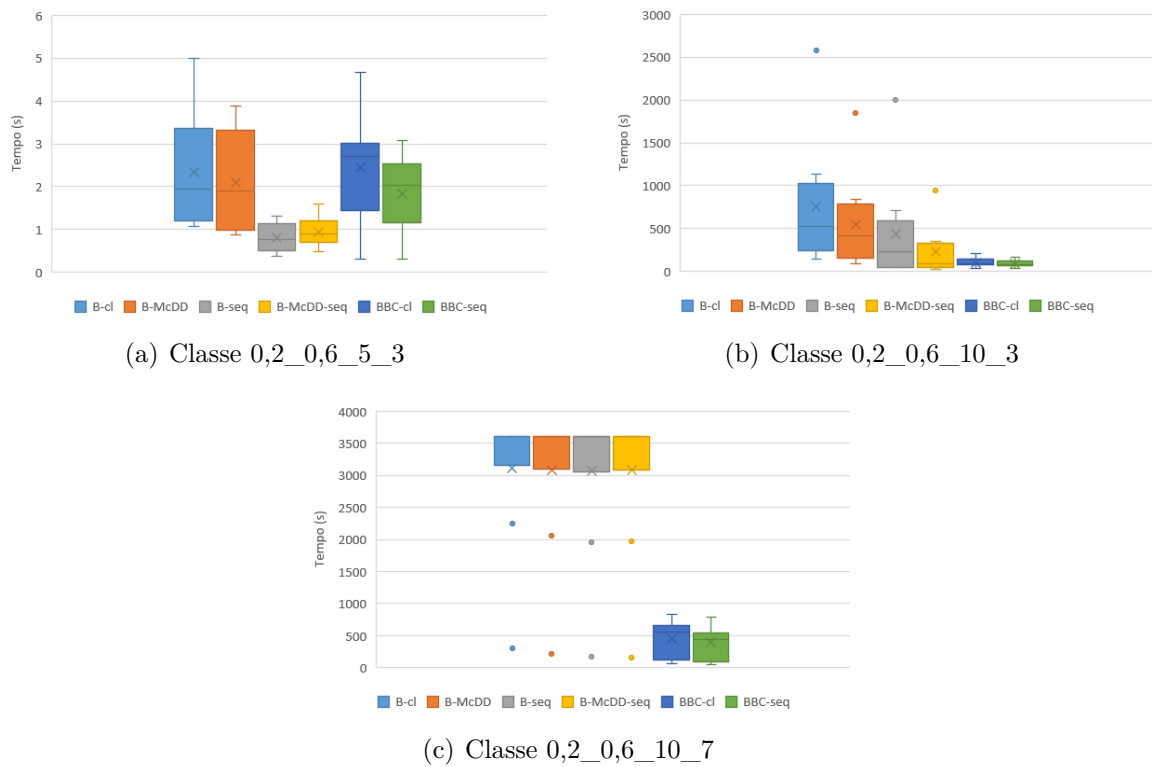


Figura 43 – Gráficos de caixas do tempo de execução do problema mestre para cada porte de instância

convergência mais rápida do método. Essa estrutura de árvore única é o que contorna a necessidade de provar a otimalidade em toda iteração, um processo custoso com o aumento do porte das instâncias.

