

JULIANO HENRIQUE SILVA

**RESOLUÇÃO SIMULTÂNEA DO PROBLEMA DE
LEIAUTE DE FACILIDADES E ROTEAMENTO
DE FLUXO POR MEIO DE UMA ABORDAGEM
EXATA**

Belo Horizonte
17 de maio de 2012

UNIVERSIDADE FEDERAL DE MINAS GERAIS
ESCOLA DE ENGENHARIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

**RESOLUÇÃO SIMULTÂNEA DO PROBLEMA DE
LEIAUTE DE FACILIDADES E ROTEAMENTO
DE FLUXO POR MEIO DE UMA ABORDAGEM
EXATA**

Dissertação apresentada ao Curso de Pós-Graduação em Engenharia de Produção da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Mestre em Engenharia de Produção.

JULIANO HENRIQUE SILVA

Belo Horizonte
17 de maio de 2012



UNIVERSIDADE FEDERAL DE MINAS GERAIS

FOLHA DE APROVAÇÃO

Resolução simultânea do problema de leiaute de facilidades e roteamento de fluxo por meio de uma abordagem exata

JULIANO HENRIQUE SILVA

Dissertação defendida e aprovada pela banca examinadora constituída por:

Ph. D. RICARDO SARAIVA DE CAMARGO – Orientador
Universidade Federal de Minas Gerais

Ph. D. GILBERTO DE MIRANDA JÚNIOR – Co-orientador
Universidade Federal de Minas Gerais

Belo Horizonte, 17 de maio de 2012

Resumo

A posição dos recursos transformadores de uma facilidade impacta diretamente no nível de serviço prestado. O Problema de Leiaute de Facilidades é um problema de otimização combinatória que consiste em determinar a posição dos recursos de forma a minimizar o custo de movimentação dos produtos na fábrica. Em ambientes voláteis, o Leiaute Distribuído tem se destacado por possuir características que auxiliam as empresas a enfrentar as flutuações presente nesses ambientes. Tradicionalmente, os modelos propostos pelos autores para resolver esse problema são modelos não lineares que, em especial, são derivados do modelo quadrático de atribuição (*Quadratic Assignment Problem - QAP*). O objetivo desse trabalho é propor um modelo linear que resolva o Problema de Leiaute Distribuído. Por possuir uma estrutura especial em sua formulação, o modelo foi resolvido segundo o método de decomposição de Benders acrescido dos procedimentos pré processamento, *core point* e *Benders-and-Cut*. Com base nos testes computacionais, observou-se porém, uma melhor performance do *software* CPLEX frente ao procedimento implementado.

Palavras-chaves: Problema de Leiaute de Facilidades; Problema de Roteamento de Fluxo; Leiaute Distribuído; Método de Decomposição de Benders.

Abstract

The placement of the processors resources in a facility directly impacts the level of service. The Facility Layout Problem is a combinatorial optimization problem which is concerned in determining the position of the resources to minimize the cost of moving goods within the factory. In volatile environments, the Distributed Layout has been highlighted because it posses features that help companies cope with the fluctuations present in these environments. Traditionally, the models proposed by the authors to solve this problem are nonlinear, in particular, are derived from the quadratic assignment model. The aim of this paper is to propose a linear model that solves the Distributed Layout Problem. By having a special structure in their formulation, the model was solved by the Benders decomposition method added the following procedures hot start, core-point and Benders-and-Cut. Based on the computational experiments, noted however, a better performance of the software CPLEX compared to the implemented procedure.

Keywords: Facility Layout Problem; Flow Assignment Problem; Distributed Layout; Benders Decomposition Method.

Sumário

1	Introdução	2
2	Exame da Literatura	6
2.1	Problema de leiaute de facilidades	6
2.1.1	Leiaute Distribuído	7
2.2	Formulações Matemáticas	9
2.2.1	Formulações Discretas	9
2.2.2	Formulações Contínuas	15
3	Problema de leiaute de facilidades e roteamento de fluxo	20
3.1	Definições e formulação	20
3.1.1	Exemplo Numérico	22
3.1.2	Formulação atribuição	24
3.2	Método de Decomposição de Benders	27
3.2.1	Problema Mestre e SubProblema	31
3.2.2	Algoritmo de Decomposição de Benders	35
3.3	Geração de cortes Pareto Ótimo	35
3.3.1	Subproblema Independente de Magnanti-Wong - SIMW	37
3.3.2	Algoritmo de Decomposição de Benders com Cortes Pareto-Ótimo	37
3.4	Pré processamento - <i>Hot Start</i>	38
3.5	<i>Benders-and-Cut</i>	39
4	Resultados Computacionais	40
4.1	Gerador de instância	40
4.2	Resultados	43
4.2.1	Modelo atribuição versus integrado	43
4.2.2	Relaxação linear	44
4.2.3	Evolução dos métodos	44
4.2.4	CPLEX versus Decomposição de Benders	48

5 Conclusão	51
Referências Bibliográficas	52

Lista de Figuras

1.1	Configurações clássicas de leiaute	4
2.1	Representação discreta e contínua do leiaute	7
2.2	Formação de célula virtual	8
3.1	Configuração do leiaute do exemplo numérico	24
4.1	Dimensões de leiaute	41
4.2	Proporção do tempo gasto pela variante BD-C para executar o PM e o SP	47
4.3	Proporção do tempo gasto pela variante BD-CP para executar o PM e o SP	48
4.4	Proporção do tempo gasto pela variante BD-CP-HS para executar o PM e o SP	49
4.5	Proporção do tempo gasto pela variante BD-CP-HS-BC para executar o PM e o SP	50

Lista de Tabelas

3.1	Parâmetro referente à máquina	23
3.2	Parâmetro referente ao produto	23
3.3	Matriz de distância	23
3.4	Roteamento do fluxo por produto	24
4.1	Resumo dos parâmetros utilizados	42
4.2	Resultados computacionais do modelo atribuição versus integrado	44
4.3	Relaxação linear da formulação proposta	45
4.4	Resultados computacionais do método Decomposição de Benders Clássico	45
4.5	Resultados computacionais do método Decomposição de Benders com o uso de corte de Pareto Ótimo	46
4.6	Resultados computacionais do método Decomposição de Benders com o uso de corte de Pareto Ótimo e Hot Start	46
4.7	Resultados computacionais do método Decomposição de Benders com o uso de corte de Pareto Ótimo, Hot Start e <i>Benders-and-Cut</i>	47
4.8	Resultados computacionais do <i>software</i> CPLEX versus variante BD-CP-HS-BC	49

Lista de Algoritmos

1	Algoritmo modelo desagregado versus integrado	27
2	Algoritmo de decomposição de Benders clássico	30
3	Algoritmo de decomposição de Benders Clássico	35
4	Algoritmo de decomposição de Benders com cortes Pareto Ótimo	37
5	Algoritmo de decomposição de Benders usando pré-processamento e cortes de Pareto Ótimo	39

Capítulo 1

Introdução

O ambiente empresarial está submetido a constantes mudanças, principalmente após a crescente globalização dos mercados. A diminuição do ciclo de vida juntamente com a tendência de demanda de médio volume dos produtos têm desafiado as empresas de manufatura a melhorar a eficiência e a sua capacidade de produção. A combinação de uma grande variedade de produtos com demanda irregular resulta em um ambiente volátil onde tanto as características produtivas quanto o volume de produção pode variar de forma expressiva entre períodos.

Em virtude dessa variação, a posição dos recursos ¹ transformadores em um determinado período pode não ser eficiente para o próximo. Uma maneira de enfrentar essa situação é por meio da terceirização da produção que pode ser de forma parcial ou até mesmo integral de certos produtos. Porém, o uso dessa prática pode se tornar inviável ao realizar uma análise de alternativa do custo/benefício ou em virtude de outros fatores.

Manter o nível de estoque mais elevado é uma outra alternativa estratégica que pode auxiliar as empresas a enfrentar esse tipo de flutuação. Entretanto, para certos ramos de atividade, essa opção pode não ser aplicável em virtude de um conjunto de restrições que inviabiliza essa prática.

Partindo do pressuposto que essas opções não serão acolhidas pela empresa, esta fica com o desafio de ser capaz de produzir os produtos em um tempo hábil, com preço competitivo e além de garantir uma boa qualidade.

Nessa situação, posicionar os recursos necessários na transformação dos produtos de forma estratégica no chão de fábrica é uma etapa crucial que irá impactar diretamente no nível de serviço prestado. As empresas deverão adotar um arranjo físico (leiaute) que seja facilmente reconfigurável entre períodos ou robusto o suficiente tal que seja capaz de oferecer uma performance aceitável para todos os períodos adotando um único

¹Durante o texto, os termos recurso, máquina e departamento são usados como sinônimos, sem erro de interpretação.

leiaute. Os tradicionais arranjos físicos funcional, por produto e celular não são capazes de atender a essas exigências.

O leiaute funcional é projetado para obter uma máxima flexibilidade sendo capaz de processar uma grande variedade de produtos em pequenos lotes. Esses produtos normalmente possuem uma sequência tecnológica distinta e uma alta variabilidade no tempo de processamento de suas operações. Os recursos são agrupados de acordo com a sua natureza de processamento como ilustrado na Figura 1.1a, formando setores funcionais. Em razão dessa forma de distribuição, as máquinas de natureza distinta estão distantes entre si. Para tornar o processo de fabricação mais econômico, a movimentação das peças se dá em lotes (*batches*), que em contrapartida resulta em um alto nível de produtos em processo e em um alto tempo de fila.

Ao contrário do leiaute funcional, o leiaute por produto possui pouca flexibilidade pois os recursos são organizados em linha de forma a otimizar o fluxo e são destinados à fabricação de um produto ou de uma família de produtos que possuem características de produção similares conforme observado na Figura 1.1b. Dessa forma é possível produzir grandes volumes e obter uma alta taxa de produtividade. A falta de flexibilidade na produção de novos produtos é uma grande limitação desse leiaute.

A Figura 1.1c ilustra o leiaute celular que, para sua concepção utiliza os princípios da Tecnologia de Grupo (TG) introduzidas por Mitrofanov (1966). A TG é uma filosofia de gestão que identifica produtos que compartilham recursos em comum e os agrupam em famílias. Os recursos de natureza distinta são fisicamente agrupados de forma sistemática, dando origem às células de manufatura. Cada célula projetada é então responsável pela fabricação de uma família de produtos. Pela dedicação e pela proximidade das máquinas dentro das células, esse leiaute oferece um fluxo eficiente e uma alta taxa de produtividade em ambientes onde os produtos são produzidos em grandes lotes possuindo pouca variabilidade. Caso contrário, o reposicionamento das máquinas dentro das células será frequente ou o fluxo de peças entre células tornará significativo (Irani et al., 1993; Irani e Huang, 2006).

Para atender a essas novas exigências de mercado, há um consenso da necessidade de uma nova geração de leiaute que deve possuir características que auxiliam as empresas de manufatura a produzir seus produtos de forma eficiente.

O leiaute distribuído é um tipo de leiaute que pertence a essa nova geração e vem se destacando por ser capaz de oferecer uma performance aceitável para todos os períodos em ambientes altamente voláteis (Benjaafar et al., 2002). Os recursos de natureza distinta são estrategicamente posicionados ² em diferentes regiões da fábrica e dessa forma a distância relativa entre os mesmos é reduzida.

²Durante o texto, os termos posicionar, alocar, atribuir e instalar são usados como sinônimos, sem erro de interpretação.

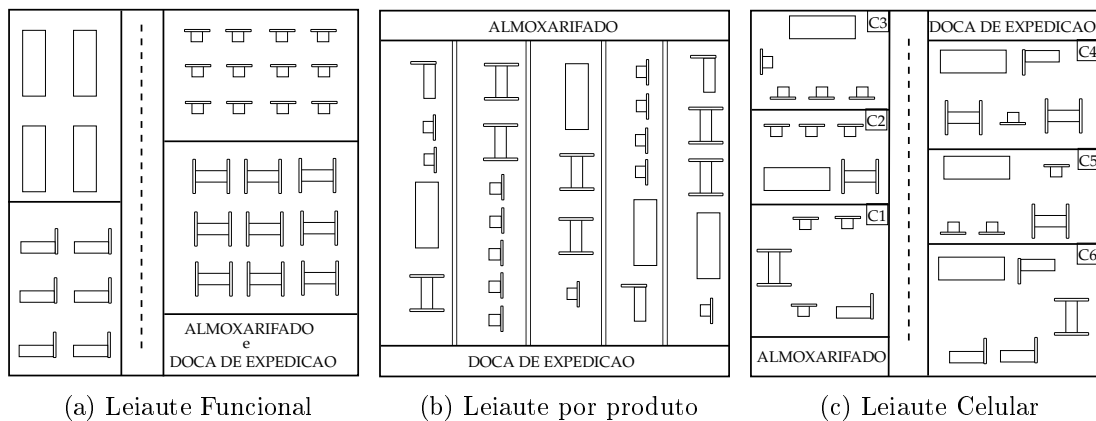


Figura 1.1: Configurações clássicas de leiaute

Com esse tipo de configuração de leiaute, a empresa estará mais apta a reagir contra a flutuação nos padrões de fluxo e no volume de produção dos produtos, pois aumenta a possibilidade de encontrar um caminho eficiente no sistema sem a necessidade de um rearranjo dos recursos. Esse caminho pode ser encontrado mediante o uso do conceito de célula virtual proposto por [McLean et al. \(1982\)](#). A célula virtual consiste em um agrupamento lógico de recursos que será temporariamente dedicado na produção de um produto. Uma vez concluída a fabricação, os recursos estarão disponíveis para realizar a produção de um próximo produto.

Projetar um leiaute distribuído consiste em resolver de forma simultânea o problema de leiaute de facilidades e o problema de roteamento de fluxo ([Taghavi e Murat, 2011](#)). O problema de leiaute de facilidades é um problema de otimização combinatória que determina a posição dos recursos no chão de fábrica de forma a minimizar o custo de movimentação dos produtos. Tradicionalmente, os autores têm formulado esse problema utilizando o modelo quadrático de atribuição (*Quadratic Assignment Problem - QAP*) proposto por [Koopmans e Beckman \(1957\)](#).

O objetivo principal desse trabalho é apresentar um modelo linear de programação inteira mista capaz de projetar um leiaute distribuído. Essa formulação foi apresentada no 42o Simpósio de Pesquisa Operacional no ano de 2010, ([Silva et al., 2010](#)), como contribuição parcial da pesquisa.

Por possuir uma estrutura especial na sua formulação, o modelo foi resolvido por meio de uma abordagem exata utilizando o método de decomposição proposto por [Benders \(1962\)](#) acrescido de outros procedimentos que auxiliam na sua convergência.

O presente trabalho está organizado da seguinte forma. No Capítulo 2 é apresentado a classificação do problema de leiaute de facilidade e suas formas de abordagem. Em seguida, é descrito as principais formulações propostas para resolver esse problema.

No Capítulo 3, é feita uma descrição do modelo proposto e suas premissas con-

sideradas. O desenvolvimento do método de decomposição de Benders aplicado ao problema e o algoritmo proposto são apresentados na Seção 3.2. Na próxima Seção 3.3 desse capítulo são descritos os procedimentos que foram incorporados ao método de Decomposição de Benders clássico.

No Capítulo 4, são descritos os parâmetros utilizados e é apresentado o resultado dos testes computacionais. A conclusão e sugestões para futuros trabalhos são apresentados no Capítulo 5.

Capítulo 2

Exame da Literatura

Nesse capítulo é apresentado a classificação do problema de leiaute de facilidades e suas formas de abordagem. Em seguida é feito um exame da literatura referente ao leiaute distribuído. As principais formulações matemáticas propostas para resolver o problema de leiaute de facilidades são descritas na seção 2.2.1. O método de decomposição de Benders é apresentado na seção 3.2.

2.1 Problema de leiaute de facilidades

Uma facilidade é um espaço físico que possui um conjunto de recursos capaz de oferecer/processar determinado serviço/produto. Como exemplo, pode-se citar escolas, hospitais, aeroportos, indústrias, entre outros. A localização desses recursos dentro da facilidade impacta diretamente no nível de serviço prestado (Tompkins et al., 1996).

