Tese apresentada à Pró-Reitoria de Pós-Graduação e Pesquisa do Instituto Tecnológico de Aeronáutica e da Universidade Federal de São Paulo, como parte dos requisitos para obtenção do título de Doutora em Ciências no Programa de Pós-Graduação em Pesquisa Operacional, Área de Engenharia de Produção/Pesquisa Operacional.

Evelyn Michelle Henrique Braga

MODELAGEM E RESOLUÇÃO DO PROBLEMA DE LAYOUT DE FACILIDADES ROBUSTO DE ÁREAS DESIGUAIS COM LOCAIS DE ENTRADA E SAÍDA

Tese aprovada em sua versão final pelos abaixo assinados:

Prof. Dr. Luiz Leduino de Salles Neto

Orientador

São José dos Campos, SP - Brasil
2022

Dados Internacionais de Catalogação-na-Publicação (CIP)
Divisão de Informação e Documentação

Braga, Evelyn Michelle Henrique
Modelagem e Resolução do Problema de Layout de Facilidades Robusto de Áreas Desiguais com Locais de Entrada e Saída / Evelyn Michelle Henrique Braga.
São José dos Campos, 2022.
119f.

Tese de Doutorado – Curso de Pesquisa Operacional. Área de Engenharia de Produção/Pesquisa Operacional – Instituto Tecnológico de Aeronáutica e Instituto de Ciência e Tecnologia da Universidade Federal de São Paulo, 2022. Orientador: Prof. Dr. Luiz Leduino de Salles Neto.

1. Planejamento e projeto de facilidades. 2. Problema de layout de facilidades de áreas desiguais. 3. Programação inteira mista. I. Instituto Tecnológico de Aeronáutica. II. Universidade Federal de São Paulo. III. Título.

REFERÊNCIA BIBLIOGRÁFICA

BRAGA, Evelyn Michelle Henrique. **Modelagem e Resolução do Problema de Layout de Facilidades Robusto de Áreas Desiguais com Locais de Entrada e Saída**. 2022. 119f. Tese de Doutorado – Instituto Tecnológico de Aeronáutica e Universidade Federal de São Paulo, São José dos Campos.

CESSÃO DE DIREITOS

Evelyn Michelle Henrique Braga
Rua Paulo Piedade Campos, 680, Ap. 402B
30.494-060 – Belo Horizonte–MG

MODELAGEM E RESOLUÇÃO DO PROBLEMA DE LAYOUT DE FACILIDADES ROBUSTO DE ÁREAS DESIGUAIS COM LOCAIS DE ENTRADA E SAÍDA

Evelyn Michelle Henrique Braga

Composição e Assinatura da Banca Examinadora:

Prof. Dr.	Antônio Augusto Chaves	Presidente	-	UNIFESP
Prof. Dr.	Luiz Leduino de Salles Neto	Orientador	-	UNIFESP
Prof. Dr.	Nei Yoshihiro Soma		-	ITA
Profª. Drª.	Kelly Cristina Poldi		-	UNICAMP
Profª. Drª.	Deisemara Ferreira		-	UFSCAR

ITA/UNIFESP

Dedico este trabalho a todos os pesquisadores brasileiros pela dedicação e determinação em prol da ciência.

Agradecimentos

Agradeço a Deus, por nunca me desamparar.

Ao meu amado esposo, por estar sempre do meu lado.

Aos meus familiares e amigos próximos, em especial à minha mãe e minha irmã, pelo grande apoio e incentivo.

Aos professores e colegas da Pós-Graduação em Pesquisa Operacional do ITA/UNIFESP por compartilharem comigo seus conhecimentos e experiências, principalmente ao meu orientador por me direcionar no caminho certo.

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

"Ninguém ignora tudo. Ninguém sabe tudo.
Todos nós sabemos alguma coisa. Todos nós ignoramos alguma coisa.
Por isso aprendemos sempre."
— Paulo Freire

Resumo

Uma das estratégias utilizadas para otimizar processos de produção é a definição do melhor arranjo físico. Neste sentido, muitas empresas realizam um estudo do posicionamento relativo dos seus diversos equipamentos, áreas ou atividades funcionais. A disposição adequada das instalações pode resultar em um menor tempo de processo e maior rendimento dos fatores de produção. Em geral, o objetivo do problema de layout de facilidades é reduzir o custo de manuseio de material, que pode ser representado por uma função que relaciona os fluxos de materiais e as distâncias entre as facilidades. O problema de layout de facilidades dinâmico considera um horizonte de planejamento multi-período no qual os fluxos de materiais entre pares de facilidades podem mudar com o tempo e pode ser resolvido com uma abordagem flexível ou robusta. Uma planta flexível é aquela capaz de aceitar alterações no posicionamento das instalações ao longo do tempo para acompanhar as mudanças de demanda desde que compense os custos de realocação, enquanto que uma planta robusta é uma solução de layout único que pode não ser excelente para nenhum dos cenários individualmente, mas busca ser a melhor quando se avalia o conjunto destes. Considerando um ambiente dinâmico de demanda, o presente trabalho propôs-se a realizar uma análise robusta, ao invés de flexível, por considerar que custos de rearranjo e interrupção da produção são altos, além de que uma abordagem adaptativa pode ser demasiadamente inconveniente na rotina de uma indústria. Dentro do problema de layout existem pontos pouco explorados que foram tratados nesta pesquisa, a saber, blocos de áreas desiguais, locais de entrada e saída nos blocos, facilidades fixas ou espaços ocupados e distância de folga em torno das facilidades. Nesse contexto foram desenvolvidos e validados novos modelos matemáticos de programação inteira mista para o problema de layout de facilidades, a partir da análise de modelos propostos na literatura. Como alternativa para a implementação em programas de modelagem e uso de *softwares* matemáticos, também foi efetuada a linearização desses modelos. A fim de oferecer soluções para os problemas modelados foram desenvolvidos e aplicados alguns métodos de resolução envolvendo programação matemática e heurísticas que obtiveram bons resultados para diferentes instâncias da literatura.

Abstract

One of the strategies used to optimize production processes is to define the best layout. In this sense, many companies do a study of the relative positioning of their various equipment, areas or functional activities. Proper arrangement of facilities can result in shorter process times and higher throughput performance. In general, the objective of the facility layout problem is to reduce the material handling cost, which can be represented by a function that relates material flows and distances between facilities. The dynamic facility layout problem considers a multi-period planning horizon in which material flows between pairs of facilities can change over time, and can be solved with a flexible or robust approach. A flexible plant is one able to accepting changes in the placement of facilities over time to keep up with changes in demand as long as it compensates for the relocation costs, whereas a robust plant is a single layout solution that may not be excellent for any of the scenarios individually, but seeks to be the best when evaluating all of them. Considering a dynamic demand environment, the present work proposed to perform a robust analysis, instead of flexible, considering that the costs of rearrangement and interruption of production are high, besides that an adaptive approach can be too inconvenient in the routine of an industry. Within the layout problem there are little explored points that were addressed in this research, such as blocks of unequal areas, places of entry and exit in the blocks, fixed facilities or unallowed areas and clearance distance around facilities. In this context, new mathematical models of mixed integer programming were developed and validated for the facility layout problem, based on the analysis of models proposed in the literature. As an alternative for the implementation in modeling programs and use of mathematical *software*, the linearization of these models was also performed. In order to offer solutions to the modeled problems, some resolution methods involving mathematical programming and heuristics were developed and applied, which obtained good results for different instances of literature.

Lista de Figuras

FIGURA 1.1 – Layout de uma cozinha da rede de *fast-food* McDonald's (RAGOV, 2018). 23

FIGURA 2.1 – Exemplo de configuração para o FLP clássico. 27

FIGURA 2.2 – Representação das distâncias Euclideana (d = D) e retilínea (d = H + V). 28

FIGURA 2.3 – Exemplo de arranjo físico de facilidades considerando layout flexível e robusto. 29

FIGURA 2.4 – Departamentos de áreas desiguais (INÁCIO; ERTHAL, 2017). 33

FIGURA 2.5 – Equipamentos de áreas desiguais (NILSON, 2017). 33

FIGURA 2.6 – Diferentes configurações de layouts em função do sistema de produção (HOSSEINI-NASAB *et al.*, 2018). 34

FIGURA 2.7 – Exemplo de facilidades com I/O *points* fixos (MELO, 2017). 37

FIGURA 2.8 – Exemplo de facilidades com I/O *points* variáveis (BRANSKI, 2008). . 37

FIGURA 2.9 – Classificação dos problemas de layout (HOSSEINI-NASAB *et al.*, 2018). 41

FIGURA 3.1 – Facilidades com locais candidatos aos I/O *points*: a) quatro possibilidades localizadas no meio das bordas do bloco; b) quatro opções localizadas nos cantos do bloco; e c) podem assumir infinitas posições ao redor do bloco nas bordas. 44

FIGURA 3.2 – Configurações de blocos. 47

FIGURA 3.3 – *Grid* com caminhos de fluxo viáveis para um problema com três facilidades e marcação da distância de contorno $d_{1,3}$ entre a saída do bloco 1 e a entrada do bloco 3. 48

FIGURA 3.4 – Facilidade de dimensões variáveis com I/O *point* no centroide. . . . 50

FIGURA 3.5 – Plantas iniciais para instâncias com a) 12 b) 15 c) 20 e d) 30 facilidades (TAM, 1992). 54

FIGURA 3.6 – Projeto final de layout para 12 facilidades sem distância de folga e com distância de folga. 55

FIGURA 3.7 – Layouts ótimos encontrados para 4 facilidades sem distância de folga e com distância de folga. 56

FIGURA 4.1 – Exemplo do processo de resolução do Algoritmo 1. 63

FIGURA 4.2 – Árvore binária. 67

FIGURA 4.3 – Processo de geração do layout por particionamento a partir da árvore binária. 67

FIGURA 4.4 – Processo 1 de construção da árvore binária a partir da matriz de tráfego. 69

FIGURA 4.5 – Árvores binárias: a) Processo 2 - Distribuição sequencial; b) Processo 3 - Distribuição equilibrada. 69

FIGURA 4.6 – Processo de particionamento em um layout com espaços ocupados. . 71

FIGURA 4.7 – Formação de layout do Algoritmo 4 por meio de uma heurística construtiva e do *solver* CPLEX (Instância Das93N4). 76

FIGURA 5.1 – Melhores layouts encontrados: a) 12 facilidades - FO = 5279,54; b) 15 facilidades - FO = 9721,07; c) 20 facilidades - FO = 21937,20; d) 30 facilidades - FO = 55742,40. 81

FIGURA 5.2 – Layouts ótimos encontrados para I/O *points* fixos. 84

FIGURA 5.3 – Soluções para problemas com I/O *points* fixos. 84

FIGURA 5.4 – Layouts ótimos encontrados para I/O *points* variáveis. 86

FIGURA 5.5 – Soluções encontradas com o Método PM-MNL para problemas com I/O *points* variavéis - Instância: WelN12. 86

FIGURA 5.6 – Soluções encontradas com o Método PM-ML para problemas com I/O *points* variavéis, Modelo UAFLP1. 87

FIGURA 5.7 – Soluções encontradas com o Método PM-ML para problemas com I/O *points* variavéis, Modelo UAFLP2. 87

FIGURA 5.8 – Soluções encontradas com o Método PM-ML para problemas com I/O *points* variavéis, Modelo UAFLP3. 88

FIGURA 5.9 – Layouts encontrados para o Modelo UAFLP2 - Instância: DunN62. 88

Lista de Tabelas

TABELA 2.1 – Notações usadas no modelo FLP 27

TABELA 2.2 – Notações usadas no modelo DFLP flexível 30

TABELA 2.3 – Notações usadas no modelo DFLP robusto 32

TABELA 3.1 – Notações usadas nos Modelos 1, 2, 3 e 4 45

TABELA 3.2 – Notações usadas no modelo para facilidades com dimensões variáveis. 50

TABELA 3.3 – Notações acrescentadas aos modelos anteriores para incluir um conjunto de facilidades fixas ou restrições geométricas. 53

TABELA 3.4 – Notações adicionais usadas nos Modelos 1, 2, 3 e 4 linearizados . . . 56

TABELA 4.1 – Comparação entre os Algoritmos 1 e 2. Instância Deb05N12 (DEB; BHATTACHARYYA, 2005) e t = 10h. 65

TABELA 5.1 – Valores médios de função objetivo e de violações das permutações de entrada e de saída da Fase 1 do algoritmo, taxa de melhoria com a Busca Local e tempo médio de execução em segundos para cada instância. 79

TABELA 5.2 – Resultados da Fase 1 por processo construtivo da árvore de base para cada instância. 80

TABELA 5.3 – Resultados da Fase 2 por processo construtivo da árvore de base e média do tempo de execução total das duas fases do algoritmo para cada instância. 80

TABELA 5.4 – Identificação das instâncias retiradas da literatura. 81

TABELA 5.5 – Resultados do Método BLAR-PM para cada instância testada. . . . 83

TABELA 5.6 – Resultados para I/O *points* fixos. 83

TABELA 5.7 – Resultados para I/O *points* variáveis 85

TABELA A.1 –Número total de facilidades e dimensões do espaço disponível (Tabela válida para Tam92N12, Tam92N15, Tam92N20 e Tam92N30). . 102

TABELA A.2 –Área e limites da razão de aspecto das facilidades (Tabela válida para Tam92N12, Tam92N15, Tam92N20 e Tam92N30). 103

TABELA A.3 –Posição das facilidades fixas no espaço inicial (Tam92N12). 104

TABELA A.4 –Matriz de fluxos entre facilidades (Tam92N12). 104

TABELA A.5 –Posição das facilidades fixas no espaço inicial (Tam92N15). 104

TABELA A.6 –Matriz de fluxos entre facilidades (Tam92N15). 105

TABELA A.7 –Posição das facilidades fixas no espaço inicial (Tam92N20). 105

TABELA A.8 –Matriz de fluxos entre facilidades (Tam92N20). 106

TABELA A.9 –Posição das facilidades fixas no espaço inicial (Tam92N30). 106

TABELA A.10 –Matriz de fluxos entre facilidades (Tam92N30). 107

TABELA A.11 –Características das facilidades (Das93N4). 108

TABELA A.12 –Matriz de fluxos entre facilidades (Das93N4). 108

TABELA A.13 –Características das facilidades (Das93N6). 108

TABELA A.14 –Matriz de fluxos entre facilidades (Das93N6). 109

TABELA A.15 –Características das facilidades (Das93N8). 109

TABELA A.16 –Matriz de fluxos entre facilidades (Das93N8). 109

TABELA A.17 –Características das facilidades (Das93N10). 110

TABELA A.18 –Matriz de fluxos entre facilidades (Das93N10). 110

TABELA A.19 –Características das facilidades (Das93N12). 111

TABELA A.20 –Matriz de fluxos entre facilidades (Das93N12). 111

TABELA A.21 –Características das facilidades (Wel93N6). 112

TABELA A.22 –Matriz de fluxos entre facilidades (Wel93N6). 112

TABELA A.23 –Características das facilidades (Wel93N12). 113

TABELA A.24 –Matriz de fluxos entre facilidades (Wel93N12). 113

TABELA A.25 –Número total de facilidades e dimensões do espaço disponível (Tabela válida para Deb05N12 e Deb05N18). 114

TABELA A.26 –Características das facilidades (Tabela válida para Deb05N12 e Deb05N18).114

TABELA A.27 –Matriz de fluxos entre facilidades (Tabela válida para Deb05N12 e Deb05N18). 115

TABELA A.28 –Características das facilidades (Dun03N62). 117

TABELA A.29 –Matriz de fluxos entre facilidades (Dun03N62). 118

Lista de Abreviaturas e Siglas

ACO	*Ant Colony Optimization*
AGV	*Automatic Guided Vehicle*
AMPL	*A Mathematical Programming Language*
BSA	*Backtracking Search Algorithm*
BRKGA	*Biased Random-Key Genetic Algorithm*
CRO	*Coral Reefs Optimizatio*n
FLP	*Facility Layout Problem*
DFLP	*Dynamic Facility Layout Problem*
ESHGA	*Elitist Strategy Hybrid Genetic Algorithm*
FMS	*Flexible Manufacturing System*
FBS	*Flexible Bay Structures*
FO	Função Objetivo
GA	*Genetic Algorithm*
I/O	*Input/Output*
IMGA	*Island Model Genetic Algorithm*
MAIN	*Machines Allocation INter-relationship*
MHC	*Material Handling Cost*
MILO	*Mixed-Integer Linear Optimization*
MIP	*Mixed-Integer Programming*
NP-*hard*	*Non-deterministic Polynomial-time hard*
QAP	*Quadratic Assignment Problem*
PDM	*Perimeter Distance Metric*
PSO	*Particle Swarm Optimization*
RDM	*Rectilinear Distance Metric*
SFLP	*Static Facility Layout Problem*
SA	*Simulated Annealing*
ST	*Slicing Tree*
TS	*Tabu Search*
UAFLP	*Unequal Area Facility Layout Problem*

Lista de Símbolos

Conjuntos e Índices	
N	número total de facilidades/locais
i e j	índices para facilidades = (1,...,N)
r e s	índices para locais = (1,...,N)
T	número total de períodos
t	índice para períodos = (1,...,T)
NF	conjunto de facilidades fixas
Dados de Entrada	
d_{rs}	distância entre os locais r e s
f_{ij}	fluxo de peças entre as facilidades i e j
f_{ijt}	fluxo de peças entre as facilidades i e j no período t
cr_{it}	custo de realocar a facilidade i no início do período t
cf_t	custo fixo de realizar qualquer rearranjo no início do período t
F_{ij}	fluxo médio de peças entre as facilidades i e j
W	largura do espaço disponível
H	altura do espaço disponível
w_i	largura original da facilidade i
h_i	altura original da facilidade i
Ix_i, Iy_i	coordenadas x e y originais da entrada da facilidade i
Ox_i, Oy_i	coordenadas x e y originais da saída da facilidade i
$aisle$	largura mínima do corredor entre facilidades (*clearence distance*)
Δ	número de suportes afins
$\overline{x}_{ik}$	ponto de suporte afim k da facilidade i
Variáveis de Decisão	
x_{ir}	1 se a facilidade i está localizada no local r, 0 cc .
x_{irt}	1 se a facilidade i está localizada no local r no período t, 0 cc.
y_{it}	1 se a facilidade i é realocada no início do período t, 0 cc.
z_t	1 se qualquer rearranjo é feito no início do período t, 0 cc.
d_{ij}	distância entre as facilidades i e j
wr_i	largura real da facilidade i (considera a rotação)

hr_i	altura real da facilidade i (considera a rotação)
x_i, y_i	coordenadas x e y do canto inferior esquerdo da facilidade i
xs_i, ys_i	coordenadas x e y do canto superior direito da facilidade i
l_{ij}	(posição relativa) 1 se a facilidade i está localizada totalmente à esquerda de j, 0 cc.
b_{ij}	(posição relativa) 1 se a facilidade i está localizada totalmente abaixo de j, 0 cc.
u_i	0 se a facilidade i está na sua orientação original; 1 se a facilidade i está rotacionada 90°
x_i^I, y_i^I	coordenadas x e y da entrada da facilidade i
x_i^O, y_i^O	coordenadas x e y da saída da facilidade i
m_i^I, n_i^I	posição perimetral nas direções x e y da entrada da facilidade i
m_i^O, n_i^O	posição perimetral nas direções x e y da saída da facilidade i
(u_i, v_i)	(0,0) se a facilidade i está na sua orientação original; (1,0) se a facilidade i está rotacionada 90° no sentido horário; (0,1) se a facilidade i está rotacionada 180° no sentido horário; (1,1) se a facilidade i está rotacionada 270° no sentido horário
(p_i^I, q_i^I)	define em qual lado a entrada da facilidade i está localizada: (0,0) se no lado A; (1,0) se no lado B; (0,1) se no lado C; (1,1) se no lado D
(p_i^O, q_i^O)	define em qual lado a saída da facilidade i está localizada: (0,0) se no lado A; (1,0) se no lado B; (0,1) se no lado C; (1,1) se no lado D
lb_i	menor dimensão permitida para a facilidade i (*lower bound*)
ub_i	maior dimensão permitida para a facilidade i (*upper bound*)
a_i	área da facilidade i
α_i	razão de aspecto da facilidade i
xf_i, yf_i	coordenadas x e y do canto inferior esquerdo da facilidade fixa i
xsf_i, ysf_i	coordenadas x e y do canto superior direito da facilidade fixa i
xf_i^I, yf_i^I	coordenadas x e y da entrada da facilidade fixa i
xf_i^O, yf_i^O	coordenadas x e y da saída da facilidade fixa i
dx_{ij}	variável artificial que substitui o binômio não-linear $\left\|x_i^O - x_j^I\right\|$
dy_{ij}	variável artificial que substitui o binômio não-linear $\left\|y_i^O - y_j^I\right\|$
lx_{ij}	variável artificial que substitui o termo não-linear $l_{ij}x_i$
by_{ij}	variável artificial que substitui o termo não-linear $b_{ij}y_i$
uv_i	variável artificial que substitui o termo não-linear u_iv_i
$p^{I(O)}m_i^{I(O)}$	variável artificial que substitui o termo não-linear $p_i^{I(O)}m_i^{I(O)}$
$p^{I(O)}n_i^{I(O)}$	variável artificial que substitui o termo não-linear $p_i^{I(O)}n_i^{I(O)}$
$p^{I(O)}wr_i$	variável artificial que substitui o termo não-linear $p_i^{I(O)}wr_i$
$q^{I(O)}hr_i$	variável artificial que substitui o termo não-linear $q_i^{I(O)}hr_i$

$p^{I(O)}q^{I(O)}wr_i$	variável artificial que substitui o termo não-linear $p_i^{I(O)}q_i^{I(O)}wr_i$
$p^{I(O)}q^{I(O)}hr_i$	variável artificial que substitui o termo não-linear $p_i^{I(O)}q_i^{I(O)}hr_i$
Outros	
A, B, C, D	lados das facilidades (inferior, esquerdo, superior e direito)

Sumário

1 Introdução 21

1.1 Contextualização 21

1.2 Objetivos 22

1.3 Motivação 23

1.4 Organização do Trabalho 24

2 Revisão Bibliográfica 26

2.1 Problema de Layout de Facilidades (FLP) 26

2.2 Problema de Layout de Facilidades Estático (SFLP) 26

2.3 Problema de Layout de Facilidades Dinâmico (DFLP) 28

2.4 Problema de Layout de Facilidades de Áreas Desiguais (UAFLP) 33

2.5 Problema de Layout de Facilidades com Locais de Entrada e Saída . . . 36

2.6 Problema de Layout com Restrições Geométricas ou Facilidades Fixas . 38

2.7 Problema de Layout com Corredores entre Facilidades 39

2.8 Classificação dos Problemas de Layout 40

3 Modelagem Matemática 42

3.1 Novos Modelos Matemáticos Propostos neste Trabalho 42

3.2 Programação Inteira Mista 43

3.3 UAFLP com Dimensões Fixas 43

3.3.1 1ª parte dos Modelos 1, 2, 3 e 4 44

3.3.2 2ª parte do Modelo 1 (UAFLPfixo) 46

3.3.3 2ª parte do Modelo 2 (UAFLP1) 46

3.3.4 2ª parte do Modelo 3 (UAFLP2) 46

3.3.5 2ª parte do Modelo 4 (UAFLP3) 46
3.3.6 Discussão dos Modelos 47
3.4 UAFLP com Dimensões Variáveis 49
3.4.1 Modelo 5 (UAFLPvar) 49
3.4.2 Discussão do Modelo 51
3.5 Restrições Geométricas ou Facilidades Fixas 52
3.6 Corredores entre Facilidades 53
3.7 Linearização dos Modelos 54
3.7.1 1ª parte dos Modelos 1, 2, 3 e 4 57
3.7.2 2ª parte do Modelo 1 (UAFLPfixo) 58
3.7.3 2ª parte dos Modelos 2 e 4 (UAFLP1 e UAFLP3) 58
3.7.4 2ª parte do Modelo 3 (UAFLP2) 59
3.7.5 Modelo 5 (UAFLPc) 59
3.7.6 Restrições Geométricas ou Facilidades Fixas 60
3.7.7 Corredores entre Facilidades 60
3.8 Modelos Matemáticos e Métodos de Resolução 60

4 Métodos de Resolução 61
4.1 Método PM-MNL (Programação Matemática com Modelos Não-Lineares) 61
4.1.1 Algoritmo 1 62
4.1.2 Algoritmo 2 64
4.2 Método PM-ML (Programação Matemática com Modelos Lineares) . . . 65
4.3 Método BLAR-CG (Busca Local e Agregação de Rankings usando Cortes Guilhotinados) 66
4.3.1 Formulação do Problema 66
4.3.2 Abordagem Proposta 73
4.4 Método BLAR-PM (Busca Local e Agregação de Rankings usando Programação Matemática) 75
4.5 Aplicação dos Métodos Propostos 76

5 Testes Computacionais 78

5.1 Condições de Implementação . . . 78
5.2 UAFLP: dimensões variáveis e restrições geométricas . . . 79
5.3 UAFLP: dimensões fixas . . . 81
5.3.1 I/O *points* Fixos . . . 82
5.3.2 I/O *points* Variáveis . . . 85

6 Conclusão . . . 89
6.1 Facilidades com dimensões variáveis . . . 90
6.2 Facilidades de dimensões fixas . . . 90
6.3 Impactos e Indicadores . . . 91
6.4 Contribuição Científica . . . 92

Referências . . . 94

Apêndice A – Instâncias da Literatura . . . 102
A.1 Instâncias de Tam (1992)/Nugent *et al.* (1968) . . . 102
A.1.1 Instância Tam92N12 . . . 104
A.1.2 Instância Tam92N15 . . . 104
A.1.3 Instância Tam92N20 . . . 105
A.1.4 Instância Tam92N30 . . . 106
A.2 Instâncias de Das (1993) . . . 108
A.2.1 Instância Das93N4 . . . 108
A.2.2 Instância Das93N6 . . . 108
A.2.3 Instância Das93N8 . . . 109
A.2.4 Instância Das93N10 . . . 110
A.2.5 Instância Das93N12 . . . 111
A.3 Instâncias de Welgama e Gibson (1993) . . . 112
A.3.1 Instância Wel93N6 . . . 112
A.3.2 Instância Wel93N12 . . . 113
A.4 Instâncias de Deb e Bhattacharyya (2005) . . . 114
A.5 Instância de Dunker *et al.* (2003) . . . 116

1 Introdução

1.1 Contextualização

Em um mercado global cada vez mais competitivo, torna-se essencial garantir a otimização de resultados dentro de uma empresa. Maximizar a produtividade é um dos principais objetivos do setor de fabricação das indústrias. Uma das análises realizadas para atingir esse potencial é a do arranjo físico mais adequado para cada situação de forma a agilizar os processos produtivos. No contexto industrial, a execução de um produto passa por diversas etapas que podem ser caracterizadas, por exemplo, por um equipamento ou um departamento. Ou seja, o produto parte de uma configuração inicial e através de uma sequência de processos atinge a configuração final.

Um arranjo físico, ou layout, pode ser definido como o posicionamento relativo das diversas facilidades (instalações, máquinas, equipamentos, setores) dentro de um recinto. A proposta de organizar o layout tem por objetivos melhorar a utilização do espaço disponível, diminuir as distâncias de movimentação de materiais, serviços e pessoas, incrementar a produção, ordenar o ambiente e reduzir os custos indiretos e o tempo de manufatura (já que a disposição das instalações influencia diretamente no tempo gasto nas atividades e processos). As áreas e locais de trabalho devem ser dispostos de forma que homens, materiais e equipamentos se movam em um fluxo contínuo, organizado e de acordo com a sequência lógica dos processos de produção. Devem ser evitados cruzamentos e retornos a fim de garantir melhores fluxos de materiais e serviços.

A iniciativa de organizar as áreas de trabalho acompanha a evolução humana, seja na fabricação de utensílios, na preparação de alimentos ou na construção de edifícios. Mas a configuração de instalações adquiriu importância principalmente a partir da Revolução Industrial, quando os proprietários de fábricas perceberam que seria vantajoso financeiramente levá-la em consideração nos seus projetos. Na perspectiva da indústria, o projeto de layout de facilidades instrui como organizar o arranjo físico dos processos de fabricação para fornecer o melhor suporte para o sistema produtivo (LENO *et al.*, 2018). Com o desenvolvimento da engenharia de produção ao longo dos anos, a distribuição física de produtos e a organização de layouts tornaram-se objetos frequentes de estudo.

Dentro da pesquisa operacional, procura-se a otimização de layouts considerando as mais diversas situações de processo e demanda, de forma a minimizar o tempo de produção e utilizar o espaço existente da forma mais eficiente possível. O problema geral, ou Problema de Layout de Instalações, designa o posicionamento das instalações na fábrica, com o objetivo de determinar a disposição mais eficaz de acordo com alguns critérios ou objetivos, sob certas restrições (GARCÍA-HERNÁNDEZ *et al.*, 2013).