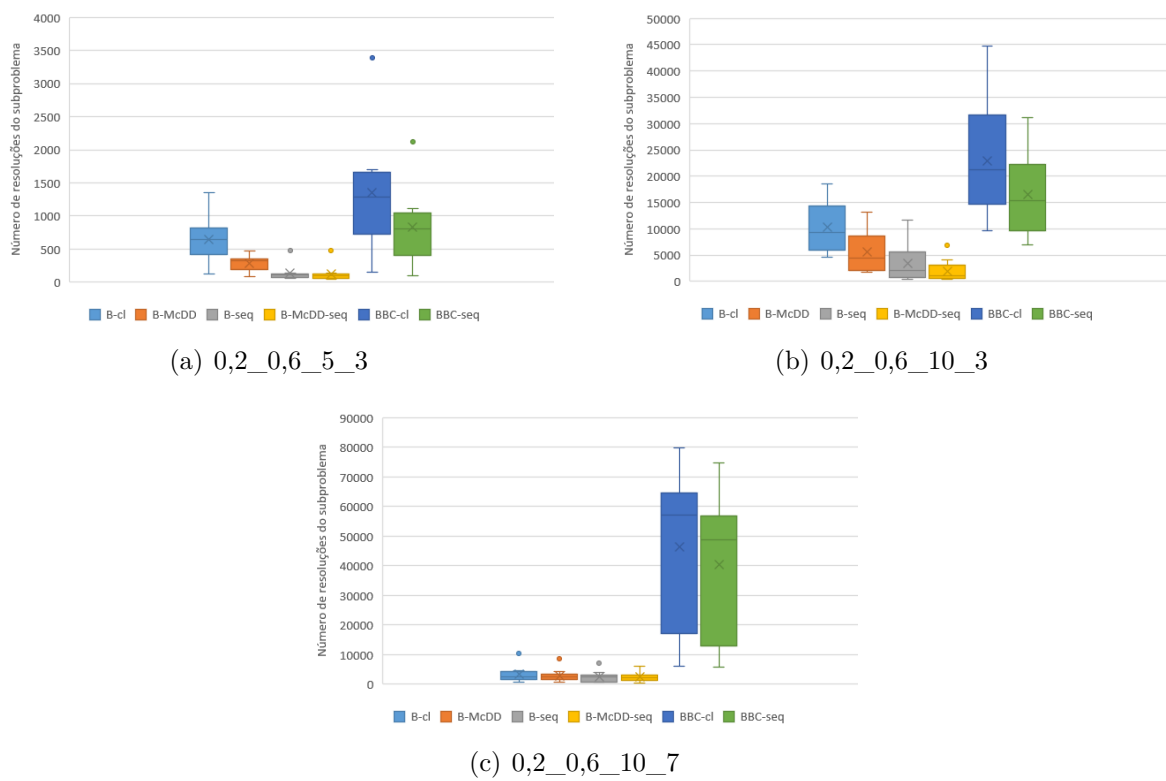


Figura 44 – Gráficos de caixas do número de subproblemas resolvidos por porte de instância

6 Considerações finais

Nesta tese foram apresentados os conceitos básicos da Otimização Matemática Multiobjetivo, bem como os principais métodos de solução: escalarização e *branch-and-bound*. Foi apresentado o problema de sequenciamento em *flow shop* permutacional com o foco nos objetivos de minimização do adiantamento total e do atraso total através de uma revisão da literatura, no qual foi identificado um tratamento monoobjetivo para um problema biobjetivo. Pela revisão, identificou-se que esse tratamento monoobjetivo é feito, por toda a literatura, através de uma simplificação do método da soma ponderada. Essa tese foi a primeira a chamar a atenção para a natureza biobjetiva do problema. Além disso, na revisão destacou-se o uso predominante de métodos heurísticos e metaheurísticos para resolver o problema mono e multiobjetivo, deixando uma lacuna pouco explorada que são os métodos exatos, o que contribuímos para preencher com os resultados desta tese.

Um modelo escalar geral foi proposto na Seção 2.1.1 e sua relaxação lagrangiana motivou a busca de métodos de resolução multiobjetivos alternativos aos de escalarização. Além disso, um modelo de otimização linear biobjetivo foi proposto e utilizado no estudo computacional do Capítulo 5. Resultados da Seção 5.1 mostraram que o sistema BIOBAB é um bom ponto de partida por disponibilizar métodos de escalarização, que se destacaram para instâncias com até 20 tarefas, e um método *branch-and-bound* que é o destaque para as instâncias com 50 e 100 tarefas.

Para explorar melhor a natureza biobjetiva do problema, foram desenvolvidas ferramentas matemáticas para analisar pontos na fronteira de Pareto. Resultados da Seção 5.2 demonstraram que o tratamento monoobjetivo feito pela literatura pode obter soluções indesejadas para o sistema de produção e que, através da abordagem biobjetivo, é possível alterar a programação das tarefas sem mudar o valor total da solução em quase 50% das 156 instâncias resolvidas. Esses resultados também evidenciaram a necessidade de utilizar métodos biobjetivos que determinam toda a fronteira de Pareto, pois se notou uma presença significativa de pontos não suportados na fronteira de Pareto, mostrando um aumento desses pontos conforme o porte das instâncias cresce, chegando a máximos de 77,78% com 10 tarefas e 10 máquinas.

No estudo dos métodos multiobjetivos GRB_{mult} , BIOBAB, caixa balanceada e ε -restrito aumentado feito na Seção 5.3, os resultados mostraram que BIOBAB foi o método intermediário obtendo pontos bem distribuídos na AFP, mas devido a uma cardinalidade significativamente menor, determinou uma AFP significativamente menos representativa em relação a caixa balanceada e ao ε -restrito aumentado. Além disso, o aumento do porte do problema impacta no aumento do tempo de execução do BIOBAB. Destacou-se nesse estudo o ε -restrito aumentado como o método que obteve AFPs mais representativas com

um bom controle de tempo de execução, resultando em uma média máxima de 135,81 segundos. Analisou-se também o *trade-off* entre manter a AFP representativa e diminuir o número de pontos como forma de auxiliar o tomador de decisão, utilizando o método ε -restrito aumentado aliado à ϵ -dominância.

Cinco variações do método de decomposição de Benders clássico foram propostas na Seção 4.2. Resultados da Seção 5.4 demonstraram que as três variações que mantiveram a estrutura clássica do método de Benders obtiveram pouca melhora no desempenho computacional. Melhoras significativas foram registradas nos métodos BBC_{cl} e BBC_{seq} que fazem uma busca em árvore única, estrutura diferente do método de Benders clássico. Dessa forma, evidenciou-se que a resolução de cada problema mestre até a otimalidade é um fator que pode prejudicar significativamente o desempenho do método.

Ainda existem lacunas a serem exploradas em relação a utilização de métodos biobjetivos exatos para resolver instâncias com cem ou mais tarefas do problema JIT-FSP. O método de decomposição de Benders e as variações avaliadas ainda apresentam dificuldades teóricas para alcançar esse objetivo no JIT-FSP. Uma dessas dificuldades está em definir inequações válidas para representar o poliedro do problema, já que em cada iteração do ε -restrito aumentado a região factível se altera por conta do método e por conta das restrições de linearização dos objetivos. Outro potencial de pesquisa está relacionado à forma de implementar a decomposição de Benders. Nesta tese o método foi aplicado para resolver cada subproblema do método ε -restrito aumentado, como é feito na literatura, porém se pode investigar outras formas de aplicar Benders, como por exemplo, aplicar o método de decomposição de Benders antes de utilizar algum método multiobjetivo, ou seja, antes de transformar o problema multiobjetivo em uma sequência de subproblemas monoobjetivos.