O problema de leiaute de facilidades é um problema de otimização combinatória que consiste em posicionar tais recursos de forma a minimizar ou maximizar determinado fator. Em um contexto industrial, que é o foco desse trabalho, o custo de movimentação dos produtos entre os recursos é o fator mais utilizado (Fu e Kaku, 1997; Alan et al., 2010).

A representação do espaço disponível de uma facilidade em uma formulação se dá de forma discreta conforme ilustrado na Figura 2.1a ou contínua, Figura 2.1b. Na formulação discreta, as máquinas são instaladas em posições/sites pré-definidos tal que cada posição comporta somente uma máquina e a distância relativa entre as posições é calculada segundo a métrica que melhor representa o percurso realizado entre as mesmas. Já na formulação contínua, a área a ser ocupada pelo recurso é conhecida e o posicionamento é realizado respeitando uma distância mínima entre os recursos e a área total disponível.

Em função da variação das necessidades produtivas no decorrer de um horizonte

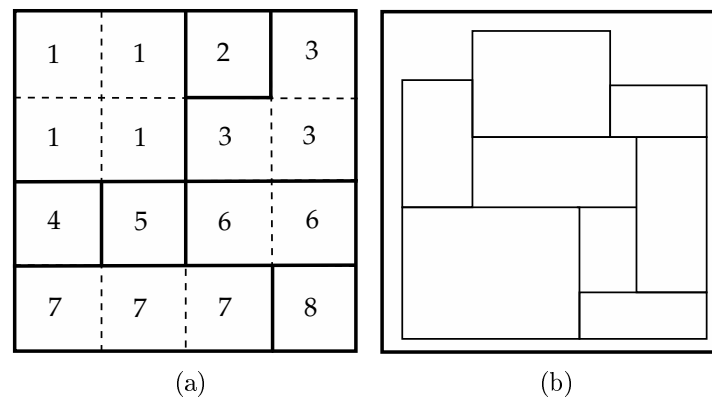


Figura 2.1: Representação discreta e contínua do leiaute

de planejamento, o problema de leiaute de facilidades é categorizado em estático ou dinâmico (Amine et al., 2007). Quando as informações sobre o processo e a demanda sofrerão pouca variação ao longo do tempo, é denominado como problema de Leiaute de Facilidades Estático (*Static Facility Layout Problem - SFLP*).

O problema se torna dinâmico quando há uma variação significativa nos requisitos de produção tais que interfere diretamente no fluxo de fabricação dos produtos. Além de uma variação expressiva da demanda, produtos com baixo ciclo de vida ou mesmo o reposicionamento de um recurso são fatores que interferem na trajetória original dos produtos (Afentakis et al., 1990; Amine et al., 2007)

O problema de Leiaute de Facilidades Dinâmico (*Dynamic Facility Layout Problem - DFLP*) pode ser abordado de duas formas distintas. A abordagem flexível ou reconfigurável sugere um novo leiaute a cada período de forma a garantir um fluxo eficiente, minimizando o custo de reposicionamento dos recursos entre períodos consecutivos. A abordagem robusta irá adotar um único leiaute que deverá oferecer um desempenho satisfatório para todos os períodos.

Entre os tipos de leiaute que suportam uma abordagem robusta, o leiaute distribuído vem se destacando por oferecer um bom desempenho em ambientes altamente voláteis (Bayakasoglu e Göçken, 2010). Na próxima seção é apresentado o seu conceito e as suas características.

2.1.1 Leiaute Distribuído

O Leiaute Distribuído desagrega um departamento funcional em departamentos menores e os distribui estrategicamente no chão de fábrica considerando o fluxo entre departamentos de forma a minimizar a distância total percorrida dos produtos. A distribuição dos recursos de mesma natureza em diferentes regiões da fábrica aumenta a acessibili-

dade e conseqüentemente reduz a distância entre departamentos distintos.

O conceito de leiaute distribuído foi primeiramente proposto por [Montreuil et al. \(1991\)](#) denominado leiaute holográfico (*Holographic layout*). Essa abordagem consiste em desagregar totalmente um departamento funcional em máquinas individuais e posicioná-las o mais distante entre si. Uma extensão de sua abordagem foi realizada por [Marcotte et al. \(1998\)](#) que incorporou ao modelo o número esperado de movimentação dos produtos entre as máquinas.

O leiaute distribuído é um tipo de leiaute obtido por meio da resolução simultânea do problema de leiaute de facilidades e o problema de roteamento de fluxo ([Benjaafar et al., 2002](#)). Embora a sua concepção não possua uma estrutura celular, o leiaute distribuído pode servir de base para utilizar o conceito de célula virtual proposto por [McLean et al. \(1982\)](#).

A célula virtual consiste em um agrupamento lógico de recursos que será temporariamente dedicado na produção de um produto. Assim que finalizar a produção, esses recursos estarão disponíveis para compor uma nova célula virtual e iniciar a produção de um próximo produto conforme ilustrado na [Figura 2.2](#).

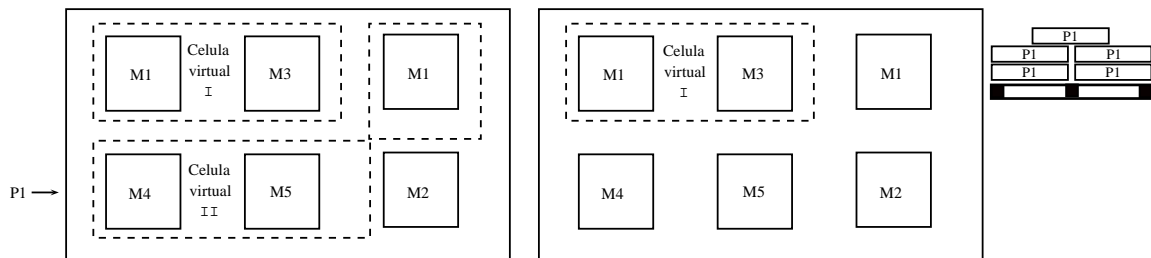


Figura 2.2: Formação de célula virtual

Nesse exemplo hipotético, irá iniciar a produção do produto 1 (P1) que requer as máquinas 4, 5 e 1. Observa-se que já há um produto em processo que requisitou as máquinas 1 e 3 formando a célula virtual I. Ao iniciar a fabricação do produto 1, criou-se uma nova célula virtual - célula virtual II - composta pelas máquinas 4, 5 e 1 que deixará de existir assim que completar a produção e tais máquinas estarão disponíveis para executar uma nova ordem.

A formulação proposta para resolver o problema de leiaute de facilidade estático considera o chão de fábrica como sendo um espaço discreto tal que é possível alocar somente um recurso em cada posição disponível.

Na próxima seção é apresentado as principais formulações referente ao problema de leiaute de facilidades.

2.2 Formulações Matemáticas

Conforme mencionado na Seção 2.1, a representação do espaço físico disponível se dá de forma contínua ou discreta. Nesta seção são apresentadas as formulações matemáticas propostas para resolver o problema de leiaute de facilidades bem como o método de resolução utilizado.

2.2.1 Formulações Discretas

2.2.1.1 Formulação de Koopmans e Beckman

Koopmans e Beckman (1957) propõe o primeiro modelo para resolver o problema de alocação de recursos indivisíveis denominado problema quadrático de atribuição (*Quadratic Assignment Problem - QAP*). O objetivo da formulação é instalar os n recursos em n posições de forma a minimizar o custo de transporte entre recursos e o custo incorrido ao instalar um dado recurso em uma determinada posição.

O modelo considera três matrizes simétricas de dimensão n^2 para representar os seguintes parâmetros:

a_{ij}	custo de instalação da facilidade i na posição j ,
f_{ik}	fluxo de material entre a facilidade i e k ,
d_{jl}	custo unitário de transporte da posição j até l ,

E a variável de atribuição é dada por:

$$x_{ij} = \begin{cases} 1, & \text{se a facilidade } i \text{ é instalada na posição } j; \\ 0, & \text{caso contrário.} \end{cases}$$

A sua formulação é dada por:

$$\min \sum_{i=1}^n \sum_{j=1}^n a_{ij} x_{ij} + \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n \sum_{l=1}^n f_{ik} d_{jl} x_{ij} x_{kl} \quad (2.1)$$

sujeito a:

$$\sum_{j=1}^n x_{ij} = 1 \quad i = 1, \dots, n \quad (2.2)$$

$$\sum_{i=1}^n x_{ij} = 1 \quad j = 1, \dots, n \quad (2.3)$$

$$x_{ij} \in \{0, 1\} \quad i = 1, \dots, n, \quad j = 1, \dots, n \quad (2.4)$$

A função objetivo (2.1) minimiza o custo de instalação e a distância relativa entre recursos. As restrições (2.2) garantem que somente um recurso seja instalado em uma posição e as restrições (2.3) garantem que em cada posição pode ser instalada somente um recurso.

2.2.1.2 Formulação de Lawler

Lawler (1963) introduziu um parâmetro com quatro índices para representar o custo unitário de transporte e o custo de instalação da seguinte forma:

$$b_{ijkl} = \begin{cases} f_{ik} c_{jl} + a_{ij}, & \text{se } i = k, j = l \\ f_{ik} c_{jl}, & \text{se } i \neq k \text{ ou } j \neq l \end{cases}$$

O modelo de Koopmans e Beckman pode ser reformulado como:

$$\min \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n \sum_{l=1}^n b_{ijkl} x_{ij} x_{kl} \quad (2.5)$$

sujeito a: (2.2), (2.3) e (2.4)

Tanto a formulação de Koopmans e Beckman quanto a de Lawler são formulações não lineares que possui somente a variável de atribuição na formulação. O fluxo entre recursos é considerado como um parâmetro de entrada do modelo o que diferente da formulação proposta que considera o fluxo como sendo uma variável de decisão.

2.2.1.3 Formulação de Benjaafar e Sheikhzadeh

A formulação proposta por Benjaafar e Sheikhzadeh (2000) permite a utilização de diversas cópias de um mesmo tipo de máquina e posicioná-las de forma independente

entre si. Essas máquinas possuem capacidade finita e a métrica de *Manhattan* foi utilizada para calcular a distância entre posições.

Seguem os parâmetros utilizados pelos autores.

K	total de posições,
N	total de tipos de máquinas,
P	total de produtos,
S	total de cenários,
N_i	número de máquinas do tipo $i = 1, \dots, N$,
c_{n_i}	capacidade da máquina $n_i = 1, \dots, N_i$,
$t_{m_j p}$	tempo unitário de processamento do produto $p = 1, \dots, P$ na máquina $m_j = 1, \dots, N_j$,
d_{kl}	distância entre as posições $k = 1, \dots, K$ e $l = 1, \dots, K$ tal que $k \neq l$,
v_{ijps}	volume de fluxo entre a máquina do tipo i e a do tipo $j = 1, \dots, N$ de um dado produto p em um dado cenário de demanda $s = 1, \dots, S$,
π_s	probabilidade de ocorrência do cenário $s = 1, \dots, S$

As variáveis de atribuição e de fluxo são denotadas por:

$$x_{n_i k} = \begin{cases} 1, & \text{se a } n\text{-ésima máquina do tipo } i \text{ é instalada na posição } k; \\ 0, & \text{caso contrário.} \end{cases}$$

$v_{n_i m_j ps} \geq 0$: informa o volume de fluxo do produto p entre a n -ésima máquina do tipo i e a m -ésima máquina do tipo j referente ao cenário de demanda s .

$$\min \sum_{s=1}^S \sum_{p=1}^P \sum_{i=1}^N \sum_{j=1}^N \sum_{n_i=1}^{N_i} \sum_{m_j=1}^{N_j} \sum_{k=1}^K \sum_{l=1}^K \pi_s v_{n_i m_j ps} d_{kl} x_{n_i k} x_{m_j l} \quad (2.6)$$

sujeito a:

$$\sum_{k=1}^K x_{n_i k} = 1 \quad \forall n_i, i \quad (2.7)$$

$$\sum_{i=1}^N \sum_{n_i=1}^{N_i} x_{n_i k} = 1 \quad \forall k \quad (2.8)$$

$$\sum_{p=1}^P \sum_{i=0}^N \sum_{n_i=1}^{N_i} v_{n_i m_j p s} t_{m_j p} \leq c_{m_j} \quad \forall m_j, j, s \quad (2.9)$$

$$\sum_{n_i=1}^{N_i} \sum_{m_j=1}^{N_j} v_{n_i m_j p s} = v_{ijps} \quad \forall i, j = 0, 1, \dots, N, \quad (2.10)$$

$$\sum_{i=0}^N \sum_{n_i=1}^{N_i} v_{n_i m_j p s} = \sum_{q=0}^N \sum_{r_q=1}^{N_q} v_{m_j r_q p s} \quad \forall m_j, j, p, s \quad (2.11)$$

$$x_{n_j k} \in \{0, 1\} \quad \forall n_i, k \quad (2.12)$$

$$v_{n_i m_j p s} \geq 0 \quad \forall i, j = 0, 1, \dots, N, \quad (2.13)$$

$$n_i, m_j, p, s$$

A função objetivo (2.6) minimiza o fluxo dos produtos entre departamentos e a distância entre máquinas. As restrições (2.7) e (2.8) asseguram que cada máquina é instalada em exatamente uma posição e cada posição comporta somente uma máquina. A capacidade das máquinas é assegurada pelas restrições (2.9). O total de fluxo requisitado entre máquinas é garantido pelas restrições (2.10) e as restrições (2.11) asseguram o balanceamento de fluxo.

Foi adotado um procedimento heurístico similar ao proposto por Cooper (1963) que consiste em resolver dois sub-problemas de forma iterativa. O primeiro passo resolve o problema de roteamento de fluxo por um sub-problema derivado do problema original dado um leiaute pré-definido, ou seja, as variáveis de atribuição x na formulação original passam a ser consideradas como parâmetro denotado por \bar{x} . Nessa etapa, os autores utilizaram o CPLEX para resolver o seguinte problema de roteamento de fluxo.

$$\min \sum_{s=1}^S \sum_{p=1}^P \sum_{i=1}^N \sum_{j=1}^N \sum_{n_i=1}^{N_i} \sum_{m_j=1}^{N_j} \sum_{k=1}^K \sum_{l=1}^K \pi_s v_{n_i m_j p s} d_{kl} \bar{x}_{n_i k} \bar{x}_{m_j l} \quad (2.14)$$

sujeito a: (2.9), (2.10), (2.11) e (2.13).

O passo seguinte consiste em resolver o problema de atribuição para cada cenário s tal que $\bar{v}_{n_i m_j p} = \sum_{s=1}^S v_{n_i m_j p s}$. Nessa etapa, os autores utilizaram a heurística de aprimoramento 2-opt com múltiplas soluções iniciais para atribuir as máquinas às posições.

$$\min \sum_{p=1}^P \sum_{i=1}^N \sum_{j=1}^N \sum_{n_i=1}^{N_i} \sum_{m_j=1}^{N_j} \sum_{k=1}^K \sum_{l=1}^K \bar{v}_{n_i m_j p} d_{kl} x_{n_i k} x_{m_j l} \quad (2.15)$$

sujeito a: (2.7), (2.8) e (2.12).

As instâncias geradas para realizar os testes computacionais foram baseadas em um leiaute com 16 ou 40 posições. A quantidade de produtos e o número de operações foram obtidos pela distribuição uniforme com parâmetros (5, 10) e (2, 8) respectivamente. O tempo computacional gasto para resolver as instâncias com 16 posições está entre 33 e 55 segundos. Já para as instâncias maiores, leiaute com 40 posições, o tempo compreende entre 260 e 524 segundos.

2.2.1.4 Formulação de Lahmar e Benjaafar

A formulação proposta por [Lahmar e Benjaafar \(2005\)](#) é uma extensão do modelo de [Benjaafar e Sheikhzadeh](#) o qual é utilizado em um contexto dinâmico. O posicionamento dos recursos estão em função do volume de fluxo de cada período e o objetivo é projetar um leiaute que minimiza o total do custo de movimentação de materiais e o custo de realocação dos departamentos entre períodos consecutivos sobre um horizonte finito de planejamento.

Considere a mesma notação utilizada por [Benjaafar e Sheikhzadeh](#), além da definição dos seguintes parâmetros adicionais: seja $r_{n_i k l}$ o custo de realocação da n -ésima máquina do tipo i entre as posições k e l , o custo unitário de movimentação entre as posições k e l é dado por c_{kl} e v_{ijpt} o total de fluxo do produto p entre as máquinas do tipo i e j no período t .