1.2 Objetivos

O presente trabalho tem como objetivo geral **a)**: estudar diferentes modelos para problemas de otimização de layouts encontrados na literatura e, a partir desses, elaborar novos modelos matemáticos para diferentes situações; e **b)** desenvolver e aplicar métodos de resolução para estes problemas.

São objetivos específicos deste projeto:

1. Realizar estudo bibliográfico sobre otimização do problema de layout de facilidades (definições, relevância, classificação, abordagens, modelos, variações, métodos de resolução);

2. Formular, implementar e validar modelos para situações específicas de demanda dinâmica robusta, blocos de áreas desiguais, locais de entrada e saída, espaços pré-ocupados e corredores entre facilidades; e

3. Desenvolver métodos de resolução que forneçam bons resultados, a saber:

   (a) Resolução em duas etapas associando programação matemática (usando um *solver* para problemas não lineares) e uma heurística de melhoramento (duas abordagens);

   (b) Linearização de modelos e resolução via programação matemática (*solver* para problemas lineares);

   (c) Resolução em duas etapas associando Busca Local e Agregação de Rankings com formação de layouts por cortes guilhotinados;

   (d) Resolução em duas etapas associando Busca Local e Agregação de Rankings com formação de layouts por combinação de heurística construtiva e programação matemática (*solver* para problemas lineares).

1.3 Motivação

Sistemas de produção e serviços devem ser operados com planejamento otimizado e boas práticas operacionais para atingir seu potencial. Além disso, definir um bom layout de instalações garante que o sistema como um todo atue de forma mais eficiente. Essa questão do arranjo físico é tão importante que pode impactar na viabilidade do processo de fabricação a longo prazo e, portanto, de preferência, deve ser pensada logo na fase inicial de projeto. Um layout mal projetado resultará em produtividade reduzida, maior trabalho em andamento, maior tempo de fabricação, manuseio de material desordenado e assim por diante (PILLAI *et al.*, 2011). Antigamente, o planejamento de instalações era considerado principalmente uma ciência, mas hoje é uma estratégia (TOMPKINS *et al.*, 2010).

Um exemplo bem interessante do impacto da organização de um layout na produtividade pode ser encontrado em uma rede de *fast-food*. Em 1948, os irmãos McDonald's modificaram todo o modelo de fornecimento de refeições do seu restaurante, inclusive alterando completamente a disposição dos equipamentos dentro da cozinha de forma a tornar a produção o mais rápida possível, o que lhes permitiu atender os clientes em até cinco minutos, comparado a 20 minutos gastos pela maioria de seus concorrentes. A Figura 1.1 ilustra a organização interna aplicada por essa cadeia mundial de *fast-food*.

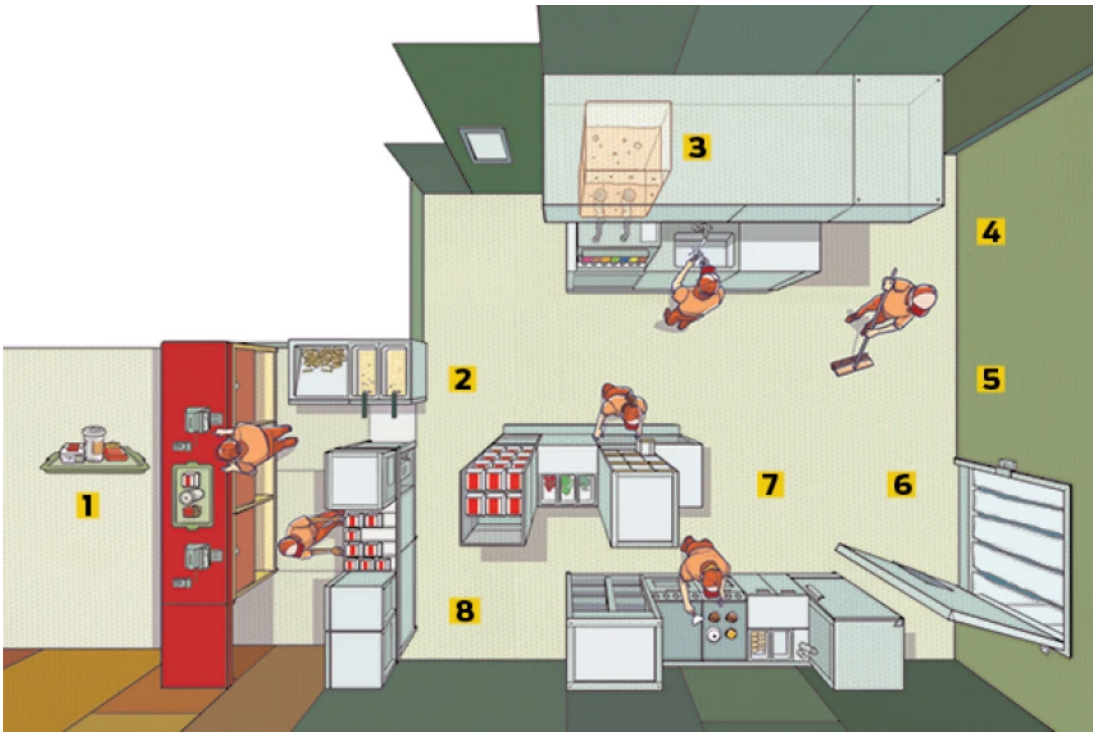

FIGURA 1.1 – Layout de uma cozinha da rede de *fast-food* McDonald's (RAGOV, 2018).

Para além da aplicação na indústria e na produção, a organização dos arranjos físicos também pode ser estudada em outros contextos. Em Qin e Zhao (2021), o problema de

layout está voltado para um parque logístico, que é uma estrutura com grandes espaços para armazenar, administrar e despachar mercadorias de diferentes empresas com eficiência e baixo custo. Rahimi e Jahanirad (2021) avaliam o arranjo físico na implementação de circuitos em FGPAs (placas de circuitos digitais, *chips*) que tem inúmeras aplicações, indo desde vídeo games até as áreas aeroespacial, médica e de computação de alto desempenho. O estudo da otimização de layouts também pode trazer benefício social, como em mais este exemplo recente: o aplicativo "Sala Planejada" de acesso gratuito, desenvolvido para auxiliar gestores no distanciamento social em salas de aula, já foi acessado em mais de 30 países e gerou mais de 500.000 layouts para a comunidade (BORTOLETE *et al.*, 2022). Um exemplo de aplicação de modelos e soluções para layouts com impacto ambiental sustentável é em fazendas de ondas, que são dispositivos de conversão de energia das ondas do oceano, sendo muito interessantes como fontes renováveis de energia: no estudo de Moarefdoost *et al.* (2017), ressalta-se que o recurso potencial total de energia das ondas ao longo da borda da plataforma continental dos EUA é estimado em quase um terço do consumo anual de eletricidade no país.

Desde a primeira modelagem publicada por Koopmans e Beckmann (1957) para o problema de layouts já se passaram quase sete décadas e, desde então, foram sugeridas muitas alterações no modelo quadrático de atribuição original proposto. Embora o problema de layout de facilidades já tenha sido amplamente abordado na literatura, de acordo com o extensivo estudo de Hosseini-Nasab *et al.* (2018), a pesquisa sob muitos aspectos do problema ainda está em estágio inicial, sendo este um campo interessante para se trabalhar.

1.4 Organização do Trabalho

O presente trabalho encontra-se organizado em cinco capítulos.

O Capítulo 1 contém uma introdução ao tema da pesquisa na qual são expostos a contextualização, os objetivos visados, as considerações que motivaram na escolha do tema e a organização do conteúdo.

No Capítulo 2 é realizada uma revisão bibliográfica da literatura relacionada ao tema do projeto. São apresentados os primeiros trabalhos em otimização de layouts, alguns conceitos pertinentes, o modelo clássico do problema de layout de facilidades, as diferentes abordagens dadas ao problema com o passar do tempo tratadas nesta tese, as justificativas para tais e os estudos relacionados a cada uma.

O Capítulo 3 apresenta os modelos matemáticos desenvolvidos pela autora para representar as diferentes situações estudadas, assim como a alternativa linearizada destes. São explicitados os conjuntos, índices, dados de entrada e variáveis desses modelos. Todas as restrições e a função objetivo são devidamente discutidas.

O Capítulo 4 discorre sobre os métodos de resolução usados, com a descrição detalhada das abordagens adotadas para cada caso, dos algoritmos, das heurísticas de construção e melhoramento e das etapas de formulação do problema.

As condições de implementação dos modelos e métodos de resolução em linguagem de programação, as instâncias testadas, os parâmetros utilizados, as soluções obtidas para diferentes configurações, as comparações com outros trabalhos e as avaliações dos resultados alcançados nesta tese encontram-se no Capítulo 5.

Finalmente, o Capítulo 6 aborda as conclusões gerais do trabalho, os potenciais impactos científicos, econômicos, sociais e ambientais da pesquisa e uma análise crítica de todo o processo.

Esta tese também contém um apêndice no qual são transcritas todas as características e dados das instâncias testadas.

2 Revisão Bibliográfica

2.1 Problema de Layout de Facilidades (FLP)

Processos produtivos passam por diferentes diferentes etapas, por vezes bem definidas, caracterizadas por setores ou equipamentos. Conforme frisado anteriormente, o estudo do layout adequado é muito relevante, pois a disposição das facilidades influencia diretamente no tempo despendido nas atividades e processos de produção. Por conta disso, o FLP é um dos problemas mais importantes na literatura de gerenciamento de produção e engenharia industrial, atraindo a atenção de muitos pesquisadores no campo de layouts estáticos e dinâmicos (HOSSEINI-NASAB *et al.*, 2018).

2.2 Problema de Layout de Facilidades Estático (SFLP)

Quando o fluxo de materiais entre as instalações não muda com o tempo, o problema é conhecido como problema de layout estático das instalações e, na forma mais simples, pode ser formulado como um Problema de Quadrático de Alocação (QAP) (EMAMI; NOOKABADI, 2013).

O problema quadrático de alocação, de classe NP-difícil (NP-*hard*), modela diversas aplicações em diferentes áreas como pesquisa operacional, computação paralela e análise estatística de dados discretos (LOIOLA *et al.*, 2004).

O modelo clássico de otimização, introduzido por Koopmans e Beckmann (1957), é definido como segue. As notações usadas na formulação constam na Tabela 2.1.

$$\min \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{r=1}^{N} \sum_{s=1}^{N} f_{ij} d_{rs} x_{ir} x_{js} \tag{2.1}$$

Conjunto e Índices	
N	número total de facilidades/locais
i e j	índices para facilidades = (1,...,N)
r e s	índices para locais = (1,...,N)
Dados de Entrada	
d_{rs}	distância entre os locais r e s
f_{ij}	fluxo de peças entre as facilidades i e j
Variáveis de Decisão	
x_{ir}	1 se a facilidade i está localizada no local r, 0 cc.

TABELA 2.1 – Notações usadas no modelo FLP

sujeito a

$$\sum_{r=1}^{N} x_{ir} = 1 \qquad \forall i \tag{2.2}$$

$$\sum_{i=1}^{N} x_{ir} = 1 \qquad \forall r \tag{2.3}$$

$$x_{ir} \in \{0, 1\} \qquad \forall i, r \tag{2.4}$$

Na formulação do QAP para layouts, todas as instalações têm a mesma área e o espaço é dividido em N locais de tamanho igual, onde cada instalação é atribuída a exatamente um local e vice-versa (KONAK *et al.*, 2006). O objetivo é minimizar o somatório dos Custos de Manuseio de Materiais (MHCs) entre cada par de instalações, sendo o MHC um produto do fluxo de materiais entre duas facilidades pela distância entre as mesmas. As restrições garantem que cada facilidade ocupa um único local (2.2) e que cada local é ocupado por uma única facilidade (2.3). A Figura 2.1 exemplifica uma possível configuração otimizada para um problema com 24 facilidades.

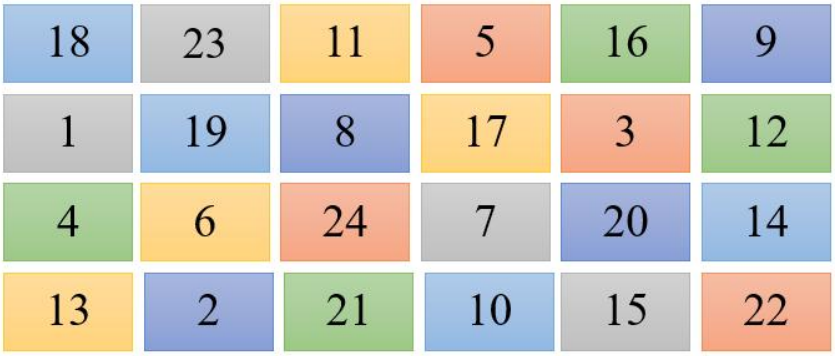

FIGURA 2.1 – Exemplo de configuração para o FLP clássico.

Nesse modelo básico, as distâncias são medidas entre os centroides dos blocos, podendo ser Euclideanas (diagonal) ou retilíneas (soma da horizontal com a vertical), conforme Figura 2.2. As distâncias retilíneas também são chamadas de retangulares (BRIMBERG;

WESOLOWSKY, 2008) e de distâncias de Manhattan (ZHENG, 2014).

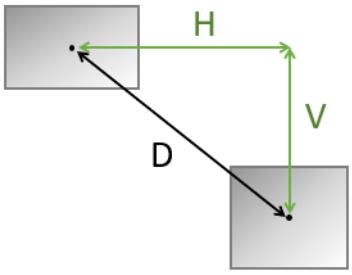

FIGURA 2.2 – Representação das distâncias Euclideana (d = D) e retilínea (d = H + V).

Embora a diagonal represente a menor distância entre os centroides, a distância retilínea é muito utilizada pelo fato de representar melhor uma situação real de deslocamento que ocorreria através dos corredores entre as facilidades.

2.3 Problema de Layout de Facilidades Dinâmico (DFLP)

É muito improvável que os fluxos de material entre as instalações permaneçam inalterados durante um longo horizonte de planejamento (HOSSEINI-NASAB *et al.*, 2018). Novas tendências e tecnologias surgem, a economia e a renda dos consumidores oscilam, os costumes da população podem mudar e a sazonalidade atinge vários mercados. A todo momento as demandas por produtos mudam, algumas até desaparecem e novas surgem. São frequentes as necessidades de alteração nos projetos dos produtos e nos métodos e sistemas de trabalho devido a incertezas, sejam essas no nível de demanda, nos custos envolvidos, no preço ou na variedade de produtos.

Tendo então demandas que variam e são incertas ao longo do tempo, é de se esperar que os dados de entrada referente aos fluxos de material entre as facilidades não sejam determinísticos, mas sim, estocásticos e possam ser gerados, por exemplo, através de distribuições de probabilidade, sendo então convertidos em dados discretos para aplicação no modelo. No FLP estocástico, o *mix* e a demanda dos produtos são considerados variáveis aleatórias com parâmetros conhecidos (por exemplo, valor esperado, variância, covariância e informações de roteamento de produtos e custos unitários de manuseio de material) (KULTUREL-KONAK, 2007). No momento de se projetar instalações, a demanda para os vários períodos pode ser obtida a partir de estimativas de padrões de produção futuros condensados em cenários discretos.

O problema de layout de facilidades dinâmico, DFLP, é estudado através de duas abordagens: a primeira sugere alterações no arranjo das facilidades ao longo do tempo de modo a se readaptar às novas condições de produção (considera plantas flexíveis), enquanto que a segunda procura pelo melhor layout possível que atenda a todas as mudanças

ao longo do tempo, pois considera custos de *setup* e re-layout muito ostensivos (layout robusto). A solução para um tratamento robusto é um único layout, mas para o DFLP flexível (adaptativo), pode ser uma série de layouts (plano de layout) com cada layout associado a um período específico (MAZINANI *et al.*, 2013). A Figura 2.3 ilustra as duas situações para um problema teórico com 9 facilidades e 3 períodos.

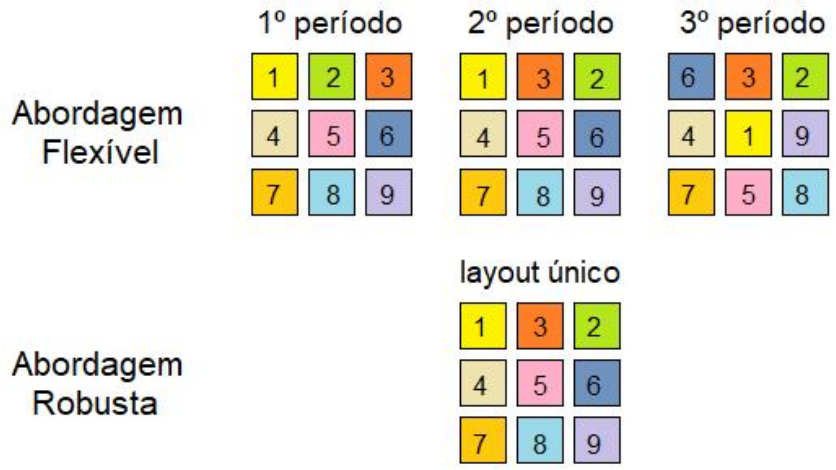

FIGURA 2.3 – Exemplo de arranjo físico de facilidades considerando layout flexível e robusto.

No contexto do problema de layout de facilidades, o termo *flexível* está associado a possibilidade de se alterar a configuração do arranjo das instalações para melhor atender as mudanças de cenário que impactam o setor produtivo da empresa, enquanto que *robusto* quer dizer que o layout definido comporta bem as mudanças de fluxo ao longo do tempo mesmo se mantendo inalterado.

A abordagem flexível ou adaptativa pressupõe que um layout acomodará mudanças de tempos em tempos com baixos custos de rearranjo e interrupção de produção e que as máquinas podem ser facilmente realocadas (PILLAI *et al.*, 2011). Um horizonte de planejamento é dividido em períodos (estados) definidos, por exemplo, por semanas, meses, trimestres ou anos (MAZINANI *et al.*, 2013).

Um modelo geral adotado para resolver DFLP flexível inclui no cálculo da Função Objetivo (FO) custos variáveis de realocar uma facilidade de um lugar a outro e custos fixos de realizar qualquer rearranjo no início de um período, sem interferência no conteúdo das restrições. Um exemplo de FO para essa situação é apresentada a seguir (Expressão 2.5), conforme trabalho de Urban (1998) (notações na Tabela 2.2):

$$\min \sum_{t=1}^{T} \left(\sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{r=1}^{N} \sum_{s=1}^{N} f_{ijt} d_{rs} x_{irt} x_{jst} + \sum_{i=1}^{N} cr_{it} y_{it} + cf_t z_t \right) \tag{2.5}$$

A abordagem robusta tem a vantagem da ausência de custo de reorganização e a

Conjuntos e Índices	
N	número total de facilidades/locais
T	número total de períodos
i e j	índices para facilidades = (1,...,N)
r e s	índices para locais = (1,...,N)
t	índice para períodos = (1,...,T)
Dados de Entrada	
d_{rs}	distância entre os locais r e s
f_{ijt}	fluxo de peças entre as facilidades i e j no período t
cr_{it}	custo de realocar a facilidade i no início do período t
cf_t	custo fixo de realizar qualquer rearranjo no início do período t
Variáveis de Decisão	
x_{irt}	1 se a facilidade i está localizada no local r no período t, 0 cc.
y_{it}	1 se a facilidade i é realocada no início do período t, 0 cc.
z_t	1 se qualquer rearranjo é feito no início do período t, 0 cc.

TABELA 2.2 – Notações usadas no modelo DFLP flexível

desvantagem de não ter um layout ótimo para cada período (MOSLEMIPOUR *et al.*, 2017). É apropriada para situações que envolvem equipamentos pesados (que dificilmente poderiam ser movidos uma vez alocados) ou de instalação complexa nas quais o custo de rearranjo das facilidades é alto ou inconveniente.

Ao considerar mudanças futuras na etapa de *design*, os gerentes da instalação podem selecionar projetos que não se degradam radicalmente com as mudanças na produção (KULTUREL-KONAK, 2007). A robustez é um critério de flexibilidade que é utilizado em casos de incerteza e pode ser definida como a frequência com que um arranjo cai dentro de uma porcentagem pré-especificada da solução ótima para diferentes conjuntos de cenário de produção (ROSENBLATT; LEE, 1987). Em um modelo robusto, embora o layout da solução encontrada possa não ser o melhor para nenhum dos períodos individualmente, ele pode ser o mais adequado considerando todos os estados (ROSENBLATT; LEE, 1987).

Rosenblatt e Lee (1987) foram os primeiros a demonstrar a aplicação da abordagem de robustez para o DFLP. Diversos autores propuseram métodos de resolução para modelos robustos: Kouvelis *et al.* (1992) criaram algoritmos para planejamentos de período único e múltiplo; Azadivar e Wang (2000) propuseram simulação e algoritmos genéticos (GAs); Smith e Norman (2000) trataram o projeto evolutivo de facilidades de áreas desiguais por GA; Aiello e Enea (2001) apresentaram um modelo difuso (*fuzzy*); Braglia *et al.* (2003) analisaram os efeitos da incerteza nas taxas de produção em um layout de linha única; Chan *et al.* (2004) desenvolvem um algoritmo heurístico de Alocação de Máquinas (MAIN) para layout intracelular.

Kulturel-Konak *et al.* (2004) abordaram rotas flexíveis e blocos de áreas desiguais com resolução por Busca Tabu (TS); Irappa e Madhusudanan (2008) desenvolveram uma heurística para DFLPs considerando QAP e a média das demandas determinísticas dos

períodos; See e Wong (2008) aplicaram um algoritmo de otimização da colônia de formigas (ACO) para resolver problemas modelados como QAPs; e Pillai *et al.* (2011) propuseram um algoritmo de Recozimento Simulado (SA) e consideraram uma matriz de fluxo esperado derivada da média aritmética dos fluxos nos vários períodos.

Além desses, Forghani *et al.* (2013) estudaram layouts em sistemas de fabricação de celulares com demanda incerta; Soolaki e Izadi (2013) resolveram um problema de projeto em célula de fabricação sob incerteza; Azadeh *et al.* (2014) abordaram o processo de injeção com demandas aleatórias aplicando análise envoltória de dados e TS com diversificação; Neghabi *et al.* (2014) propuseram um novo modelo e algoritmo adaptável considerando instalações de dimensões definidas; Nematian (2014) projetou um layout de fileira única assumindo variáveis aleatórias *fuzzy*; Pourvaziri e Naderib (2014) desenvolveram um GA de multi-população híbrida; Lee *et al.* (2015) usaram GA para resolver um modelo formulado como QAP; Fazlelah *et al.* (2016) sugeriram um modelo para ambiente dinâmico e um GA baseado em permutação.

Asl e Wong (2017) lidaram com SFLPs e DFLPs de áreas desiguais usando um algoritmo modificado de Otimização de Enxame de Partículas (PSO); Moslemipour *et al.* (2017) propuseram um novo modelo matemático para o FLP estocástico tendo como base a formulação do QAP; Zha *et al.* (2017) apresentaram uma formulação para o FLP robusto com base na teoria aleatória *fuzzy* na qual um método de valor esperado primeiro transformava uma variável aleatória *fuzzy* em variável *fuzzy* e depois em valor determinístico; Peng *et al.* (2018) estabeleceram um modelo matemático para o DFLP robusto considerando a atribuição de dispositivos de transporte (que considera diferentes custos unitários para cada veículo associados ao MHC) e usaram como procedimento de solução um GA adaptativo melhorado.

Vitayasak *et al.* (2019) integraram o *design* robusto do DFLP com um planejamento de manutenção das máquinas com resolução via GA; Xiao *et al.* (2019) propuseram um modelo híbrido de otimização robusta considerando áreas desiguais e resolveram um caso real usando um algoritmo PSO; e Salimpour *et al.* (2021) trataram a incerteza de demanda com uma abordagem celular semi-robusta que não permite o layout das instalações se alterar de um período para o outro, mas permite que as posições dos pontos de embarque/desembarque das células mudem.

A seguir é apresentada a FO (e uma equação associada) para um modelo do DFLP robusto cuja ideia principal (aderida para os modelos desenvolvidos nesta tese) é a mesma adotada pela maioria dos autores nos artigos mais recentes sobre o tema (por exemplo, Vitayasak *et al.* (2019), Xiao *et al.* (2019) e Salimpour *et al.* (2021)), que também consideram no cálculo do MHC total o acumulado dos fluxos no tempo. No artigo de Pillai *et al.* (2011), o fluxo médio entre facilidades nos diferentes períodos é considerado como o fluxo de peças para todo o horizonte de planejamento, matematicamente levando ao mesmo

resultado que a abordagem anterior em termos de layout. Os resultados de Pillai *et al.* (2011) mostraram que esse modelo robusto tem bom desempenho para FLPs e é computacionalmente eficiente comparado ao modelo adaptativo. Eles ainda ressaltam que, embora em alguns casos o MHC para os layouts do método robusto não tenha sido significativamente diferente dos melhores resultados para a abordagem adaptativa, a abordagem robusta tem a vantagem de não contar com a realocação de instalações nos períodos do horizonte de planejamento e, portanto, ocorrer sem interrupções das operações (PILLAI *et al.*, 2011). As notações utilizadas estão na Tabela 2.3.

Conjuntos e Índices	
N	número total de facilidades/locais
T	número total de períodos
i e j	índices para facilidades = (1,...,N)
r e s	índices para locais = (1,...,N)
t	índice para períodos = (1,...,T)
Dados de Entrada	
d_{rs}	distância entre os locais r e s
f_{ijt}	fluxo de peças entre as facilidades i e j no período t
F_{ij}	fluxo total de peças entre as facilidades i e j
Variáveis de Decisão	
x_{ir}	1 se a facilidade i está localizada em r, 0 cc.

TABELA 2.3 – Notações usadas no modelo DFLP robusto

$$\min \sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{r=1}^{N}\sum_{s=1}^{N} F_{ij} d_{rs} x_{ir} x_{js} \tag{2.6}$$

$$\text{sendo } F_{ij} = \sum_{t=1}^{T} f_{ijt} \qquad \forall i, j \tag{2.7}$$

Também poderiam ser adotadas outras alternativas para o modelo. Na proposta de Moslemipour *et al.* (2017), o modelo robusto para o DFLP é desenhado de forma que o fluxo entre facilidades calculado para um único período (F_{ij}) seja considerado como o fluxo de peças para todo o horizonte de planejamento; porém, ao invés de considerar F_{ij} igual a média dos fluxos f_{ijt} como feito por Pillai *et al.* (2011), ele é calculado a partir de dados das demandas de produtos que são consideradas variáveis aleatórias normalmente distribuídas com valor esperado, variância e covariância conhecidos que mudam aleatoriamente de período para período (em geral, os autores consideram esses cálculos probabilísticos já embutidos nos valores dos fluxos f_{ijt} fornecidos nos dados de entrada). Já nos trabalhos de Chan *et al.* (2004) e Peng *et al.* (2018), o MHC é calculado individualmente para cada período e então todas essas soluções são aplicadas nos demais períodos, sendo escolhida aquela que resulta no melhor MHC total.

2.4 Problema de Layout de Facilidades de Áreas Desiguais (UAFLP)

Em uma situação real é improvável que as várias facilidades de uma empresa possuam todas a mesma área. Uma FLP particularmente interessante, devido a sua aplicação direta em casos reais, é conhecida como *Unequal Area Facility Layout Problem* (UAFLP) (GARCÍA-HERNÁNDEZ *et al.*, 2019). As Figuras 2.4 e 2.5 mostram duas situações aplicadas nas quais fica evidente a diversificação de tamanho das facilidades, respectivamente, a divisão de departamentos dentro de uma instituição de ensino e o posicionamento de equipamentos em uma marcenaria.

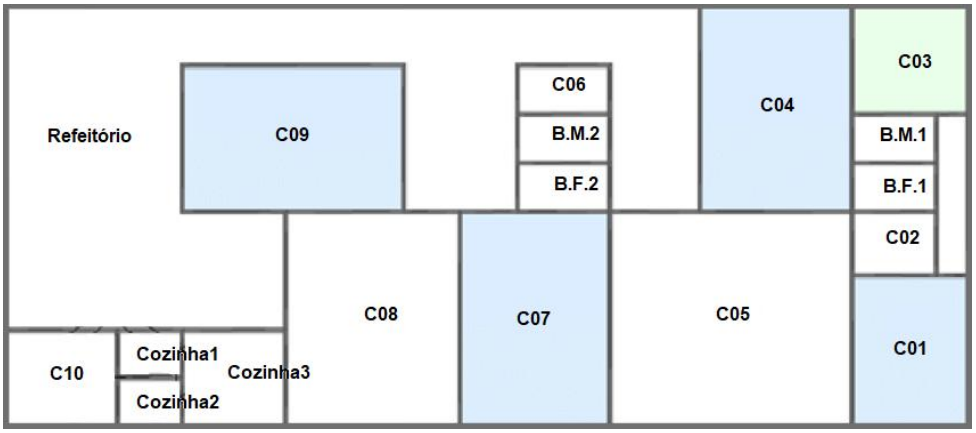

FIGURA 2.4 – Departamentos de áreas desiguais (INÁCIO; ERTHAL, 2017).