Outras lacunas a serem exploradas em pesquisas futuras sobre o uso da decomposição de Benders para resolver instâncias com cem ou mais tarefas do JIT-FSP são:

- Utilizar métodos heurísticos como *warm-start*;
- Utilizar métodos heurísticos para aprimorar as soluções do problema mestre e gerar múltiplos cortes;
- Realizar um estudo comparativo entre diferentes decomposições de Benders e a decomposição disponível em resolvedores comerciais.

Outros temas de pesquisa que podem ser explorados considerando o JIT-FSP são:

- Estudo de diferentes modelagens para o JIT-FSP, como modelos de precedência, modelos com variáveis de tempos ociosos e de espera e modelos baseados no problema do caixeiro viajante/roteamento de veículos;

- Estudo do *trade-off* entre generalizar o modelo (3.28)-(3.36), ou seja, dificultar o modelo e melhorar os objetivos adiantamento total e atraso total, já que programações não permutacionais geram soluções melhores para o adiantamento total (*e.g.* [2]);
- Utilizar outros métodos multiobjetivos para resolver o problema, como o Tchebycheff aumentado (*e.g.* [93]), que reduzem o espaço critério (*e.g.* [94]) e baseados em busca em árvores (*e.g.* [95]);
- Variações do problema: com restrições de precedência entre as tarefas, com tempos de *setup* dependentes da sequência das tarefas, sem *buffers* entre as máquinas, com múltiplas máquinas em paralelo, etc.

Durante o desenvolvimento desta tese foi realizada uma apresentação oral intitulada “Ferramentas de comparação entre os problemas mono e multiobjetivo de sequenciamento *just-in-time flow shop* permutacional” na XXV Oficina Nacional de Problemas de Corte, Empacotamento, Planejamento e Programação da Produção e Correlatos sobre parte dos resultados obtidos na Seção 5.2. Também foi submetido um artigo para publicação intitulado “Multi-objective versus mono-objective just-in-time permutation flow shop: tools for analysis of different conflict scenarios” contemplando demonstrações teóricas sobre métodos multiobjetivos aplicados ao JIT-FSP e resultados da Seção 5.2. Também está em produção um artigo com os resultados obtidos no Estudo 4 sobre o método de decomposição de Benders (Seção 5.4).

Referências

- 1 HOU, Y.; WANG, H.; FU, Y.; GAO, K.; ZHANG, H. Multi-objective brain storm optimization for integrated scheduling of distributed flow shop and distribution with maximal processing quality and minimal total weighted earliness and tardiness. *Computers & Industrial Engineering*, Elsevier, v. 179, p. 109217, 2023.
- 2 TOKTAŞ, B.; AZIZOĞLU, M.; KÖKSALAN, S. K. Two-machine flow shop scheduling with two criteria: Maximum earliness and makespan. *European Journal of Operational Research*, Elsevier, v. 157, n. 2, p. 286–295, 2004.
- 3 SUN, Y.; ZHANG, C.; GAO, L.; WANG, X. Multi-objective optimization algorithms for flow shop scheduling problem: a review and prospects. *The International Journal of Advanced Manufacturing Technology*, Springer, v. 55, p. 723–739, 2011.
- 4 MURATA, T.; ISHIBUCHI, H.; TANAKA, H. Multi-objective genetic algorithm and its applications to flowshop scheduling. *Computers & industrial engineering*, Elsevier, v. 30, n. 4, p. 957–968, 1996.
- 5 PARRAGH, S. N.; TRICOIRE, F. Branch-and-bound for bi-objective integer programming. *INFORMS Journal on Computing*, INFORMS, v. 31, n. 4, p. 805–822, 2019.
- 6 GRECO, S.; FIGUEIRA, J.; EHRGOTT, M. *Multiple criteria decision analysis*. [S.l.]: Springer, 2016. v. 37.
- 7 EHRGOTT, M.; FIGUEIRA, J.; GRECO, S. *Multiple criteria decision analysis: state of the art surveys*. [S.l.]: Springer, 2005.
- 8 GAL, T.; STEWART, T.; HANNE, T. *Multicriteria decision making: advances in MCDM models, algorithms, theory, and applications*. [S.l.]: Springer Science & Business Media, 2013. v. 21.
- 9 TENHUISEN, M. L.; WIECEK, M. M. Vector optimization and generalized lagrangian duality. *Annals of Operations Research*, Springer, v. 51, p. 15–32, 1994.
- 10 WIERZBICKI, A. P. On the completeness and constructiveness of parametric characterizations to vector optimization problems. *Operations-Research-Spektrum*, Springer, v. 8, n. 2, p. 73–87, 1986.
- 11 PARETO, V. *Manual d'économie politique (in French)*. [S.l.], 1896.
- 12 EHRGOTT, M. *Multicriteria optimization*. [S.l.]: Springer Science & Business Media, 2005. v. 491.
- 13 BANDYOPADHYAY, S.; MAULIK, U.; CHAKRABORTY, R. Incorporating epsilon-dominance in amosa: Application to multiobjective 0/1 knapsack problem and clustering gene expression data. *Applied Soft Computing*, Elsevier, v. 13, n. 5, p. 2405–2411, 2013.