Foi adicionado um índice nas variáveis de decisão para representar o período de planejamento indexado por $t = 1, \dots, T$. Dessa forma, as variáveis de atribuição são denotadas por $x_{n_i k t}$ e irá assumir valor 1 se a n -ésima máquina do tipo i é instalada na posição k no período t e 0 caso contrário. As variáveis de fluxo são dadas por $v_{n_i m_j p t}$ que representa o volume de fluxo do produto p entre a n -ésima máquina do tipo i e a m -ésima máquina do tipo j no período t .

$$\min \sum_{t=1}^T \sum_{p=1}^P \sum_{i=1}^N \sum_{n=1}^{N_i} \sum_{j=1}^N \sum_{m=1}^{N_j} \sum_{k=1}^M \sum_{l=1}^M v_{n_i m_j p t} x_{n_i k t} x_{m_j l t} c_{kl} d_{kl} + \quad (2.16)$$

$$\sum_{t=1}^{T-1} \sum_{i=1}^N \sum_{n=1}^{N_i} \sum_{k=1}^M \sum_{l=1}^M x_{n_i k t} x_{n_i l (t+1)} r_{n_i k l}$$

sujeito a:

$$\sum_{i=1}^N \sum_{n=1}^{N_i} x_{n_i k t} = 1 \quad \forall k, t \quad (2.17)$$

$$\sum_{k=1}^M x_{n_i k t} = 1 \quad \forall i, n, t \quad (2.18)$$

$$\sum_{m=1}^{N_j} \sum_{n=1}^{N_i} v_{n_i m_j p t} = v_{ij p t} \quad \forall i, j, p, t \quad (2.19)$$

$$\sum_{i=0}^N \sum_{n=1}^{N_i} v_{n_i m_j p t} = \sum_{q=0}^N \sum_{r_q=1}^{N_q} v_{m_j r_q p t} \quad \forall j, m, p, t \quad (2.20)$$

$$\sum_{p=1}^P \sum_{i=0}^N \sum_{n_i=1}^{N_i} v_{n_i m_j p t} t_{m_j p} \leq c_{m_j} \quad \forall j, m, t \quad (2.21)$$

$$x_{n_i k t} \in \{0, 1\} \quad \forall i, n, k, t \quad (2.22)$$

$$v_{n_i m_j p t} \geq 0 \quad \forall i, n, j, m, p, t \quad (2.23)$$

A função objetivo (2.16) minimiza o custo de movimentação dos produtos e o custo de realocação dos recursos entre períodos consecutivos. As restrições (2.17) e (2.18) asseguram que cada recurso deverá ser instalado em uma posição e cada posição comporta somente uma única máquina em cada período. A quantidade necessária de fluxo entre recursos para cada período é garantida pelas restrições (2.19). As restrições (2.20) garantem o balanço de fluxo entre os recursos. E a quantidade de fluxo que passa em um recurso m instalado na posição j não deverá exceder a sua capacidade, restrições (2.21).

Lahmar e Benjaafar utilizaram o mesmo procedimento heurístico adotado por Benjaafar e Sheikhzadeh e utilizaram tamanhos de leiaute que variaram entre 16 e 48 posições. Diversas faixas de valores foram utilizadas para representar o custo de rearranjo, a demanda e o número de operações de cada produto. Pelos experimentos realizados, os autores comprovaram uma melhora na performance do leiaute ao de-

sagregar um leiaute funcional e distribuir os recursos utilizando o modelo proposto.

2.2.2 Formulações Contínuas

2.2.2.1 Formulação de Montreuil

A formulação proposta por [Montreuil \(1990\)](#) utiliza as seguintes definições: seja L e W o comprimento máximo na direção x e y respectivamente e N o conjunto de departamentos. A área necessária para instalar um determinado departamento $i \in N$ é denotada a_i .

Os limites inferior e superior de cada departamento é dado por ub_i e lb_i respectivamente e a distância mínima e máxima a ser respeitada é denotada por p_i e P_i . O conjunto de fluxo entre departamentos é dado por $F = \{f_{ij}\}$ tal que $|F| = M$ e o m -ésimo fluxo originado no departamento $i(m)$ e terminado no departamento $j(m)$.

A distância linear entre departamentos na direção x e y é denotado por d_{ij}^x e d_{ij}^y respectivamente. Seja (x_i, y_i) , a localização do departamento i e considerando que cada departamento possui forma retangular, departamento i possui dimensão $2l_i$ na direção x e $2w_i$ em y . As variáveis z_{ij}^x e z_{ij}^y determinam a localização relativa entre dois departamentos. A formulação matemática é dada por:

$$\min \sum_m f_m(d_m^x + d_m^y) \quad (2.24)$$

sujeito a:

$$d_m^x \geq x_{i(m)} - x_{j(m)} \quad \forall m \quad (2.25)$$

$$d_m^x \geq x_{j(m)} - x_{i(m)} \quad \forall m \quad (2.26)$$

$$d_m^y \geq y_{i(m)} - y_{j(m)} \quad \forall m \quad (2.27)$$

$$d_m^y \geq y_{j(m)} - y_{i(m)} \quad \forall m \quad (2.28)$$

$$l_i \leq x_i \leq L - l_i \quad \forall i \quad (2.29)$$

$$w_i \leq y_i \leq W - w_i \quad \forall i \quad (2.30)$$

$$lb_i \leq 2l_i \leq ub_i \quad \forall i \quad (2.31)$$

$$lb_i \leq 2w_i \leq ub_i \quad \forall i \quad (2.32)$$

$$2 \leq z_{ij}^x + z_{ji}^x + z_{ij}^y + z_{ji}^y \leq 3 \quad \forall i, j : i < j \quad (2.33)$$

$$x_i + l_i \leq x_j - l_j + Lz_{ij}^x \quad \forall i, j \quad (2.34)$$

$$x_j + l_j \leq x_i - l_i + Lz_{ji}^x \quad \forall i, j \quad (2.35)$$

$$y_i + w_i \leq y_j - w_j + Wz_{ij}^y \quad \forall i, j \quad (2.36)$$

$$y_j + w_j \leq y_i - w_i + Wz_{ji}^y \quad \forall i, j \quad (2.37)$$

$$p_i \leq 4(l_i + w_i) \leq P_i \quad \forall i \quad (2.38)$$

$$x_i, y_i \geq 0 \quad \forall i \quad (2.39)$$

$$z_{ij}^x, z_{ij}^y \geq 0 \quad \forall i, j \quad (2.40)$$

A função objetivo (2.24) minimiza a distância entre departamentos e as inequações (2.25) – (2.28) são utilizadas para linearização do modelo. As restrições (2.29) e (2.30) asseguram o posicionamento dos departamentos dentro do espaço limite disponível, enquanto que as restrições (2.31) e (2.32) garantem que o comprimento mínimo e máximo seja respeitado. As restrições (2.33) permitem o posicionamento dos recursos em duas ou três direções. O não sobreposicionamento dos recursos instalados é garantido pelas restrições (2.34)–(2.37). As restrições (2.38) garantem que a distância mínima e máxima seja respeitada, e as restrições de domínio das variáveis são dadas pelas restrições (2.39) e (2.40).

2.2.2.2 Formulação de Solimanpur e Jafari

A formulação proposta por Solimanpur e Jafari (2008) aloca os recursos no chão de fábrica e de forma simultânea realiza o roteamento do fluxo dos produtos além de

considerar que os recursos possuem capacidade finita.

Considere I o número de máquinas de natureza distinta, N_i a quantidade disponível de cada tipo $i = 1, \dots, I$ e M o total de máquinas disponíveis. A capacidade de cada máquina é dada por C_i . Seja l_i e b_i o comprimento e a largura de cada máquina e h e v o comprimento horizontal e vertical total disponível respectivamente.

O número total de produtos a serem processados é dado por P e seja D_p a demanda de cada produto $p = 1, \dots, P$. O tempo de processamento de um produto em um dado recurso do tipo i é denotado por t_{ip} . O parâmetro b_{ip} informa se o produto p requisita a máquina do tipo i e a_{ij}^p se a máquina do tipo $j = 1, \dots, I$ é requisitada imediatamente após a máquina do tipo i referente à rota do produto p .

A variável de decisão Z_{inp} irá assumir 1 se o produto p será produzido pela n -ésima máquina do tipo i tal que $n = 1, \dots, N_i$, e 0 caso contrário. Já as variáveis x_{in} e y_{in} informa a localização da n -ésima máquina instalada na coordenada horizontal e vertical respectivamente. Segue abaixo a formulação:

$$\min \sum_{p=1}^P \sum_{i=1}^I \sum_{\substack{j=1 \\ i \neq j}}^I \sum_{n=1}^{N_i} \sum_{r=1}^{N_j} a_{ij}^p D_p Z_{inp} Z_{jrp} (|x_{in} - x_{jr}| + |y_{in} - y_{jr}|) \quad (2.41)$$

sujeito a:

$$0 \leq |x_{in} - x_{jr}| \leq h \quad \forall i, j, n, r \quad (2.42)$$

$$0 \leq |y_{in} - y_{jr}| \leq v \quad \forall i, j, n, r \quad (2.43)$$

$$|x_{in} - x_{jr}| + M\alpha_{injr} \geq 0.5(l_i + l_j) + d_{ij} \quad \forall i, j, n, r \quad (2.44)$$

$$|y_{in} - y_{jr}| + M(1 - \alpha_{injr}) \geq 0.5(b_i + b_j) + d_{ij} \quad \forall i, j, n, r \quad (2.45)$$

$$\alpha_{injr}(1 - \alpha_{injr}) = 0 \quad \forall i, j, n, r \quad (2.46)$$

$$\sum_{n=1}^{N_i} Z_{inp} = b_{ip} \quad \forall i, p \quad (2.47)$$

$$\sum_{p=1}^P Z_{inp} t_{ip} D_p \leq C_i \quad \forall i, n \quad (2.48)$$

A função objetivo (2.41) minimiza a distância total percorrida pelos materiais no chão de fábrica. As restrições (2.42) e (2.43) garantem que o posicionamento das máquinas estejam em uma região viável. O não sobreposicionamento das máquinas é garantido pelas restrições (2.44) e (2.45), e as restrições (2.46) garantem que somente uma delas estejam ativa. Cada produto deve visitar todas as máquinas pertencentes à sua rota, (2.47). E a capacidade de cada máquina é garantida pelas restrições (2.48)

O modelo foi resolvido de forma exata por meio de um procedimento baseado no *Branch & Bound*. Mediante a resolução de um problema pequeno que possui 4 tipos de máquinas e 3 produtos a serem processados, os autores demonstraram uma redução de 41,8% do total de distância percorrida pelos produtos comparado ao leiaute funcional. Para um problema médio que possui 10 tipos de máquinas e 5 produtos, o algoritmo proposto encontrou a solução ótima em menos de 24 horas de processamento. Já para um problema grande, 20 tipos de máquinas e 8 produtos, o algoritmo não foi capaz de encontrar a solução ótima em um tempo limite de 24 horas.

2.2.2.3 Taghavi e Murat

Taghavi e Murat (2011) aplicaram a técnica padrão de linearização ao modelo proposto por Solimanpur e Jafari. Para tanto, considere a mesma notação utilizada por Solimanpur e Jafari. As variáveis Z_{inp} e Z_{jrp} foram substituídas pela variável $Z_{injr p}$ a qual assumirá valor 1 somente quando ambas também assumirem. As variáveis $D_X_{injr p}$ e $D_Y_{injr p}$ correspondem a posição das máquinas nas coordenadas x e y respectivamente. Segue o modelo proposto por Taghavi e Murat.

$$\min \sum_{p=1}^P \sum_{i=1}^I \sum_{\substack{j=1 \\ (j \neq i)}}^I \sum_{n=1}^{N_i} \sum_{r=1}^{N_j} a_{ij}^p D_p Z_{injr p} (D_X_{injr p} + D_Y_{injr p}) \quad (2.49)$$

sujeito a:

$$0.5l_i \leq x_{in} \leq h - 0.5l_i \quad \forall i, n \quad (2.50)$$

$$0.5b_i \leq y_{in} \leq v - 0.5b_i \quad \forall i, n \quad (2.51)$$

$$\sum_{n=1}^{N_i} Z_{inp} = b_{ip} \quad \forall i, p \quad (2.52)$$

$$\sum_{p=1}^P Z_{inp} t_{ip} D_p \leq C_i \quad \forall i, n \quad (2.53)$$

$$Z_{injr p} \leq Z_{inp} + Z_{jrp} - 1 \quad \forall i, j : i \neq j, n, r, p \quad (2.54)$$

$$D_X_{injr p} \leq (x_{in} - x_{jr}) - M(1 - Z_{injr p}) \quad \forall i, j : i \neq j, n, r, p \quad (2.55)$$

$$D_X_{injr p} \leq -(x_{in} - x_{jr}) - M(1 - Z_{injr p}) \quad \forall i, j : i \neq j, n, r, p \quad (2.56)$$

$$D_Y_{injr p} \leq (y_{in} - y_{jr}) - M(1 - Z_{injr p}) \quad \forall i, j : i \neq j, n, r, p \quad (2.57)$$

$$D_Y_{injr p} \leq -(y_{in} - y_{jr}) - M(1 - Z_{injr p}) \quad \forall i, j : i \neq j, n, r, p \quad (2.58)$$

$$+ (x_{in} - x_{jr}) + M\alpha_{injr} + M\beta_{injr}^x \leq 0.5(l_i + l_j) + d_{ij} \quad \forall i, j : i \neq j, n, r \quad (2.59)$$

$$-(x_{in} - x_{jr}) + M\alpha_{injr} + M(1 - \beta_{injr}^x) \leq 0.5(l_i + l_j) + d_{ij} \quad \forall i, j : i \neq j, n, r \quad (2.60)$$

$$+(y_{in} - y_{jr}) + M(1 - \alpha_{injr}) + M\beta_{injr}^y \leq 0.5(b_i + b_j) + d_{ij} \quad \forall i, j : i \neq j, n, r \quad (2.61)$$

$$-(y_{in} - y_{jr}) + M(1 - \alpha_{injr}) + M(1 - \beta_{injr}^y) \leq 0.5(b_i + b_j) + d_{ij} \quad \forall i, j : i \neq j, n, r \quad (2.62)$$

$$Z_{inp}, Z_{injr} \in \{0, 1\} \quad \forall i, j : i \neq j, n, r, p \quad (2.63)$$

$$\alpha_{injr}, \beta_{injr}^x, \beta_{injr}^y \in \{0, 1\} \quad \forall i, j : i \neq j, n, r, p \quad (2.64)$$

$$x_{in}, y_{in}, D_X_{injr}, D_Y_{injr} \geq 0 \quad \forall i, j : i \neq j, n, r, p \quad (2.65)$$

A formulação original foi decomposta em dois sub-problemas que são resolvidos conforme a heurística proposta por [Cooper \(1963\)](#). Além do uso dessa heurística, os autores incorporaram um algoritmo de perturbação baseado nas decisões de atribuição para escapar de ótimos locais.

Os problemas testes foram classificados em quatro níveis de complexidade. O que apresenta menor nível, possui 5 máquinas e cada máquina possui 2 cópias com 4 produtos a serem processados. Já o maior, possui 20 máquinas com três cópias e um total de 9 produtos.

O método proposto se mostrou eficiente e resolveu todos os problemas testes. Para o mais simples, foi necessário 213 segundos e para o mais complexo 11,2 horas de processamento.

Capítulo 3

Problema de leiaute de facilidades e roteamento de fluxo

A Seção 3.1 apresenta as definições e a formulação proposta para projetar um leiaute distribuído. Para tanto, é necessário que algumas informações sejam conhecidas a priori, tais como a sequência tecnológica dos produtos, o número de posições disponíveis e o total de máquinas disponíveis. É apresentado um exemplo numérico para ilustrar as informações extraídas pelas variáveis de decisão. A Seção apresenta uma segunda formulação destinada a resolver o mesmo problema. A diferença entre as formulações é dado pela forma da contabilização do fluxo, em que na formulação apresentada na Seção 3.1.2, o fluxo é dado entre máquinas já na formulação 3.1, o fluxo é contabilizado por produto. Na Seção é apresentado o desenvolvimento do método de decomposição de Benders Clássico para a formulação apresentada na Seção 3.1. Em seguida é apresentado as técnicas que foram incorporadas ao método clássico.