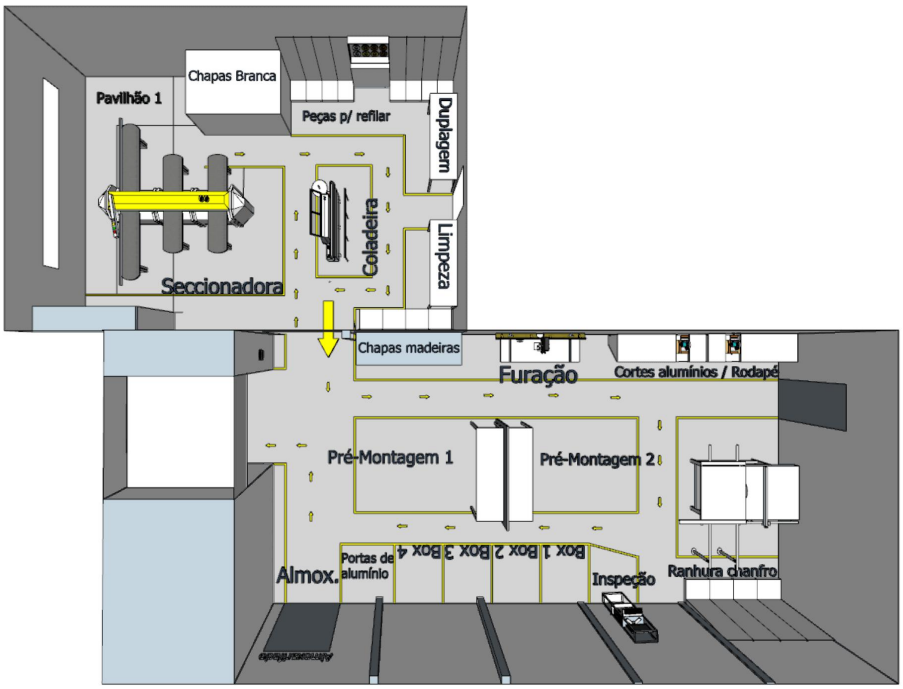

FIGURA 2.5 – Equipamentos de áreas desiguais (NILSON, 2017).

Quando as facilidades têm áreas diferentes, o problema não pode mais ser resolvido atribuindo N facilidades a N localizações de centroides distintas. As complicações adicionais dos requisitos de áreas desiguais, com posições de facilidades contínuas que podem estar em qualquer lugar de uma área retangular e áreas variadas (largura e altura), tornam o FLP extremamente difícil de resolver (CASTILLO; WESTERLUND, 2005).

O UAFLP é estudado sob muitos aspectos. A começar, em relação à forma, as facilidades podem ser descritas como regulares (por exemplo, blocos retangulares) ou irregulares (geralmente polígonos). Já em relação às dimensões, uma facilidade pode ser representada por medidas fixas de altura e largura (bloco rígido) ou por uma área fixa e uma razão entre dimensões (*aspect ratio*). No que diz respeito à configuração, os FLPs são classificados em sete categorias distinguidas pelo formato de seu caminho de manuseio de material, a saber, layouts de a) linha única, b) múltiplas linhas, c) linha dupla, d) linhas paralelas, e) *loops*, f) campo aberto (*open-field*) e g) pisos múltiplos (HOSSEINI-NASAB *et al.*, 2018), conforme ilustrado na Figura 2.6.

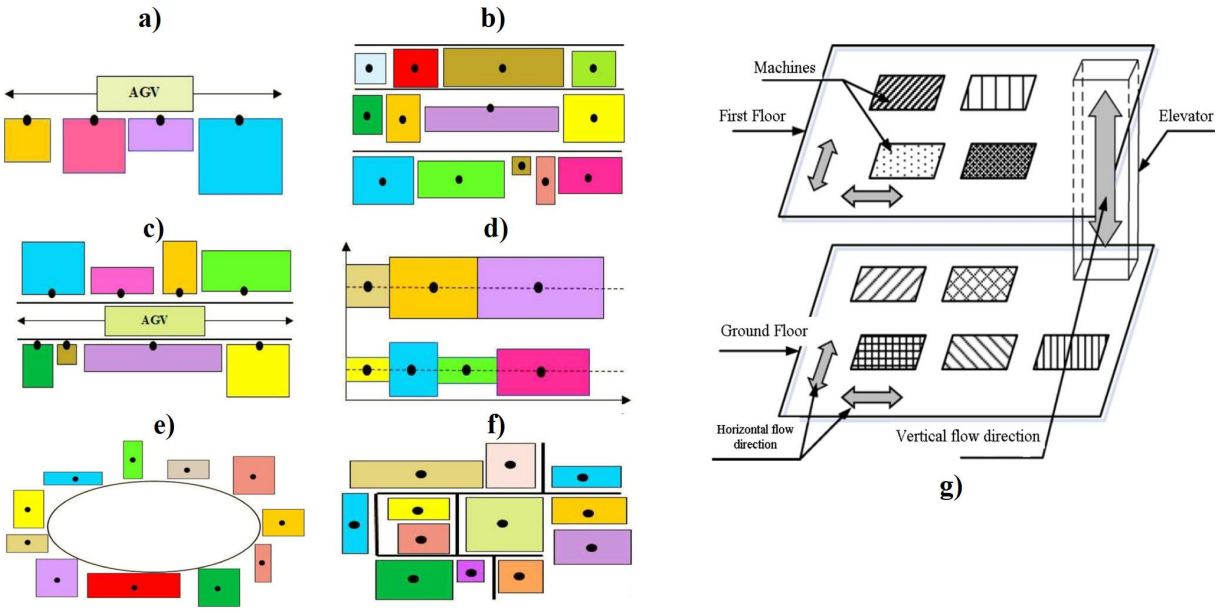

FIGURA 2.6 – Diferentes configurações de layouts em função do sistema de produção (HOSSEINI-NASAB *et al.*, 2018).

De acordo com Hosseini-Nasab *et al.* (2018), a maior parte dos autores do campo de layouts considera em seus estudos projetos com as seguintes características: facilidades regulares, de dimensões fixas e sistema *open-field*. Isso leva a crer que tais circunstâncias têm boa relevância no contexto prático justificando sua adoção. No presente trabalho, além de projetos com essas características, também foi analisado o caso com facilidades de área fixa e dimensões variáveis.

Armour e Buffa (1963) foram os primeiros a abordar o problema de layout de facilidades de áreas desiguais. Alguns dos pesquisadores de DFLP robusto, já citados anteriormente,

também consideraram blocos de áreas desiguais em seus estudos como Smith e Norman (2000), Kulturel-Konak *et al.* (2004), Asl e Wong (2017), e Xiao *et al.* (2019). Outros vários autores propuseram métodos de resolução para o UAFLP: Bazaraa (1975) dividiu o espaço disponível em pequenas áreas unitárias formulando o problema de layout como um problema quadrático de cobertura de conjuntos resolvendo-o com uma abordagem *Branch and Bound* (B&B).

Tate e Smith (1995) apresentaram uma versão restritiva da formulação de Árvore B&B (ST) conhecida como Estrutura de Baias Flexíveis (FBS) e resolveram o problema com GA; Yang e Peters (1998) consideraram um layout robusto para ambientes de produção dinâmicos e incertos; Deb e Bhattacharyya (2003) se basearam em uma metodologia de tomada de decisão multi-critério *fuzzy*; Sherali *et al.* (2003) aprimoraram formulações de modelos para o problema com facilidades de tamanhos variados; Konak *et al.* (2006) sugeriram uma nova formulação do modelo de Programação Inteira Mista (MIP *model*) para FBS; e Liu e Meller (2007) propuseram uma heurística baseada em GA que combina a representação de pares de sequência com o MIP *model* para o caso de facilidades de área fixa e dimensões variáveis.

Na última década, vários autores propuseram métodos de resolução para abordar UAFLP: Kulturel-Konak (2012) usou programação linear e TS com relaxação das restrições de localização e formato das facilidades para resolver um problema com FBS, conquistando reduções significativas no MHC em comparação às instâncias da literatura testadas; Chang e Ku (2013) representaram uma ST e desenvolveram uma heurística que encontrou soluções ótimas ou quase ótimas para a maioria dos problemas, desde que tivessem menos de 20 facilidades; García-Hernández *et al.* (2013) observaram aspectos qualitativos do UAFLP como, por exemplo, preferências do tomador de decisão sobre a localização de instalações específicas, distribuição do espaço restante e localização relativa das instalações.

Gonçalves e Resende (2015) resolveram o problema propondo duas versões de um algoritmo genético com chaves aleatórias tendenciosas (BRKGA) alcançando soluções de alta qualidade em tempos computacionais relativamente pequenos com a versão irrestrita da abordagem, mas não tão boas com a restrita; Kang e Chae (2017) usaram uma heurística de busca harmônica; Palomo-Romero *et al.* (2017) fizeram um Algoritmo Genético de Modelo de Ilha (IMGA) como uma abordagem alternativa a demais métodos de resolução da literatura que ajudou a evitar problemas de convergência prematura e tempo de execução excessivo; García-Hernández *et al.* (2019) utilizaram um algoritmo de otimização de recifes de coral (CRO) que apresentou excelente desempenho ao considerar exclusivamente a representação de FBS, mas não tão bom ao considerar tanto a estrutura ST quanto a FBS; Hou *et al.* (2019) aplicaram GA de codificação em camadas; e García-Hernández *et al.* (2020) desenvolveram um novo modelo de ilha baseado em CRO.

2.5 Problema de Layout de Facilidades com Locais de Entrada e Saída

As restrições consideradas para os FLPs são das mais diversas, mas entre essas, são de muita importância as que definem pontos de entrada e saída das facilidades. Em uma facilidade, é de se esperar que o fluxo de materiais, produtos e/ou pessoas chegue por um determinado local e saia também por um certo local (que pode ser o mesmo pelo qual chegou ou não). Nos FLPs tradicionais, os pontos de entrada e saída das facilidades não são considerados no modelo quando um layout está sendo construído, e sim são determinados após a obtenção de um layout (KIM; KIM, 2000).

Na literatura, essas localidades são chamadas por diferentes nomes como: pontos de entrada e saída (input/output - I/O); estações de coleta, ou recebimento, e entrega (pick-up/drop-off ou delivery - P/D); e locais de carga e descarga ou embarque e desembarque (load e unload). Nos artigos escritos nos últimos anos, priorizou-se usar o termo I/O *points*, portanto, adiante eles estão referenciados dessa forma.

Os I/O *points* podem ser fixos (predeterminados) ou variáveis com restrições, a depender do tipo de facilidade que está sendo tratada no problema. Quando as facilidades, por exemplo, são máquinas ou equipamentos que possuem I/O *points* definidos, ou seja, que não podem ser alterados, é interessante considerar pontos fixos, como pode ser observado na Figura 2.7 que apresenta como facilidades esteiras rolantes que possuem locais de entrada e saída de material determinados. Por outro lado, uma situação com cômodos pode permitir mais flexibilidade no projeto: a Figura 2.8 apresenta um conjunto de departamentos para os quais é possível definir as melhores posições para instalar as portas das salas através das quais ocorre entrada e saída de pessoal (nesse caso, os I/O *points* são variáveis, porém com a restrição de estarem localizados nas bordas dos blocos).

O'brien e Barr (1980) foram os primeiros a considerar I/O *points* nos dados básicos de entrada exigidos pelo seu procedimento. Muitos autores propuseram métodos de resolução para a situação de I/O *points* no FLP: Das (1993) sugeriu uma heurística de quatro etapas que combina métodos de particionamento variável e programação inteira para sistemas de fabricação flexíveis (FMS) com I/O *points* fixos; Welgama e Gibson (1993) utilizaram um algoritmo construtivo considerando I/O *points* fixos; Rajasekharan *et al.* (1998) encontraram bons resultados para o problema por meio de GA considerando a formulação de Das (1993).

Kim e Kim (2000) consideraram blocos de dimensões fixas no problema e desenvolveram um MIP *model*, o qual foi resolvido com um algoritmo heurístico de duas fases (construção e melhoramento); Aiello *et al.* (2002) estudaram a representação de FBS para um sistema de veículos autoguiados (AGV); Wu e Appleton (2002) avaliaram o problema

FIGURA 2.7 – Exemplo de facilidades com I/O *points* fixos (MELO, 2017).

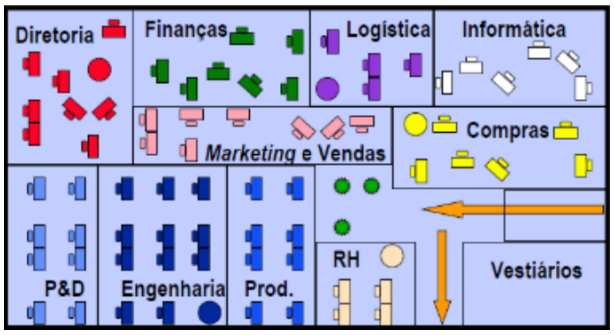

FIGURA 2.8 – Exemplo de facilidades com I/O *points* variáveis (BRANSKI, 2008).

de layout com I/O *points* considerando uma estrutura de corredores entre blocos e fileiras de blocos, utilizando GA como método de resolução; Dunker *et al.* (2003) propuseram um algoritmo co-evolucionário para um MIP *model*.

Deb e Bhattacharyya (2005) utilizaram técnicas de busca aleatória e soluções com GA, SA e algoritmo híbrido; Dunker *et al.* (2005) combinaram computação evolutiva e programação dinâmica para um DFLP flexível; Bock e Hoberg (2007) detalharam o planejamento de layout para máquinas de formato irregular com projeto do caminho de transporte; Hu *et al.* (2007) desenvolveram um GA para a situação de I/O *points* livres nas bordas (UAFLP3, Seção 3.3); Kelachankuttu *et al.* (2007) trabalharam com a construção de linhas de contorno para uma nova instalação retangular em um layout existente com facilidades retangulares.

Na última década, Jolai *et al.* (2012) desenvolveram um algoritmo de PSO multi-objetivo; Xiao *et al.* (2013) sugeriram uma heurística de duas fases, sendo a primeira com algoritmos de zona interconectada e SA, e a segunda com as informações de posição relativa

obtidas na primeira etapa aplicadas em um MIP *model*, resultando em melhores soluções do que as fornecidas por outros algoritmos da literatura; Zheng (2014) gerou gráficos de conectividade para o cálculo do caminho de Manhattan em layouts detalhados; Leno *et al.* (2016) desenvolveram um GA de estratégia elitista usando SA como busca local.

Friedrich *et al.* (2018) fizeram uma abordagem integrada de *slicing tree* para resolver o problema com base na distância de contorno; Leno *et al.* (2018) criaram um MIP *model* e um algoritmo híbrido GA-SA de estratégia elitista (ESHGA) que gerou bons layouts em relação a demais resultados reportados; Kulturel-Konak (2019) considerou facilidades atribuídas a zonas flexíveis com um posicionamento pré-estruturado e DFLP adaptativo; Park e Seo (2019) sugeriram uma heurística construtiva de duas fases que produziu bons resultados em um tempo muito menor do que pesquisas anteriores.

2.6 Problema de Layout com Restrições Geométricas ou Facilidades Fixas

Idealmente, arranjos físicos são definidos durante a fase de projeto de uma construção, porém às vezes é necessário alterar uma planta de layout já existente, na qual constam facilidades fixas tanto em localização quanto em tamanho. E mesmo em layouts concebidos previamente, ainda é possível se deparar com a existência de espaços no interior da área disponível que não possam ser usados para alocar as facilidades por já estarem ocupados, por exemplo, com elevadores, escadas, pilares estruturais, entre outras barreiras ou obstáculos.

A maior parte dos trabalhos que tratam de áreas desiguais só consideram facilidades no formato retangular (de dimensões fixas ou variadas) e não são capazes de incorporar restrições geométricas como as citadas. Mas existem artigos como o de Tam (1992) que consideram essas restrições no problema. Tam (1992) desenhou plantas contendo espaços vazios cuja área útil total é igual a área total das facilidades combinadas de determinadas instâncias encontradas na literatura e solucionou o problema usando uma abordagem do tipo GA. Posteriormente, Furtado e Lorena (1997), usando como estrutura de dados uma árvore binária e soluções iniciais geradas por um procedimento de aglomeração ou aleatoriamente, resolveram o problema com uma heurística de melhoramento por TS.

Gau e Meller (1999) desenvolveram um algoritmo iterativo que mescla GA e programação matemática, considerando uma razão de aspecto fixa e uma função de penalidade para restrição do formato das facilidades similar a encontrada em Coit *et al.* (1996). Xie e Sahinidis (2008) examinaram várias particularidades como I/O *points* e facilidades fixas em modelos matemáticos e resolveram alguns problemas com um algoritmo B&B, mas nenhum com restrições geométricas. Por fim, Anjos e Vieira (2016) desenvolveram uma

estrutura aprimorada baseada na combinação de dois modelos de otimização matemática para resolver UAFLP, entre estes um com facilidades fixas, considerando razões de aspecto fixas, mas desconsiderando o fator de irregularidade da forma.

2.7 Problema de Layout com Corredores entre Facilidades

Outra situação estudada neste trabalho foi a possibilidade de inclusão nos modelos de corredores entre blocos, também chamados de distância de folga (na literatura encontra-se principalmente o termo *clearence distance*). Tratando-se de equipamentos, a distância de folga é o afastamento necessário ao redor das máquinas para carga e descarga de materiais ou produtos, acesso dos trabalhadores às máquinas para operá-las ou realizar manutenção e reparo, armazenar temporariamente materiais em processo, ventilação adequada e minimizar os efeitos de vibração de máquinas vizinhas (ZUO *et al.*, 2016).

É razoável esperar que em torno de cada instalação haja um espaço mínimo dedicado ao trânsito de pessoas e materiais de forma a atender requisitos operacionais e de segurança, mas em geral, no projeto de layout estabelece-se que os corredores estão passando entre as facilidades por suas bordas, sem incluir a largura desses. Nesses casos, considera-se que a distância de folga está embutida nas dimensões individuais de largura e comprimento dos blocos, o que simplifica o modelo, mas não oferece a possibilidade de compartilhar corredores entre as facilidades quando aplicável (SALIMPOUR *et al.*, 2021). Outro fato que justifica considerar essa distância como um dado de entrada à parte no modelo é que a largura do corredor pode ser significativa para o cálculo do MHC.

Alguns autores exploraram esse aspecto no FLP, por exemplo: Das (1993) reconheceu a importância de um espaço de folga para operações de carga e descarga, porém, ao invés de incluir o intervalo no modelo, projetou o afastamento no próprio desenho da instância, ficando os I/O *points* alocados no interior dos blocos. Xie e Sahinidis (2008) apresentaram várias extensões aos modelos matemáticos de layout, como a referente a corredores (associada a adição de um novo conjunto de restrições ao modelo), e resolveram alguns problemas com um algoritmo B&B, mas nenhum para este caso em particular.

Wang *et al.* (2015) formularam o problema para a situação de configuração em linha dupla e demanda dinâmica flexível considerando uma folga mínima entre as facilidades e resolveram com uma combinação de SA e programação matemática. Vitayasak *et al.* (2017) resolveram o problema para uma configuração em múltiplas linhas e demanda dinâmica flexível usando GA e um Algoritmo de Busca por Retrocesso (BSA) modificado. Salimpour *et al.* (2021) consideraram o planejamento conjunto de células de produção e máquinas no layout e incluíram no modelo tanto uma distância de folga em torno das máquinas quanto em torno das células.

2.8 Classificação dos Problemas de Layout

Os problemas de layout são diversos e podem ser estudados sob vários aspectos. O trabalho de revisão da classificação do problema de layout mais novo é o de Hosseini-Nasab *et al.* (2018), no qual é realizada uma ampla e extensa investigação e análise da literatura existente sobre o tema. Além de recente, o trabalho de Hosseini-Nasab *et al.* (2018) tem a vantagem de não se limitar a considerações específicas sobre projeto de layout.

A Figura 2.9 traz a representação em árvore desenvolvida por Hosseini-Nasab *et al.* (2018) na qual são categorizados os problemas de layout. Ainda nessa figura estão realçados todos os aspectos relacionados à pesquisa da presente tese.

Neste capítulo foram analisadas as principais características do problema vinculadas aos modelos desta tese, como demanda dinâmica robusta, facilidades de áreas desiguais, locais de entrada e saída nas facilidades, restrições geométricas e corredores. O próximo capítulo traz os modelos desenvolvidos em si.

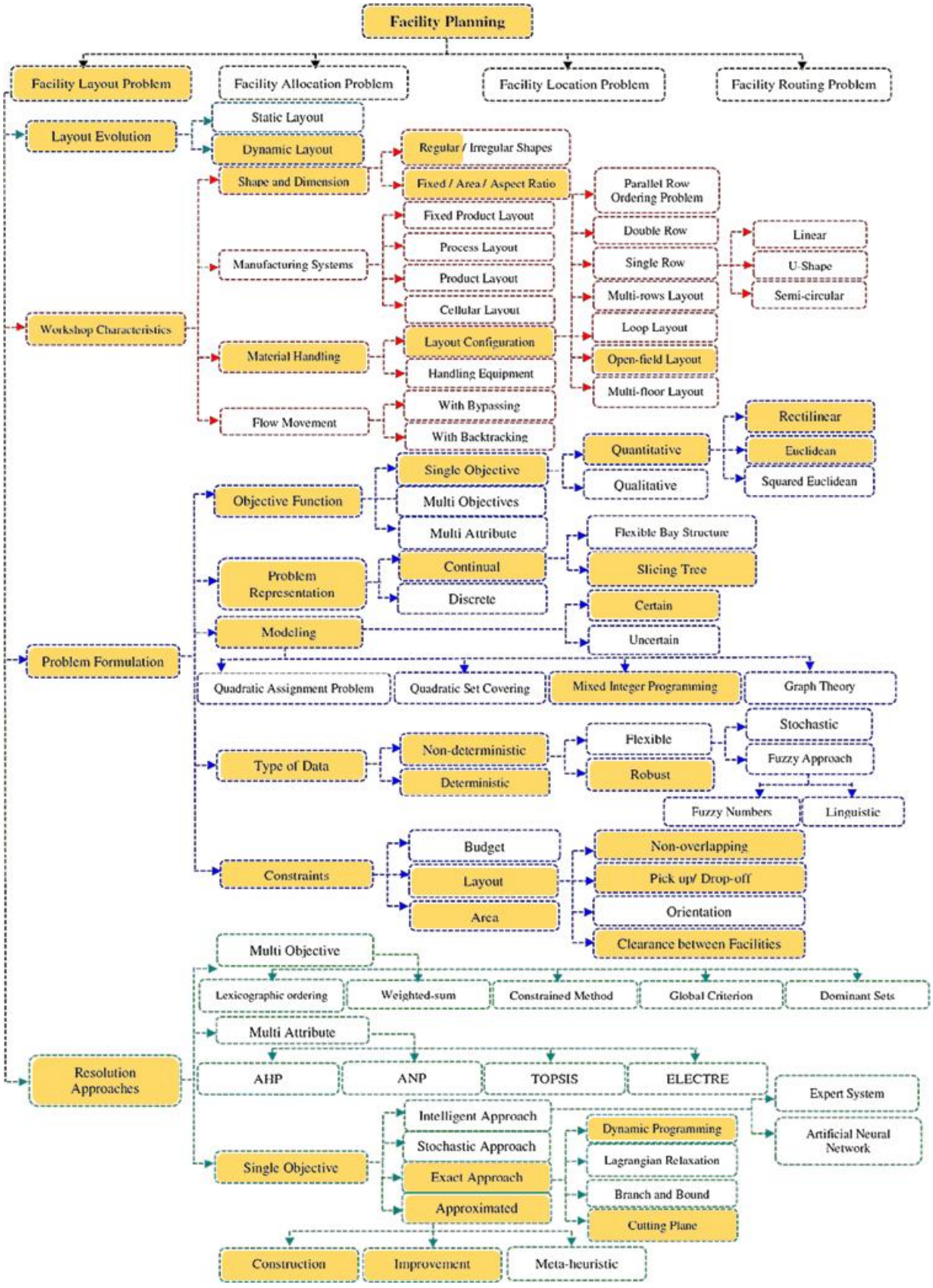

FIGURA 2.9 – Classificação dos problemas de layout (HOSSEINI-NASAB *et al.*, 2018).

3 Modelagem Matemática

3.1 Novos Modelos Matemáticos Propostos neste Trabalho

Este capítulo apresenta em detalhes os novos modelos matemáticos para otimização de layouts, principal contribuição desta pesquisa para a área de concentração de interesse.

Para esse trabalho foram formulados cinco modelos de Programação Inteira Mista para o Problema de Layout de Facilidades Robusto de Áreas Desiguais com Locais de Entrada e Saída com as seguintes características:

- Medida de distância retilínea (consultar Figura 2.2);
- Demanda dinâmica com abordagem robusta considerando informações de fluxo convertidas em dados discretos, conforme apresentado na Seção 2.3;
- Facilidades regulares do tipo retangulares (blocos) sujeitas a um sistema *open-field*;
- Locais de entrada e saída fixos e variáveis.

Os primeiros quatro modelos contemplam a situação de facilidades com dimensões fixas de altura e largura e o último (Seção 3.4) o caso de áreas fixas, mas dimensões variáveis sujeitas a uma razão máxima entre elas (razão de aspecto). Como base para os MIP *models* desenvolvidos nesta pesquisa foram utilizados os modelos propostos por Salimpour *et al.* (2021) (abordagem robusta), Hu *et al.* (2007) e Leno *et al.* (2018) (dimensões fixas e I/O points), e Liu e Meller (2007) e Sherali *et al.* (2003) (dimensões variáveis).

Além dos cinco modelos completos, este capítulo também traz duas possibilidades de incremento aos modelos, referentes a restrições geométricas ou facilidades fixas e corredores entre facilidades. Por fim, também traz como contribuição a apresentação dos modelos em um formato alternativo linear para aplicação de *solvers* especializados em resolver problemas de otimização linear.

3.2 Programação Inteira Mista

Quando algumas ou todas as variáveis de um problema de otimização pertencem ao conjunto dos números inteiros, esse é representado por meio de um modelo de Programação Inteira. Quando todas as variáveis são inteiras o modelo é denominado Programação Inteira Pura; caso contrário, é denominado Programação Inteira Mista. Embora produzir soluções para programas inteiros possa parecer simples, por vezes trata-se de um problema NP-difícil.

Uma das primeiras formulações de Programação Inteira Mista para resolver problemas de layout contínuo foi introduzida por Montreuil (1991). Posteriormente, diversos autores formularam Problemas de Layout de Facilidades em superfícies contínuas como modelos de Programação Inteira Mista (MIP *models*; referências no capítulo anterior).

3.3 UAFLP com Dimensões Fixas

A princípio, concentrou-se em modelos para representar situações nas quais as dimensões das instalações são dadas e, no máximo, se alteram entre si (bloco rotacionado). Embora os quatro modelos considerando facilidades de dimensões fixas apresentados adiante sejam similares, cada um representa uma situação particular. A diferença entre eles está na condição de localização dos pontos de entrada e saída, como explicado a seguir:

- Modelo 1 (UAFLPfixo) - Locais de Entrada e Saída Fixos (em qualquer ponto do bloco, seja no centroide, em algum outro ponto no interior ou nas bordas);
- Modelo 2 (UAFLP1) - Locais de Entrada e Saída Variáveis restritos aos pontos médios das bordas;
- Modelo 3 (UAFLP2) - Locais de Entrada e Saída Variáveis restritos aos pontos de quinas;
- Modelo 4 (UAFLP3) - Locais de Entrada e Saída Variáveis restritos a qualquer ponto nas bordas.

A Figura 3.1 ilustra as três possibilidades para I/O *points* variáveis.

Como os modelos possuem função objetivo e diversas restrições idênticas entre si, considerou-se por bem dividi-los em duas partes, uma comum e uma particular, e apresentá-los em cinco subseções. A subseção 3.3.1 traz exatamente a função objetivo e as equações que se repetem para todos os modelos, ou seja, pode ser considerada como a primeira parte de qualquer um dos quatro modelos individualmente. As demais subseções subsequentes trazem a segunda parte de cada um dos modelos separadamente, pois nesse ponto

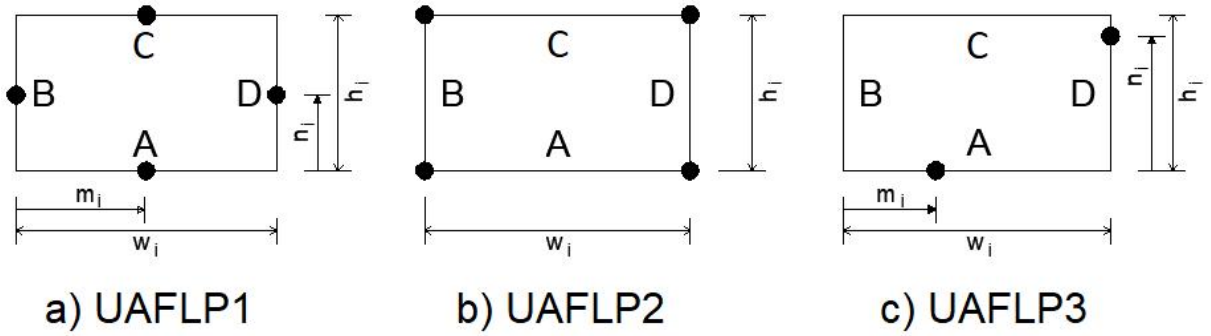

FIGURA 3.1 – Facilidades com locais candidatos aos I/O *points*: a) quatro possibilidades localizadas no meio das bordas do bloco; b) quatro opções localizadas nos cantos do bloco; e c) podem assumir infinitas posições ao redor do bloco nas bordas.

as equações são diferentes e caracterizam condições específicas de localização das estações de entrada e saída. As notações, válidas para todos os modelos, constam na Tabela 3.1.