- 14 WANG, J.; ZHONG, C.; ZHOU, Y.; ZHOU, Y. Multiobjective optimization algorithm with objective-wise learning for continuous multiobjective problems. *Journal of Ambient Intelligence and Humanized Computing*, Springer, v. 6, n. 5, p. 571–585, 2015.
- 15 LAUMANN, M.; THIELE, L.; DEB, K.; ZITZLER, E. Combining convergence and diversity in evolutionary multiobjective optimization. *Evolutionary computation*, MIT Press, v. 10, n. 3, p. 263–282, 2002.
- 16 BOLAND, N.; CHARKHGARD, H.; SAVELSBERGH, M. A criterion space search algorithm for biobjective integer programming: The balanced box method. *INFORMS Journal on Computing*, INFORMS, v. 27, n. 4, p. 735–754, 2015.
- 17 HAIMES, Y. Y.; LASDON, L. S.; WISMER, D. A. On a bicriterion formulation of the problems of integrated system identification and system optimization. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-1, n. 3, p. 296–297, 1971.
- 18 BORGES, J. C. *Um estudo sobre métodos de solução para o problema de corte de estoque biobjetivo*. Tese (Doutorado) — Universidade Estadual Paulista (Unesp), 2021.
- 19 SANTIS, M. D.; GRANI, G.; PALAGI, L. Branching with hyperplanes in the criterion space: The frontier partitioner algorithm for biobjective integer programming. *European Journal of Operational Research*, Elsevier, v. 283, n. 1, p. 57–69, 2020.
- 20 BENSON, H.; SUN, E. A weight set decomposition algorithm for finding all efficient extreme points in the outcome set of a multiple objective linear program. *European Journal of Operational Research*, v. 139, p. 26–41, 02 2002.
- 21 MITTEN, L. Branch-and-bound methods: General formulation and properties. *Operations Research*, INFORMS, v. 18, n. 1, p. 24–34, 1970.
- 22 ARENALES, M.; ARMENTANO, V.; MORABITO, R.; YANASSE, H. *Pesquisa Operacional para cursos de Engenharia*. [S.l.]: GEN LTC, 2015. v. 2.
- 23 MORRISON, D. R.; JACOBSON, S. H.; SAUPPE, J. J.; SEWELL, E. C. Branch-and-bound algorithms: A survey of recent advances in searching, branching, and pruning. *Discrete Optimization*, Elsevier, v. 19, p. 79–102, 2016.
- 24 EHRGOTT, M.; GANDIBLEUX, X. Bound sets for biobjective combinatorial optimization problems. *Computers & Operations Research*, v. 34, n. 9, p. 2674–2694, 2007.
- 25 RIQUELME, N.; LÜCKEN, C. V.; BARAN, B. Performance metrics in multi-objective optimization. In: IEEE. *2015 Latin American computing conference (CLEI)*. [S.l.], 2015. p. 1–11.
- 26 AUDET, C.; BIGEON, J.; CARTIER, D.; DIGABEL, S. L.; SALOMON, L. Performance indicators in multiobjective optimization. *European journal of operational research*, Elsevier, v. 292, n. 2, p. 397–422, 2021.
- 27 ZITZLER, E.; THIELE, L. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE transactions on Evolutionary Computation*, IEEE, v. 3, n. 4, p. 257–271, 1999.
- 28 PINEDO, M. L. *Scheduling: Theory, Algorithms, and Systems*. [S.l.]: Springer, 2012. v. 29.