3.1 Definições e formulação

Seja P o número de posições disponíveis para instalar um determinado recurso. A distância entre posições é dado pela matriz d_{pr} no qual $p = 1, \dots, P$ e $r = 1, \dots, P$ tal que $p \neq r$. Define-se ainda duas posições fictícias 0 e n para representar o almoxarifado e a doca de expedição respectivamente.

O total de produtos a serem produzidos é dado por I e a demanda de cada um é dada pelo parâmetro w_i em que $i = 1, \dots, I$. Cada produto possui uma sequência tecnológica ou roteiro de produção que é dada pela matriz s_{ij} . Essa matriz retorna o tipo de recurso necessário para processar um determinado produto em uma dada operação $j = 1, \dots, o_i$ da sequência, sendo o_i o número total de operações de cada produto.

O número de recursos de natureza distinta é dado por M . Cada tipo de recurso possui um número total de cópias disponíveis denotado por c_m e a cada aquisição é incorrido um custo a_m , em que $m = 1, \dots, M$. Seja ainda g_{mp} o custo de se instalar um determinado recurso m em uma posição específica p .

Os parâmetros Γ , Φ e Θ representam o custo de movimentação do material. No qual Γ_{0p} é o custo unitário de retirada do material no almoxarifado e enviá-lo a uma posição p , Φ_{pr} o custo de movimentação entre recursos instalados nas posições p e r e Θ_{pn} o custo de enviar o produto da posição p até à doca de expedição n .

O modelo possui as seguintes variáveis de decisão: a variável $x_{mp} \in \{0, 1\}$, irá assumir valor 1 quando o recurso m for instalado na posição p e 0, caso contrário. Se uma cópia do recurso m for adquirida, a variável $y_{mc} \in \{0, 1\}$ irá assumir valor 1 e 0 caso contrário. A variável h_{ip} representa a operação de retirada de material no almoxarifado e determina em que posição irá iniciar as operações. Já a variável t_{ip} decide em qual posição será realizada a última operação para então ser enviado à doca de expedição. E a variável $f_{ijpr} \geq 0$ representa a fração de fluxo da operação $j = 1, \dots, o_i$, pertencente ao produto i realizada na posição p e enviada a r .

Segue abaixo a formulação matemática:

$$\min \sum_i \left[\sum_{j=1}^{(o_i-1)} \sum_p \sum_{\substack{r \\ r \neq p}} w_i \Phi_{pr} d_{pr} f_{ijpr} + \sum_p (w_i \Gamma_{0p} d_{0p} h_{ip} + w_i \Theta_{pn} d_{pn} t_{ip}) \right] + \quad (3.1)$$

$$\sum_m \left[\sum_p g_{mp} x_{mp} + \sum_{c=1}^{c_m} a_m y_{mc} \right]$$

sujeito a:

$$\sum_p h_{ip} = 1 \quad \forall i \quad (3.2)$$

$$\sum_r f_{i1pr} = h_{ip} \quad \forall i, p \quad (3.3)$$

$$\sum_{\substack{r \\ r \neq p}} f_{ijrp} = \sum_{\substack{r \\ r \neq p}} f_{i(j+1)pr} \quad \forall i, j = 1, \dots, (o_i - 2), p : o_i > 1 \quad (3.4)$$

$$\sum_p f_{i(o_i-1)pr} = t_{ir} \quad \forall i, r : o_i > 1 \quad (3.5)$$

$$\sum_r t_{ir} = 1 \quad \forall i \quad (3.6)$$

$$h_{ip} \leq x_{s(i,1)p} \quad \forall i, p \quad (3.7)$$

$$f_{ijpr} \leq x_{s(i,j)p} \quad \forall i, j = 1, \dots, (o_i - 1), p, r : r \neq p \quad (3.8)$$

$$f_{ijpr} \leq x_{s(i,j+1)r} \quad \forall i, j = 1, \dots, (o_i - 1), p, r : r \neq p \quad (3.9)$$

$$t_{ir} \leq x_{s(i,o_i)r} \quad \forall i, r \quad (3.10)$$

$$\sum_m x_{mp} \leq 1 \quad \forall p \quad (3.11)$$

$$\sum_p x_{mp} \leq 1 + \sum_{c=1}^{c_m} y_{mc} \quad \forall m \quad (3.12)$$

$$y_{m(c+1)} \leq y_{mc} \quad \forall m, c = 1, \dots, (c_m - 1) \quad (3.13)$$

$$x_{mp} \in \{0, 1\} \quad \forall m, p \quad (3.14)$$

$$y_{mc} \in \{0, 1\} \quad \forall m, c = 1 \dots c_m \quad (3.15)$$

$$f_{ijpr} \geq 0 \quad \forall i, j, p, r : r \neq p \quad (3.16)$$

$$h_{ip} \geq 0 \quad \forall i, p \quad (3.17)$$

$$t_{ir} \geq 0 \quad \forall i, r \quad (3.18)$$

A função objetivo (3.1) minimiza o custo de movimentação dos produtos, o custo de instalação e o custo de aquisição.

A conservação de fluxo é garantida pelas restrições (3.2)–(3.6). As restrições (3.7) – (3.10) asseguram que somente terá fluxo do produto em uma determinada posição se o recurso necessário para realizar a operação estiver alocada na mesma.

As restrições (3.11) garantem que cada posição comporta no máximo um recurso. O número máximo de recurso de um determinado tipo é limitado pelas restrições (3.12). As restrições (3.13) asseguram que a duplicação de um recurso somente será realizada se o recurso anterior já estiver em uso.

E as restrições (3.14)–(3.18) são de integralidade e não negatividade das variáveis.

3.1.1 Exemplo Numérico

Considere que em um sistema de manufatura há 5 tipos de máquina. Cada uma possui 2 cópias e o custo de cada cópia foi gerado por meio de uma distribuição uniforme com mínimo de 10.000 e máximo de 20.000. Ao instalar uma dada máquina em uma posição é incorrido um custo conforme os valores podem ser observados na Tabela 3.1 que resume os dados associado às máquinas.

A demanda anual de cada produto foi gerada por meio de uma distribuição uniforme entre os limites 20.000 e 70.000 unidades. De acordo com a Tabela 3.2, o produto 1 demanda 52.300 unidades e é produzido em 4 etapas que requerem as máquinas 1, 2, 3 e 5 nessa ordem.

A distância entre duas posições consecutivas é de 1 unidade e para cada par de

Tabela 3.1: Parâmetro referente à máquina

Tipo Máq. M	# Cópias c_m	Custo a_m	Custo de instalação - g_{mp}								
			1	2	3	4	5	6	7	8	9
1	2	16.615	440	324	622	308	618	625	390	560	337
2	2	10.855	540	311	532	650	562	582	573	482	629
3	2	18.120	366	447	330	313	326	648	353	630	448
4	2	12.775	345	435	333	376	321	433	457	307	368
5	2	10.686	507	407	321	694	667	513	619	617	663

Tabela 3.2: Parâmetro referente ao produto

# Produtos I	Demanda w_i	# operações o_i	operação - s_{ij}			
			1	2	3	4
1	52.300	4	5	2	1	3
2	64.900	4	1	3	5	4
3	50.600	3	2	4	5	-
4	21.600	3	1	4	2	-

posição, a distância foi calculada pela métrica de *Manhatan*.

Tabela 3.3: Matriz de distância

	1	2	3	4	5	6	7	8	9	n
0	1	2	3	2	3	4	3	4	5	-
1	-	1	2	1	2	3	2	3	4	5
2		-	1	2	1	2	3	2	3	4
3			-	3	2	1	4	3	2	3
4				-	1	2	1	2	3	4
5					-	1	2	1	2	3
6						-	3	2	1	2
7							-	1	2	3
8								-	1	2
9									-	1

A configuração do leiaute resultante desse exemplo é ilustrado na figura 3.1. Pode-se observar que, em função dos custos associados, é vantajoso adquirir uma cópia da máquina 4 e uma cópia da máquina 2. Vale destacar que as máquinas adquiridas foram alocadas em posições estratégicas e não em posições adjacentes.

O roteamento do fluxo de cada produto é informado pela Tabela 3.4. O produto 1, que requer a máquina 5 na sua primeira operação, inicia a sua produção na posição 1 (P1) e finaliza na posição 8. Com essa configuração de leiaute, observa-se que é possível

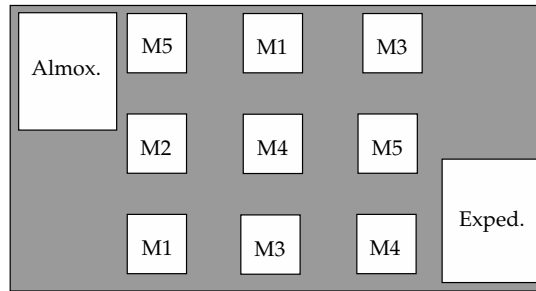


Figura 3.1: Configuração do leiaute do exemplo numérico

Tabela 3.4: Roteamento do fluxo por produto

Produto	Rota
1	P1 - P4 - P7 - P8
2	P2 - P3 - P6 - P9
3	P4 - P5 - P6
4	P2 - P5 - P4

produzir os produtos 1 e 2 simultaneamente.

3.1.2 Formulação atribuição

Nesta Seção é apresentado a formulação em que as máquinas são alocadas em função do fluxo entre as mesmas. Para tanto, considere a seguinte notação e definição do modelo:

Conjuntos:

- M Conjunto de tipos de máquinas;
 P Conjunto de posições disponíveis;

Considere que os índices $m, q \in M$, e os índices $0, p, r, n \in P$, em que 0 representa o almoxarifado e n a doca de expedição.

Parâmetros:

- AI_m 1 se há fluxo entre o almoxarifado e a máquina m , 0 caso contrário;
 A_{mq} 1 se há fluxo entre as máquinas m e q , 0 caso contrário;
 AF_m 1 se há fluxo entre a máquina m e a doca de expedição, 0 caso contrário;
 s_{mq} demanda de fluxo entre as máquinas m e q ;
 d_{pr} distância entre as posições p e r ;
 γ_m quantidade de demanda entre o almoxarifado e a máquina m ;

Γ_{0p}	custo unitário de deslocamento entre o almoxarifado e a posição p ;
ρ_m	quantidade de demanda entre a máquina m e a doca de expedição;
Θ_m	custo unitário de deslocamento entre a posição p e a doca de expedição;
g_{mp}	custo de instalação da máquina m na posição p ;
a_m	custo de aquisição da máquina m .

Variáveis:

$$x_{mp} = \begin{cases} 1, & \text{se a máquina } m \text{ for instalada na posição } p; \\ 0, & \text{c.c.} \end{cases}$$

$$y_{mc} = \begin{cases} 1, & \text{se a cópia da máquina } m \text{ for adquirida;} \\ 0, & \text{c.c.} \end{cases}$$

$f_{mqpr} \geq 0$, parcela do fluxo direcionada da máquina m instalada na posição p para a máquina q instalada na posição r .

$h_{mp} \geq 0$, parcela do fluxo direcionada do almoxarifado para a máquina m instalada na posição p .

$t_{mp} \geq 0$, parcela do fluxo direcionada da máquina m instalada na posição p para a doca de expedição.

$$\begin{aligned} \min \sum_m \sum_{m \neq q: A_{mq}=1}^q \sum_p \sum_{\substack{r \\ p \neq r}}^r s_{mq} d_{pr} f_{mqpr} + \sum_m \sum_{A I_m=1}^p \gamma_m d_{0p} \Gamma_{0p} h_{mp} + \\ \sum_m \sum_{A F_m=1}^p \rho_m d_{pn} \Theta_{pn} t_{mp} + \sum_m \left[\sum_p g_{mp} x_{mp} + \sum_{c=1}^{c_m} a_m y_{mc} \right] \end{aligned} \quad (3.19)$$

$$\sum_p \sum_{\substack{r \\ p \neq r}} f_{mqpr} = 1 \quad \forall m, q : A_{mq} = 1 \quad (3.20)$$

$$\sum_p h_{mp} = 1 \quad \forall m : AI_m = 1 \quad (3.21)$$

$$\sum_p t_{mp} = 1 \quad \forall m : AF_m = 1 \quad (3.22)$$

$$\sum_q \sum_{\substack{r \\ p \neq r : A_{mq}=1}} s_{mq} f_{qmrp} + \begin{cases} \gamma_m h_{mp}, & \text{se } AI_m = 1; \\ 0, & \text{c.c.} \end{cases} =$$

$$\sum_q \sum_{\substack{r \\ p \neq r : A_{mq}=1}} s_{mq} f_{mqpr} + \begin{cases} \theta_m t_{mp}, & \text{se } AF_m = 1; \\ 0, & \text{c.c.} \end{cases} \quad \forall m, p \quad (3.23)$$

$$f_{mqpr} \leq x_{mp} \quad \forall m, q, p, r : m \neq q$$

$$A_{mq} = 1, p \neq r \quad (3.24)$$

$$f_{mqpr} \leq x_{qr} \quad \forall m, q, r, p : m \neq q$$

$$A_{mq} = 1, p \neq r \quad (3.25)$$

$$h_{mp} \leq x_{mp} \quad \forall m, p : AI_m = 1 \quad (3.26)$$

$$t_{mp} \leq x_{mp} \quad \forall m, p : AF_m = 1 \quad (3.27)$$

$$\sum_m x_{mp} \leq 1 \quad \forall p \quad (3.28)$$

$$\sum_p x_{mp} \leq 1 + \sum_{c=1}^{c_m} y_{mc} \quad \forall m \quad (3.29)$$

$$y_{m(c+1)} \leq y_{mc} \quad \forall m,$$

$$c = 1, \dots, (c_m - 1) \quad (3.30)$$

$$x_{mp} \in \{0, 1\} \quad \forall m, p \quad (3.31)$$

$$y_{mc} \in \{0, 1\} \quad \forall m, c = 1 \dots c_m \quad (3.32)$$

$$f_{mqpr} \geq 0 \quad \forall m, q, p, r :$$

$$m \neq q, r \neq p \quad (3.33)$$

$$h_{mp} \geq 0 \quad \forall m, p \quad (3.34)$$

$$t_{mr} \geq 0 \quad \forall m, r \quad (3.35)$$

A função objetivo 3.19 minimiza os custos de fluxo entre máquinas e os custos de instalação e aquisição das mesmas.

As restrições 3.20 garantem que o total de fluxo alocado nas posições deverá ser

completo, tal que haja fluxo entre as mesmas. As restrições 3.21 asseguram que o fluxo a ser alocado nas posições deverá totalizar em 1 unidade, que representa 100 % do fluxo, para todas as máquinas tal haja fluxo entre o almoxarifado e a respectiva máquina. As restrições 3.22 é análoga as restrições 3.21, o que difere é que a análise é feita para todas as últimas máquinas, ou seja, máquinas que possuem fluxo com a doca de expedição.

O balanço de fluxo é garantido pelas restrições 3.23, em que uma parcela de fluxo inicial é adicionada se há fluxo entre o almoxarifado e a máquina. Caso a máquina pertence a última operação, é também adicionada uma parcela do fluxo entre a última máquina e a doca de expedição.

As restrições 3.24 – 3.27, garantem que somente terá fluxo da máquina em uma dada posição se a mesma estiver instalada na respectiva posição. As restrições 3.28 asseguram que cada posição comporta no máximo uma máquina.

O número total de máquinas disponíveis é assegurado pelas restrições 3.29. As restrições (3.30) asseguram que a duplicação de um recurso somente será realizada se o recurso anterior já estiver em uso. E as restrições (3.31)–(3.35) são de integralidade e não negatividade das variáveis.

Usando a formulação acima, foi criado um procedimento com o intuito de verificar a possível diferença das soluções quando o roteamento de fluxo dos produtos é feito simultaneamente com a atribuição das máquinas e quando o roteamento é realizado de forma desagregada, ou seja, quando as máquinas são atribuídas em um primeiro momento e posteriormente é feito o roteamento dos produtos.

Esse procedimento consiste em resolver o modelo 3.1.2 e armazenar o valor das variáveis de atribuição x e das variáveis y e utilizá-las como parâmetro na formulação 3.1. O algoritmo 1 descreve esse procedimento.