3.3.1 1ª parte dos Modelos 1, 2, 3 e 4

$$\min MHC = \sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{j=1}^{N} f_{ijt}d_{ij} \tag{3.1}$$

sujeito a

$$wr_i = (1-u_i)w_i + u_i h_i \qquad \forall i \tag{3.2}$$

$$hr_i = (1-u_i)h_i + u_i w_i \qquad \forall i \tag{3.3}$$

$$xs_i = x_i + wr_i \qquad \forall i \tag{3.4}$$

$$ys_i = y_i + hr_i \qquad \forall i \tag{3.5}$$

$$l_{ij} + l_{ji} + b_{ij} + b_{ji} \geq 1 \qquad \forall i,j; i<j \tag{3.6}$$

$$xs_i \leq l_{ij}x_j + W(1-l_{ij}) \qquad \forall i,j \tag{3.7}$$

$$ys_i \leq b_{ij}y_j + H(1-b_{ij}) \qquad \forall i,j \tag{3.8}$$

$$d_{ij} = \left|x_i^O - x_j^I\right| + \left|y_i^O - y_j^I\right| \qquad \forall i,j \tag{3.9}$$

$$d_{ij}, wr_i, hr_i, x_i, y_i, xs_i, ys_i, x_i^{I(O)}, y_i^{I(O)}, m_i^{I(O)}, n_i^{I(O)} \geq 0 \qquad \forall i \tag{3.10}$$

$$u_i, v_i, p_i^{I(O)}, q_i^{I(O)} \in \{0,1\} \qquad \forall i \tag{3.11}$$

$$l_{ij}, b_{ij} \in \{0,1\} \qquad \forall i,j \tag{3.12}$$

Conjuntos e Índices	
N	número total de facilidades
T	número total de períodos
i e j	índices para facilidades = (1,...,N)
t	índice para períodos = (1,...,T)
Dados de Entrada	
f_{ijt}	fluxo de peças entre as facilidades i e j no período t
W	largura do espaço disponível
H	altura do espaço disponível
w_i	largura original da facilidade i
h_i	altura original da facilidade i
Ix_i, Iy_i	coordenadas x e y originais da entrada da facilidade i
Ox_i, Oy_i	coordenadas x e y originais da saída da facilidade i
Variáveis Contínuas	
d_{ij}	distância retilínea entre a saída da facilidade i e a entrada de j
wr_i	largura real da facilidade i (considera a rotação)
hr_i	altura real da facilidade i (considera a rotação)
x_i, y_i	coordenadas x e y do canto inferior esquerdo da facilidade i
xs_i, ys_i	coordenadas x e y do canto superior direito da facilidade i
x_i^I, y_i^I	coordenadas x e y da entrada da facilidade i
x_i^O, y_i^O	coordenadas x e y da saída da facilidade i
m_i^I, n_i^I	posição perimetral nas direções x e y da entrada da facilidade i
m_i^O, n_i^O	posição perimetral nas direções x e y da saída da facilidade i
Variáveis Binárias	
l_{ij}	(posição relativa) 1 se a facilidade i está localizada totalmente à esquerda de j, 0 cc.
b_{ij}	(posição relativa) 1 se a facilidade i está localizada totalmente abaixo de j, 0 cc.
(u_i, v_i)	(0,0) se a facilidade i está na sua orientação original; (1,0) se a facilidade i está rotacionada 90° no sentido horário; (0,1) se a facilidade i está rotacionada 180° no sentido horário; (1,1) se a facilidade i está rotacionada 270° no sentido horário
(p_i^I, q_i^I)	define em qual lado a entrada da facilidade i está localizada: (0,0) se no lado A; (1,0) se no lado B; (0,1) se no lado C; (1,1) se no lado D
(p_i^O, q_i^O)	define em qual lado a saída da facilidade i está localizada: (0,0) se no lado A; (1,0) se no lado B; (0,1) se no lado C; (1,1) se no lado D
Marcadores	
A, B, C, D	lados das facilidades (inferior, esquerdo, superior e direito)
Observação:	cada restrição com a notação I(O) equivale a duas restrições distintas, uma com a notação I e uma com a notação O. Por exemplo, $x_i^{I(O)} \geq 0$ quer dizer que $x_i^I \geq 0$ e $x_i^O \geq 0$.

TABELA 3.1 – Notações usadas nos Modelos 1, 2, 3 e 4

3.3.2 2ª parte do Modelo 1 (UAFLPfixo)

$$m_i^{I(O)}, n_i^{I(O)}, p_i^{I(O)}, q_i^{I(O)} = 0 \qquad \forall i \tag{3.13}$$

$$\begin{aligned} x_i^{I(O)} = x_i + (1-u_i)(1-v_i)I(O)x_i + u_i(1-v_i)I(O)y_i \\ +(1-u_i)v_i(w_i - I(O)x_i) + u_i v_i(h_i - I(O)y_i) \qquad \forall i \end{aligned} \tag{3.14}$$

$$\begin{aligned} y_i^{I(O)} = y_i + (1-u_i)(1-v_i)I(O)y_i + u_i(1-v_i)(w_i - I(O)x_i) \\ +(1-u_i)v_i(h_i - I(O)y_i) + u_i v_i I(O)x_i \qquad \forall i \end{aligned} \tag{3.15}$$

3.3.3 2ª parte do Modelo 2 (UAFLP1)

$$m_i^{I(O)} = \frac{wr_i}{2} \qquad \forall i \tag{3.16}$$

$$n_i^{I(O)} = \frac{hr_i}{2} \qquad \forall i \tag{3.17}$$

$$x_i^{I(O)} = x_i + (1 - p_i^{I(O)})m_i^{I(O)} + p_i^{I(O)} q_i^{I(O)} wr_i \qquad \forall i \tag{3.18}$$

$$y_i^{I(O)} = y_i + p_i^{I(O)} n_i^{I(O)} + (1 - p_i^{I(O)}) q_i^{I(O)} hr_i \qquad \forall i \tag{3.19}$$

3.3.4 2ª parte do Modelo 3 (UAFLP2)

$$m_i^{I(O)} = 0 \qquad \forall i \tag{3.20}$$

$$n_i^{I(O)} = 0 \qquad \forall i \tag{3.21}$$

$$x_i^{I(O)} = x_i + p_i^{I(O)} wr_i \qquad \forall i \tag{3.22}$$

$$y_i^{I(O)} = y_i + q_i^{I(O)} hr_i \qquad \forall i \tag{3.23}$$

3.3.5 2ª parte do Modelo 4 (UAFLP3)

$$m_i^{I(O)} \leq wr_i \qquad \forall i \tag{3.24}$$

$$n_i^{I(O)} \leq hr_i \qquad \forall i \tag{3.25}$$

$$x_i^{I(O)} = x_i + (1 - p_i^{I(O)})m_i^{I(O)} + p_i^{I(O)} q_i^{I(O)} wr_i \qquad \forall i \tag{3.26}$$

$$y_i^{I(O)} = y_i + p_i^{I(O)} n_i^{I(O)} + (1 - p_i^{I(O)}) q_i^{I(O)} hr_i \qquad \forall i \tag{3.27}$$

3.3.6 Discussão dos Modelos

A função objetivo dos modelos (Expressão 3.1) busca minimizar o MHC total que é igual a somatória do produto dos fluxos de material pelas distâncias entre a saída de uma facilidade e a entrada de outra.

As restrições (3.2) e (3.3) definem a configuração das facilidades. Se $u_i = 0$, então a facilidade continua em sua configuração original; se $u_i = 1$, então o bloco é rotacionado 90°, assim a largura real (ou seja, a largura do bloco no layout definido pelo modelo) passa a ser a altura original e a altura real passa a ser a largura original. A Figura 3.2 ilustra as duas situações.

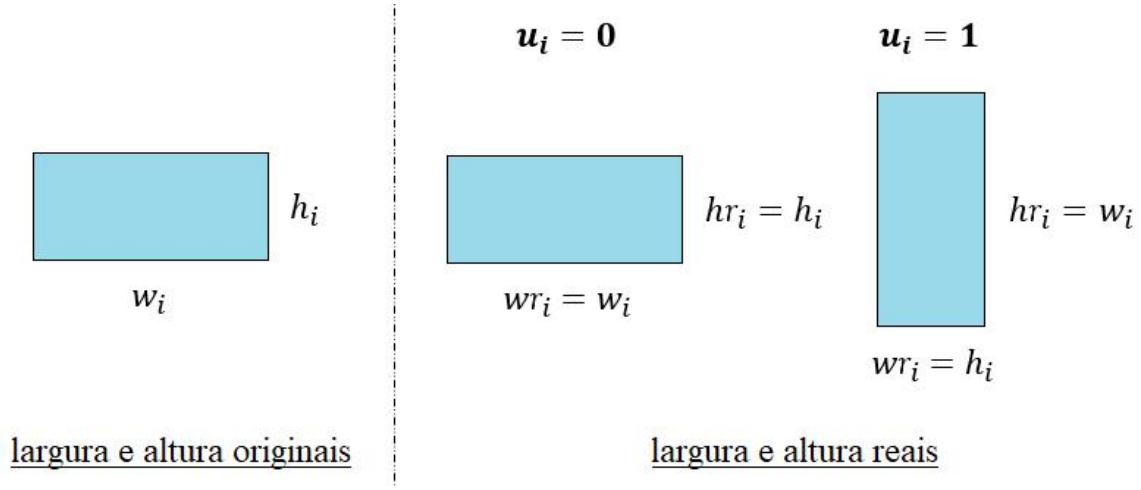

FIGURA 3.2 – Configurações de blocos.

As equações (3.4) e (3.5) delimitam os blocos por meio das coordenadas x_i,y_i do canto inferior esquerdo e xs_i,ys_i do canto superior direito. As inequações (3.6), (3.7) e (3.8) garantem que não ocorre sobreposição de peças, pois um bloco sempre está, pelo menos, ou totalmente à esquerda, ou totalmente à direita, ou totalmente acima ou totalmente abaixo de algum outro bloco. Além disso, (3.7) e (3.8) também delimitam os blocos ao espaço total disponível para o layout, tanto em largura quanto em altura. Em Leno *et al.* (2018), a restrição (3.6) é apresentada como uma equação, o que ressignifica as notações l_{ij} e b_{ij}, mas não interfere no resultado do modelo. Neste trabalho essa restrição foi escrita como uma desigualdade de forma similar à formulação do modelo de Hu *et al.* (2007).

Nos modelos propostos foi considerado como tipo de medida adotada a Métrica de Distância Retilínea (RDM). Porém, no caso de considerar distância Euclideana, basta trocar a restrição (3.9) pela seguinte:

$$d_{ij} = \sqrt{\left[x_i^O - x_j^I\right]^2 + \left[y_i^O - y_j^I\right]^2} \qquad \forall i, j \tag{3.28}$$

Outra opção é considerar a Métrica de Distância Perimetral (PDM), ou Distância

de Contorno (conceito muito bem apresentado por Leno *et al.* (2018) e reproduzido na sequência), que procura representar melhor uma situação real na qual não é possível transportar materiais passando por dentro de facilidades, atravessando-as, mas, sim, somente por corredores ou áreas vazias. Para calcular essa distância é preciso programar um algoritmo para construir um grafo que represente os caminhos de fluxo viáveis. Para isso, primeiramente constroi-se uma malha gráfica (*grid*) na qual as linhas horizontais e verticais atravessam todos os pontos de quinas e também de entrada e saída dos blocos. A interseção de cada linha horizontal e vertical define os pontos de *grid*. Os pontos que aparecem dentro das facilidades são removidos do conjunto de pontos de *grid* original e os restantes formam o conjunto V de vértices (*vertices*) ou nós do grafo. As linhas de grade que tocavam pontos removidos também são extintas; as demais formam o conjunto E de arestas (*edges*) do grafo (conectando todos os pontos adjacentes). A cada aresta é atribuído um peso correspondente ao comprimento da aresta em questão (que é a distância real). Dado o grafo $G = (V, E)$, a menor distância d_{ij} da estação de saída da facilidade i até a estação de entrada da facilidade j (e entre todos os pares de facilidades) é calculada usando o algoritmo de Dijkstra criado para resolver problemas de caminho mínimo (*shortest-path problem*). O algoritmo desenvolvido por Dijkstra (1959) é amplamente conhecido na área de ciência da computação e está bem explicado no livro de Cormen *et al.* (2009) que serviu como base para a programação desta etapa nesta pesquisa. A Figura 3.3 traz a representação de uma malha (geradora do grafo) para um layout com três facilidades e mostra o caminho mínimo entre dois pontos.

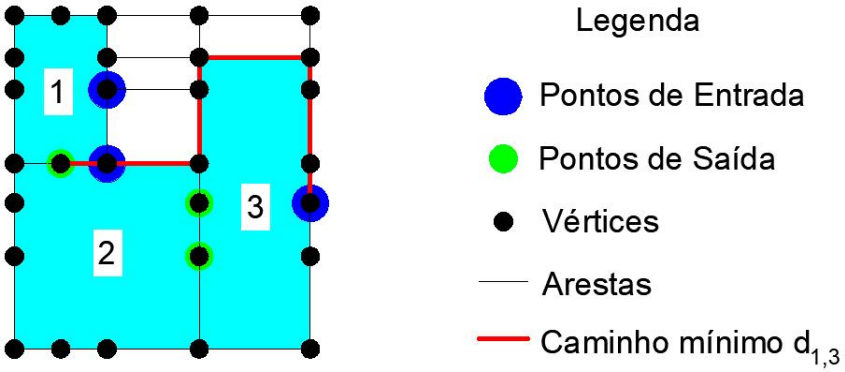

FIGURA 3.3 – *Grid* com caminhos de fluxo viáveis para um problema com três facilidades e marcação da distância de contorno $d_{1,3}$ entre a saída do bloco 1 e a entrada do bloco 3.

Como inicialmente não se sabe a posição dos retângulos e a localização dos I/O *points*, o grafo para o cálculo da distância de contorno só é montado depois de gerado o layout. Em (3.10), (3.11) e (3.12) são definidos os tipos de variáveis, sejam contínuas positivas ou binárias.

No Modelo 1, os I/O *points* são fixos, o que justifica a equação (3.13), pois as variáveis m, n, p e q não são necessárias. As restrições (3.14) e (3.15) definem as coordenadas dos I/O *points* dos blocos com base em como esses foram rotacionados para a forma-

ção do layout final (rotação de 90°, 180°, 270° ou não rotacionados). No caso dos I/O *points* do problema estarem fixados nos centroides dos blocos, essas restrições podem ser substituídas pelas seguintes:

$$x_i^{I(O)} = x_i + \frac{wr_i}{2} \qquad \forall i \tag{3.29}$$

$$y_i^{I(O)} = y_i + \frac{hr_i}{2} \qquad \forall i \tag{3.30}$$

No Modelo 2, os I/O *points* devem ser alocados aos pontos médios das bordas dos blocos. As equações (3.16), (3.17), (3.18) e (3.19) garantem o posicionamento desses em um dos quatro locais possíveis. No Modelo 3, as equações (3.20), (3.21), (3.22) e (3.23) estabelecem que a alocação dos I/O *points* em uma das quatro quinas dos blocos correspondentes. As restrições (3.18), (3.19), (3.22) e (3.23) foram modeladas de forma diferente da apresentada em Leno *et al.* (2018).

Por fim, no Modelo 4, as restrições (3.24), (3.25), (3.26) e (3.27) garantem que os I/O *points* estão posicionados em algum local nas bordas dos blocos. É interessante notar que o Modelo 4 tem duas equações semelhantes às do Modelo 2. O que diferencia os modelos são as restrições que definem $m_i^{I(O)}$ e $n_i^{I(O)}$, pois enquanto no Modelo 2 têm-se equações que forçam essas variáveis a serem iguais a metade das dimensões reais, no Modelo 4 têm-se inequações que dão liberdade para o valor de $m_i^{I(O)}$ e $n_i^{I(O)}$ desde que sejam menores que as dimensões reais. No modelo apresentado em Hu *et al.* (2007), as equações que definem $x_i^{I(O)}$ e $y_i^{I(O)}$ são diferentes de (3.26) e (3.27) e semelhantes às equações (3.14), (3.15) considerados neste trabalho para o UAFLPfixo. O caso UAFLP3 não é abordado em Leno *et al.* (2018).

3.4 UAFLP com Dimensões Variáveis

3.4.1 Modelo 5 (UAFLPvar)

No problema de layout de áreas desiguais com facilidades de dimensões variáveis são fornecidos previamente os valores das áreas das facilidades, porém as medidas de altura e largura podem variar, desde que respeitando uma razão máxima entre elas, chamada de *aspect ratio* ou razão de aspecto, que também deve estar predefinida nos dados de entrada.

O problema com facilidades de dimensões variáveis estudado neste trabalho pode ser representado pelo MIP *model* não-linear apresentado a seguir. Para não ficar repetitivo, ao invés de apresentar quatro modelos com as quatro opções de localização dos I/O *points*, é apresentado um único modelo considerando I/O *points* fixos localizados no centro dos blocos, como ilustrado na Figura 3.4. As notações do modelo constam na Tabela 3.2.

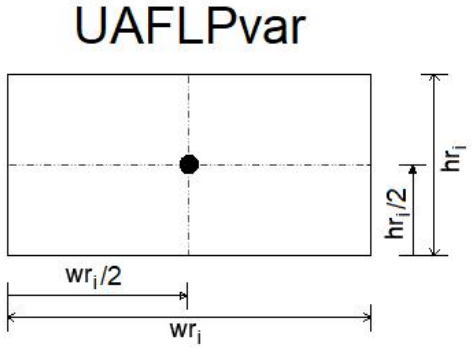

FIGURA 3.4 – Facilidade de dimensões variáveis com I/O *point* no centroide.

Conjuntos e Índices	
N	número total de facilidades
T	número total de períodos
i e j	índices para facilidades = (1,...,N)
t	índice para períodos = (1,...,T)
Dados de Entrada	
f_{ijt}	fluxo de peças entre as facilidades i e j no período t
W	largura do espaço disponível
H	altura do espaço disponível
a_i	área da facilidade i
α_i	razão de aspecto da facilidade i
Variáveis Contínuas	
d_{ij}	distância retilínea entre os centroides das facilidades i e j
wr_i	largura final da facilidade i
hr_i	altura final da facilidade i
x_i, y_i	coordenadas x e y do canto inferior esquerdo da facilidade i
xs_i, ys_i	coordenadas x e y do canto superior direito da facilidade i
x_i^I, y_i^I	coordenadas x e y da entrada da facilidade i
x_i^O, y_i^O	coordenadas x e y da saída da facilidade i
Variáveis Binárias	
lb_i	menor dimensão permitida para a facilidade i (*lower bound*)
ub_i	maior dimensão permitida para a facilidade i (*upper bound*)
l_{ij}	posição relativa: é igual a 1 se a facilidade i está localizada totalmente à esquerda de j; e igual a 0 caso contrário
b_{ij}	posição relativa: é igual a 1 se a facilidade i está localizada totalmente abaixo de j; e igual a 0 caso contrário
Observação:	cada restrição com a notação I(O) equivale a duas restrições distintas, uma com a notação I e uma com a notação O. Por exemplo, $x_i^{I(O)} \geq 0$ quer dizer que $x_i^I \geq 0$ e $x_i^O \geq 0$.

TABELA 3.2 – Notações usadas no modelo para facilidades com dimensões variáveis.

$$\min \sum_{t=1}^{T} \sum_{i=1}^{n} \sum_{j=1}^{n} f_{ijt} d_{ij} \tag{3.31}$$

sujeito a

$$a_i = wr_i hr_i \qquad \forall i \tag{3.32}$$

$$ub_i = \sqrt{\alpha_i a_i} \qquad \forall i \tag{3.33}$$

$$lb_i = \frac{\sqrt{\alpha_i a_i}}{\alpha_i} \qquad \forall i \tag{3.34}$$

$$lb_i \leq wr_i \leq ub_i \qquad \forall i \tag{3.35}$$

$$lb_i \leq hr_i \leq ub_i \qquad \forall i \tag{3.36}$$

$$xs_i = x_i + wr_i \qquad \forall i \tag{3.37}$$

$$ys_i = y_i + hr_i \qquad \forall i \tag{3.38}$$

$$l_{ij} + l_{ji} + b_{ij} + b_{ji} \geq 1 \qquad \forall i, j; i < j \tag{3.39}$$

$$xs_i \leq l_{ij} x_j + W(1 - l_{ij}) \qquad \forall i, j \tag{3.40}$$

$$ys_i \leq b_{ij} y_j + H(1 - b_{ij}) \qquad \forall i, j \tag{3.41}$$

$$d_{ij} = \left| x_i^O - x_j^I \right| + \left| y_i^O - y_j^I \right| \qquad \forall i, j \tag{3.42}$$

$$x_i^{I(O)} = x_i + \frac{wr_i}{2} \qquad \forall i \tag{3.43}$$

$$y_i^{I(O)} = y_i + \frac{hr_i}{2} \qquad \forall i \tag{3.44}$$

$$d_{ij}, wr_i, hr_i, x_i, y_i, xs_i, ys_i, x_i^{I(O)}, y_i^{I(O)}, lb_i, ub_i \geq 0 \qquad \forall i \tag{3.45}$$

$$l_{ij}, b_{ij} \in \{0, 1\} \qquad \forall i, j \tag{3.46}$$

3.4.2 Discussão do Modelo

É possível observar que esse modelo tem diversas similaridades com os modelos para facilidades de dimensões fixas, com a função objetivo e muitas restrições sendo as mesmas.

A função objetivo consiste em minimizar o MHC que é igual a somatória do produto dos fluxos de material (parâmetro) pelas distâncias entre as facilidades (variável) para todos os períodos (consonante com os modelos da seção anterior).

A restrição (3.32) garante que o produto das dimensões finais de cada facilidade corresponda a área desejada. Por definição, temos que:

$$a_i = ub_i lb_i \qquad \forall i \tag{3.47}$$

$$\alpha_i = \frac{ub_i}{lb_i} \qquad \forall i \tag{3.48}$$

Com estes dois conjuntos de equações é possível calcular ub_i e lb_i em função de valores fornecidos nos dados de entrada, conforme consta nas equações (3.33) e (3.34).

As restrições (3.35) e (3.36) delimitam a largura e altura de cada facilidade de forma a respeitar a razão de aspecto. Esta pode ser considerada a principal modificação em relação aos modelos para facilidades de dimensões fixas. Nos modelos anteriores, wr e hr (a largura e a altura real da facilidade) tinham como base as dimensões originais fornecidas nos dados de entrada e a rotação da peça. Mas neste modelo (3.31-3.46), wr e hr possuem maior liberdade de variação, sendo a única exigência respeitarem os limites mínimo e máximo definidos pelo *lower bound* e pelo *upper bound*.

As equações (3.37) e (3.38) delimitam os blocos por meio das coordenadas x_i,y_i do canto inferior esquerdo e xs_i,ys_i do canto superior direito. As inequações (3.39), (3.40) e (3.41) garantem que não ocorre sobreposição de peças, pois um bloco sempre está, pelo menos, ou totalmente à esquerda, ou totalmente à direita, ou totalmente acima ou totalmente abaixo de algum outro bloco. Além disso, (3.40) e (3.41) também delimitam os blocos ao espaço total disponível para o layout, tanto em largura quanto em altura.

No conjunto de restrições (3.42), é calculada a distância entre os I/O *points* (localizados nos centroides das peças conforme imposto nas equações (3.43) e 3.44). O tipo de medida considerada é a Métrica de Distância Retilínea, podendo esta ser substituída por outra medida, como a distância Euclidiana, se desejado. Em (3.45) e (3.46) são definidos os tipos de variáveis, sejam contínuas positivas ou binárias.

3.5 Restrições Geométricas ou Facilidades Fixas

Com o desenvolvimento do trabalho, achou-se interessante estudar a possibilidade de considerar facilidades pré-fixadas ou restrições geométricas nos modelos. Como comentado na Seção 2.6, quando consideramos um espaço retangular total disponível para alocação, é possível que existam regiões no layout que não podem ser disponibilizadas integralmente às facilidades por já estarem parcialmente ocupadas, seja com construções, tais como elevadores, escadas, pilares, entre outros, ou mesmo com outras facilidades previamente fixadas.

Para acrescentar essa possibilidade nos modelos é criado um novo conjunto NF contendo o índice das facilidades fixas, sendo suas coordenadas incluídas nos dados de entrada, conforme Tabela 3.3.

Nos modelos, algumas restrições são adicionadas de forma a associar as facilidades de índice pertencente ao conjunto NF aos parâmetros fixados, como apresentado a seguir. No caso de a facilidade fixa ser um espaço vazio/ocupado (restrição geométrica), basta considerar os fluxos de materiais relacionados iguais a zero.

Conjuntos e Índices	
NF	conjunto de facilidades fixas
Dados de Entrada	
xf_i, yf_i	coordenadas x e y do canto inferior esquerdo da facilidade fixa i
xsf_i, ysf_i	coordenadas x e y do canto superior direito da facilidade fixa i
xf_i^I, yf_i^I	coordenadas x e y da entrada da facilidade fixa i
xf_i^O, yf_i^O	coordenadas x e y da saída da facilidade fixa i

TABELA 3.3 – Notações acrescentadas aos modelos anteriores para incluir um conjunto de facilidades fixas ou restrições geométricas.

$$x_i = xf_i \qquad \forall i \in NF \tag{3.49}$$
$$y_i = yf_i \qquad \forall i \in NF \tag{3.50}$$
$$xs_i = xsf_i \qquad \forall i \in NF \tag{3.51}$$
$$ys_i = ysf_i \qquad \forall i \in NF \tag{3.52}$$
$$x_i^{I(O)} = xf_i^{I(O)} \qquad \forall i \in NF \tag{3.53}$$
$$y_i^{I(O)} = yf_i^{I(O)} \qquad \forall i \in NF \tag{3.54}$$

As instâncias de Tam (1992) reproduzidas na Figura 3.5 exemplificam como seria um cenário inicial contemplando restrições geométricas.

3.6 Corredores entre Facilidades

Outra possibilidade disponibilizada para inclusão nos modelos foram os corredores, ou *clearence distance*, entre facilidades. Na Figura 3.6 foram acrescentados corredores manualmente ao projeto de layout para exemplo visual.