- 29 NEUFELD, J. S.; SCHULZ, S.; BUSCHER, U. A systematic review of multi-objective hybrid flow shop scheduling. *European Journal of Operational Research*, v. 309, n. 1, p. 1–23, 2023.
- 30 de ABREU, A. P.; FUCHIGAMI, H. Y. An efficiency and robustness analysis of warm-start mathematical models for idle and waiting times optimization in the flow shop. *Computers & Industrial Engineering*, v. 166, p. 107976, 2022.
- 31 TYAGI, N.; VARSHNEY, R.; CHANDRAMOULI, A. Six decades of flowshop scheduling research. *International Journal of Scientific & Engineering Research*, v. 4, n. 9, p. 854–864, 2013.
- 32 POTTS, C. N.; STRUSEVICH, V. A. Fifty years of scheduling: a survey of milestones. *Journal of the Operational Research Society*, Springer, v. 60, p. S41–S68, 2009.
- 33 GUPTA, J. N.; JR, E. F. S. Flowshop scheduling research after five decades. *European Journal of Operational Research*, Elsevier, v. 169, n. 3, p. 699–711, 2006.
- 34 MINELLA, G.; RUIZ, R.; CIAVOTTA, M. A review and evaluation of multiobjective algorithms for the flowshop scheduling problem. *Inform's Journal on Computing*, v. 20, p. 451–471, 08 2008.
- 35 SINGH, H.; OBEROI, J. S.; SINGH, D. Multi-objective permutation and non-permutation flow shop scheduling problems with no-wait: a systematic literature review. *RAIRO-Operations Research*, EDP Sciences, v. 55, n. 1, p. 27–50, 2021.
- 36 FUCHIGAMI, H. Y.; SARKER, R.; RANGEL, S. Near-optimal heuristics for just-in-time jobs maximization in flow shop scheduling. *Algorithms*, MDPI, v. 11, n. 4, p. 43, 2018.
- 37 TAVARES, D. M. L.; BASTOS, L. d. S. L.; REIS, K. A. dos. Uma proposta de heurísticas e meta-heurísticas aplicadas ao problema de flow shop scheduling. *Brazilian Journal of Development*, v. 6, n. 6, p. 38266–38282, 2020.
- 38 PRATA, B. de A.; FUCHIGAMI, H. Y. A genetic iterated greedy algorithm for the blocking flowshop to minimize total earliness and tardiness. *Journal of Intelligent Manufacturing*, Springer, p. 1–14, 2023.
- 39 YAVUZ, M.; TUFEKCI, S. Dynamic programming solution to the batching problem in just-in-time flow-shops. *Computers & Industrial Engineering*, Elsevier, v. 51, n. 3, p. 416–432, 2006.
- 40 SHABTAY, D. The just-in-time scheduling problem in a flow-shop scheduling system. *European Journal of Operational Research*, Elsevier, v. 216, n. 3, p. 521–532, 2012.
- 41 GRAHAM, R. L.; LAWLER, E. L.; LENSTRA, J. K.; KAN, A. R. Optimization and approximation in deterministic sequencing and scheduling: a survey. In: *Annals of discrete mathematics*. [S.l.]: Elsevier, 1979. v. 5, p. 287–326.
- 42 BIRGIN, E. G.; FERREIRA, J. E.; RONCONI, D. P. A filtered beam search method for the m-machine permutation flowshop scheduling problem minimizing the earliness and tardiness penalties and the waiting time of the jobs. *Computers & Operations Research*, Elsevier, v. 114, p. 104824, 2020.

- 43 SCHALLER, J.; VALENTE, J. M. Minimizing total earliness and tardiness in a nowait flow shop. *International Journal of Production Economics*, Elsevier, v. 224, p. 107542, 2020.
- 44 GUEVARA-GUEVARA, A.; GÓMEZ-FUENTES, V.; POSOS-RODRÍGUEZ, L.; REMOLINA-GÓMEZ, N.; GONZÁLEZ-NEIRA, E. Earliness/tardiness minimization in a no-wait flow shop with sequence-dependent setup times. *Journal of Project Management*, v. 7, n. 3, p. 177–190, 2022.
- 45 TAVANA, M.; HAJIPOUR, V.; ALAGHEBANDHA, M.; CAPRIO, D. D. A bi-objective hybrid vibration damping optimization model for synchronous flow shop scheduling problems. *Machine Learning with Applications*, Elsevier, v. 11, p. 100445, 2023.
- 46 RONCONI, D. P.; BIRGIN, E. G. Mixed-integer programming models for flowshop scheduling problems minimizing the total earliness and tardiness. *Just-in-Time systems*, Springer, p. 91–105, 2012.
- 47 HAMDI, I.; LOUKIL, T. The permutation flowshop scheduling problem with exact time lags to minimise the total earliness and tardiness. *International Journal of Operational Research*, Inderscience Publishers (IEL), v. 28, n. 1, p. 70–86, 2017.
- 48 M'HALLAH, R. Minimizing total earliness and tardiness on a permutation flow shop using vns and mip. *Computers & Industrial Engineering*, Elsevier, v. 75, p. 142–156, 2014.
- 49 CHANDRA, P.; MEHTA, P.; TIRUPATI, D. Permutation flow shop scheduling with earliness and tardiness penalties. *International Journal of Production Research*, Taylor & Francis, v. 47, n. 20, p. 5591–5610, 2009.
- 50 FERNANDEZ-VIAGAS, V.; DIOS, M.; FRAMINAN, J. M. Efficient constructive and composite heuristics for the permutation flowshop to minimise total earliness and tardiness. *Computers & Operations Research*, Elsevier, v. 75, p. 38–48, 2016.
- 51 SCHALLER, J.; VALENTE, J. M. Heuristics for scheduling jobs in a permutation flow shop to minimize total earliness and tardiness with unforced idle time allowed. *Expert Systems with Applications*, Elsevier, v. 119, p. 376–386, 2019.
- 52 SCHALLER, J.; VALENTE, J. M. A comparison of metaheuristic procedures to schedule jobs in a permutation flow shop to minimise total earliness and tardiness. *International Journal of Production Research*, Taylor & Francis, v. 51, n. 3, p. 772–779, 2013.
- 53 PRATA, B. d. A.; FUCHIGAMI, H. Y. A genetic iterated greedy algorithm for the blocking flowshop to minimize total earliness and tardiness. *Journal of Intelligent Manufacturing*, Springer, p. 1–14, 2023.
- 54 KARACAN, I.; SENVAR, O.; BULKAN, S. A novel parallel simulated annealing methodology to solve the no-wait flow shop scheduling problem with earliness and tardiness objectives. *Processes*, MDPI, v. 11, n. 2, p. 454, 2023.
- 55 BEHNAMIAN, J. A parallel competitive colonial algorithm for jit flowshop scheduling. *Journal of Computational Science*, Elsevier, v. 5, n. 5, p. 777–783, 2014.