Algoritmo 1: Algoritmo modelo desagregado versus integrado

Resolva o modelo atribuição (3.19) – (3.35), obtenha valor das variáveis x e y .
 Resolva o modelo integrado (3.1) – (3.18) considerando x e y como parâmetro
 Guarde o valor da função objetivo - (FO-1)
 Resolva o modelo integrado (3.1) – (3.18) considerando x e y como variável
 Guarde o valor da função objetivo - (FO-2)

3.2 Método de Decomposição de Benders

A decomposição de Benders (1962) é um método para resolver problemas de programação linear inteiro misto (PLIM) que possuem uma estrutura especial no conjunto de restrições. Essa estrutura deve ser de tal forma que, ao fixar as variáveis com-

plicantes (variáveis inteiras), o modelo matemático original é decomposto em subproblemas menores que são computacionalmente mais fáceis de se resolver. O método é baseado na implantação dos conceitos de projeção de variáveis, seguido pela técnica de dualização, linearização e relaxação (Lasdon, 1972; Minoux e Vajda, 1986).

Em termos gerais, as variáveis complicantes do problema original são projetadas, resultando em um modelo equivalente que possui um número menor de variáveis porém com um número elevado de restrições que, geralmente, é uma função exponencial do número de variáveis projetadas. Essas restrições são adicionadas à medida que são necessárias seguindo um método sistemático. Nesse método o problema original é decomposto em dois problemas que iteragem até que a solução ótima seja encontrada.

O problema mestre (PM), que é uma versão relaxada do problema original, contém o conjunto de variáveis inteiras e suas respectivas restrições associadas. O PM é responsável por fixar os valores dessas variáveis inteiras e por fornecer um limite inferior (LI) para o problema. O subproblema (SP) é responsável por gerar novos cortes ou cortes de Benders que serão adicionados ao PM e por fornecer o limite superior (LS) para o problema. O algoritmo cicla, resolvendo o problema PM e o SP separadamente até que os limites superior e inferior converjam para uma solução ótima.

Considere o seguinte problema:

$$P_{original} \min c^\top x + f^\top y \quad (3.36)$$

$$\text{s. a: } Ax + By \geq b \quad (3.37)$$

$$Dy \geq d \quad (3.38)$$

$$x \geq 0 \quad (3.39)$$

$$y \in \mathbb{Z}^+ \quad (3.40)$$

Em que A é uma matriz $m \times n$, x e c são vetores de tamanho n , B e D são matriz $m \times p$ e $q \times p$ respectivamente e os vetores y e f possuem dimensão p .

Projetando o problema 3.36 - 3.40 no espaço definido pelas variáveis y , resulta no seguinte problema:

$$\min_{y \in Y} \left\{ f^\top y + \underbrace{\min_{x \geq 0} \{ c^\top x : Ax \geq b - By \}}_I \right\} \quad (3.41)$$

em que $Y = \{y : Dy \geq d, y \in \mathbb{Z}^+\}$.

O problema I é uma problema de programação linear que pode ser dualizado pela

associação de variáveis duais a cada restrição.

Associando as variáveis duais u às restrições $Ax \geq b - By$ do problema linear I temos o seguinte problema dual:

$$\max_{u \geq 0} \{(b - By)^\top u : A^\top u \leq c\} \quad (3.42)$$

Pela teoria da dualidade, os problemas (3.42) e o problema primal I são equivalentes. Dessa forma, podemos reescrever (3.41) como:

$$\min_{y \in Y} \left\{ f^\top y + \max_{u \geq 0} \{(b - By)^\top u : A^\top u \leq c\} \right\} \quad (3.43)$$

A vantagem em usar o problema dual se dá pelo fato da sua região de viabilidade não depender do valor da variável inteira y .

Seja $\Omega = \{u : A^\top u \leq c\}$ o espaço de viabilidade gerado pelas restrições $A^\top u \leq c$ e assumindo que $\Omega \neq \emptyset$ caso contrário o problema primal seria ilimitado ou inviável, o conjunto Ω é composto por pontos extremos $u^i (i = 1, \dots, |H|)$, pontos obtidos mediante as soluções limitadas do problema dual e por raios extremos $u^r (r = 1, \dots, |G|)$ obtidos pelas soluções ilimitadas do problema dual.

Soluções duais ilimitadas implica em soluções primais inviáveis. Como Ω é um poliedro e possui um número finito de geradores, utilizando o lema de Farkas pode-se então eliminar os valores de y que geram tais soluções.

$$0 \geq (b - By)^\top u^r \quad r = 1, \dots, |G| \quad (3.44)$$

Assim sendo, o problema (3.43) pode ser reformulado como:

$$\min \left\{ f^\top y + \max_{u \geq 0} \{(b - By)^\top u^i : i = 1, \dots, |H|\} \right\} \quad (3.45)$$

$$\text{s. a: } Dy \geq d \quad (3.46)$$

$$0 \geq (b - By)^\top u^r \quad r = 1, \dots, |G| \quad (3.47)$$

$$y \in \mathbb{Z}^+ \quad (3.48)$$

Com o auxílio da variável contínua η , o problema acima pode ser reformulado da

seguinte forma:

$$P_{mestre} \min f^\top y + \eta \quad (3.49)$$

$$\text{s. a: } Dy \geq d \quad (3.50)$$

$$\eta \geq u^i (b - By)^\top \quad i = 1 \dots |H| \quad (3.51)$$

$$0 \geq u^r (b - By)^\top \quad r = 1 \dots |G| \quad (3.52)$$

$$y \in \mathbb{Z}^+ \quad (3.53)$$

$$\eta \geq 0 \quad (3.54)$$

O problema acima é um problema equivalente ao problema original (3.36) – (3.40) denominado problema mestre (PM) de Benders. Esse problema possui um número menor de variáveis em relação ao problema original. A grande vantagem desse método está relacionada às restrições (3.51) e (3.52) que são adicionadas a cada iteração do método. Essas restrições foram denominadas por Benders como corte de otimalidade (3.51) e a restrição (3.52) como corte de viabilidade.

Sejam S_{PM}^* e SP^* a solução ótima do problema mestre e a do subproblema respectivamente. Considere LS o limite superior e LI o limite inferior e ϵ um valor pequeno, como por exemplo 0,01. O algoritmo de decomposição de Benders clássico é dado por:

Algoritmo 2: Algoritmo de decomposição de Benders clássico

Inicialização: $LS \leftarrow +\infty$, $LI \leftarrow -\infty$.

enquanto ($LS - LI > \epsilon$) **faça**

 Resolva o PM (3.49) – (3.54), obtendo S_{PM}^* e o valor das variáveis y .

$LI \leftarrow S_{PM}^*$ e atualize $y \in Y$ no problema dual (3.42).

 Resolva o SP (3.42).

se SP *for limitado* **então**

 Adicione ao PM um corte de otimalidade (3.51).

senão

 Adicione o corte de viabilidade (3.52).

fim

fim

$LS \leftarrow \min \{LS, SP^* + f^\top y\}$.

fim

O trabalho pioneiro de Geoffrion e Graves (1974) comprovou eficiência computacional da decomposição de Benders na resolução de algumas classes de problemas em larga escala.

3.2.1 Problema Mestre e SubProblema

Nesta seção é apresentado a decomposição da formulação original em dois subproblemas conforme o método de decomposição de Benders.

Ao fixar o valor das variáveis inteiras x e y e o das variáveis h e t no modelo original, resulta em um problema linear de fluxo que pode ser decomposto para cada produto i . Em outras palavras, seja $\mathcal{Z} = \{(x, y) \in \{0, 1\} : x \text{ e } y \text{ satisfazem as restrições (3.11) – (3.13)}\}$. Fixando as variáveis inteiras $x = x^h$ e $y = y^h$, tais que $(x^h, y^h) \in \mathcal{Z}$, e fixando a fração de fluxo inicial e final $h_{ip} = h_{ip}^h$ e $t_{ip} = t_{ip}^h$, obtém-se o seguinte subproblema primal de Benders.

$$\min \sum_i \sum_{j=1}^{(o_i-1)} \sum_p \sum_{\substack{r \\ r \neq p}} w_i \Phi_{pr} d_{pr} f_{ijpr} \quad (3.55)$$

sujeito a:

$$\sum_r f_{i1pr} = h_{ip}^h \quad \forall i, p \quad (3.56)$$

$$\sum_{\substack{r \\ r \neq p}} f_{ijrp} = \sum_{\substack{r \\ r \neq p}} f_{i(j+1)pr} \quad \forall i, j = 1, \dots, (o_i - 2), p \quad (3.57)$$

$$\sum_p f_{io_i pr} = t_{ir}^h \quad \forall i, r \quad (3.58)$$

$$f_{ijpr} \leq x_{s(i,j)p}^h \quad \forall i, j = 1, \dots, (o_i - 1), p, r : r \neq p \quad (3.59)$$

$$f_{ijpr} \leq x_{s(i,j+1)r}^h \quad \forall i, j = 1, \dots, (o_i - 1), p, r : r \neq p \quad (3.60)$$

$$f_{ijpr} \geq 0 \quad \forall i, j, p, r : r \neq p \quad (3.61)$$

Fixada as variáveis complicantes, o subproblema acima pode ser decomposto em

subproblemas independentes em função de cada produto i , dado por:

$$\min \sum_{j=1}^{(o_i-1)} \sum_p \sum_{\substack{r \\ r \neq p}} w_i \Phi_{pr} d_{pr} f_{ijpr} \quad (3.62)$$

sujeito a:

$$\sum_r f_{i1pr} = h_{ip}^h \quad \forall p \quad (3.63)$$

$$\sum_{\substack{r \\ r \neq p}} f_{ijrp} - \sum_{\substack{r \\ r \neq p}} f_{i(j+1)pr} = 0 \quad \forall j = 1, \dots, (o_i - 2), p \quad (3.64)$$

$$\sum_p f_{io_i pr} = t_{ir}^h \quad \forall r \quad (3.65)$$

$$f_{ijpr} \leq x_{s(i,j)p}^h \quad \forall j = 1, \dots, (o_i - 1), p, r : r \neq p \quad (3.66)$$

$$f_{ijpr} \leq x_{s(i,j+1)r}^h \quad \forall j = 1, \dots, (o_i - 1), p, r : r \neq p \quad (3.67)$$

$$f_{ijpr} \geq 0 \quad \forall j, p, r : r \neq p \quad (3.68)$$

Associando as variáveis duais α_{ijp} , μ_{ijpr} , ρ_{ijpr} , às restrições (3.63) – (3.65), (3.66) e (3.67) respectivamente, tem-se o seguinte problema dual para cada produto i .

$$\max \sum_p \left(h_{ip}^h \alpha_{i1p} + t_{ip}^h \alpha_{i(o_i-1)p} \right) - \sum_{j=1}^{(o_i-1)} \sum_p \sum_{\substack{r \\ r \neq p}} \left(x_{s(i,j)p}^h \mu_{ijpr} - x_{s(i,j+1)r}^h \rho_{ijpr} \right) \quad (3.69)$$

sujeito a:

$$\alpha_{i1p} + \alpha_{i1r} - \mu_{ijpr} - \rho_{ijpr} \leq w_i d_{pr} \Phi_{pr} \quad \forall p, r : r \neq p \quad (3.70)$$

$$\begin{aligned} -\alpha_{i(j-1)p} + \alpha_{ijr} - \mu_{ijpr} - \rho_{ijpr} &\leq w_i d_{pr} \Phi_{pr} & \forall j = 2, \dots, (o_i - 1), \\ & & p, r : r \neq p \end{aligned} \quad (3.71)$$

$$-\alpha_{i(o_i-1)p} + \alpha_{i(o_i-1)r} - \mu_{i(o_i-1)pr} - \rho_{i(o_i-1)pr} \leq w_i d_{pr} \Phi_{pr} \quad \forall p, r : r \neq p \quad (3.72)$$

$$\alpha_{ijp} \in \mathcal{R} \quad \forall i, j, p \quad (3.73)$$

$$\mu_{ijpr} \geq 0 \quad \forall i, j, p, r : r \neq p \quad (3.74)$$

$$\rho_{ijpr} \geq 0 \quad \forall i, j, p, r : r \neq p \quad (3.75)$$

Dado que o valor das variáveis duais tenham sido calculados em uma dada iteração h e com o auxílio da variável contínua η_i , pode-se construir a seguinte restrição conhecida como corte de otimalidade de Benders.

$$\eta_i \geq \sum_p \left(\alpha_{i1p}^h h_{ip} + \alpha_{i(o_i-1)p}^h t_{ip} \right) - \sum_{j=1}^{(o_i-1)} \sum_p \sum_{\substack{r \\ r \neq p}} \left(\mu_{ijpr}^h x_{s(i,j)p} - \rho_{ijpr}^h x_{s(i,j+1)r} \right) \quad (3.76)$$

No qual $(\alpha_{ijp}^h, \mu_{ijpr}^h, \rho_{ijpr}^h)$ é o ponto extremo correspondente à solução do SP em uma dada iteração h e η_i é uma variável contínua que subestima o valor da função objetivo (3.69) para cada produto $i \in I$.

Logo, o problema mestre é dado por:

$$\begin{aligned} \min \sum_i \left[\sum_p (w_i d_{0p} \Gamma_{0p} h_{ip} + w_i d_{pn} \Theta_{pn} t_{ip}) \right] + \\ \sum_m \left[\sum_p g_{mp} x_{mp} + \sum_{c=1}^{c_m} a_m y_{mc} \right] + \sum_i \eta_i \end{aligned} \quad (3.77)$$

sujeito a:

$$\sum_p h_{ip} = 1 \quad \forall i \quad (3.78)$$

$$\sum_r t_{ir} = 1 \quad \forall i \quad (3.79)$$

$$h_{ip} \leq x_{s_{(i,1)}p} \quad \forall i, p \quad (3.80)$$

$$t_{ir} \leq x_{s_{(i,o_i)}r} \quad \forall i, r \quad (3.81)$$

$$\sum_m x_{mp} \leq 1 \quad \forall p \quad (3.82)$$

$$\sum_p x_{mp} \leq 1 + \sum_{c=1}^{c_m} y_{mc} \quad \forall m \quad (3.83)$$

$$y_{m(c+1)} \leq y_{mc} \quad \forall m, \\ c = 1, \dots, (c_m - 1) \quad (3.84)$$

$$\sum_p x_{mp} \geq 1 \quad \forall m \quad (3.85)$$

$$\begin{aligned} \eta_i \geq \sum_p (\alpha_{i1p}^h h_{ip} + \alpha_{i(o_i-1)p}^h t_{ip}) \\ - \sum_{j=1}^{(o_i-1)} \sum_p \sum_{\substack{r \\ r \neq p}} \left(\mu_{ijpr}^h x_{s_{(i,j)}p} - \rho_{ijpr}^h x_{s_{(i,j+1)}r} \right) \quad \forall i, h = 1, \dots, H \end{aligned} \quad (3.86)$$

$$\eta_i \geq 0 \quad \forall i \quad (3.87)$$

$$x_{mp} \in \{0, 1\} \quad \forall m, p \quad (3.88)$$

$$y_{mc} \in \{0, 1\} \quad \forall m, c = 1 \dots c_m \quad (3.89)$$

$$h_{ip} \geq 0 \quad \forall i, p \quad (3.90)$$

$$t_{ip} \geq 0 \quad \forall i, p \quad (3.91)$$

no qual H é o número máximo de iterações do algoritmo. A restrição (3.85) é adicionada ao PM para assegurar que somente soluções viáveis sejam geradas. Essa restrição garante que ao menos uma máquina de cada tipo estará disponível.

3.2.2 Algoritmo de Decomposição de Benders

Sejam S_{PM}^* e S_{SP}^* a solução ótima do problema mestre e a solução ótima do subproblema respectivamente e considere o limite superior e o limite inferior denotado por LS e LI . O algoritmo de decomposição de Benders com múltiplos cortes é dado por:

Algoritmo 3: Algoritmo de decomposição de Benders Clássico

Inicialização: $LS \leftarrow +\infty$, $LI \leftarrow -\infty$.

enquanto $(LS - LI > \epsilon)$ **faça**

Resolva o PM (3.77) – (3.91).

$LI \leftarrow S_{PM}^*$ e atualize x_{mp} , h_{ip} e t_{ip} no problema dual (3.69) – (3.72).

Resolva o SP (3.69) – (3.72) $\forall i \in I$.

Adicione ao PM (3.77) – (3.91) cortes de otimalidade usando (3.76).