A modificação nos modelos consistiu em criar o parâmetro $aisle$ com a largura do corredor e alterar duas inequações acrescentando esse novo parâmetro. Sendo assim, seja o dado de entrada $aisle$ igual a largura mínima do corredor entre facilidades, substituem-se as inequações:

$$xs_i \leq l_{ij} x_j + W(1 - l_{ij}) \qquad \forall i, j \tag{3.55}$$
$$ys_i \leq b_{ij} y_j + H(1 - b_{ij}) \qquad \forall i, j \tag{3.56}$$

por:

$$xs_i \leq l_{ij}(x_j - aisle) + W(1 - l_{ij}) \qquad \forall i, j \tag{3.57}$$
$$ys_i \leq b_{ij}(y_j - aisle) + H(1 - b_{ij}) \qquad \forall i, j \tag{3.58}$$

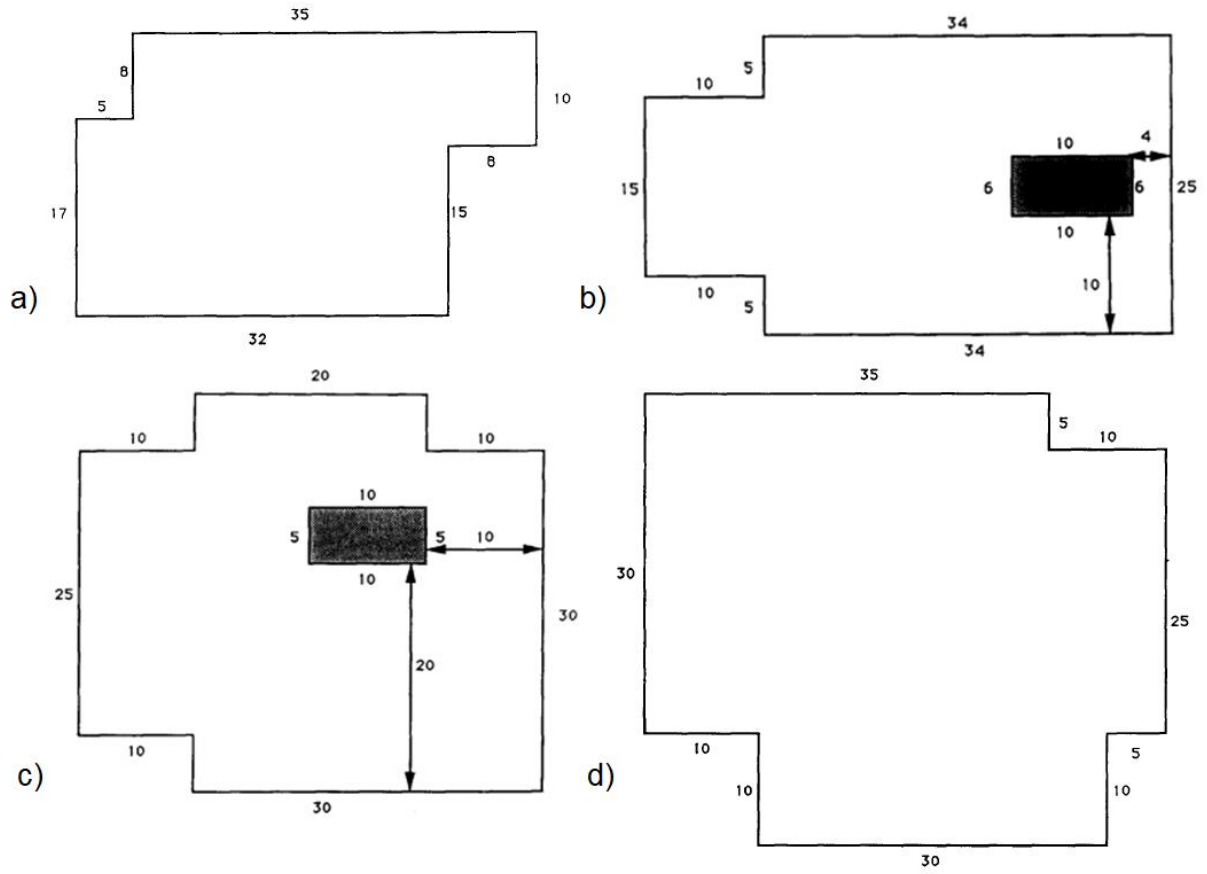

FIGURA 3.5 – Plantas iniciais para instâncias com a) 12 b) 15 c) 20 e d) 30 facilidades (TAM, 1992).

Intuitivamente, poderia se pensar que incluir corredores no modelo não modificaria a posição relativa das facilidades entre si, pois bastaria afastá-las umas das outras como feito na Figura 3.6. Mas na prática, o resultado pode acabar sendo diferente, como podemos ver na Figura 3.7 na qual são comparadas as soluções ótimas com e sem distância de folga.

3.7 Linearização dos Modelos

A linearização é uma técnica usada para a obtenção de um modelo linear a partir do modelo não linear equivalente, de forma a fornecer soluções iguais ou aproximadas àquelas do problema original. Existem *softwares* desenvolvidos especificamente para resolver problemas lineares (no máximo, quadráticos) usando algoritmos de métodos exatos, como simplex, barreira de pontos interiores e *branch-and-bound*. Como, dependendo do caso, eles podem se sobressair em relação a *softwares* para problemas não lineares, uma das intenções da linearização é justamente adaptar a modelagem para a aplicação desses otimizadores na busca por resultados melhores e/ou mais rápidos. Pensando assim, os modelos de Programação Inteira Mista anteriormente apresentados foram linearizados e,

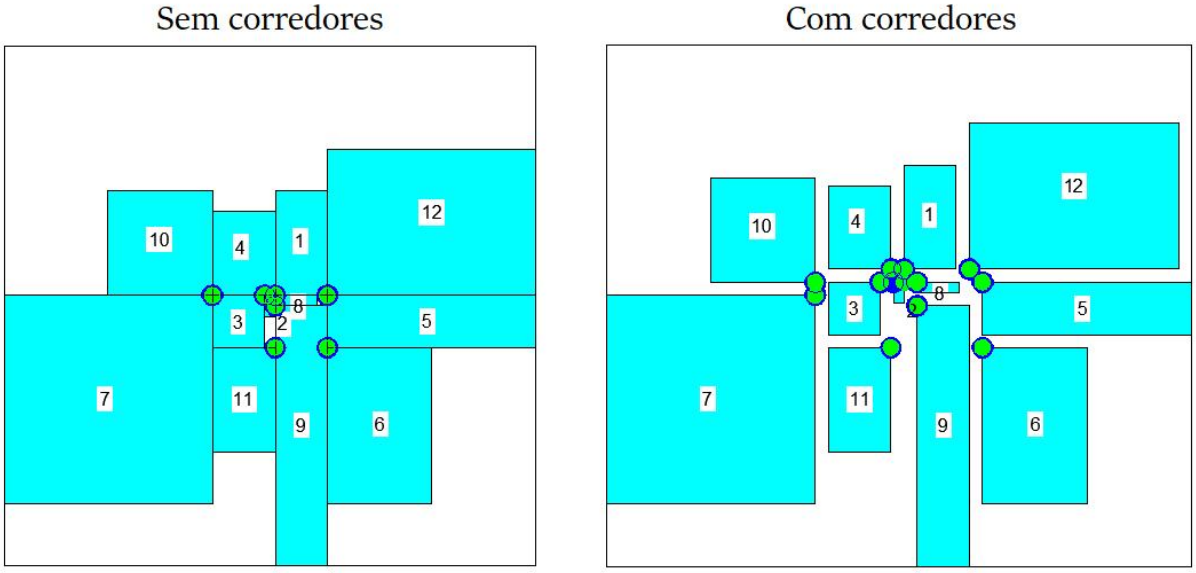

FIGURA 3.6 – Projeto final de layout para 12 facilidades sem distância de folga e com distância de folga.

assim, transformados em modelos de Otimização Linear Inteira Mista (MILO).

Como referência, para os modelos do UAFLP com dimensões fixas foi utilizada a técnica de linearização idealizada por Glover e Woolsey (1974) e apresentada por Mauri e Lorena (2009) na aplicação para Problemas Quadráticos Binários. Adaptando a técnica aos modelos de Layout de Facilidades, foram substituídos os termos quadráticos $l_{ij}x_i$, $b_{ij}y_i$, u_iv_i, $p_i^{I(O)}m_i^{I(O)}$, $p_i^{I(O)}n_i^{I(O)}$, $p_i^{I(O)}wr_i$ e $q_i^{I(O)}hr_i$ e cúbicos $p_i^{I(O)}q_i^{I(O)}wr_i$ e $p_i^{I(O)}q_i^{I(O)}hr_i$ (que nada mais são do que multiplicações de variáveis que aparecem nos modelos originais) pelas variáveis contínuas lx_{ij}, by_{ij}, uv_i, $p^{I(O)}m_i^{I(O)}$, $p^{I(O)}n_i^{I(O)}$, $p^{I(O)}wr_i$, $q^{I(O)}hr_i$, $p^{I(O)}q^{I(O)}wr_i$ e $p^{I(O)}q^{I(O)}hr_i$, e por restrições que garantissem a igualdade entre cada termo e sua variável correspondente.

Por exemplo, vamos considerar a desigualdade do modelo não linear na qual $xs_i \leq l_{ij}x_j + W(1-l_{ij})\ \forall i,j$. Para linearizar o termo quadrático preservando as regras do modelo original, a desigualdade pode ser substituída por estas restrições:

$$xs_i \leq lx_{ij} + W(1 - l_{ij}) \qquad \forall i,j \tag{3.59}$$

$$lx_{ij} - Wl_{ij} \leq 0 \qquad \forall i,j \tag{3.60}$$

$$lx_{ij} - x_j \leq 0 \qquad \forall i,j \tag{3.61}$$

$$W(l_{ij} - 1) + x_j - lx_{ij} \leq 0 \qquad \forall i,j \tag{3.62}$$

As restrições (3.60), (3.61) e (3.62) garantem que quando $l_{ij} = 0$ ou $x_j = 0$, então $lx_{ij} = 0$, e que quando $l_{ij} = 1$ e $x_j > 0$, então $lx_{ij} = x_j$. Portanto, garante-se que lx_{ij} em (3.59) será igual ao produto $l_{ij}x_j$.

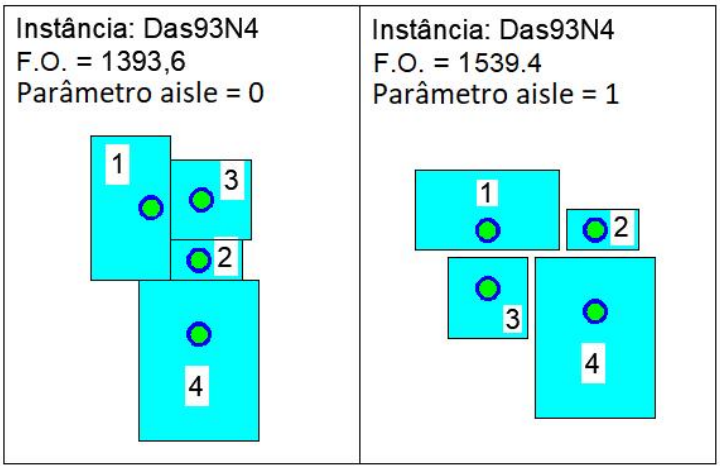

FIGURA 3.7 – Layouts ótimos encontrados para 4 facilidades sem distância de folga e com distância de folga.

Quanto à restrição referente ao cálculo da distância d_{ij}, foi utilizada uma técnica comum de linearização de módulos. Assim, os módulos das diferenças entre x_i^O e x_j^I e entre y_i^O e y_j^I foram convertidos em restrições equivalentes com a criação das variáveis dx_{ij} e dy_{ij}, cuja soma resulta em d_{ij}.

Os modelos linearizados são apresentados a seguir em quatro subseções, em formato próximo ao adotado na Seção 3.3, com a diferença de que a 2ª parte dos Modelos 2 e 4 foi condensada por ser muito semelhante. As notações são as mesmas que constam na Tabela 3.1, além das adicionais criadas que constam na Tabela 3.4.

Variáveis Contínuas	
dx_{ij}	variável artificial que substitui o binômio não-linear $\left\|x_i^O - x_j^I\right\|$
dy_{ij}	variável artificial que substitui o binômio não-linear $\left\|y_i^O - y_j^I\right\|$
lx_{ij}	variável artificial que substitui o termo não-linear $l_{ij}x_i$
by_{ij}	variável artificial que substitui o termo não-linear $b_{ij}y_i$
uv_i	variável artificial que substitui o termo não-linear $u_i v_i$
$p^{I(O)}m_i^{I(O)}$	variável artificial que substitui o termo não-linear $p_i^{I(O)}m_i^{I(O)}$
$p^{I(O)}n_i^{I(O)}$	variável artificial que substitui o termo não-linear $p_i^{I(O)}n_i^{I(O)}$
$p^{I(O)}wr_i$	variável artificial que substitui o termo não-linear $p_i^{I(O)}wr_i$
$q^{I(O)}hr_i$	variável artificial que substitui o termo não-linear $q_i^{I(O)}hr_i$
$p^{I(O)}q^{I(O)}wr_i$	variável artificial que substitui o termo não-linear $p_i^{I(O)}q_i^{I(O)}wr_i$
$p^{I(O)}q^{I(O)}hr_i$	variável artificial que substitui o termo não-linear $p_i^{I(O)}q_i^{I(O)}hr_i$

TABELA 3.4 – Notações adicionais usadas nos Modelos 1, 2, 3 e 4 linearizados

3.7.1 1ª parte dos Modelos 1, 2, 3 e 4

$$\min MHC = \sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{j=1}^{N} f_{ijt}d_{ij} \tag{3.63}$$

sujeito a

$$wr_i = (1-u_i)w_i + u_i h_i \qquad \forall i \tag{3.64}$$
$$hr_i = (1-u_i)h_i + u_i w_i \qquad \forall i \tag{3.65}$$
$$xs_i = x_i + wr_i \qquad \forall i \tag{3.66}$$
$$ys_i = y_i + hr_i \qquad \forall i \tag{3.67}$$
$$l_{ij} + l_{ji} + b_{ij} + b_{ji} \geq 1 \qquad \forall i,j; i<j \tag{3.68}$$
$$xs_i \leq lx_{ij} + W(1-l_{ij}) \qquad \forall i,j \tag{3.69}$$
$$ys_i \leq by_{ij} + H(1-b_{ij}) \qquad \forall i,j \tag{3.70}$$
$$lx_{ij} - Wl_{ij} \leq 0 \qquad \forall i,j \tag{3.71}$$
$$lx_{ij} - x_j \leq 0 \qquad \forall i,j \tag{3.72}$$
$$W(l_{ij}-1) + x_j - lx_{ij} \leq 0 \qquad \forall i,j \tag{3.73}$$
$$by_{ij} - Hb_{ij} \leq 0 \qquad \forall i,j \tag{3.74}$$
$$by_{ij} - y_j \leq 0 \qquad \forall i,j \tag{3.75}$$
$$H(b_{ij}-1) + y_j - by_{ij} \leq 0 \qquad \forall i,j \tag{3.76}$$
$$d_{ij} = dx_{ij} + dy_{ij} \qquad \forall i,j \tag{3.77}$$
$$dx_{ij} \geq x_i^O - x_j^I \qquad \forall i,j \tag{3.78}$$
$$dx_{ij} \geq x_j^I - x_i^O \qquad \forall i,j \tag{3.79}$$
$$dy_{ij} \geq y_i^O - y_j^I \qquad \forall i,j \tag{3.80}$$
$$dy_{ij} \geq y_j^I - y_i^O \qquad \forall i,j \tag{3.81}$$
$$\begin{aligned} &d_{ij}, wr_i, hr_i, x_i, y_i, xs_i, ys_i, x_i^{I(O)}, y_i^{I(O)}, m_i^{I(O)}, n_i^{I(O)}, \\ &lx_{ij}, by_{ij}, dx_{ij}, dy_{ij}, uv_i, p^{I(O)}m_i^{I(O)}, p^{I(O)}n_i^{I(O)}, \\ &p^{I(O)}wr_i, q^{I(O)}hr_i, p^{I(O)}q^{I(O)}wr_i, p^{I(O)}q^{I(O)}hr_i \geq 0 \qquad \forall i \end{aligned} \tag{3.82}$$
$$u_i, v_i, p_i^{I(O)}, q_i^{I(O)} \in \{0,1\} \qquad \forall i \tag{3.83}$$
$$l_{ij}, b_{ij} \in \{0,1\} \qquad \forall i,j \tag{3.84}$$

3.7.2 2ª parte do Modelo 1 (UAFLPfixo)

$$m_i^{I(O)} = n_i^{I(O)} = p_i^{I(O)} = q_i^{I(O)} = 0 \qquad \forall i \tag{3.85}$$

$$\begin{aligned} x_i^{I(O)} = x_i + (1 - u_i - v_i + uv_i)I(O)x_i + (u_i - uv_i)I(O)y_i \\ +(v_i - uv_i)(w_i - I(O)x_i) + uv_i(h_i - I(O)y_i) \qquad \forall i \end{aligned} \tag{3.86}$$

$$\begin{aligned} y_i^{I(O)} = y_i + (1 - u_i - v_i + uv_i)I(O)y_i + (u_i - uv_i)(w_i - I(O)x_i) \\ +(v_i - uv_i)(h_i - I(O)y_i) + uv_iI(O)x_i \qquad \forall i \end{aligned} \tag{3.87}$$

$$uv_i - u_i \leq 0 \qquad \forall i \tag{3.88}$$

$$uv_i - v_i \leq 0 \qquad \forall i \tag{3.89}$$

$$u_i + v_i - uv_i \leq 1 \qquad \forall i \tag{3.90}$$

3.7.3 2ª parte dos Modelos 2 e 4 (UAFLP1 e UAFLP3)

$$m_i^{I(O)} \leq wr_i \text{ (se } m_i^{I(O)} = \frac{wr_i}{2}\text{, temos o UAFLP1)} \qquad \forall i \tag{3.91}$$

$$n_i^{I(O)} \leq hr_i \text{ (se } n_i^{I(O)} = \frac{hr_i}{2}\text{, temos o UAFLP1)} \qquad \forall i \tag{3.92}$$

$$x_i^{I(O)} = x_i + m_i^{I(O)} - p^{I(O)}m_i^{I(O)} + p^{I(O)}q^{I(O)}wr_i \qquad \forall i \tag{3.93}$$

$$y_i^{I(O)} = y_i + p^{I(O)}n_i^{I(O)} + q^{I(O)}hr_i - p^{I(O)}q^{I(O)}hr_i \qquad \forall i \tag{3.94}$$

$$p^{I(O)}m_i^{I(O)} - (w_i + h_i)p_i^{I(O)} \leq 0 \qquad \forall i,j \tag{3.95}$$

$$p^{I(O)}m_i^{I(O)} - m_i^{I(O)} \leq 0 \qquad \forall i,j \tag{3.96}$$

$$(w_i + h_i)(p_i^{I(O)} - 1) + m_i^{I(O)} - p^{I(O)}m_i^{I(O)} \leq 0 \qquad \forall i,j \tag{3.97}$$

$$p^{I(O)}n_i^{I(O)} - (w_i + h_i)p_i^{I(O)} \leq 0 \qquad \forall i,j \tag{3.98}$$

$$p^{I(O)}n_i^{I(O)} - n_i^{I(O)} \leq 0 \qquad \forall i,j \tag{3.99}$$

$$(w_i + h_i)(p_i^{I(O)} - 1) + n_i^{I(O)} - p^{I(O)}n_i^{I(O)} \leq 0 \qquad \forall i,j \tag{3.100}$$

$$q^{I(O)}hr_i - (w_i + h_i)q_i^{I(O)} \leq 0 \qquad \forall i,j \tag{3.101}$$

$$q^{I(O)}hr_i - hr_i \leq 0 \qquad \forall i,j \tag{3.102}$$

$$(w_i + h_i)(q_i^{I(O)} - 1) + hr_i - q^{I(O)}hr_i \leq 0 \qquad \forall i,j \tag{3.103}$$

$$p^{I(O)}q^{I(O)}wr_i - (w_i + h_i)p_i^{I(O)} \leq 0 \qquad \forall i,j \tag{3.104}$$

$$p^{I(O)}q^{I(O)}wr_i - (w_i + h_i)q_i^{I(O)} \leq 0 \qquad \forall i,j \tag{3.105}$$

$$p^{I(O)}q^{I(O)}wr_i - wr_i \leq 0 \qquad \forall i,j \tag{3.106}$$

$$(w_i + h_i)(p_i^{I(O)} + q_i^{I(O)} - 2) + wr_i - p^{I(O)}q^{I(O)}wr_i \leq 0 \qquad \forall i,j \tag{3.107}$$

$$p^{I(O)}q^{I(O)}hr_i - (w_i + h_i)p_i^{I(O)} \leq 0 \qquad \forall i, j \tag{3.108}$$

$$p^{I(O)}q^{I(O)}hr_i - (w_i + h_i)q_i^{I(O)} \leq 0 \qquad \forall i, j \tag{3.109}$$

$$p^{I(O)}q^{I(O)}hr_i - hr_i \leq 0 \qquad \forall i, j \tag{3.110}$$

$$(w_i + h_i)(p_i^{I(O)} + q_i^{I(O)} - 2) + hr_i - p^{I(O)}q^{I(O)}hr_i \leq 0 \qquad \forall i, j \tag{3.111}$$

3.7.4 2ª parte do Modelo 3 (UAFLP2)

$$m_i^{I(O)} = 0 \qquad \forall i \tag{3.112}$$

$$n_i^{I(O)} = 0 \qquad \forall i \tag{3.113}$$

$$x_i^{I(O)} = x_i + p^{I(O)}wr_i \qquad \forall i \tag{3.114}$$

$$y_i^{I(O)} = y_i + q^{I(O)}hr_i \qquad \forall i \tag{3.115}$$

$$p^{I(O)}wr_i - (w_i + h_i)p_i^{I(O)} \leq 0 \qquad \forall i, j \tag{3.116}$$

$$p^{I(O)}wr_i - wr_i \leq 0 \qquad \forall i, j \tag{3.117}$$

$$(w_i + h_i)(p_i^{I(O)} - 1) + wr_i - p^{I(O)}wr_i \leq 0 \qquad \forall i, j \tag{3.118}$$

$$q^{I(O)}hr_i - (w_i + h_i)q_i^{I(O)} \leq 0 \qquad \forall i, j \tag{3.119}$$

$$q^{I(O)}hr_i - hr_i \leq 0 \qquad \forall i, j \tag{3.120}$$

$$(w_i + h_i)(q_i^{I(O)} - 1) + hr_i - q^{I(O)}hr_i \leq 0 \qquad \forall i, j \tag{3.121}$$

3.7.5 Modelo 5 (UAFLPc)

Como já comentado, o modelo para facilidades de dimensões variáveis tem diversas equivalências com os modelos para facilidades de dimensões fixas, portanto nesta subseção é destacada somente a linearização da restrição não linear que é própria desse modelo.

Para fazer a linearização do modelo, no lugar da restrição de área exata $a_i = wr_i hr_i$, foi adotada uma aproximação externa poliédrica com base em um número adequado de suportes afins definido por Δ. Sendo Δ um número inteiro ≥ 2, o valor para o ponto de suporte afim $\overline{x}_{ik}$ é calculado da seguinte forma:

$$\overline{x}_{ik} = lb_i + \frac{k-1}{\Delta - 1}(ub_i - lb_i) \qquad \forall i; k = 1, 2, ..., \Delta \tag{3.122}$$

$$a_i\frac{wr_i}{2} + \overline{x}_{ik}^2\frac{hr_i}{2} \geq a_i\overline{x}_{ik} \qquad \forall i; k = 1, 2, ..., \Delta \tag{3.123}$$

$$\overline{x}_{ik} \geq 0 \qquad \forall i, k \tag{3.124}$$

Esta aproximação, retirada do artigo de Sherali *et al.* (2003) (onde está bem deta-

lhada), é puramente linear e não envolve nenhuma variável binária. Além disso, ela pode fornecer uma representação tão precisa quanto desejada, quanto maior o valor de Δ (em seu estudo, para $\Delta = 20$, o erro máximo de área gerado para resolver um problema de nove facilidades foi de só $0,3\%$ (SHERALI *et al.*, 2003)).

3.7.6 Restrições Geométricas ou Facilidades Fixas

No caso do problema com restrições geométricas, não é necessário nenhum tratamento especial em relação à linearização, pois as mesmas restrições válidas para os modelos não-lineares (Seção 3.5) também valem para os modelos linearizados.

3.7.7 Corredores entre Facilidades

Em relação à inclusão de corredores nos modelos linearizados, basta considerar, no lugar das inequações apresentadas para os modelos não-linearizados (Seção 3.6), as seguintes:

$$xs_i \leq lx_{ij} - l_{ij}aisle + W(1 - l_{ij}) \qquad \forall i, j \tag{3.125}$$

$$ys_i \leq by_{ij} - b_{ij}aisle + H(1 - b_{ij}) \qquad \forall i, j \tag{3.126}$$

3.8 Modelos Matemáticos e Métodos de Resolução

Neste capítulo foram apresentados os modelos de Programação Inteira Mista desenvolvidos para este trabalho, sendo discutidas todas as condições e características dos mesmos. Foi analisado tanto o Problema de Layout de Facilidades de Áreas Desiguais com facilidades de dimensões fixas quanto o com dimensões variáveis.

Também foram exploradas situações particulares do problema de layouts, como a de restrições geométricas no espaço disponível e a de corredores entre facilidades. Todos os modelos e as situações particulares foram linearizados expandindo assim as possibilidades de resolução computacional dos problemas.

No próximo capítulo são apresentados os diferentes métodos de resolução propostos para encontrar soluções para os problemas de layout estudados e a discussão desses métodos.

4 Métodos de Resolução

Uma vez definidos os modelos que representam matematicamente a situação dos problemas de layout estudados, elaborados para a pesquisa desta tese, o próximo passo adotado foi desenvolver e aplicar alguns métodos para resolver estes problemas.

Foram desenvolvidos quatro métodos com diferentes abordagens envolvendo tanto programação matemática quanto heurísticas. Para cada método foi adotada uma nomenclatura, a saber:

1. PM-MNL (Programação Matemática com Modelos Não-Lineares) - Envolve resolução em duas etapas associando programação matemática com o *solver* Baron e uma heurística de melhoramento (dois algoritmos);
2. PM-ML (Programação Matemática com Modelos Lineares) - Envolve resolução unicamente via programação matemática com o *solver* CPLEX considerando os modelos linearizados;
3. BLAR-CG (Busca Local e Agregação de Rankings usando Cortes Guilhotinados) - Envolve resolução em duas etapas associando Busca Local e Agregação de Rankings com formação de layouts por cortes guilhotinados;
4. BLAR-PM (Busca Local e Agregação de Rankings usando Programação Matemática) - Envolve resolução em duas etapas associando Busca Local e Agregação de Rankings com formação de layouts por combinação de heurística construtiva e programação matemática com o *solver* CPLEX;

A seguir os métodos são explicados em detalhes.

4.1 Método PM-MNL (Programação Matemática com Modelos Não-Lineares)

A medida que o tempo passa, os otimizadores computacionais para problemas matemáticos e de otimização se tornam mais poderosos, fornecendo soluções eficazes em um

tempo cada vez menor. Também os computadores evoluem, com melhores processadores e sistemas operacionais. Por conta disso, apesar de problemas UAFLP serem da classe NP-*hard*, a primeira ideia adotada nesta pesquisa para resolver os modelos para o problema de layout (antes destes serem linearizados) foi a de usar um *solver* para problemas não lineares, sendo escolhido para esse propósito o *solver* Baron.

Porém, embora o *solver* Baron seja eficiente, ele não se mostrou suficiente para fornecer soluções ótimas em tempo hábil para as instâncias testadas. Mesmo para problemas pequenos (de $N = 6$) e deixando o programa rodar por alguns dias, o otimizador não conseguiu chegar a um resultado comprovadamente ótimo (em um dos casos ele alcançou um resultado já conhecido da literatura como ótimo, mas não conseguiu provar que era o ótimo e, assim, continuou tentando melhorar a solução indefinidamente), o que é plausível visto que o problema é NP-*hard* como já comentado anteriormente. Para problemas maiores, os resultados se mostraram ruins mesmo após o programa rodar por muitas horas.

Por conta disso, foram desenvolvidas duas heurísticas para complementar os resultados obtidos com o Baron, resultando em um método de resolução com duas fases. Na primeira fase, os problemas são então pré-resolvidos pelo *solver* Baron que fornece determinados resultados depois de um tempo estipulado. A partir dessas soluções, na segunda fase é executada uma heurística de melhoramento cuja função se repete até não haver mais melhora da função objetivo. Os algoritmos para as duas abordagens são detalhados na sequência.

4.1.1 Algoritmo 1

A primeira ideia de algoritmo colocada em prática é simples e foi capaz de fornecer resultados em um tempo razoavelmente curto, mas acabou não se mostrando eficiente para algumas instâncias. A seguir é apresentado o Algoritmo 1.

Algoritmo 1: *PM-MNL1*

$s^0 =$ gerar solução inicial (Baron, t) ;
$k_0 \leftarrow 0$;
$r \leftarrow 0$;
repetir
  $dt_i = \sum_{j=1}^{N}(d_{ij} + d_{ji}) \qquad \forall i$;
  $k_r =$ Índice da facilidade com maior dt_i ;
  $s^* =$ Gerar solução ótima (Baron) ;
  $r = r + 1$;
até *atingir critério de parada* ($k_r = k_{r-1}$);
Saída: s^*.

Na primeira fase, o programa recebe como dados de entrada as informações originais das instâncias tiradas da literatura e roda o otimizador por um certo tempo (simbolizado por t) gerando uma solução inicial (1). k representa o índice de uma determinada facilidade e começa com o valor zero, pois a princípio não está associado a nenhuma facilidade (2). r representa o número de rodagens do *loop* e também inicia com o valor zero (3).

Em seguida, o algoritmo começa o processo de reposicionar as peças mais distantes através de uma heurística, da seguinte forma: primeiro ele calcula o somatório das distâncias de cada peça em relação às demais (5) e coloca um marcador naquela que tem sua entrada e saída mais distantes das outras (6); depois ele roda o *solver* até atingir um resultado ótimo considerando um cenário no qual todas as peças e seus I/O *points* estão fixos menos a peça marcada, reposicionando-a de forma otimizada (7). O contador r marca a iteração (8).