- 56 MADHUSHINI, N.; RAJENDRAN, C. Branch-and-bound algorithms for scheduling in an m-machine permutation flowshop with a single objective and with multiple objectives. *European Journal of Industrial Engineering*, Inderscience Publishers Ltd, v. 5, n. 4, p. 361–387, 2011.
- 57 GENG, X.-N.; SUN, X.; WANG, J.; PAN, L. Scheduling on proportionate flow shop with job rejection and common due date assignment. *Computers & Industrial Engineering*, Elsevier, v. 181, p. 109317, 2023.
- 58 RAMEZANIAN, R.; ARIANEZHAD, M.; HEYDARI, M. A mathematical programming model for flow shop scheduling problems for considering just in time production. *International Journal of Industrial Engineering and Production Research*, 2010.
- 59 MOHAMMADI, G. Multi-objective flow shop production scheduling via robust genetic algorithms optimization technique. *International Journal of Service Science, Management and Engineering*, v. 2, n. 1, p. 1–8, 2015.
- 60 REZAEIAN, J.; HOSSEINI-KIA, S.; MAHDAVI, I. The preemptive just-in-time scheduling problem in a flow shop scheduling system. *Journal of Optimization in Industrial Engineering*, QIAU, v. 12, n. 2, p. 79–92, 2019.
- 61 BRANDA, A.; CASTELLANO, D.; GUIZZI, G.; POPOLO, V. Metaheuristics for the flow shop scheduling problem with maintenance activities integrated. *Computers & Industrial Engineering*, Elsevier, v. 151, p. 106989, 2021.
- 62 AKHSHABI, M.; AKHSHABI, M.; KHALATBARI, J. Bi-criteria flow shop scheduling with fuzzy simulated annealing algorithm. *African Journal of Business Management*, Academic Journals, v. 6, n. 25, p. 7478, 2012.
- 63 MISSAOUI, A.; BOUJELBENE, Y. An improved iterated local search for solving the total weighted earliness tardiness blocking flowshop scheduling problem. In: IEEE. *2022 International Conference on Data Analytics for Business and Industry (ICDABI)*. [S.l.], 2022. p. 717–722.
- 64 FUCHIGAMI, H. Y.; PRATA, B. d. A. Coronavirus optimization algorithms for minimizing earliness, tardiness, and anticipation of due dates in permutation flow shop scheduling. *Arabian Journal for Science and Engineering*, Springer, p. 1–33, 2023.
- 65 ARABAMERI, S.; SALMASI, N. Minimization of weighted earliness and tardiness for no-wait sequence-dependent setup times flowshop scheduling problem. *Computers & Industrial Engineering*, Elsevier, v. 64, n. 4, p. 902–916, 2013.
- 66 KESHAVARZ, T.; SALMASI, N.; VARMAZYAR, M. Flowshop sequence-dependent group scheduling with minimisation of weighted earliness and tardiness. *European Journal of Industrial Engineering*, Inderscience Publishers (IEL), v. 13, n. 1, p. 54–80, 2019.
- 67 WILSON, J. M. Alternative formulations of a flow-shop scheduling problem. *The Journal of the Operational Research Society*, Palgrave Macmillan Journals, v. 40, n. 4, p. 395–399, 1989.

- 68 BEKTAŞ, T.; HAMZADAYI, A.; RUIZ, R. Benders decomposition for the mixed no-idle permutation flowshop scheduling problem. *Journal of Scheduling*, Springer, v. 23, p. 513–523, 2020.
- 69 WOLSEY, L. A. *Integer programming*. [S.l.]: John Wiley & Sons, 1998.
- 70 MACAMBIRA, A. F. U.; SIMONETTI, L.; RODRIGUES, R. d. F.; MACULAN, N. et al. Tópicos em otimização inteira. *UFRJ*, 2022.
- 71 GENDRON, B.; SCUTELLÀ, M. G.; GARROPPO, R. G.; NENCIONI, G.; TAVANTI, L. A branch-and-benders-cut method for nonlinear power design in green wireless local area networks. *European Journal of Operational Research*, Elsevier, v. 255, n. 1, p. 151–162, 2016.
- 72 MORENO, A.; MUNARI, P.; ALEM, D. A branch-and-benders-cut algorithm for the crew scheduling and routing problem in road restoration. *European Journal of Operational Research*, Elsevier, v. 275, n. 1, p. 16–34, 2019.
- 73 PARRAGH, S. N.; TRICOIRE, F.; GUTJAHR, W. J. A branch-and-benders-cut algorithm for a bi-objective stochastic facility location problem. *Or Spectrum*, Springer, p. 1–41, 2021.
- 74 ISERMANN, H. On some relations between a dual pair of multiple objective linear programs. *Zeitschrift für Operations Research*, Springer, v. 22, p. 33–41, 1978.
- 75 HU, X.; WANG, L. Lagrangian duality for multiobjective programming problems in lexicographic order. In: HINDAWI. *Abstract and Applied Analysis*. [S.l.], 2013. v. 2013.
- 76 MARDAN, E.; GOVINDAN, K.; MINA, H.; GHOLAMI-ZANJANI, S. M. An accelerated benders decomposition algorithm for a bi-objective green closed loop supply chain network design problem. *Journal of Cleaner Production*, Elsevier, v. 235, p. 1499–1514, 2019.
- 77 WU, P.; WANG, Y.; CHU, C. Logic-based benders decomposition for bi-objective parallel machine selection and job scheduling with release dates and resource consumption. *Computers & Operations Research*, Elsevier, v. 164, p. 106528, 2024.
- 78 BENDERS, J. Partitioning procedures for solving mixed-variables programming problems. *Numerische mathematik*, v. 4, n. 1, p. 238–252, 1962.
- 79 RAHMANIANI, R.; CRAINIC, T. G.; GENDREAU, M.; REI, W. The benders decomposition algorithm: A literature review. *European Journal of Operational Research*, Elsevier, v. 259, n. 3, p. 801–817, 2017.
- 80 SENNA, F.; COELHO, L. C.; MORABITO, R.; MUNARI, P. An exact method for a last-mile delivery routing problem with multiple deliverymen. *European Journal of Operational Research*, Elsevier, v. 317, n. 2, p. 550–562, 2024.
- 81 MCDANIEL, D.; DEVINE, M. A modified benders' partitioning algorithm for mixed integer programming. *Management Science*, INFORMS, v. 24, n. 3, p. 312–319, 1977.
- 82 HO, M. H.; HNAIEN, F.; DUGARDIN, F. Exact method to optimize the total electricity cost in two-machine permutation flow shop scheduling problem under time-of-use tariff. *Computers & Operations Research*, Elsevier, v. 144, p. 105788, 2022.