$LS \leftarrow \min \{LS, \sum_i S_{SP_i}^* + \sum_m \sum_p g_{mp} x_{mp} + \sum_m \sum_{c=1}^{c_m} a_m y_{mc} +$

$\sum_i \sum_p (w_i d_{0p} h_{ip} \Gamma_{0p} + w_i d_{pn} t_{ip} \Theta_{pn})\}$.

fim

em que ϵ é um valor pequeno, como por exemplo 0,01.

É importante ressaltar que ao adicionar a restrição (3.81) ao PM, toda solução gerada são soluções viáveis para o problema original. Dessa forma, somente cortes de otimalidade serão adicionados ao PM.

3.3 Geração de cortes Pareto Ótimo

Magnanti e Wong (1981) observaram quando o subproblema de Benders possui mais de uma solução ótima, ou seja, o problema é degenerado, vários cortes diferentes podem ser gerados. O método propõe selecionar e adicionar um único corte ao problema mestre que melhor subestime a todos. Esse corte é denominado corte Pareto-ótimo e a sua definição é dada por:

Definição 1. *Um corte é mencionado ser Pareto-ótimo se ele domina todos os outros cortes. Seja H , o conjunto de valores viáveis para as variáveis duais u . Então o corte de benders associado às variáveis $u^1 \in H$ domina o corte associado a $u^2 \in H$, se:*

$$(b - By)^\top u^1 \geq (b - By)^\top u^2. \quad (3.92)$$

Para construir o problema de geração de cortes pareto-ótimo, é utilizado o conceito de *core point* cuja definição é dada por:

Definição 2. *Um ponto y^0 é um core point se ele pertence ao interior relativo da casca convexa $ir(Z^c)$.*

Seja \bar{u} e $f(\bar{u})$, a solução ótima e o valor da função objetivo do SP, associado à solução \bar{y} do *PM*. O subproblema de *Magnanti e Wong (SMW)* é dado por:

$$\max (b - By^0)^\top u \quad (3.93)$$

$$\text{s. a: } (b - B\bar{y})^\top u = f(\bar{u}) \quad (3.94)$$

$$A^\top u \leq c \quad (3.95)$$

$$u \geq 0 \quad (3.96)$$

O algoritmo original de [Magnanti e Wong \(1981\)](#) resolve dois subproblemas a cada iteração do método de decomposição de Benders. Um *SP* associado à solução corrente do *PM* e do *SMW* e um outro *SP* associado a um dado *core-point* y^0 .

A dependência existente entre o *SMW* em relação à solução do *SP* representada pela restrição (3.94) torna o problema propenso a instabilidade numérica. Além dessa possível instabilidade, outro fator é relacionado à dificuldade em obter um novo *core point* a cada iteração, uma vez que encontrar um *core point* pode ser uma tarefa muito difícil ([Papadakos, 2008](#)).

[Papadakos \(2008\)](#) propõe um subproblema alternativo ao proposto por [Magnanti e Wong](#) para encontrar os cortes pareto-ótimo. Este problema, nomeado subproblema independente de [Magnanti e Wong \(SIMW\)](#), independe da solução do *SP*.

Seja $y^0 \in \text{ir}(Z^c)$ um core point válido, então o *SIMW* é dado por:

$$\max (b - By^0)^\top u \quad (3.97)$$

$$\text{s. a: } A^\top u \leq c \quad (3.98)$$

$$u \geq 0 \quad (3.99)$$

Além de propor um problema mais simples e eficaz para a geração de cortes pareto-ótimo, ao excluir a restrição (3.94), [Papadakos](#) apresenta ainda uma nova forma de atualização dos *core points* a cada iteração. Ele mostra que a combinação convexa entre um *core point* e o ponto inteiro viável referente a solução do *PM* também é um *core point* válido. Seja $y^h \in \mathbb{Z}$, a solução ótima do *PM* na iteração h , então, o core point dessa iteração pode ser atualizado pela combinação convexa entre o *core point* e a solução ótima do *PM*.

$$y_{h+1}^0 = \lambda y_h^0 + (1 - \lambda)y_h \quad (3.100)$$

Em que, $0 \leq \lambda \leq 1$. Portanto, dado um core point inicial, a atualização do *core point* a cada iteração é trivial.

3.3.1 Subproblema Independente de Magnanti-Wong - SIMW

A formulação para o SIMW decomposta para cada produto é dada por:

$$\max \sum_p (h_{ip}^0 \alpha_{i1p} + t_{ip}^0 \alpha_{i(o_i-1)p}) - \sum_{j=1}^{(o_i-1)} \sum_p \sum_{r \neq p}^r (x_{s(i,j)p}^0 \mu_{ijpr} - x_{s(i,j+1)r}^0 \rho_{ijpr}) \quad (3.101)$$

sujeito a:

$$\alpha_{i1p} + \alpha_{i1r} - \mu_{ijpr} - \rho_{ijpr} \leq w_i d_{pr} \Phi_{pr} \quad \forall p, r : r \neq p \quad (3.102)$$

$$\begin{aligned} -\alpha_{i(j-1)p} + \alpha_{ijr} - \mu_{ijpr} - \rho_{ijpr} &\leq w_i d_{pr} \Phi_{pr} & \forall j = 2, \dots, (o_i - 1), \\ & & p, r : r \neq p \end{aligned} \quad (3.103)$$

$$-\alpha_{i(o_i-1)p} + \alpha_{i(o_i-1)r} - \mu_{i(o_i-1)pr} - \rho_{i(o_i-1)pr} \leq w_i d_{pr} \Phi_{pr} \quad \forall p, r : r \neq p \quad (3.104)$$

Nos quais x^0 , h^0 e $t^0 \in \mathbf{ir}(Z^c)$ são *core points* válidos.

3.3.2 Algoritmo de Decomposição de Benders com Cortes Pareto-Ótimo

O algoritmo de decomposição de Benders com cortes pareto-ótimo é dado por:

Algoritmo 4: Algoritmo de decomposição de Benders com cortes Pareto Ótimo

Inicialização: $LS \leftarrow +\infty$, $LI \leftarrow -\infty$.

enquanto $(LS - LI > \epsilon)$ **faça**

 Resolva *SIMW* (3.101) – (3.104) $\forall i = 1, \dots, I$.

 Adicione ao PM cortes de otimalidade usando (3.101).

 Resolva o PM (3.77) – (3.91).

$LI \leftarrow S_{PM}^*$ e atualize x_{mp} , h_{ip} e t_{ip} no SP (3.69)– (3.72).

 Resolva o SP (3.69)– (3.72) $\forall i = 1, \dots, I$.

 Adicione ao PM cortes de otimalidade usando (3.76).

$LS \leftarrow \min \{LS, \sum_i S_{SP_i}^* + \sum_m \sum_p g_{mp} x_{mp} + \sum_m \sum_{c=1}^{c_m} a_m y_{mc} + \sum_i \sum_p (w_i d_{0p} h_{ip} \Gamma_{0p} + w_i d_{pn} t_{ip} \Theta_{pn})\}$.

 Atualize o *core point* usando (3.100).

fm

Para iniciar as iterações do algoritmo de decomposição de Benders com cortes pareto

ótimo é necessário determinar o *core point* inicial e o valor de λ . O *core point* inicial é calculado de acordo com cada problema e deverá, segundo a sua definição 2, pertencer ao interior relativo da casca convexa gerada pelas restrições do problema.

3.4 Pré processamento - *Hot Start*

McDaniel e Devine (1977) sugerem relaxar as restrições de integralidade das variáveis do problema mestre nas primeiras iterações do método de Benders. Dessa forma são gerados cortes com base em soluções de um problema linear, sem a necessidade de resolver um problema inteiro. Esse procedimento é conhecido como *hot start* ou pré-processamento.

O número de iterações de pré processamento é o único parâmetro a ser calibrado. Esse número pode ser determinado em função da convergência do limite inferior, que nesse caso, as iterações são realizadas até que não se perceba uma melhora entre as iterações. Sendo essa a estratégia adotada nesse trabalho.

Outra forma de determinar o critério de parada é na escolha de um valor inteiro k de forma empírica, tal que o problema mestre seja resolvido nas primeiras k iterações de forma relaxada.

Considere a notação cHS o número de ciclos de pré processamento. Segue abaixo o procedimento.

Algoritmo 5: Algoritmo de decomposição de Benders usando pré-processamento e cortes de Pareto Ótimo

Inicialização: $LI \leftarrow -\infty$, $x_{mp} \in \mathcal{R}^+ \forall m, p$;

enquanto $(LI^h - LI^{h-1} > 0.01)$ **faça**

Resolva *SIMW* (3.101) – (3.104) $\forall i = 1, \dots, I$.

Adicione ao PM um corte de otimalidade usando (3.76).

Resolva o PM (3.77) – (3.91).

$LI \leftarrow S_{PM}^*$ e atualize x_{mp} , h_{ip} e t_{ip} no SP (3.69)– (3.72).

Resolva o SP (3.69)– (3.72) $\forall i = 1, \dots, I$.

Adicione ao PM um corte de otimalidade (3.76).

Atualize o *core point* usando (3.100).

Incremente *cHS*.

fim

Faça: $x_{mp} \in \mathcal{Z}^+ \forall m, p$, $LS \leftarrow +\infty$.

enquanto $(LS - LI > \epsilon)$ **faça**

Resolva *SIMW* (3.101) – (3.104) $\forall i = 1, \dots, I$.

Adicione ao PM um corte de otimalidade usando (3.76).

Resolva o PM (3.77) – (3.91).

$LI \leftarrow S_{PM}^*$ e atualize x_{mp} , h_{ip} e t_{ip} no SP (3.69)– (3.72).

Resolva o SP (3.69)– (3.72) $\forall i = 1, \dots, I$.

Adicione ao PM um corte de otimalidade (3.76).

$LS \leftarrow \min \{LS, \sum_i S_{SP_i}^* + \sum_m \sum_p g_{mp} x_{mp} + \sum_m \sum_{c=1}^{c_m} a_m y_{mc} + \sum_i \sum_p (w_i d_{0p} h_{ip} \Gamma_{0p} + w_i d_{pn} t_{ip} \Theta_{pn})\}$

Atualize o *core point* usando (3.100).

fim

3.5 *Benders-and-Cut*

O problema mestre, por conter variáveis inteiras, necessita de um tempo superior para ser resolvido em relação à resolução do subproblema. Reduzir o tempo de resolução do problema mestre é uma estratégia que auxilia na performance geral do método.

Adicionar cortes de Benders ao longo da árvore de *branch-and-bound* é uma técnica que vem sendo utilizada com bons resultados, como observado nos trabalhos de Fortz e Poss (2009) e Naoum-Sawaya e Elhedhli (2010).

Os cortes serão adicionados conforme a melhor estratégia a ser definida. Esses cortes podem ser adicionados sempre que encontrar um nó viável, quando o limite inferior melhorar baseado em um valor pré-definido ou quando encontrar uma solução incumbente ao longo do processo de busca.

Capítulo 4

Resultados Computacionais

4.1 Gerador de instância

Os problemas testes foram baseados no trabalho de [Taghavi e Murat \(2011\)](#) por ser um trabalho correlacionado com o escopo do projeto. Porém, por ser uma formulação contínua, não há a informação referente ao número de posições disponíveis. Para tanto, o número de posições foi definido como sendo o número total de máquinas disponíveis.

A distância entre duas posições é calculada segundo a métrica de *Manhattan*. Posições adjacentes tanto vertical quanto horizontal dista uma unidade. A distância entre dois pontos $u = (x_1, y_1)$ e $v = (x_2, y_2)$, é dada por:

$$DM(u, v) = |x_2 - x_1| + |y_2 - y_1|$$

Dessa forma, a distância entre as posições 1 e 14 demonstrada na Figura [4.1b](#) é de 5 unidades.

Há duas variações em relação à posição do almoxarifado e da doca de expedição. Em uma situação, esses departamentos estão localizados em posição oposta entre si conforme Figura [4.1a](#) que foi definido como leiaute tipo I. Já em outra, leiaute tipo II conforme ilustrado na Figura [4.1b](#), o almoxarifado e a doca estão localizados em um mesmo lado da fábrica. Essas variações irão interferir tanto na decisão da localização dos recursos quanto no roteamento do fluxo.

O número de produtos e a sequência tecnológica são os mesmos utilizados em [Taghavi e Murat \(2011\)](#). Os produtos poderão assumir três níveis de demanda, baixa, média e alta. O nível baixo é dado por uma distribuição uniforme que possui um mínimo de 100 e máximo de 300 unidades, *i.e.* $U(100; 300)$. Já o nível médio é dado por $U(300; 600)$ e o nível de demanda alto é dado por $U(600; 1.800)$ unidades.

Foi considerado duas faixas de valores para o custo de duplicação das máquinas, tal

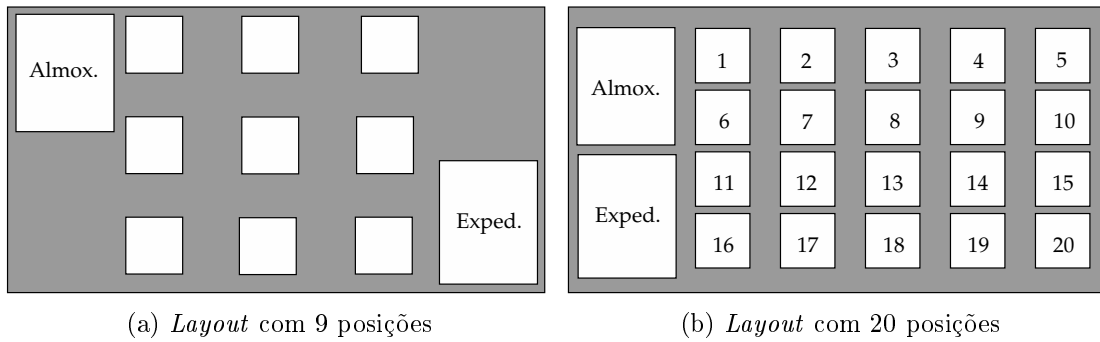
(a) *Layout* com 9 posições(b) *Layout* com 20 posições

Figura 4.1: Dimensões de leiaute

que o custo denominado baixo é dado pela distribuição $U(1.000; 5.000)$ e o custo alto, pode assumir qualquer valor entre 10.000 e 20.000 unidades monetárias. O número de tipos de máquinas está entre 5 e 20 e o número de cópias de cada uma variou entre 2 e 3 cópias. O custo de instalação de uma dada máquina em uma posição específica é dado pelo parâmetro g_{mp} e esse valor é obtido por meio de uma distribuição uniforme que representa 3 % do valor de aquisição da máquina. A Tabela 4.1 resume as instâncias consideradas.

Tabela 4.1: Resumo dos parâmetros utilizados

Prob. num.	Sites P	Layout Tipo	Máq. M	Cópias c_m	Custo a_m	Sequencia Tecnologica								
						P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9
1	5×2	I	5	2	Alto	1-2-3-5	1-3-4-5	2-4-5	2-3-5					
2					Baixo									
3		II			Alto									
4					Baixo									
5	5×4	I	10	2	Alto	1-2-5-7-10	1-4-6-7-8	2-3-4-5-9-10	3-7-8-9	1-5-7-9				
6					Baixo									
7		II			Alto									
8					Baixo									
9	9×5	I	15	3	Alto	1-4-9-10-13	3-7-11-12	3-6-12-14	1-2-5-15	3-4-8-12	3-6-9	2-10-12		
10					Baixo									
11		II			Alto									
12					Baixo									
13	6×10	I	20	3	Alto	2-6-11	4-7-8-14-7	3-5-7-10-12-13	8-9-13-17-18-19	2-8-12-18	1-5-7-9-15-1	8-11-16-19-20	4-6-9-10-12-19	1-2-14-16
14					Baixo									
15		II			Alto									
16					Baixo									

4.2 Resultados

O desempenho computacional do método de decomposição de Benders Clássico e suas variantes é apresentado a seguir. Os algoritmos foram implementados na linguagem C/C^{++} usando o *ILOG CPLEX 12.1 Concert Technology*. Todos os testes foram feitos usando um computador Intel(R) Core(TM) 2 Quad CPU Q8400 2.66Ghz com 4GB de memória RAM e sistema operacional Linux. O tempo máximo de processamento para cada teste foi limitado em 1 hora (3600s).