Realizada a primeira rodagem, o programa repete o processo. A segunda fase para quando a peça mais distante já foi reposicionada de forma ótima, não havendo mais melhoria na função objetivo (9) e a solução é entregue (10). A Figura 4.1 mostra o desenvolvimento do Algoritmo 1 em um exemplo com $N = 12$ e $t = 10h$ (instância tirada de Deb e Bhattacharyya (2005)).

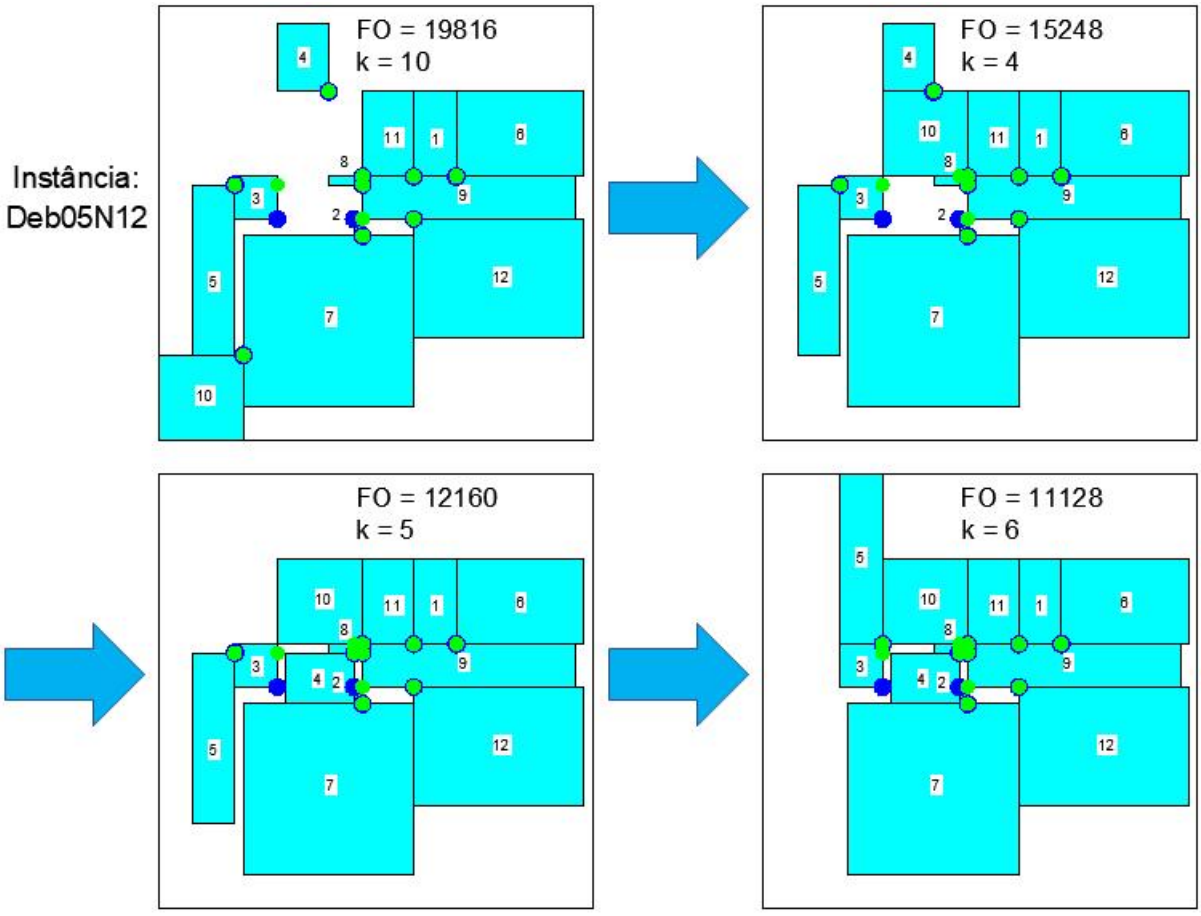

FIGURA 4.1 – Exemplo do processo de resolução do Algoritmo 1.

4.1.2 Algoritmo 2

Infelizmente, o Algoritmo 1 se mostrou ineficiente em alguns casos não contribuindo para uma melhora significativa de resultados. Na tentativa de corrigir essa situação, foi desenvolvido um algoritmo um pouco mais elaborado, o Algoritmo 2 a seguir:

Algoritmo 2: *PM-MNL2*

1 s^0 = Gerar solução inicial (Baron, t) ;
2 $r \leftarrow 0$;
3 $loop \leftarrow -1$;
4 **repetir**
5 $\quad k \leftarrow 0$;
6 $\quad$ **para** i = *1 até* N **faça**
7 $\quad\quad k \leftarrow k + 1$;
8 $\quad\quad s_i^*$ = gerar solução ótima (Baron) ;
9 $\quad$ **fim**
10 $\quad s^* = s_i^*$ com menor FO ;
11 $\quad loop = FO^* - \frac{\sum_{i=1}^{N} FO_i^*}{N}$;
12 $\quad r = r + 1$;
13 **até** *atingir critério de parada (loop = 0)*;
14 **Saída:** s*.

A primeira fase é idêntica à do outro algoritmo: o programa recebe como dados de entrada os dados originais das instâncias tiradas da literatura e roda o *solver* gerando uma solução inicial depois de um tempo t (1). r inicia valendo zero (2) e *loop* representa a melhoria na função objetivo começando com um valor diferente de zero (3).

Na segunda fase, a heurística já difere bastante da apresentada na primeira proposta. Ao invés de selecionar apenas a facilidade mais distante para ser reposicionada, o algoritmo faz esse reposicionamento com cada uma delas, uma por vez, aumentando as chances de encontrar um melhor resultado para o problema.

Funciona da seguinte maneira: partindo de $k = 0$ (5), sendo k o índice da facilidade flexível, o procedimento marca a primeira facilidade (7), roda o *solver* até atingir um resultado ótimo considerando um cenário no qual todas as facilidades e seus I/O *points* estão fixos menos a que foi marcada (8) e guarda a solução s_i^*. Depois repete o mesmo processo, porém marcando a segunda facilidade e, assim, sucessivamente, até guardar N resultados diferentes (6-9).

Dos N resultados, aquele com o menor valor de FO é salvo como a nova solução base (10), e os valores de *loop* (11) e r (12) são atualizados, finalizando a primeira rodagem. O processo continua a partir dessa nova solução. A segunda fase para quando nenhuma melhora de função objetivo é alcançada independente da peça sendo reposicionada (13). Por fim, obtém-se uma solução (14).

Para exemplificar a melhoria de resultado do Algoritmo 2 sobre o Algoritmo 1, partiu-se da mesma solução inicial apresentada na Figura 4.1 com $FO = 19816$. Enquanto o Algoritmo 1 alcançou uma $FO = 11128$, o Algoritmo 2 atingiu uma $FO = 10164$, ou seja, houve uma melhoria de $8,66\%$ no valor de função objetivo do layout final. A Tabela 4.1 traz esses resultados com mais detalhes. É interessante notar que a 2ª fase do Algoritmo 2, embora mais eficiente, também demanda mais tempo para ser executada, isto porque cada rodagem envolve N vezes mais otimizações, além de que acabam sendo realizadas mais rodagens do que no Algoritmo 1.

Algoritmo	FO	Rodagens	Tempo 2ª fase (s)
nenhum	19816	-	-
1	11128	4	50,016
2	10164	9	1144,79

TABELA 4.1 – Comparação entre os Algoritmos 1 e 2. Instância Deb05N12 (DEB; BHATTACHARYYA, 2005) e t = 10h.

Apesar de rodar indefinidamente mesmo para os problemas menores, não houve problema de memória na execução do programa para resolução com o Baron, porém não viu-se vantagem em deixar o programa rodando por um tempo excessivamente grande, visto que as heurísticas da segunda etapa já eram capazes de retornar melhores soluções a partir dos resultados gerados dentro de um certo tempo; assim ficou definido $t = 10h$.

4.2 Método PM-ML (Programação Matemática com Modelos Lineares)

Depois de resolver problemas com o *solver* Baron e constatar que os resultados não foram muito interessantes mesmo com apoio das heurísticas na segunda etapa dos algoritmos, foi proposta outra resolução via programação matemática, porém dessa vez usando um otimizador próprio para problemas lineares.

O primeiro passo foi linearizar os modelos de programação inteira inicialmente formulados. Feita a linearização, bastou implementar os modelos linearizados no programa e executá-lo com o *solver* escolhido, no caso, o CPLEX. O Algoritmo 3 ilustra a situação.

Algoritmo 3: *PM-ML*

1 s^0 = gerar solução inicial (CPLEX, t) ;
2 $s^* \leftarrow s^0$;
3 **Saída:** s^*.

Embora o processo de linearizar os modelos tenha sido mais trabalhoso, depois de

pronto tem-se um algoritmo de resolução que é extremamente simples, pois basta executar o programa definido um tempo limite.

Assim como no caso do PM-MNL, foi estabelecido um limite de tempo para o programa com o *solver* CPLEX para fins de padronização e esse foi definido após alguns testes de desempenho. Para a instância maior ($N = 62$) o programa apresentava erro de estouro de memória quando rodava por muito tempo, por isso fixou-se $t = 2h$.

4.3 Método BLAR-CG (Busca Local e Agregação de Rankings usando Cortes Guilhotinados)

Depois de dois métodos voltados para programação matemática, decidiu-se pesquisar uma abordagem totalmente diferente e desenvolver um método de resolução com procedimentos heurísticos não baseado nos modelos matemáticos.

Essa proposta foi direcionada especificamente para o Problema de Layout de Facilidades de Áreas Desiguais de dimensões variáveis limitadas por razões de aspecto, considerando Locais de Entrada e Saída posicionados nos centroides dos blocos e um espaço inicial sujeito a restrições geométricas. Como já discutido anteriormente, o caso de áreas ocupadas no interior da região inicial disponível para alocação das facilidades torna o problema mais complexo, não tendo sido muito estudado na literatura, o que aumenta o interesse.

Nessa resolução em duas etapas associa-se uma heurística de Busca Local e uma regra de Agregação de Rankings com formação de layouts por cortes guilhotinados.

4.3.1 Formulação do Problema

4.3.1.1 Formação do Layout

Assim como em outros estudos que consideram facilidades de áreas desiguais e dimensões variáveis, a estrutura de dados usada no Método BLAR-CG para gerar layouts é uma árvore binária. As folhas, ou nós externos, da árvore são preenchidas com números positivos referentes às facilidades do problema e sua ordenação define o posicionamento dos blocos no espaço disponível. Os nós internos são preenchidos com números negativos como resultado de um processo de construção. A árvore representada na Figura 4.2 é usada para exemplificar o processo de construção do layout.

Para formar um layout a partir da árvore binária, usou-se a ideia de cortes guilhotinados sucessivos em um retângulo maior. A Figura 4.3 ilustra o processo de geração de

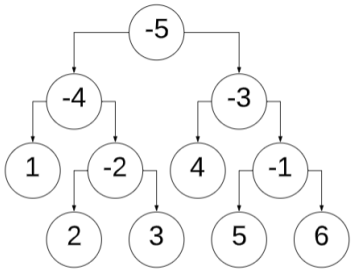

FIGURA 4.2 – Árvore binária.

layout descrito a seguir. Considere-se um espaço total de 15×10 e a área de cada uma das seis facilidades respectivamente igual a 50, 36, 24, 32, 6 e 2. O particionamento da região inteira se dá percorrendo recursivamente a árvore binária a partir de sua origem, verificando as áreas correspondentes aos ramos. Tomando como estrutura a árvore da Figura 4.2, vamos iniciar na raiz (nó -5) e calcular o somatório das áreas das facilidades que estão no seu ramo esquerdo ($50 + 36 + 24 = 110$). Considerando o valor obtido, a região é percorrida (da esquerda para a direita ou de baixo para cima, a depender da direção de particionamento) até ser obtida a área correspondente e então particionada. Na sequência, continuamos percorrendo a árvore padronizadamente da esquerda para a direita. Quando a partição for referente a área de uma única facilidade, esta facilidade é identificada no layout.

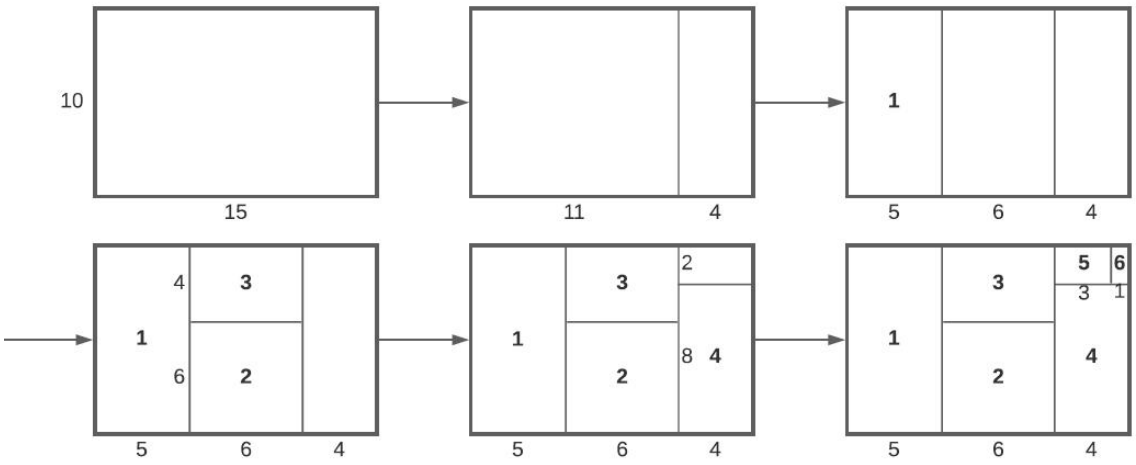

FIGURA 4.3 – Processo de geração do layout por particionamento a partir da árvore binária.

O processo continua até todos os ramos serem percorridos gerando enfim um layout com todas as facilidades. Como mencionado, a direção de particionamento interfere em como a região é percorrida e particionada. Procurando obter layouts que violem menos as razões de aspecto dos blocos, foi adotado um critério de definição da direção similar ao de Furtado e Lorena (1997). Para tanto, são verificadas as dimensões da região a ser cortada naquele momento: se a altura é menor do que a largura é realizado um corte vertical, caso

contrário, o corte executado é horizontal.

4.3.1.2 Construção da Árvore Binária

Para o Método BLAR-CG foram utilizados três processos de construção da árvore binária. O primeiro usa um método de aglomeração similar ao apresentado em Furtado e Lorena (1997), no qual facilidades com maior fluxo de peças entre si formam um aglomerado (representado por um grafo/árvore) e está explicitado nos passos a seguir. Seja $N = \{1, ..., n\}$ o conjunto de facilidades para alocação, $R = \{-1, ..., -(n-1)\}$ o conjunto de raízes e f_{ij} o fluxo de peças entre as facilidades i e j:

1. Inicialização da matriz de tráfego $\Theta_{n\times n} = [\theta_{ij}]$, com $\theta_{ij} = f_{ij} + f_{ji}$, $\forall i < j$ ($\theta_{ij} = 0$, $\forall i \geq j$) e de n aglomerados (vértices) formados pelas facilidades individuais. $k = 1$.

2. Geração de um novo aglomerado (com vértices e arestas) de raiz $-k$ combinando os aglomerados i e j para os quais θ_{ij} é máximo.

3. Atualização da matriz de tráfego: Linha e coluna i de θ_{ij} agora representam o novo aglomerado gerado e para tanto recebem novos valores de tráfego dados por $\theta_{(ij),h}$ que relaciona os antigos fluxos dos aglomerados i e j com os fluxos dos demais aglomerados remanescentes (representados por h); Linha e coluna j de θ_{ij} são zeradas.

4. Inspeção do novo aglomerado gerado: se este engloba todas as facilidades, ou seja, é o aglomerado final e, portanto, $k = n - 1$, então parar; se não, fazer $k = k + 1$ e voltar ao passo 2.

Os novos valores de tráfego mencionados no passo 3 são calculados relacionando os pesos dos aglomerados associados. Na fórmula a seguir, n_i é o número de facilidades no aglomerado i, n_j é o número de facilidades no aglomerado j e $n_{(ij)}$ é a soma de n_i com n_j:

$$\theta_{(ij),h} = \frac{n_i\theta_{ih} + n_j\theta_{jh}}{n_{(ij)}} \qquad \forall h \tag{4.1}$$

A Figura 4.4 ilustra um exemplo da aplicação do Processo 1 para formação da árvore a partir de uma matriz de fluxo. Seguindo os passos anteriores, inicialmente tem-se a matriz de tráfego $\Theta_{6\times6}$ e n aglomerados. Na primeira iteração, os aglomerados **2** e **3** vinculados ao maior valor de fluxo (63) geram um novo aglomerado **23** cuja raiz é **-1** conectada aos aglomerados **2**, pela esquerda, e **3**, pela direita, ou seja, um aglomerado com três vértices e duas arestas. A matriz de tráfego é atualizada. Na segunda iteração, os aglomerados **5** e **6** vinculados ao maior valor de fluxo (54) geram um novo aglomerado **56** cuja raiz é **-2** conectada aos aglomerados **5**, pela esquerda, e **6**, pela direita. A matriz de tráfego é

atualizada. Na terceira iteração, os aglomerados **1** e **23** vinculados ao maior valor de fluxo (45) geram um novo aglomerado **123** cuja raiz é **-3** conectada aos aglomerados **1**, pela esquerda, e **23**, pela direita, ou seja, um aglomerado com cinco vértices e quatro arestas. E assim por diante até a árvore estar completa.

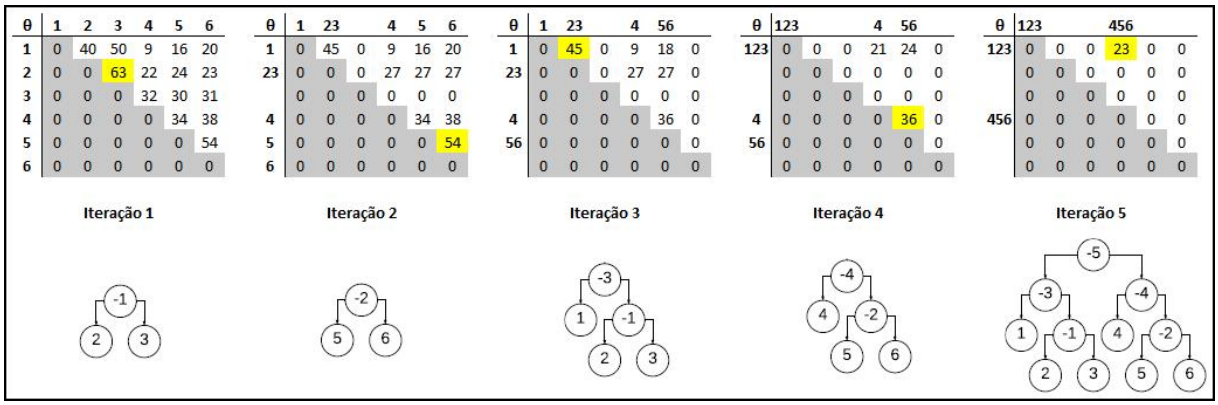

FIGURA 4.4 – Processo 1 de construção da árvore binária a partir da matriz de tráfego.

Note-se que este processo gera, além da árvore, também uma permutação, que é a sequência de facilidades que aparece nos nós externos (ver Seção 4.3.1.3).

O segundo processo é simples e consiste em preencher os nós internos a partir da raiz (sempre priorizando os nós mais internos) em ordem crescente, sequencialmente da esquerda para a direita. No terceiro processo, também se preenchem os nós internos a partir da raiz em ordem crescente, porém buscando uma distribuição mais equilibrada entre os ramos, com um preenchimento simétrico que gera uma imagem espelhada. Os Processos 2 e 3 não dependem da matriz de fluxo, somente do número de facilidades do problema, e fornecem exclusivamente estruturas de árvore, ao contrário do processo anterior que também gera uma permutação para as folhas. A Figura 4.5 mostra as árvores obtidas pelos dois processos para um problema com seis facilidades.

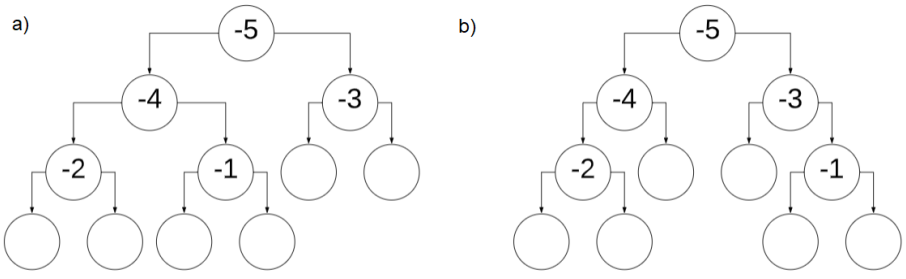

FIGURA 4.5 – Árvores binárias: a) Processo 2 - Distribuição sequencial; b) Processo 3 - Distribuição equilibrada.

4.3.1.3 Geração das Permutações de Entrada

Para a geração de um layout é necessário uma árvore binária completa com estrutura (nós internos) e folhas (nós externos). As folhas são preenchidas com facilidades, sendo que a ordem de preenchimento interfere diretamente em como se dá a alocação destas na organização do layout. A sequência de facilidades que compõem as folhas é tratada como uma permutação. Para aplicação nos testes computacionais foram propostos alguns procedimentos, resultando na geração de dez permutações a partir de heurísticas construtivas que analisam a matriz de fluxo, como detalhado a seguir:

- Permutação 1: gerada como resultado do Processo 1 descrito na Seção 4.3.1.2.
- Permutação 2: Inicia-se com a matriz de fluxo; Passo 1 - calcula-se o somatório dos fluxos associados a cada facilidade ($soma_i = \sum_{i=1}^{N}(f_{ij} + f_{ji})\ \forall i$); Passo 2 - ordenam-se os somatórios do maior para o menor, associando às respectivas facilidades; Passo 3 - alocam-se as facilidade no vetor da permutação partindo das posições centrais para as laterais.
- Permutação 3: Inicia-se com a matriz de fluxo e $iter = 0$; Passo 1 - dado $iter = iter + 1$, calcula-se o somatório dos fluxos associados a cada facilidade ($soma_i = \sum_{i=1}^{N}(f_{ij} + f_{ji})\ \forall i$), insere-se a facilidade com o maior somatório na posição $iter$ do vetor auxiliar e excluem-se linha e coluna da matriz de fluxo referentes à facilidade selecionada; Passo 2 - repete-se o Passo 1 até $iter = n$; Passo 3 - alocam-se as facilidade no vetor da permutação partindo das posições centrais para as laterais.
- Permutação 4: Inicia-se com a matriz de fluxo Θ_{nxn} e $iter = 0$; Passo 1 - dado $iter = iter+1$, inserem-se as facilidades i e j associadas ao maior θ_{ij} respectivamente nas posições $2iter-1$ e $2iter$ do vetor auxiliar e excluem-se linhas e colunas da matriz de fluxo Θ_{nxn} referentes às facilidades selecionadas; Passo 2 - repete-se o Passo 1 até $iter = n/2$; Passo 3 - alocam-se as facilidade no vetor da permutação partindo das posições centrais para as laterais.
- Permutação 5: semelhante à Permutação 4, porém no Passo 2 inserem-se as facilidades i e j associadas ao maior θ_{ij} respectivamente nas posições $2iter$ e $2iter - 1$ do vetor auxiliar.
- Permutações 6 a 10: equivalem as permutações de 1 a 5, porém adotando a ordem inversa no vetor de permutação.

Considerando como exemplo a matriz de fluxo da Figura 4.4 para seis facilidades, as

permutações geradas, de 1 a 10, ficam assim:

Permutação	1	1	2	3	4	5	6
Permutação	2	1	6	3	2	5	4
Permutação	3	1	2	3	6	5	4
Permutação	4	5	6	2	3	1	4
Permutação	5	6	5	3	2	4	1
Permutação	6	6	5	4	3	2	1
Permutação	7	4	5	2	3	6	1
Permutação	8	4	5	6	3	2	1
Permutação	9	4	1	3	2	6	5
Permutação	10	1	4	2	3	5	6

(4.2)

4.3.1.4 Restrições Geométricas

Como dito inicialmente, a proposta deste método inclui a possibilidade de restrições geométricas no espaço inicial destinado ao layout de facilidades. Estas barreiras podem ser construções prévias, facilidades já alocadas e que não podem ser removidas ou deslocadas, pilares de sustentação, elevadores, escadas, *shafts*, entre outros obstáculos, e resultam em regiões no layout que não permitem a alocação integral de uma facilidade no formato retangular durante o processo de particionamento. Essas áreas ocupadas que interferem no caminho normal do processo, precisam ser subtraídas da área a ser destinada à facilidade. Vamos considerar o exemplo da Seção 4.3.1.1, porém agora com um espaço total de 17×10 e dois espaços vazios (um em uma quina e um interno). A Figura 4.6 mostra onde seriam realizadas as novas partições, de forma a compensar as áreas perdidas (a linha pontilhada representa a partição no caso de um layout livre de restrições geométricas).

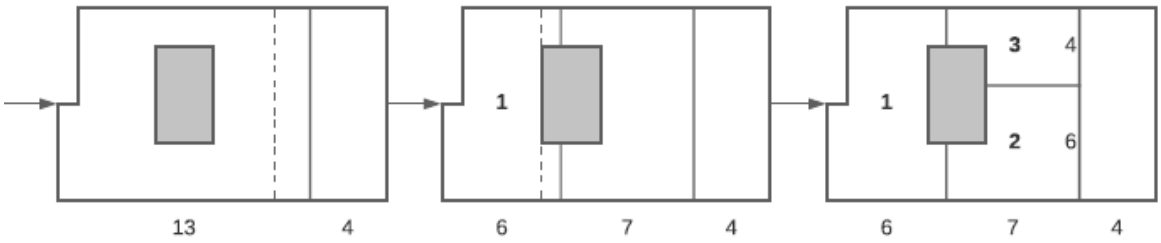

FIGURA 4.6 – Processo de particionamento em um layout com espaços ocupados.

Assim, neste exemplo, a área alocada ao bloco 1 é igual à área da partição retangular menos a porção de espaço já ocupada ($10 \times 6 - 2(5 \times 1)$). O mesmo vale para os blocos 2 ($7 \times 6 - 2 * 3$) e 3 ($7 \times 4 - 2 \times 2$).

Cada facilidade i recebe como dado de entrada uma razão de aspecto máxima e mínima dadas por α_i^{max} e α_i^{min}. Considerando wr_i e hr_i como largura e altura da facilidade i na

solução final encontrada, o cálculo da razão de aspecto é:

$$\alpha_i = \frac{wr_i}{hr_i} \qquad \forall i \tag{4.3}$$

Como a existência de áreas ocupadas dentro da facilidade acaba por distorcer a forma retangular da área utilizável, além da razão de aspecto é necessário providenciar uma nova medida, dessa vez que reflita o grau de distorção da forma. A razão de área morta (denotada por σ) é introduzida com esse propósito e pode ser definida como a área vazia sobre a área do retângulo, mais detalhadamente:

$$\sigma_i = \frac{\text{área do espaço ocupado na partição vinculada à facilidade } i}{\text{área da partição vinculada à facilidade } i} \qquad \forall i \tag{4.4}$$

No caso do exemplo da Figura 4.6, as razões de área morta dos blocos 1, 2 e 3 são, respectivamente, $\frac{10}{60} = 0,17$, $\frac{6}{42} = 0,14$ e $\frac{4}{28} = 0,14$. Se o layout não possuir áreas ocupadas, a razão de área morta das facilidades é igual a zero. Para cada facilidade podemos definir um limite superior σ_i^{max}, restringindo σ_i ao intervalo $[0, \sigma_i^{max}]$.

4.3.1.5 Função Objetivo

A função objetivo consiste em minimizar o MHC e tipo de medida considerada para o cálculo da distância entre os centroides das peças é a Métrica de Distância Retilínea, relembrando:

$$d_{ij} = \left|\left(x_i + \frac{wr_i}{2}\right) - \left(x_j + \frac{wr_j}{2}\right)\right| + \left|\left(y_i + \frac{hr_i}{2}\right) - \left(y_j + \frac{hr_j}{2}\right)\right| \qquad \forall i, j \tag{4.5}$$

No caso da formulação do problema usada na abordagem proposta, foi necessário converter as restrições geométricas de área e forma das facilidades em funções de penalidades e incorporá-las no cálculo do objetivo, dessa forma:

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} f_{ij} d_{ij} + \sum_{i=1}^{n} (w_i^{\gamma} \gamma_i + w_i^{\beta} \beta_i) \tag{4.6}$$

com

$$\gamma_i = \max\left[0, \max\left[\left(\alpha_i - \max\left[\alpha_i^{max}, \frac{1}{\alpha_i^{min}}\right]\right), \left(\min\left[\alpha_i^{min}, \frac{1}{\alpha_i^{max}}\right] - \alpha_i\right)\right]\right] \qquad \forall i \tag{4.7}$$

$$\beta_i = \max\left[0, \sigma_i - \sigma_i^{max}\right] \qquad \forall i \tag{4.8}$$

onde α_i^{max} e α_i^{min} são os limites superior e inferior de α_i, γ_i mede a extensão de violação da razão de aspecto e β_i mede a extensão de violação da razão de irregularidade da forma.