- 83 HAMZADAYI, A. An effective benders decomposition algorithm for solving the distributed permutation flowshop scheduling problem. *Computers & Operations Research*, Elsevier, v. 123, p. 105006, 2020.
- 84 ZHANG, Z.; SONG, X.; HUANG, H.; ZHOU, X.; YIN, Y. Logic-based benders decomposition method for the seru scheduling problem with sequence-dependent setup time and dejong's learning effect. *European Journal of Operational Research*, Elsevier, v. 297, n. 3, p. 866–877, 2022.
- 85 SUN, D.; TANG, L.; BALDACCI, R. A benders decomposition-based framework for solving quay crane scheduling problems. *European Journal of Operational Research*, Elsevier, v. 273, n. 2, p. 504–515, 2019.
- 86 WALLRATH, R.; FRANKE, M. B. Integration of milp and discrete-event simulation for flow shop scheduling using benders cuts. *Computers & Chemical Engineering*, Elsevier, v. 189, p. 108809, 2024.
- 87 GHORBANZADEH, M.; DAVARI, M.; RANJBAR, M. Energy-aware flow shop scheduling with uncertain renewable energy. *Computers & Operations Research*, Elsevier, v. 170, p. 106741, 2024.
- 88 Gurobi Optimization, LLC. *Gurobi Optimizer Reference Manual*. 2023. Disponível em: <<https://www.gurobi.com>>.
- 89 FOUNDATION, P. S. *Python 3.8.12 documentation*. 2021. Disponível em: <https://www.python.org>. Acessado: 10 de julho de 2023.
- 90 TAILLARD, E. Benchmarks for basic scheduling problems. *European Journal of Operational Research*, v. 64, n. 2, p. 278–285, 1993.
- 91 BLASCO, X.; HERRERO, J. M.; SANCHIS, J.; MARTÍNEZ, M. A new graphical visualization of n-dimensional pareto front for decision-making in multiobjective optimization. *Information Sciences*, Elsevier, v. 178, n. 20, p. 3908–3924, 2008.
- 92 TUŠAR, T.; FILIPIČ, B. Visualization of pareto front approximations in evolutionary multiobjective optimization: A critical review and the prosecution method. *IEEE Transactions on Evolutionary Computation*, IEEE, v. 19, n. 2, p. 225–245, 2014.
- 93 FILHO, A. A.; MORABITO, R. An effective approach for bi-objective multi-period touristic itinerary planning. *Expert Systems with Applications*, Elsevier, v. 240, p. 122437, 2024.
- 94 LOKMAN, B.; KÖKSALAN, M. Finding all nondominated points of multi-objective integer programs. *Journal of Global Optimization*, Springer, v. 57, p. 347–365, 2013.
- 95 FREITAS, J. C. de; CANTANE, D. R.; ROCHA, H.; DIAS, J. A multiobjective beam angle optimization framework for intensity-modulated radiation therapy. *European Journal of Operational Research*, Elsevier, 2024.
- 96 SLYKE, R. M. V.; WETS, R. L-shaped linear programs with applications to optimal control and stochastic programming. *SIAM journal on applied mathematics*, SIAM, v. 17, n. 4, p. 638–663, 1969.

APÊNDICE A - Decomposição de Benders

O método de decomposição de Benders [78] é uma técnica que surgiu simultaneamente em duas comunidades. Em [96] o método de Benders foi chamado de *L-shaped*, exatamente pela estrutura em L das restrições obtida pelos modelos com recurso em dois estágios, que são utilizados na otimização estocástica. Em Benders [78] o autor propõe o método para diferentes problemas, e entre eles, problemas em que parte da matriz de restrições possui uma estrutura em blocos. O autor cita a decomposição Dantzig e Wolfe como sendo outro método que aproveita da estrutura dos problemas.