Considere as seguintes notações que identifica os procedimentos inseridos em cada variante do método de decomposição de Benders clássico:

- BD-C - Método de decomposição de Benders Clássico;
- BD-CP - Método de decomposição de Benders usando corte de Pareto;
- BD-CP-HS - Método de decomposição de Benders usando corte de Pareto e pré-processamento (*hot start*);
- BD-CP-HS-BC - Método de decomposição de Benders usando corte de Pareto, pré-processamento (*hot start*) e *Benders-and-Cut*;

Foram realizados quatro conjunto de testes. O primeiro avalia o resultado da formulação integrada apresentada na Seção 3.1 frente à formulação que realiza o roteamento do fluxo de forma desagregada. O segundo conjunto tem por objetivo verificar a qualidade da relaxação linear da formulação proposta. A evolução da performance de cada variante implementada é comprovada pelo terceiro conjunto de teste. E o último conjunto foi realizado com o objetivo de comparar a performance do método frente ao *software* CPLEX.

4.2.1 Modelo atribuição versus integrado

Esse conjunto de teste foi realizado com o objetivo de verificar a diferença da solução obtida ao resolver o problema de roteamento de fluxo juntamente com o problema de atribuição de recursos e ao resolver o problema de atribuição e posteriormente o problema de roteamento.

Foi criado um segundo conjunto de instâncias, instâncias a - h, que foi baseado nas primeiras 8 instâncias que estão resumidas na Tabela 4.1. A diferença desse novo conjunto de instância está na sequencia tecnológica dos produtos, em que esta sequencia foi alterada de tal forma que as máquinas finais e iniciais são sempre diferentes para cada produto. Outra alteração é em relação ao custo de aquisição das máquinas, que foi considerado custo zero de aquisição para cada máquina e a terceira e última alteração

está relacionada à distância entre posições. O valor da distância relativa entre posições foi aumentada.

A primeira coluna da Tabela 4.2 informa a instância, a coluna FO-1 informa o valor da função objetivo obtido conforme algoritmo 1 descrito na Seção 3.1.2, já a coluna FO-2 informa o valor da função objetivo obtida pela formulação integrada e a última coluna informa o valor relativo da função objetivo duas formulações que é dada por FO-1/FO-2.

Tabela 4.2: Resultados computacionais do modelo atribuição versus integrado

INST.	FO-1	FO-2	GAP REL.
a	96911	95704	1.013
b	93176	92766	1.004
c	80154	79153	1.013
d	84775	84411	1.004
e	75729	75508	1.003
f	459772	459286	1.001
g	1298465	1296230	1.002
h	236751	236605	1.001

Na coluna GAP REL., pode-se observar que o modelo integrado gera soluções inferiores comparado ao modelo desagregado.

4.2.2 Relaxação linear

O método de decomposição de Benders é aplicável em formulações que possuam uma estrutura especial. Além dessa estrutura, a formulação também deverá possuir uma boa relaxação linear.

Conforme observado na Tabela 4.3, a formulação possui uma boa relaxação linear, apresentando um GAP de otimalidade de até 10% para o problema teste 12. Para as instâncias 13-16, o tempo limite de processamento foi ultrapassado representado por um *, inviabilizando o cômputo do GAP de otimalidade.

4.2.3 Evolução dos métodos

O terceiro conjunto de teste foi realizado para verificar a evolução dos procedimentos adicionados ao método de decomposição de Benders Clássico. A Tabela 4.4 apresenta o limite inferior (LB), limite superior (UB), GAP de otimalidade (GAP), número de cortes de otimalidade inseridos no problema mestre (CORTES), o número de ciclos ou iterações do algoritmo de decomposição de Benders (CICLOS) e o tempo necessário para encontrar a solução ótima dado em segundos (TEMPO). Pela Tabela 4.4 pode-se observar que a partir da instância 5, o método clássico de Benders não foi capaz de resolver os problemas em um tempo inferior ao tempo limite.

Tabela 4.3: Relaxação linear da formulação proposta

INST.	SOL. ÓTIMA	SOL. RELAXADO	GAP [%]	TEMPO [s]
1	20999	20999	0	0.02
2	20296	20109	0.92	0.01
3	18471	18456	0.08	0.01
4	18490	18490	0.00	0.01
5	35551	34620	2.62	0.52
6	62365	59352	4.83	0.39
7	60042	56114	6.54	0.94
8	66970	63849	4.66	0.67
9	63680	60868	4.41	12.09
10	67588	63924	5.42	14.67
11	93881	85083	9.37	14.61
12	70016	63187	9.75	11.57
13	-	83228	*	314.65
14	-	86212	*	243.12
15	-	99459	*	236.01
16	-	93017	*	162.24

Tabela 4.4: Resultados computacionais do método Decomposição de Benders Clássico

INST.	LB	UB	GAP [%]	CORTES	CICLOS	TEMPO [S]
1	20999	20999	0.00	132	34	5.17
2	20296	20296	0.00	210	54	23.68
3	18471	18471	0.00	11	6	0.06
4	18490	18490	0.00	45	13	0.57
5	22935	40596	43.50	855	172	*
6	38648	70048	44.82	685	138	*
7	42974	62852	31.62	680	137	*
8	47511	73258	35.14	619	125	*
9	44514	79936	44.31	469	68	*
10	44702	88587	49.53	273	40	*
11	60885	118882	48.78	315	46	*
12	46229	85453	45.90	371	54	*
13	55664	123824	55.04	288	33	*
14	57732	151211	61.82	522	59	*
15	60490	149112	59.43	540	61	*
16	56738	139676	59.37	639	72	*

A primeira variante do método implementado a ser analisada é a BD-CP. A Tabela 4.5 resume os resultados obtidos dessa variante que foi descrita anteriormente na Seção 3.3.2. Pode-se observar uma melhora em relação ao BD-C, sendo que essa variante foi capaz de resolver até a instância 8. Tanto o número de iterações quanto o número de cortes adicionados também foram reduzidos. Analisando o problema 4, a variante DB-C inseriu 45 cortes de otimalidade e foram necessárias 13 iterações para alcançar o valor ótimo. Já a variante BD-CP adicionou apenas 6 cortes ao problema mestre que necessitou de 3 iterações, além de resolver em um tempo inferior à variante BD-C.

Ao utilizar a técnica de pré-processamento, pode-se observar pela Tabela 4.6 uma redução no tempo de execução para a maior parte dos problemas resolvidos anteriormente, comprovando um melhor desempenho do algoritmo.

A variante BD-CP-HS-BC foi capaz de resolver as mesmas instâncias das variantes

Tabela 4.5: Resultados computacionais do método Decomposição de Benders com o uso de corte de Pareto Ótimo

INST.	LB	UB	GAP [%]	CORTES		CICLOS		TEMPO [S]
				PP.	INT.	PP.	INT.	
1	20999	20999	0.00	35	0	8	1	0.2
2	20296	20296	0.00	66	37	11	1	0.31
3	18471	18471	0.00	6	9	4	1	0.02
4	18490	18490	0.00	6	0	3	1	0.04
5	35551	35551	0.00	289	56	31	3	65.25
6	62365	62365	0.00	361	59	37	1	359.57
7	60042	60042	0.00	481	351	51	26	200.34
8	66970	66970	0.00	338	162	36	6	46.81
9	63028	67144	6.13	522	198	39	15	*
10	65313	76296	14.39	445	1341	33	2	*
11	88875	96125	7.54	516	152	38	111	*
12	65927	73983	10.88	495	107	37	100	*
13	82032	117013	29.89	486	193	28	3	*
14	84868	124654	31.97	472	184	27	86	*
15	96078	125252	23.29	397	120	23	16	*
16	90178	124227	27.40	393	311	23	22	*

Tabela 4.6: Resultados computacionais do método Decomposição de Benders com o uso de corte de Pareto Ótimo e Hot Start

INST.	LB	UB	GAP [%]	CORTES		CICLOS		TEMPO [S]
				PP.	INT.	PP.	INT.	
1	20999	20999	0.00	92	0	18	1	0.15
2	20296	20296	0.00	167	37	32	1	0.37
3	18471	18471	0.00	4	9	6	1	0.04
4	18490	18490	0.00	17	0	7	1	0.05
5	35551	35551	0.00	554	56	64	3	24.35
6	62365	62365	0.00	732	59	84	1	29.26
7	60042	60042	0.00	327	351	44	26	257.64
8	66970	66970	0.00	343	162	47	6	25.73
9	63631	63680	0.07	850	198	70	15	*
10	66564	80900	17.72	1341	152	111	2	*
11	89937	102415	12.18	1193	107	100	3	*
12	67421	76275	11.61	1016	184	86	3	*
13	82905	128039	35.25	199	120	16	4	*
14	87763	126375	30.55	311	146	22	5	*
15	99763	135191	26.21	457	62	31	1	*
16	93856	140004	32.96	420	83	28	2	*

BD-CP-HS e BD-CP em um tempo inferior. Observa-se na Tabela 4.7 que essa variante obteve um melhor desempenho para a maior parte das instâncias que ultrapassaram o tempo limite, obtendo um GAP inferior, exceto para as instâncias 11, 13 e 14.

A diferença do desempenho computacional de cada variante pode ser justificada ao analisar a proporção do tempo gasto para resolver o PM e o SP conforme ilustrada pelas Figuras 4.2, 4.3, 4.4, 4.5. A proporção de tempo gasto para resolver o PM foi reduzida gradualmente à medida que outros procedimentos foram adicionados ao método de decomposição Benders Clássico.

Tabela 4.7: Resultados computacionais do método Decomposição de Benders com o uso de corte de Pareto Ótimo, Hot Start e *Benders-and-Cut*

INST.	LB	UB	GAP [%]	CORTES		CICLOS		TEMPO [S]
				PP.	INT.	PP.	INT.	
1	20999	20999	0.00	90	2	16	1	0.11
2	20296	20296	0.00	120	10	19	1	0.28
3	18471	18471	0.00	4	7	5	1	0.04
4	18490	18490	0.00	16	1	5	1	0.04
5	35551	35551	0.00	341	95	39	4	24.48
6	62365	62365	0.00	419	28	48	1	25.71
7	60042	60042	0.00	195	436	26	10	110.03
8	66970	66970	0.00	251	126	31	4	20.84
9	63483	63786	0.48	525	384	44	9	*
10	66636	73602	9.46	745	209	60	3	*
11	90333	103359	12.6	669	170	57	2	*
12	67524	71342	5.35	618	203	51	2	*
13	82243	142797	42.41	125	40	11	1	*
14	86486	135898	36.36	187	44	14	1	*
15	92293	109520	15.73	227	48	17	1	*
16	92910	126604	26.61	152	40	12	1	*

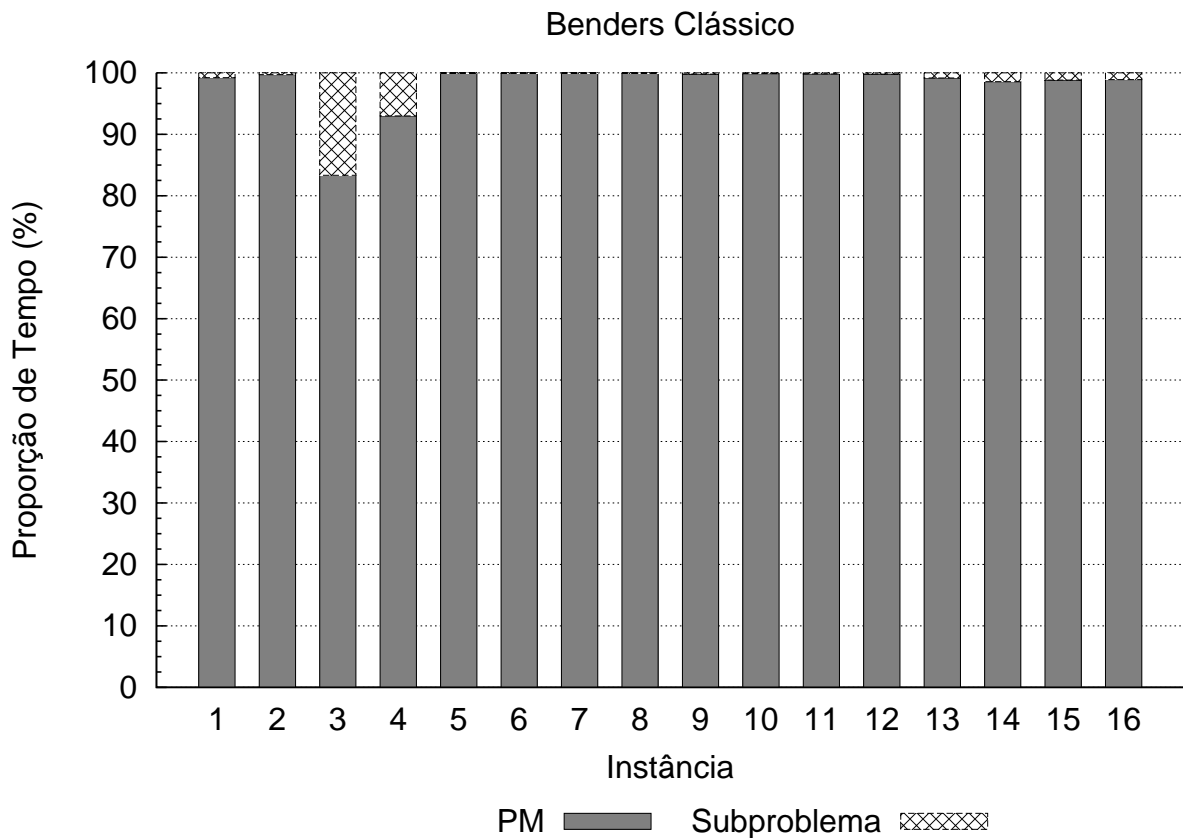


Figura 4.2: Proporção do tempo gasto pela variante BD-C para executar o PM e o SP

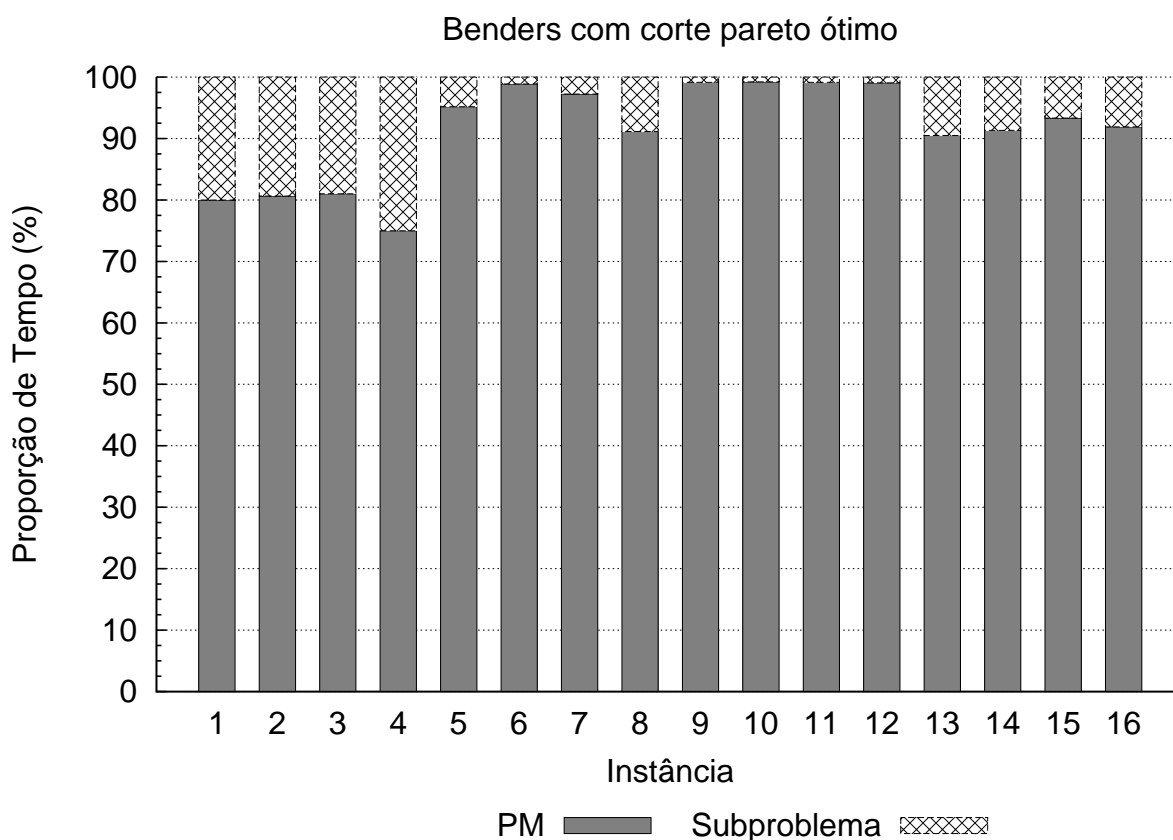


Figura 4.3: Proporção do tempo gasto pela variante BD-CP para executar o PM e o SP

4.2.4 CPLEX versus Decomposição de Benders

O próximo teste foi realizado com objetivo de verificar a performance do algoritmo de Benders frente ao *software* CPLEX. Pela Tabela 4.8, pode-se observar que o CPLEX obteve vantagem em todos os problemas testes. A partir do problema 9, a variante BD-CP-HS-BC não foi capaz de encontrar a solução ótima dentro do tempo limite estabelecido. Para esses problemas, foi registrado o *GAP* de otimalidade, que no caso do problema 9 foi de 0,5%.