Considerando que cada facilidade tem um nível de tolerância para essas violações, γ_i e β_i são multiplicados, respectivamente, pelos fatores w_i^γ e w_i^β que atuam como pesos no segundo termo da função objetivo que representa a função de penalidades.

4.3.2 Abordagem Proposta

A abordagem proposta para o UAFLP de dimensões variáveis com restrições geométricas foi a resolução em duas etapas que consistem em uma Busca Local exaustiva e posterior Agregação de Rankings.

O método de Busca Local consiste em explorar o espaço de soluções movendo-se de uma solução para outra que seja um vizinho melhor. Seja S o conjunto de soluções do problema, no qual cada solução $s \in S$ tem um conjunto de vizinhos $M(s) \subset S$, a vizinhança $M(s)$ de s consiste então no conjunto de movimentos que levam de uma solução s para uma outra solução $s' \in M(s)$. O método de melhoramento começa a partir de uma solução inicial s_0 e a cada iteração explora a vizinhança $M(s)$ escolhendo uma nova solução corrente s' de $M(s)$ que apresente um valor melhor de função objetivo (no caso da minimização, um valor menor), até atingir um critério de parada. A ideia por trás da heurística de Busca Local é simples, mas possui a desvantagem de ficar presa em ótimos locais.

Aplicado ao problema de layout tratado, considere-se uma situação com 6 facilidades cuja permutação inicial associada às folhas da árvore seja o vetor $v_n = [1, 2, 3, 4, 5, 6]$. S é o espaço de dimensão $n!$ de todas as soluções possíveis. A vizinhança $M(s)$ é dada por todas as possíveis trocas entre duas facilidades dentro do vetor, sendo sua dimensão igual a soma $(n - 1 + n - 2 + ... + 1) = \frac{(n-1)n}{2}$. Um possível vizinho para v_n seria, por exemplo, $[2, 1, 3, 4, 5, 6]$, mediante o movimento de troca entre 1 e 2. Cada movimento no vetor corresponde a uma permutação de folhas da árvore binária e gera um novo layout com um valor diferente para a função objetivo.

A técnica conhecida como Agregação de Rankings (ou *Kemeny Ranking*) consiste em uma regra de classificação de um perfil de resultados e, mais especificamente, é considerada um modelo de agregação de preferências para encontrar soluções de consenso. É utilizada em uma variedade de áreas, especialmente nos de teoria da escolha social, aprendizagem de máquinas e informática teórica (ALI; MEILĂ, 2012).

Suponha-se que as cinco primeiras permutações do conjunto de vetores em (4.2), Seção 4.3.1.3, estejam associadas aos resultados da aplicação de heurísticas podendo ser consideradas como *clusters* hierárquicos. A partir dessas permutações encontramos, através da Agregação de Rankings (AR), um layout de consenso que represente os vários layouts que foram gerados. Primeiramente gera-se a matriz de precedências correspondente às permutações $Q = [Q_{ij}]$, dado $Q_{ij} = \frac{1}{n} \sum_{k=1}^{n} I(i \prec_{\pi_k} j)$, onde $i \prec_{\pi_k} j$ significa que i precede j

na permutação π_k:

$$Q = \begin{pmatrix} 1 & 0.6 & 0.6 & 0.8 & 0.6 & 0.6 \\ 0.4 & 1 & 0.6 & 1 & 0.6 & 0.4 \\ 0.4 & 0.4 & 1 & 1 & 0.6 & 0.4 \\ 0.2 & 0 & 0 & 1 & 0.2 & 0.2 \\ 0.4 & 0.4 & 0.4 & 0.8 & 1 & 0.4 \\ 0.4 & 0.6 & 0.6 & 0.8 & 0.6 & 1 \end{pmatrix} \tag{4.9}$$

Obtida a matriz de precedências, o problema AR pode ser formulado como um modelo de programação linear inteira (ALI; MEILĂ, 2012):

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} (Q_{ij} X_{ji} + Q_{ji} X_{ij}) \tag{4.10}$$

sujeito a

$$X_{ij} \in 0, 1 \qquad \forall i, j \tag{4.11}$$

$$X_{ij} + X_{ji} = 1 \qquad \forall i, j; i \neq j \tag{4.12}$$

$$X_{ij} + X_{jk} + X_{ki} \geq 1 \qquad \forall i, j, k \tag{4.13}$$

O primeiro conjunto de restrições define que X_{ij} são variáveis binárias (recebem o valor de 1 quando $i \prec_{\pi_0} j$); o segundo conjunto de restrições captura o fato de que ou $i \prec j$ ou $j \prec i$ na AR; e o terceiro conjunto de restrições reforça a transitividade lógica $i \prec j \wedge j \prec k \Rightarrow i \prec k$. Este modelo pode ser resolvido de forma exata para problemas médios (até 150 permutações). Resolvendo o modelo para a matriz de precedências (4.9), a saída é dada por:

$$X = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \tag{4.14}$$

Logo, a permutação correspondente é $\pi_0^* = [1, 3, 4, 6, 5, 2]$, sendo π_0^* o ranking que minimiza o número total de desacordos com as classificações contidas no perfil, chamado de *Kemeny Ranking* desse perfil. Encontrada a permutação de consenso, basta transformá-la em layout e verificar se a nova solução encontrada é melhor do que as soluções contidas no perfil.

No algoritmo de duas fases que foi usado para os testes computacionais (Algoritmo 4), foi aplicada primeiro a Busca Local em cada uma das dez soluções iniciais (permutações

geradas conforme detalhado na Seção 4.3.1.3). A Busca Local é interrompida quando não é mais possível encontrar uma nova melhor solução na vizinhança (critério de parada, linha (10) do código). As soluções obtidas compuseram o perfil que foi analisado com a técnica de AR (12), resultando em um layout de consenso, no qual foi aplicada uma Busca Local (13). No fim, retorna-se como dado de saída a melhor solução s^* entre as 11 encontradas (dez obtidas com a Busca Local das permutações iniciais e uma obtida com a Busca Local do layout de consenso).

Algoritmo 4: *Busca Local e Agregação de Rankings*

1 **Entrada:** Estrutura da árvore binária e Conjunto de permutações iniciais;
2 $k \leftarrow 0$;
3 **repetir**
4 $\quad k \leftarrow k + 1$;
5 $\quad s_k^0 =$ Gerar solução inicial(Permutação k) ;
6 $\quad s_k^* \leftarrow s_k^0$;
7 $\quad$ **repetir**
8 $\quad\quad s' =$ Busca Local(s_k^*) ;
9 $\quad\quad s_k^* =$ Critério de Aceitação(s_k^*, s') ;
10 $\quad$ **até** *atingir critério de parada*;
11 **até** $k = 10$;
12 $s^{*\prime} =$ Agregação de Rankings(Soluções $s_k^*[k = 1..10]$) ;
13 $s^{*\prime} =$ Busca Local($s^{*\prime}$) ;
14 $s^* =$ Critério de Aceitação(Soluções $s_k^*[k = 1..10], s^{*\prime}$) ;
15 **Saída:** Melhor solução encontrada.

4.4 Método BLAR-PM (Busca Local e Agregação de Rankings usando Programação Matemática)

O último método desenvolvido uniu características dos três métodos anteriores. Para começar, aproveitou-se a ideia de resolução em duas etapas associando Busca Local e Agregação de Rankings do método anterior. Porém, ao invés de gerar os layouts através da estratégia de cortes guilhotinados, adotou-se uma combinação de heurística construtiva e programação matemática (usando nesse caso o *solver* CPLEX com os modelos linearizados, já que esse se mostrou bem mais eficiente em relação ao Baron).

A formação do layout acontece da seguinte maneira: dada uma permutação (com as facilidades dispostas em uma certa sequência), o algoritmo, usando a ideia de uma heurística construtiva, determina a posição das facilidades no layout, bloco a bloco, posicionando um de cada vez com o *solver*. A Figura 4.7 exemplifica esse processo para uma instância com 4 facilidades considerando a permutação inicial = [1,2,3,4].

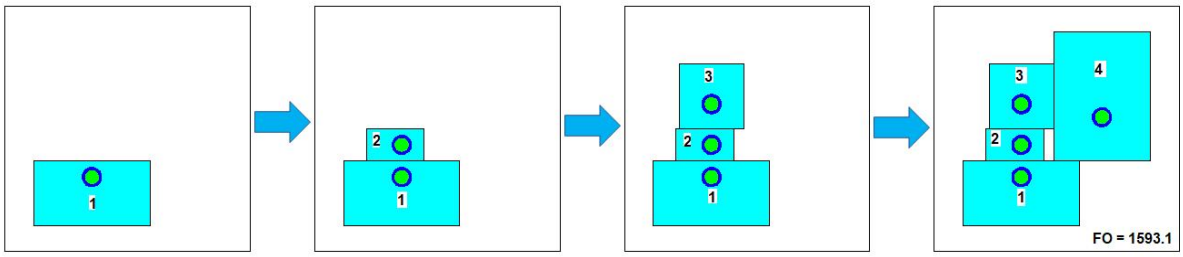

FIGURA 4.7 – Formação de layout do Algoritmo 4 por meio de uma heurística construtiva e do *solver* CPLEX (Instância Das93N4).

Uma vez que um bloco é posicionado ele é fixado naquela posição, ou seja, o CPLEX só precisa definir a melhor posição para um bloco de cada vez em relação aos outros. No exemplo, sendo o bloco 1 alocado em um espaço irrestrito (para maior liberdade de alocação) pelo algoritmo, o *solver* é executado para o bloco seguinte, 2. Com os blocos 1 e 2 fixados, o *solver* é executado novamente para posicionar o bloco 3. E, por fim, considerando as facilidades 1 e 2 e 3 fixas no modelo linearizado embutido no programa, executa-se o CPLEX para o último bloco, 4.

Como o CPLEX só precisa determinar a posição relativa de um bloco de cada vez, isso torna o programa muito mais rápido, sendo compatível com a estratégia de Busca Local que vai testar várias permutações. Em relação ao espaço irrestrito considerado, para as facilidades não ultrapassarem o espaço disponível W e H, devem ser acrescentadas as seguintes restrições:

$$xs_j \leq x_i + W \qquad \forall i, j \tag{4.15}$$

$$ys_j \leq y_i + H \qquad \forall i, j \tag{4.16}$$

O Algoritmo 4 (Seção 4.3.2) pode ser usado perfeitamente para representar o presente método, visto que a diferença entre os métodos está na forma de gerar as soluções e não na estrutura do algoritmo em si. As permutações iniciais são geradas nos mesmos preceitos da Seção 4.3.1.3, exceto a permutação 1: como o processo de formação de layout não depende de uma estrutura de árvore binária, adotou-se como primeira permutação a sequência numérica $1..N$. Ao contrário do método BLAR-CG, o método BLAR-PM não se restringe ao UAFLP de dimensões variáveis e neste trabalho foi aplicado para o caso de facilidades com dimensões fixas e diferentes condições de localização dos I/O *points*.

4.5 Aplicação dos Métodos Propostos

Neste capítulo foram apresentados e devidamente explicados os quatro métodos de resolução propostos para resolver problemas de layout de facilidades.

Como o foco desta tese são os modelos matemáticos desenvolvidos e apresentados no capítulo anterior, os primeiros métodos explorados, PM-MNL e PM-ML, foram escolhidos por envolverem procedimentos intrinsecamente relacionados aos modelos. A justificativa para os demais métodos propostos, BLAR-CG e BLAR-PM, é a de explorar abordagens diferentes, de forma a contemplar na pesquisa desta tese tanto métodos exatos como heurísticos, ampliando o estudo acerca do problema de layout.

O próximo capítulo apresenta os resultados encontrados para algumas instâncias da literatura com a aplicação dos métodos propostos. São expostas as condições de implementação dos algoritmos no computador, a linguagem de programação e os otimizadores utilizados. As soluções são analisadas, comparadas com a de outros autores e representadas como figuras para uma visualização clara dos melhores layouts encontrados.

5 Testes Computacionais

5.1 Condições de Implementação

Toda a fase de implementação, desde a validação aos testes de instâncias da literatura, foi realizada nas seguintes condições:

- Linguagem de programação matemática AMPL - Esse sistema é um excelente ambiente para a finalidade deste trabalho, pois por meio dele podem ser desenvolvidos modelos matemáticos que representam problemas de otimização de forma muito parecida com a representação de modelos escritos;
- *Solver* Baron versão 19.7.13 - É um *software* matemático especializado em resolver problemas de otimização não linear, eficaz para modelos de Programação Inteira Mista como é o caso dos modelos não lineares deste trabalho;
- *Solver* CPLEX versão 12.10.0.0 - É um *software* matemático de alto desempenho para programação linear, eficaz para o caso dos modelos linearizados deste trabalho;
- Processador Intel Core i7;
- Memória RAM de 8GB;
- Sistema operacional Windows 10.

Desenvolver modelos matemáticos que retratem de maneira mais fiel possível situações-problema da vida real é de extrema importância, mas também o é encontrar soluções para esses modelos.

No que tange à aplicação dos algoritmos, foram utilizados os quatro métodos de resolução de problemas de otimização desenvolvidos em diferentes problemas. Os resultados foram obtidos para diversas instâncias encontradas na literatura. As instâncias foram selecionadas da literatura de forma a abranger tanto problemas mais simples quanto os de maior tamanho que são, consequentemente, mais difíceis de resolver computacionalmente.

Vale ressaltar que, embora os modelos e métodos sirvam para resolver problemas de layout multi-período, para esta tese só foram testadas instâncias de período único, abrindo-se espaço para este trabalho ser explorado futuramente com foco em instâncias para demanda dinâmica.

Os resultados foram divididos em seções de acordo com a particularidade do Problema de Layout de Facilidades de Áreas Desiguais avaliado.

5.2 UAFLP: dimensões variáveis e restrições geométricas

Foram obtidos resultados para UAFLPs com dimensões variáveis e restrições geométricas (com I/O *points* nos centroides) para quatro instâncias de 12, 15, 20 e 30 facilidades, considerando as matrizes de fluxo de Nugent *et al.* (1968) e as áreas, razões de aspecto e dimensões do espaço disponível (Figura 3.5) de Tam (1992) (para mais detalhes desses dados de entrada, consultar o Apêndice A).

O algoritmo de duas fases do Método BLAR-CG foi executado para os três processos de construção da árvore binária (apresentados na Seção 4.3.1.2). Na primeira fase, de Busca Local, totalizaram 30 execuções para cada instância (3 árvores com 10 permutações de entrada). Os parâmetros considerados foram $\sigma_{(max)i} = 0,5\ \forall i$, $w_i^{\gamma} = 100\ \forall i$ e $w_i^{\beta} = 100\ \forall i$. A Tabela 5.1 resume as informações coletadas.

Número de facilidades	12	15	20	30
Valor médio de FO da solução inicial	7650,94	13651,31	28279,27	73707,92
Valor médio de FO da solução final	5532,06	10129,82	22718,48	56953,58
Melhoria da FO com a Busca Local	27,7%	25,8%	19,7%	22,7%
Violações da solução inicial (média)	8,1	10,0	13,3	20,9
Violações da solução final (média)	8,5	8,7	13,0	18,5
Tempo médio (s)	8,69	27,94	120,76	1326,48

TABELA 5.1 – Valores médios de função objetivo e de violações das permutações de entrada e de saída da Fase 1 do algoritmo, taxa de melhoria com a Busca Local e tempo médio de execução em segundos para cada instância.

Como consta, a estratégia de Busca Local foi capaz de fornecer uma boa taxa de melhoria em relação aos resultados obtidos com as permutações iniciais. De acordo com os valores médios de violações (de razões de aspecto e de irregularidade da forma) nas permutações de entrada e de saída, a aplicação da Busca Local não foi muito eficiente em reduzi-los, tendo inclusive fornecido soluções com mais violações para a instância menor.

Na Tabela 5.2 estão reunidos os resultados médios e melhores da fase de Busca Local por árvore binária (obtida para cada um dos três diferentes processos construtivos). De acordo com a tabela, os melhores valores de função objetivo foram alcançados pelos processos construtivos de árvore 1 (instâncias maiores) e 3 (instâncias menores). As melhores

médias de resultados (alcançados a partir das 10 permutações de entrada) também foram obtidas através das árvores 1 e 3.

	Média Busca Local			Melhor resultado Busca Local		
N	Árvore 1	Árvore 2	Árvore 3	Árvore 1	Árvore 2	Árvore 3
12	5637,76	5567,37	**5391,06**	5373,39	5383,03	**5279,54**
15	10407,87	10005,99	**9975,59**	9920,66	9758,76	**9721,07**
20	**22681,60**	22702,65	22771,18	**21937,20**	22102,90	22195,00
30	57162,53	56860,15	**56838,05**	**55742,40**	56458,90	56415,10

TABELA 5.2 – Resultados da Fase 1 por processo construtivo da árvore de base para cada instância.

Depois dos resultados da fase de Busca Local, foram analisadas as soluções obtidas na segunda fase do algoritmo relativa à Agregação de Rankings. Os resultados estão na Tabela 5.3.

	FO do layout de consenso			FO pós aplicação da Busca Local			Tempo
N	Árvore 1	Árvore 2	Árvore 3	Árvore 1	Árvore 2	Árvore 3	médio (s)
12	7037,32	7370,75	6906,44	5693,02	5348,22	5491,78	93,50
15	12923,80	12952,60	11899,80	9834,36	9821,33	10068,60	306,89
20	28187,50	27993,30	32129,10	22614,90	22567,80	22422,00	1342,21
30	72411,30	73321,50	75710,40	56272,70	56728,80	56703,10	14822,10

TABELA 5.3 – Resultados da Fase 2 por processo construtivo da árvore de base e média do tempo de execução total das duas fases do algoritmo para cada instância.

A permutação de consenso foi tratada como uma nova permutação de entrada na qual foi realizada uma Busca Local assim como nas demais anteriores. Os valores obtidos foram melhores que as médias apresentadas na Tabela 5.2 em 75% dos casos, sendo que em dois dos 12 cenários, a Fase 2 foi capaz de alcançar um resultado melhor que as dez soluções da Fase 1 que compunham o perfil usado na Agregação de Rankings.

Os tempos médios totais de execução do algoritmo de duas fases estão na última coluna da Tabela 5.3. O custo computacional exigido pelo programa pertence quase que exclusivamente aos procedimentos de busca local. O tempo gasto para construção das árvores binárias, geração das permutações de entrada, formação dos layouts iniciais, geração da matriz de precedências e formação do layout de consenso foi ínfimo para as instâncias testadas, variando entre centésimos e décimos de segundo dependendo da instância.

A Figura 5.1 traz a representação dos melhores layouts obtidos para os problemas testados.

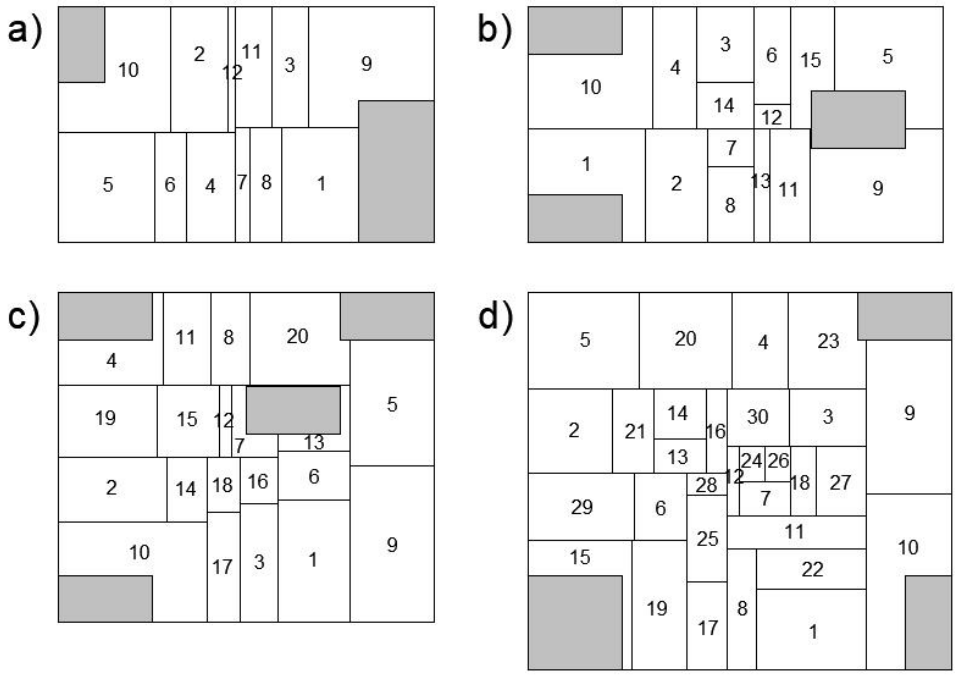

FIGURA 5.1 – Melhores layouts encontrados: a) 12 facilidades - FO = 5279,54; b) 15 facilidades - FO = 9721,07; c) 20 facilidades - FO = 21937,20; d) 30 facilidades - FO = 55742,40.

5.3 UAFLP: dimensões fixas

Para os Problemas de Layout de Facilidades de Áreas Desiguais com dimensões fixas e I/O *points* em posições variadas, foram aplicados os Métodos PM-MNL, PM-ML e BLAR-PM.

A Tabela 5.4 apresenta as dez instâncias testadas nesta etapa, com seus respectivos artigos de referência, posição dos I/O *points* de acordo com os dados de entrada, número de facilidades (diretamente relacionado ao tamanho do problema) e o código usado para se referir a cada instância nesta pesquisa.

lightgray Referência	Dados de entrada	Nº de facilidades	Código
Das (1993)	I/O *points* fixos	4	DasN4
		6	DasN6
		8	DasN8
		10	DasN10
		12	DasN12
Welgama e Gibson (1993)	I/O *points* fixos	6	WelN6
		12	WelN12
Deb e Bhattacharyya (2005)	sem I/O *points* prévios	12	DebN12
		18	DebN18
Dunker *et al.* (2003)	I/O *points* nos centroides	62	DunN62

TABELA 5.4 – Identificação das instâncias retiradas da literatura.

Como consta na Tabela 5.4, originalmente alguns problemas definem a posição dos I/O *points* em relação aos blocos e outros não apresentam essa informação (nesse último caso os I/O *points* podem ser considerados variáveis, sendo sua localização definida na solução do problema). Como é próprio de UAFLPs, a rotação de blocos é permitida em todos os casos testados.

Vale a observação de que nenhuma das instâncias especificava a largura e o comprimento do espaço total disponível para alocação das facilidades, portanto essas foram definidas de acordo com o arredondamento dos valores obtidos com a seguinte fórmula:

$$W = H = 1,5\sqrt{\sum_{i=1}^{N} w_i h_i} \tag{5.1}$$

O valor de $1,5$ da fórmula foi definido após testes empíricos. Valores maiores pioraram o desempenho do algoritmo em relação ao tempo de execução e valores menores interferiram no alcance de alguns resultados ótimos. Além disso, analisando os layouts finais conquistados por diferentes autores para essas instâncias, ficou evidenciado que nenhum ultrapassou as medidas alcançadas com a fórmula, ou seja, as medidas adotadas não interferem na comparação dos resultados, sendo viável adotá-las.

Para mais detalhes dos dados de entrada das instâncias com todas as medidas de comprimento e largura das facilidades, coordenadas dos I/O *points*, dimensões adotadas para espaço disponível e fluxos entre facilidades, consultar o Apêndice A.

A análise dos resultados foi dividida em duas subseções, a primeira referente à situação de I/O *points* inicialmente fixos contemplada pelo Modelo 1 (UAFLPfixo) e a segunda referente à situação de I/O *points* variáveis tratada nos Modelos 2, 3 e 4 (UAFLP1-3).

5.3.1 I/O *points* Fixos

Das dez instâncias apresentadas na Tabela 5.4, vimos que duas não possuem dados de coordenadas de entrada e saída iniciais relativos, portanto nesta subseção foram avaliados os resultados para oito instâncias encontrados pelos autores Das (1993), Dunker *et al.* (2003), Rajasekharan *et al.* (1998) e Welgama e Gibson (1993).

Na primeira fase do algoritmo do Método BLAR-PM, de Busca Local (BL), foi realizado um total de 10 execuções para cada instância (referentes às 10 permutações de entrada). A Tabela 5.5 resume as informações coletadas nas duas fases. A estratégia de Busca Local foi capaz de fornecer melhoria em relação aos resultados obtidos com as permutações iniciais, mas não tão significativa quanto nos testes da seção anterior (Tabela 5.1). Os valores obtidos com a Busca Local do layout de consenso foram melhores que as

médias da Busca Local da Fase 1 em $62,5\%$ dos casos, sendo que em um dos 8 cenários, a Fase 2 foi capaz de alcançar um resultado melhor que as dez soluções da Fase 1 que compunham o perfil usado na Agregação de Rankings.

Instância	DasN4	DasN6	DasN8	DasN10	DasN12	WelN6	WelN12	DunN62
Valor médio de FO^0	1566,1	3597,3	11059,9	19234,3	42443,1	503,0	6641,8	4721234
Valor médio de FO BL	1469,2	3019,4	10081,6	17121,7	38528,8	432,1	5861,5	4359990
Melhoria Busca Local	6,2%	16,1%	8,8%	11,0%	9,2%	14,1%	11,7%	7,7%
Melhor valor de FO BL	1432,6	2570,6	9251,7	16121,9	36953,4	409,5	5522,5	4005480
Tempo médio BL (s)	1,55	13,78	30,02	71,54	178,37	13,96	243,44	3620,40
FO da BL do consenso	1554,7	2622,8	9932,0	18252,6	39211,5	409,5	5431,0	4215180
Tempo (s) total	17,6	152,2	328,0	795,5	2085,9	155,2	2501,3	39824,4

TABELA 5.5 – Resultados do Método BLAR-PM para cada instância testada.

O Método BLAR-PM mostrou-se muito demorado para a maior instância de 62, sendo que a execução de uma única permutação dessas instância com esse método demorou vários dias. Portanto, para esse caso de 62 facilidades, foi adotado um novo critério de parada no algoritmo relativo ao tempo de execução, ficando esse limitado a $1h$ para cada permutação. O tempo médio de formação do layout inicial foi de $40,79s$. Os resultados obtidos estão na última coluna da Tabela 5.5.

Os valores de função objetivo de alguns autores foram comparados com os encontrados pelos métodos propostos neste trabalho e, para isso, a medida de distância adotada foi a mesma: distância retilínea. Observa-se que Xiao *et al.* (2013) encontraram resultados interessantes para algumas instâncias de Das (1993), porém inverteram a posição dos I/O *points* nos blocos com constatado pela observação dos layouts finais obtidos, criando uma instância nova que não poderia ser comparada com as demais. A compilação de dados consta na Tabela 5.6 que segue.

Instância	Melhor resultado da literatura			Métodos deste trabalho		
	Referência	Método	Resultado	PM-MNL	PM-ML	BLAR-PM
DasN4	Rajasekharan	Algoritmo	**1393,6***	**1393,6***	**1393,6***	1432,6
DasN6	*et al.* (1998)	genético	2612,7	2581,9	**2556,0***	2570,6
DasN8	Dunker	Algoritmo	**8778,3***	9186,1	**8778,3***	9251,7
DasN10	*et al.* (2003)	coevolucionário	15694,5	17231,8	**15222,9**	16121,9
DasN12			37396,1	40063,4	**36622,8**	36953,4
WelN6	Kim & Kim	Algoritmo	**398,5***	**398,5***	**398,5***	409,5
WelN12	(2000)	de duas fases	**5016,5**	5784,0	5458,5	5431,0
DunN62	Dunker *et al.* (2003)	Algoritmo coevolucionário	**3939362**	-	5484080	4005480

*Resultado ótimo

TABELA 5.6 – Resultados para I/O *points* fixos.

Os resultados conquistados com o *solver* Baron associado a heurística de melhoramento foram melhores em relação aos encontrados na literatura para somente uma das

instâncias testadas (DasN6). Já com a linearização dos modelos e uso do *solver* CPLEX, os resultados encontrados foram melhores ou iguais para quase todas as instâncias, sendo a exceção a WelN12 e a maior de todas com 62 facilidades.

Os resultados encontrados com o Método BLAR-PM foram bem competitivos, inclusive superando os métodos estritamente construídos com Programação Matemática para duas das instâncias (WelN12 e DunN62), com a vantagem de ser mais rápido. Ainda em relação a esse método, mesmo restringindo o tempo do algoritmo para a resolução da instância DunN62, ele foi capaz de encontrar um resultado com valor só $1,68\%$ maior do que o da literatura para esse caso, evidenciando seu potencial de aplicação.

Para as instâncias menores (com $N \leq 8$), o Método PM-ML conseguiu retornar o resultado ótimo do problema antes de atingir o tempo limite. Na Figura 5.2 temos a representação dos resultados ótimos alcançados, assim como o tempo gasto na solução dos problemas. Os layouts das melhores soluções encontradas para as instâncias DasN10 e DasN12 estão na Figura 5.3.