Considere o problema (1)-(4) em que D possui estrutura facilitadora.

$$\text{Min } c^T x + h^T y \quad (1)$$

$$\text{S.a : } Ax = b \quad (2)$$

$$Tx + Dy = d \quad (3)$$

$$x \in \mathbb{B}, y \geq 0 \quad (4)$$

Nesse problema, fixando as variáveis x , aproveita-se da boa estrutura da matriz D . Então a ideia é reescrever o problema (1)-(4) de forma que parte do problema retorne um valor de x e outra parte do problema irá informar a qualidade desse x . O problema resultante é exibido em (5)-(7), em que $\Phi(x) = \text{Min}_y \{h^T y \mid Dy = d - Tx, y \geq 0\}$.

$$\text{Min}_x c^T x + \Phi(x) \quad (5)$$

$$\text{S.a : } Ax = b \quad (6)$$

$$x \in \mathbb{B} \quad (7)$$

Analisando o problema (5)-(7) nem todo $\bar{x}$ que satisfaça $A\bar{x} = b$ irá resultar que o sistema $Dy = d - T\bar{x}$ seja factível. Ou seja, nem sempre é possível avaliar a função $\Phi(x)$. Para contornar isso, Benders utilizou do dual do problema da $\Phi(x)$ obtendo $\Phi(x) = \text{Max}_\lambda \{(d - Tx)^T \lambda \mid D^T \lambda \leq h\}$. Dessa forma $\Phi(x)$ ou tem solução ótima ou é ilimitado.

Para que o problema (5)-(7) seja equivalente ao problema (1)-(4) é preciso eliminar os raios extremos (Definição 19) de subida da $\Phi(x)$. Esse processo pode ser feito utilizando o Teorema da Representatividade e eliminando raios de subida. O resultado desse processo é apresentado em (8)-(11), onde $\mathcal{R}$ e $\mathcal{Q}$ são, respectivamente, os conjuntos de raios extremos e pontos extremos e a restrição (10) elimina direções de subida.

Definição 19. Considere o poliedro $P = \{x \in \mathbb{R}^n \mid Ax = b, x \geq 0\}$.

- O vetor $r \in P$ é chamado de raio quando satisfaz $x + r\varepsilon \in P$ para todo $x \in P$ e todo $\varepsilon \geq 0$;
- O vetor $r \in P$ é chamado de raio extremo quando não pode ser obtido pela combinação linear positiva de dois raios distintos.

$$\text{Min}_x c^T x + \text{Max}_{q \in \mathcal{Q}} \{(d - Tx)^T \bar{\lambda}_q\} \quad (8)$$

$$\text{S.a : } Ax = b \quad (9)$$

$$(d - Tx)^T \bar{\lambda}_r \leq 0, \forall r \in \mathcal{R} \quad (10)$$

$$x \in \mathbb{B} \quad (11)$$

O problema (8)-(11) é equivalente ao problema (1)-(4). Linearizando o objetivo (8) é dado o Problema Mestre de Benders (PMB) em (12)-(16). As restrições (14) e (15) são chamadas, respectivamente, de cortes de factibilidade e cortes de otimalidade. Como o conjunto $\mathcal{R}$ e $\mathcal{Q}$ costumam ser muito grandes o PMB é resolvido pelo método de planos de corte. Além disso, sendo o PMB um problema de otimização inteira, é necessário utilizar por exemplo o método *branch-and-bound*.

$$\text{Min}_x c^T x + v \quad (12)$$

$$\text{S.a : } Ax = b \quad (13)$$

$$(d - Tx)^T \bar{\lambda}_r \leq 0, \forall r \in \mathcal{R} \quad (14)$$

$$(d - Tx)^T \bar{\lambda}_q \leq v, \forall q \in \mathcal{Q} \quad (15)$$

$$x \in \mathbb{B} \quad (16)$$

O módulo *main* é o de execução do sistema. Nele é possível escolher diferentes configurações de execução, as principais são:

1. Método de solução:

- a) BIOBAB;
- b) ε -restrito;
- c) Caixa balanceada.

2. Busca da árvore:

- a) Profundidade;
- b) Largura;
- c) Melhor.

3. Tempo limite;

4. Relaxação linear.

Na configuração relaxação linear, quando escolhe-se não relaxar o domínio das variáveis, os métodos de escalarização funcionam igualmente discutido na Seção 2.1.1. Quando a escolha é de relaxar o domínio das variáveis, os métodos de escalarização são usados para resolver os subproblemas associados aos nós da árvore *branch-and-bound*.

Os módulos *basicdata*, *basiclp* e *integersolution* são utilizados para implementar um problema de otimização próprio. Para fazer essa implementação, é necessário criar dois módulos. O primeiro módulo será o *nomeinstance* que deverá conter uma classe que herde a classe *BasicData* que está no módulo *basicdata*. Essa será a classe responsável pelos dados de entrada e por definir limitantes iniciais para o problema. O segundo módulo será o *nomemodel* que deverá conter uma classe que herde a classe *BasicLP* que está no módulo *basiclp*. Essa será a classe responsável pelo modelo matemático de otimização resolvido. O módulo *integersolution* é utilizado para modificar a saída do sistema. Como o BIOBAB já possui um formato padrão, implementar uma classe que herde a classe *Integersolution* é opcional.