Já o CPLEX resolveu quase todos os problemas dentro do tempo limite. O *GAP* de otimalidade também foi registrado para os problemas não resolvidos, ficando esse sempre inferior à variante BD-CP-HS-BC.

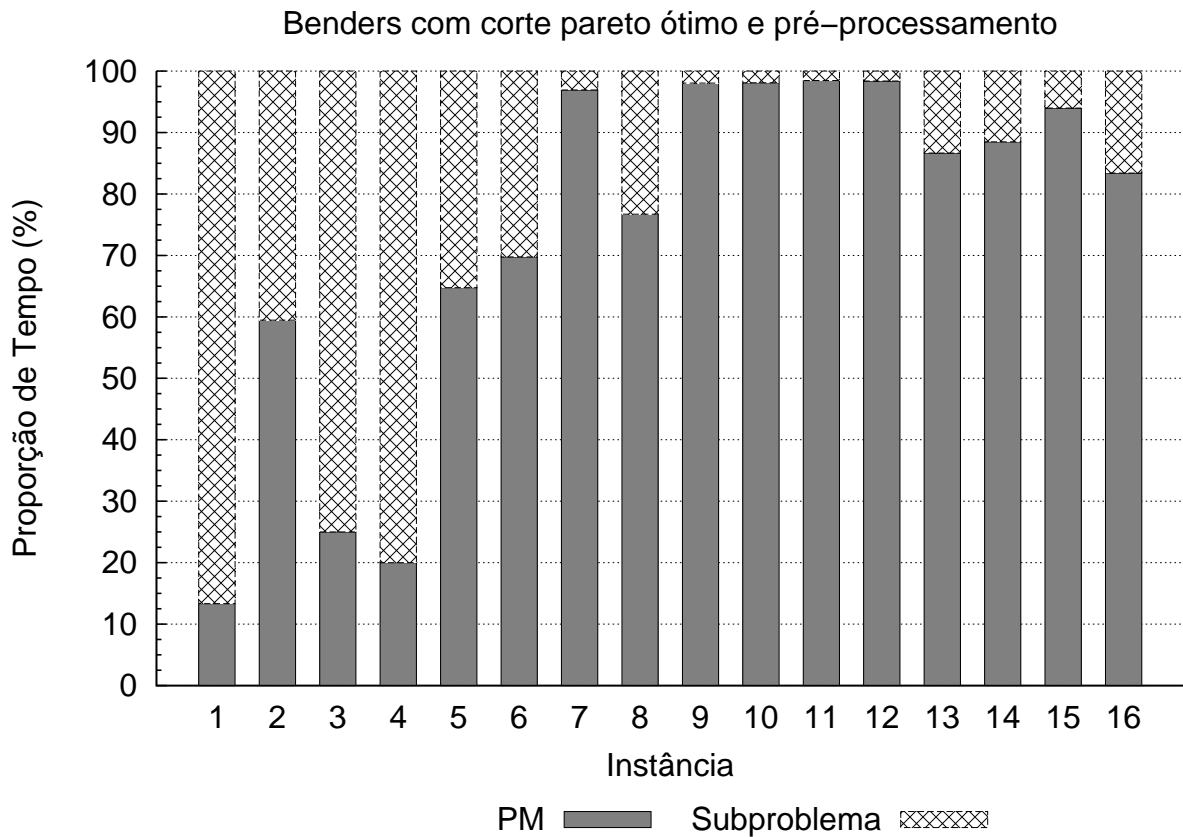


Figura 4.4: Proporção do tempo gasto pela variante BD-CP-HS para executar o PM e o SP

Tabela 4.8: Resultados computacionais do *software* CPLEX versus variante BD-CP-HS-BC

INST.	CPLEX				BENDERS - BD-CP-HS-BC			
	LB	UB	GAP[%]	TEMPO[S]	LB	UB	GAP[%]	TEMPO[S]
1	20999	20999	0.00	0.01	20999	20999	0.00	0.11
2	20296	20296	0.00	0.02	20296	20296	0.00	0.28
3	18471	18471	0.00	0.12	18471	18471	0.00	0.04
4	18490	18490	0.00	0.25	18490	18490	0.00	0.04
5	35551	35551	0.00	5.26	35551	35551	0.00	24.48
6	62365	62365	0.00	8.27	62365	62365	0.00	25.71
7	60042	60042	0.00	26.18	60042	60042	0.00	110.03
8	66970	66970	0.00	8.30	66970	66970	0.00	20.84
9	63680	63680	0.00	283.56	63483	63786	0.47	*
10	67588	67588	0.00	618.87	66636	73602	9.46	*
11	93881	93881	0.00	686.46	90333	103359	12.60	*
12	70016	70016	0.00	1660.25	67524	71342	5.35	*
13	83228	98843	15.89	*	82243	142797	42.40	*
14	86212	98079	12.17	*	86486	135898	36.35	*
15	99459	106787	6.94	*	92294	109520	15.72	*
16	93018	101087	8.09	*	92910	126604	26.61	*

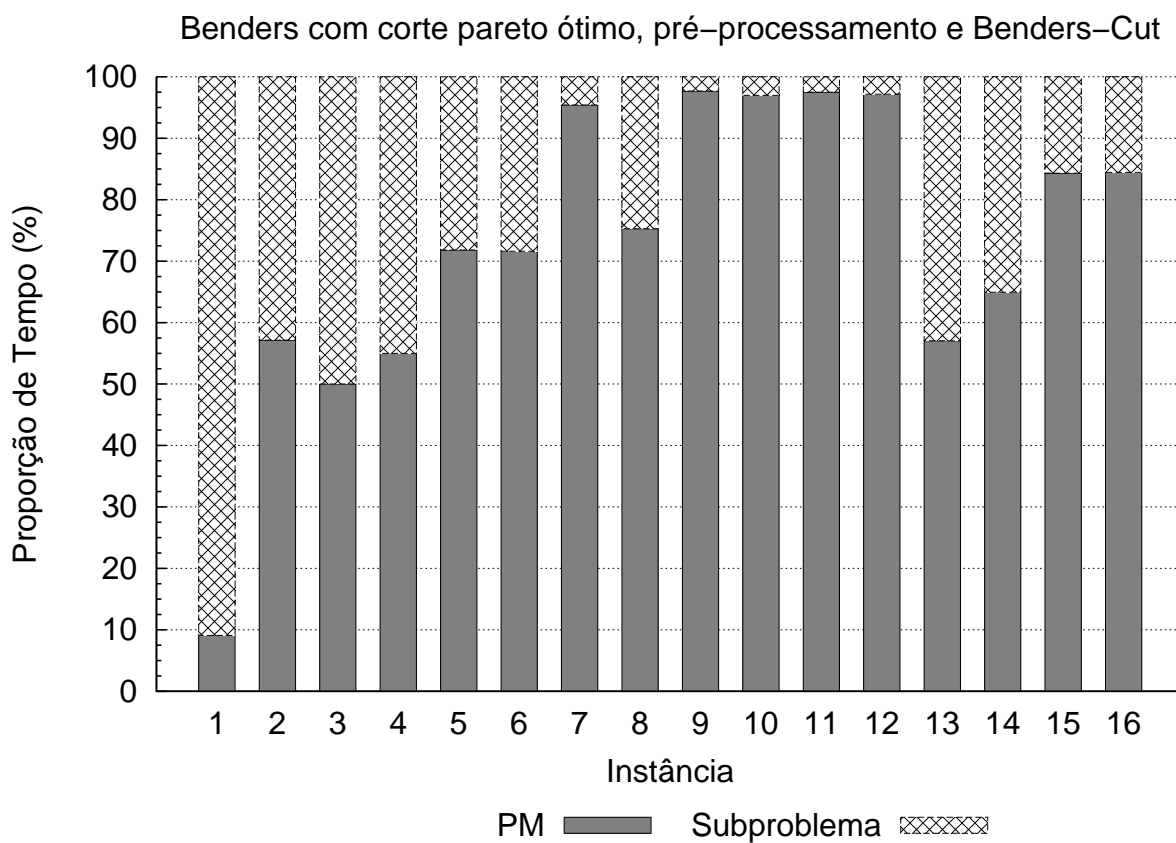


Figura 4.5: Proporção do tempo gasto pela variante BD-CP-HS-BC para executar o PM e o SP

Capítulo 5

Conclusão

O objetivo central deste trabalho foi propor uma formulação linear inteira mista para resolver o problema de leiaute de facilidades, em especial o leiaute distribuído. Pelo primeiro conjunto de testes, foi observado que a formulação proposta gera resultados inferiores por resolver de forma integrada o problema de roteamento de fluxo e o problema de atribuição de recursos.

Além de possuir uma estrutura especial em sua formulação que é necessária na implementação do método de decomposição de Benders, diante do segundo conjunto dos testes computacionais foi observado que a formulação proposta possui uma boa relaxação linear.

O uso do método de decomposição de Benders Clássico possui uma convergência muito lenta, fato observado no terceiro conjunto de testes. Observou-se neste mesmo conjunto de testes que a inserção dos procedimentos corte de Pareto Ótimo, pré-processamento e *Benders-and-Cut* auxiliou o método a encontrar soluções em um tempo inferior, além de ser capaz de resolver problemas maiores.

Porém, o método implementado apresentou uma pior performance comparado ao *software* CPLEX, observado no último conjunto de teste. Essa realidade despertou uma motivação para continuar a pesquisar outros procedimentos que poderão ser adicionados ao método clássico de Benders, uma vez que esses resultados são contra intuitivos.

Além da inserção de novos procedimentos ao método de Benders, outras perspectivas de trabalhos futuros relacionados com o presente trabalho poderão ser realizados. Em relação ao modelo, este pode se tornar capacitado mediante a inserção de uma restrição que limita o fluxo de acordo com a capacidade da máquina instalada. O modelo também pode ser transformado em um modelo que resolve o problema de leiaute dinâmico por meio da inserção de uma dimensão nas variáveis de atribuição para representar os diferentes períodos de planejamento.

Referências Bibliográficas

- Afentakis, P.; Millen, R. A. e Solomon, M. N. (1990). Dynamic layout strategies for flexible manufacturing systems. *International Journal of Production Research*, 28:311–323.
- Alan, R.; McKendall, J. e H., A. (2010). Heuristics for the dynamic facility layout problem with unequal-area departments. *European Journal of Operational Research*, 201:171–182.
- Amine, D.; Henri, P. e Sonia, H. G. (2007). Facility layout problems: A survey. *Annual Reviews in Control*, 31:255–267.
- Bayakasoglu, A. e Göçken, M. (2010). Capability-based distributed layout and its simulation based analyses. *Journal of Intelligent Manufacturing*, 21:471–485.
- Benders, J. F. (1962). Partitioning procedures for solving mixed-variables programming problems. *Numerisch Mathematik*, 4:238–252.
- Benjaafar, S.; Heragu, S. S. e Irani, S. (2002). Next generation factory layouts: Research challenges and recent progress. *Interfaces*, 32(6):58–76.
- Benjaafar, S. e Sheikhzadeh, S. (2000). Design of flexible plant layouts. *IIE Transactions*, 32:309–322.
- Cooper, L. (1963). Location-allocation problems. *Operations Research*, 11(2):331–344.
- Fortz, B. e Poss, M. (2009). An improved benders decomposition applied to a multi-layer network design problem. *Operations Research Letters*, 37(5):359–364.
- Fu, M. e Kaku, B. (1997). Minimizing work-in-process and material handling in the facilities layout problem. *IIE Transactions*, 29:29–36.
- Geoffrion, A. M. e Graves, G. W. (1974). Multicommodity distribution system design by Benders decomposition. *Management Science*, 20:822–844.

- Irani, S.; Cavalier, T. e Cohen, P. (1993). Virtual manufacturing cells: Exploiting layout design and intercell flows for the machine sharing problem. *International Journal of Production Research*, 31(4):791–810.
- Irani, S. e Huang, H. (2006). Cascading flowlines and layout modules: Practical strategies for machine duplication in facility layouts. *International Journal of Flexible Manufacturing Systems*, 17:119–149.
- Koopmans, T. C. e Beckman, M. (1957). Assignment problems and the location of economic activities. *Econometrica*, 25:53–76.
- Lahmar, M. e Benjaafar, S. (2005). Design of distributed layouts. *IIE Transactions*, 37:303–318. Working paper, Mechanical Engineering Department, University of Minnesota.
- Lasdon, L. S. (1972). *Optimization Theory for Large Systems*. The Macmillan Company, New York.
- Lawler, E. (1963). The quadratic assignment problem. *Management Science*, 9.
- Magnanti, T. L. e Wong, R. T. (1981). Accelerating benders decomposition: Algorithmic enhancement and model selection criteria. *Operations Research*, 29(3):464–483.
- Marcotte, S.; Montreuil, B. e Lefrançois, P. (1998). Holographic layout design: Towards manufacturing agility. In *7th annual IERC, Banff, Alberta, Canada*.
- McDaniel, D. e Devine, M. (1977). A modified Benders partitioning algorithm for mixed integer programming. *Management Science*, 24(3):312–319.
- McLean, C. R.; Bloom, H. M. e Hopp, T. H. (1982). The virtual cell. In *Fourth IFAC/IFIP Conference of Information Control Problems in Manufacturing Technology*, p.105-111.
- Minoux, M. e Vajda, S. (1986). *Mathematical Programming: Theory and Algorithms*. John Wiley & Sons Ltd, Wiley-Interscience Series in Discrete Mathematics and Optimization.
- Mitrofanov, S. (1966). The scientific principles of group technology. National Landing Library Translation, Boston Spa, Yorkshire, UK.
- Montreuil, B. (1990). *A Modelling Framework for Integrating Layout Network Design*. Material Handling Research Colloquium. Hebron, KY: 1990.

- Montreuil, B.; Venkatadri, U. e Lefrançois, P. (1991). Holographic layout of manufacturing systems. Technical Report 91-76, Faculty of Management, Laval University, Québec, Canada.
- Naoum-Sawaya, J. e Elhedhli, S. (2010). An interior-point benders based branch-and-cut algorithm for mixed integer programs. *Annals of Operations Research*, pp. 1–23.
- Papadakos, N. (2008). Practical enhancements to the magnanti–wong method. *Operations Research Letters*, 36:444–449.
- Silva, J.; de CAMARGO, R. S. e MIRANDA, G. (2010). Desenho de células de manufatura minimizando os movimentos inter-celulares e considerando a sequencia tecnológica de produção. In *In: 42o SBPO, 2010, Bento Gonçalves. Anais do 42o SBPO, 2010*.
- Solimanpur, M. e Jafari, A. (2008). Optimal solution for the two-dimensional facility layout problem using a branch-and-bound algorithm. *Computers & Industrial Engineering*, 55:606–619.
- Taghavi, A. e Murat, A. (2011). A heuristic procedure for the integrated facility layout design and flow assignment problem. *Computers & Industrial Engineering*, pp. 55–63.
- Tompkins, J. A.; White, J. A.; Boyer, Y. A.; Frazelle, E. H.; Tanchoco, J. M. e Trevino, J. (1996). Facilities planning. New York: Wiley.