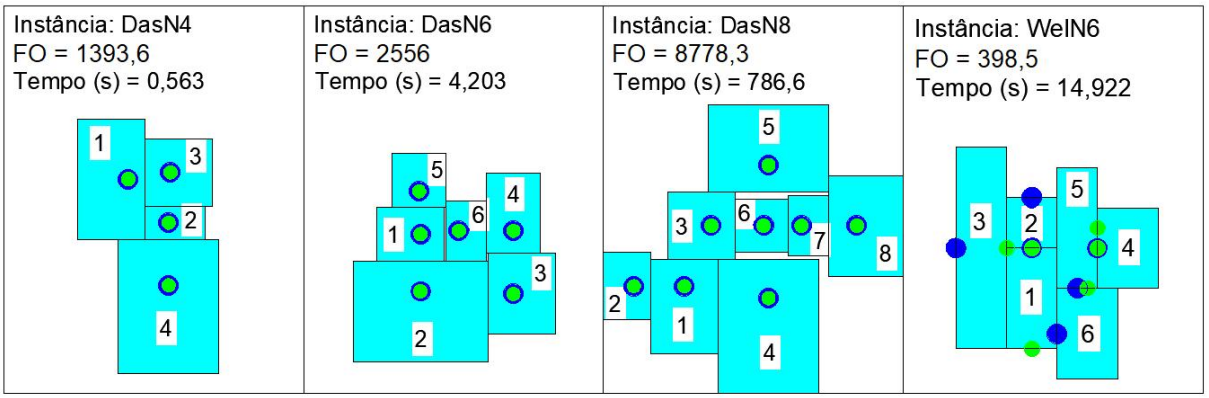

FIGURA 5.2 – Layouts ótimos encontrados para I/O *points* fixos.

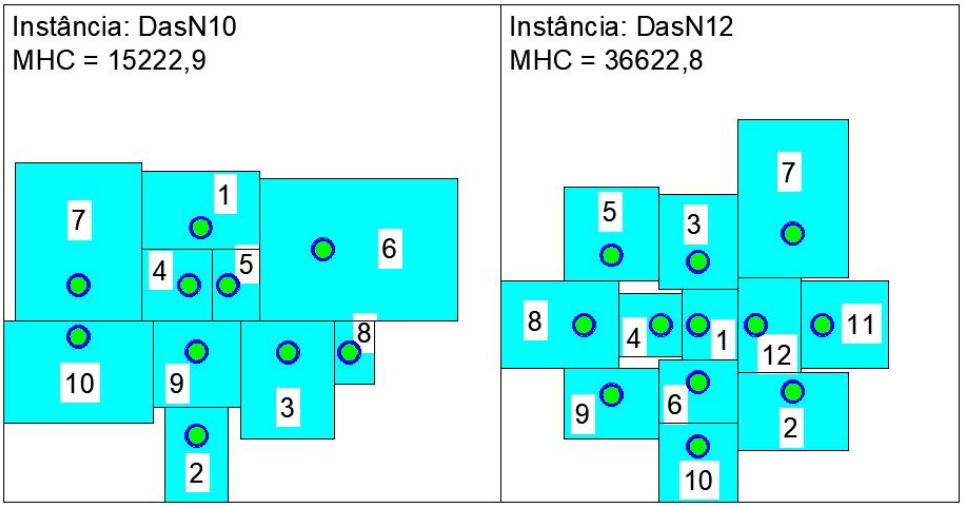

FIGURA 5.3 – Soluções para problemas com I/O *points* fixos.

5.3.2 I/O *points* Variáveis

Para o caso de I/O *points* variáveis, mesmo instâncias propostas inicialmente com I/O *points* fixos podem ser testadas, pois para tanto basta desconsiderar essas coordenadas nos dados de entrada do problema. Nesta parte da pesquisa foram testados os métodos PM-MNL e PM-ML.

Foram avaliados resultados da literatura de dois artigos (os melhores encontrados) referentes a cinco instâncias, que variam quanto às restrições aplicadas aos locais de implantação dos pontos de entrada e saída das facilidades que definem os modelos UAFLP1, UAFLP2 e UAFLP3 (conforme Figura 3.1) e quanto às medidas de distância adotadas.

A comparação dos resultados pode ser vista na Tabela 5.7 abaixo. A Referência [1] é do artigo de Leno *et al.* (2018) que desenvolveram o método ESHGA, um algoritmo genético híbrido baseado em estratégia elitista que usa recozimento simulado como busca local. Já [2] é referente ao valores encontrados pelos autores Deb e Bhattacharyya (2005), que usaram o método GA-SA de busca aleatória associada a um algoritmo híbrido que integra algoritmo genético e recozimento simulado. Em alguns casos a medida considerada pelos autores foi a distância de contorno (PDM) e em outros casos a distância retilínea (RDM); para cada caso tomou-se o cuidado de usar a mesma medida de distância na aplicação dos métodos desta pesquisa para uma comparação justa.

Modelo	Instância	Melhor resultado da literatura			Métodos deste trabalho	
		Ref./Método	Medida	Resultado	PM-MNL	PM-ML
UAFLP1	WelN6	[1] ESHGA	PDM	325	333	**237***
	WelN12			5750	**4264,5**	5288
	DebN12			37894	27314	**12404**
	DebN18			72451	47432	**40754**
	DebN12	[2] GA-SA	RDM	21142	19014	**11504**
	DebN18			62902	43100	**37382**
UAFLP2	DebN12	[1] ESHGA	RDM	8676	8368	**6804**
	DebN18			44012	36056	**27780**
	WelN6	[1] ESHGA	PDM	266	209	**115***
	WelN12			5454	**1985**	2889
	DebN12			16404	8368	**7028**
	DebN18			58324	36128	**28540**
	DunN62			4921320	-	**3972940**
UAFLP3	DebN12	[2] GA-SA	RDM	19892	8820	**6924**
	DebN18			61756	38612	**27092**

*Resultado ótimo

TABELA 5.7 – Resultados para I/O *points* variáveis

Como pode ser notado nas tabelas, os resultados conquistados com o método PM-ML foram todos melhores que os encontrados na literatura e somente foram superados pelo PM-MNL em duas situações.

Para as instâncias menores (com $N = 6$), o método PM-ML foi capaz de retornar o resultado ótimo do problema antes do tempo limite. Na Figura 5.4 temos a representação dos resultados ótimos alcançados, assim como o tempo gasto na solução dos problemas.

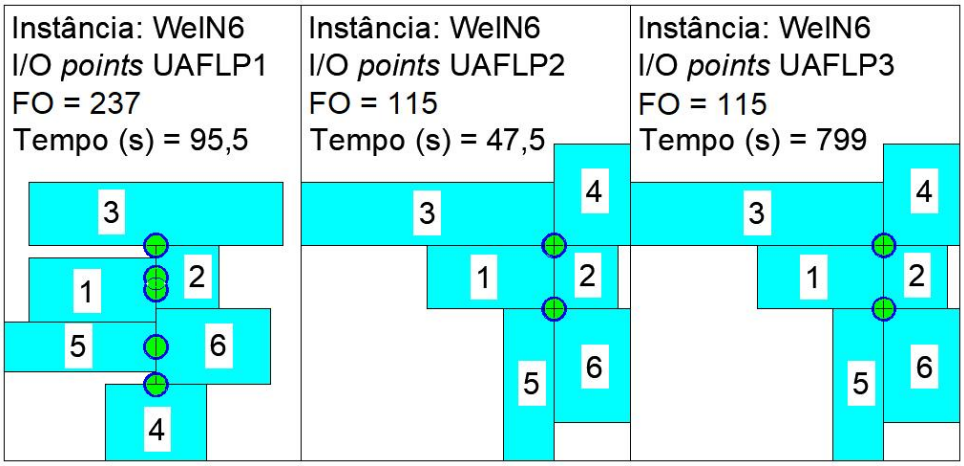

FIGURA 5.4 – Layouts ótimos encontrados para I/O *points* variáveis.

Os layouts das melhores soluções encontradas para as instâncias da Tabela 5.7 estão nas Figuras 5.5, 5.6, 5.7 e 5.8. Para efeito de comparação visual, na Figura 5.9, estão representados os layouts finais gerados por Leno *et al.* (2018) com o método ESHGA e pelo método PM-ML do presente trabalho, para a instância DunN62, Modelo UAFLP2.

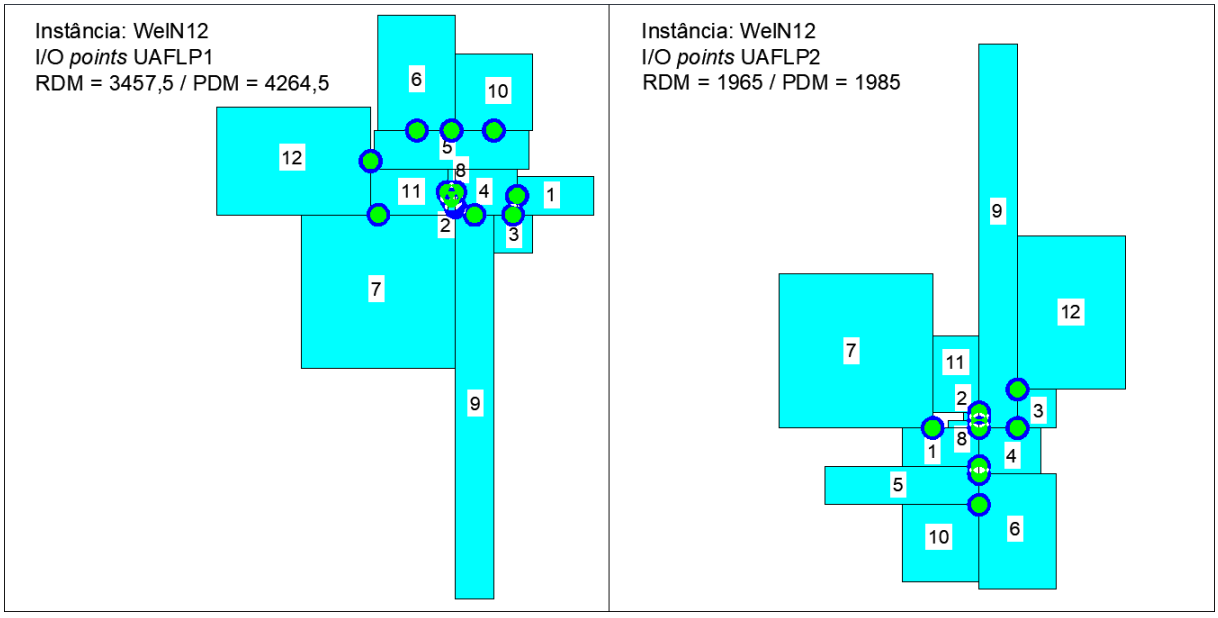

FIGURA 5.5 – Soluções encontradas com o Método PM-MNL para problemas com I/O *points* variavéis - Instância: WelN12.

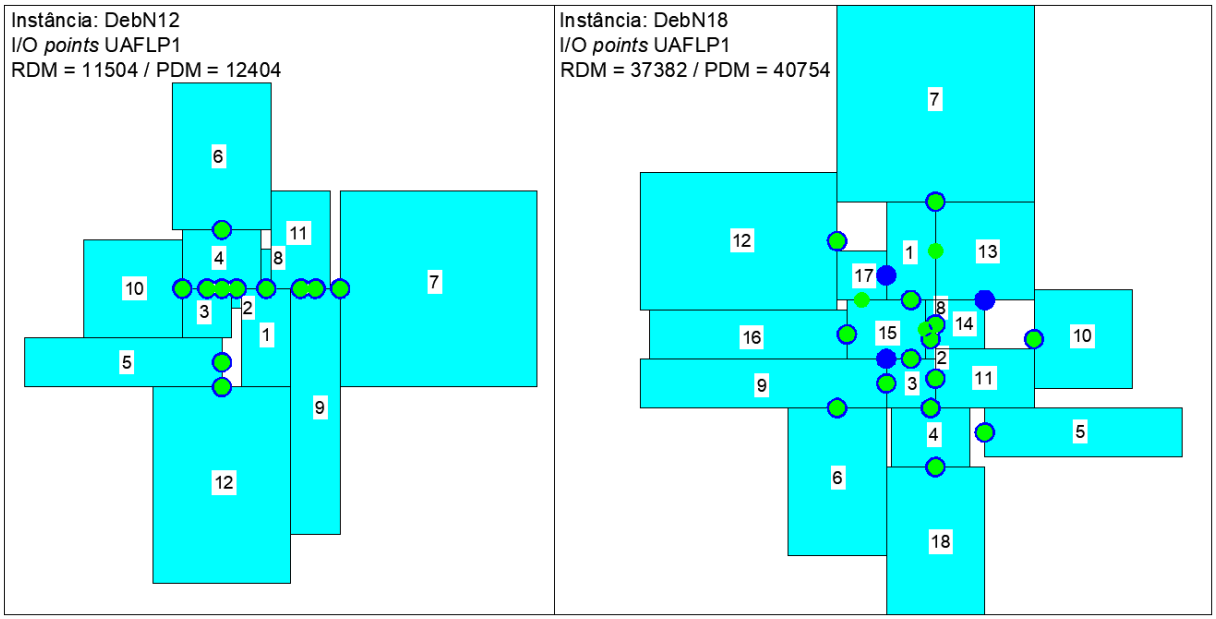

FIGURA 5.6 – Soluções encontradas com o Método PM-ML para problemas com I/O *points* variavéis, Modelo UAFLP1.

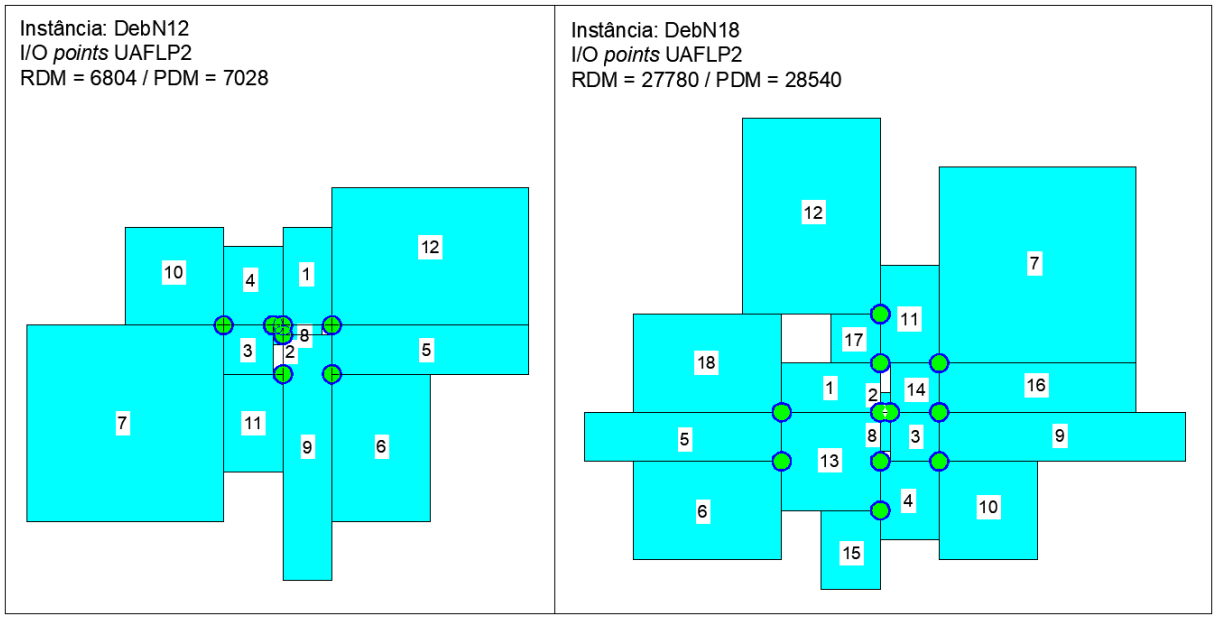

FIGURA 5.7 – Soluções encontradas com o Método PM-ML para problemas com I/O *points* variavéis, Modelo UAFLP2.

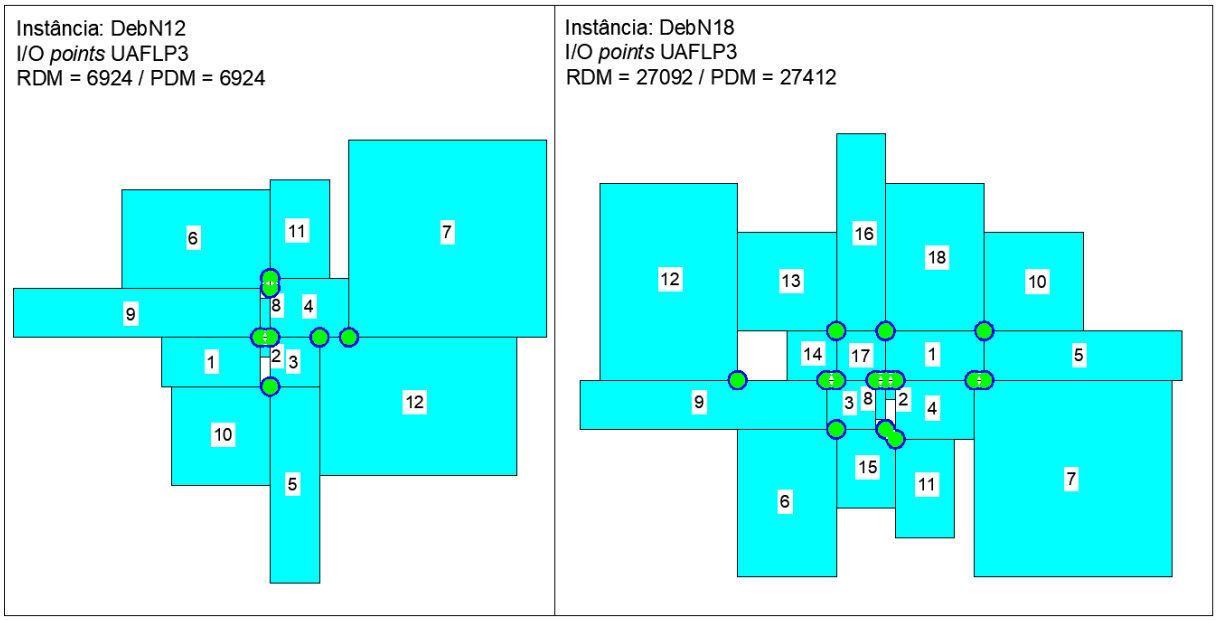

FIGURA 5.8 – Soluções encontradas com o Método PM-ML para problemas com I/O *points* variavéis, Modelo UAFLP3.

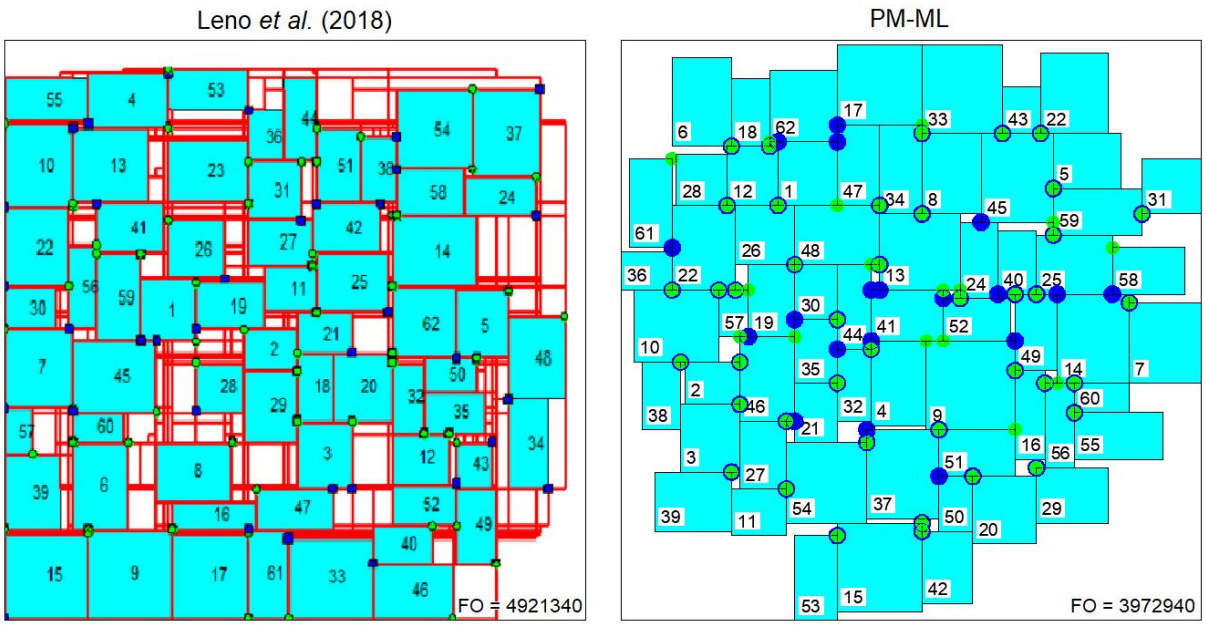

FIGURA 5.9 – Layouts encontrados para o Modelo UAFLP2 - Instância: DunN62.

6 Conclusão

Neste trabalho, abordou-se o Problema de Layout de Facilidades Robusto de Áreas Desiguais com Locais de Entrada e Saída. Foram formulados cinco modelos matemáticos, apresentados no formato não-linear e linear. Por fim, foram desenvolvidos e testados quatro métodos de resolução, sendo os resultados comparados a outros da literatura.

Quanto ao tipo de demanda de produção considerada, foram avaliadas duas propostas da literatura para demanda dinâmica que são as plantas flexíveis e as plantas robustas. A partir dessa análise, optou-se por aprofundar na tática de layout robusto, visto que é menos inconveniente para uma fábrica ou indústria implementar um único layout que receba bem as previsões de mudança no cenário de produção do que ter que parar toda a sua operação de tempos em tempos para satisfazer as mudanças, ainda que os cálculos de layout flexível considerem custos de realocação. Observando as tendências de outros estudos atuais no campo de layout dinâmico robusto, escolheu-se uma abordagem simples, mas eficiente, para adoção nos modelos matemáticos desenvolvidos.

Quanto a posição dos pontos de entrada e saída de materiais e pessoas das facilidades, tradicionalmente considera-se que estes estão posicionados no centro do bloco, porém nesta pesquisa foram consideradas não só essa como também outras possibilidades. Assim sendo, os modelos desenvolvidos valem para quatro situações de disposição dos I/O *points* sendo estas: a) fixas (opcionalmente no centroide), b) variáveis nos pontos médios das bordas, c) variáveis nas quinas e d) variáveis em qualquer lugar nas bordas do bloco.

Quanto ao formato das facilidades, foram considerados blocos de áreas desiguais sujeitos a dois tipos diferentes de situação: a) blocos de área fixa e dimensões variáveis sujeitas a limites de razão de aspecto e b) blocos de área e dimensões fixas, sendo testados métodos diferentes para cada situação.

Nas próximas duas seções, comenta-se a respeito das soluções obtidas com a aplicação dos métodos de resolução desenvolvidos, de acordo com a situação particular de cada problema no que se refere ao formato das facilidades.

6.1 Facilidades com dimensões variáveis

Situações contemplando facilidades de dimensões variáveis foram analisadas considerando ainda um espaço inicial com restrições geométricas. Foi executado um método de duas fases, identificado como BLAR-CG, composto por uma heurística de Busca Local aplicada a cenários cujos resultados compuseram o perfil analisado pela técnica de Agregação de Rankings na segunda fase. Foram desenvolvidos três processos construtivos para definição da estrutura de base no formato de árvore binária e dez estratégias de organização das facilidades para composição dos vetores de permutação da solução inicial. Como busca local, foi usada a ideia de troca de facilidades na ordem de alocação para exploração da vizinhança, duas por vez e de forma exaustiva (encerrando o programa só após serem testadas todas as possibilidades sem encontrar melhora). Na função objetivo que minimiza o MHC do layout final foi incluída uma função de penalidades para as violações da razão de aspecto e da razão de irregularidade de forma das facilidades.

Os resultados encontrados para o método de Busca Local foram bons com melhoria em torno de 24% em relação aos valores iniciais. A última permutação de entrada submetida ao processo de Busca Local (gerada na etapa de Agregação de Rankings, referente ao layout de consenso) apresentou um resultado melhor que a média na maioria dos casos, sendo inclusive capaz de superar os resultados das dez permutações anteriores em algumas situações.

A aplicação do Método BLAR-CG resultou em muitas violações de formato das facilidades. Uma perspectiva futura desta pesquisa é procurar contornar essa situação. Valores mais elevados dos parâmetros w^{γ} e w^{β} podem evitar que as razões de aspecto e forma sejam tão violadas. Visto que na análise da Tabela 5.1 foi contado o número absoluto de violações, uma opção para reduzir esse número é mudar o cálculo da função de penalidades por intensidade para o método de Coit *et al.* (1996). Outra possibilidade é considerar valores mais flexíveis de razão de aspecto das facilidades como foi feito nos trabalhos de Gau e Meller (1999) e Anjos e Vieira (2016).

O estudo de layout considerando espaços com restrições geométricas ainda é muito pouco explorado na literatura o que reforça a relevância desta análise realizada nesta tese.

6.2 Facilidades de dimensões fixas

O UAFLP de dimensões fixas com locais de entrada e saída já foi resolvido na literatura com o uso dos mais variados métodos. Ainda assim, os resultados encontrados na literatura até então foram superados pelas soluções encontradas nesta pesquisa com o Método PM-

ML, o que evidencia que o método usado e os resultados alcançados nesta tese foram bem significativos.

Embora alguns dos resultados comparados durante os testes computacionais remetam a artigos que datam de muitos anos, estes ainda foram os melhores encontrados na literatura para cada situação, o que confirma a necessidade de novos estudos e propostas para resolver problemas como os analisados.

Na maioria dos artigos consultados durante a revisão bibliográfica, constatou-se que os autores utilizam heurísticas e metaheurísticas para resolver o problema de UAFLP com locais de entrada e saída, fixos ou variáveis. Neste trabalho mesmo, foram desenvolvidos métodos focados em heurísticas (BLAR-CG) ou associando heurística e programação matemática (BLAR-PM).

Métodos heurísticos são aplicados em problemas de otimização com o objetivo de encontrar bons resultados em um tempo razoável, feito que nem sempre pode ser alcançado por meio de métodos exatos. Ainda assim, os resultados encontrados com os métodos PM-MNL e PM-ML (especialmente com o segundo) nesta pesquisa foram quase todos melhores do que os encontrados na literatura. E, embora o tempo demandado por esses métodos pareça considerável, deve-se lembrar que estão sendo resolvidos problemas de layout que definirão uma planta de fábrica ou um projeto de instalação que funcionará por meses ou anos, logo é perfeitamente plausível que se gastem algumas horas ou dias definindo-se o melhor arranjo físico.

6.3 Impactos e Indicadores

Nesta pesquisa foram exploradas várias peculiaridades como:

- Análise robusta, considerando ambiente de produção dinâmico com incerteza de demanda;
- Facilidades na forma de blocos retangulares de áreas desiguais;
- Locais de entrada e saída dos processos nos blocos;
- Espaços iniciais com restrições geométricas; e
- Inclusão de corredores.

O estudo dessas problemáticas se justifica, pois ajustam melhor o problema dentro da realidade encontrada no setor de produção. Nesta tese, a autora desenvolveu e forneceu uma análise minuciosa de modelos matemáticos para estes casos, além de aplicar uma

estratégia de linearização e métodos de resolução que associam diferentes abordagens, podendo esta pesquisa servir de referência para futuras produções científicas no tema de layouts. Como indicador de produção e impacto científico, esta pesquisa resultou, até então, em dois artigos publicados, um em periódico internacional (BRAGA; SALLES-NETO, 2022) e um nos anais do Simpósio Brasileiro de Pesquisa Operacional (BRAGA; SALLES-NETO, 2021).

Como consta nesta tese, convencionalmente a função objetivo do problema de layout de instalações consiste na redução do Custo de Manuseio de Material (MHC), um custo sem valor agregado que relaciona os fluxos de materiais e as distâncias entre as facilidades. Estima-se que o planejamento eficiente das instalações pode reduzir o MHC total em até 30% (JUNG *et al.*, 2017), diminuindo assim o custo operacional total do projeto. Em contrapartida, pesquisas mostram que é provável que mais de 35% da eficiência do sistema seja perdida aplicando *designs* de localização incorretos (IZADINIA; ESHGHI, 2016). Sendo assim, fica claro o potencial de impacto econômico que o estudo realizado pode gerar quando adotado.

As contribuições deste trabalho no que se referem à modelagem de problemas e aos métodos de resolução também têm impacto social e ambiental no sentido que contribuem para melhorar o desempenho dos processos produtivos. A capacidade de implementar um sistema de produção mais eficiente pode ser o ponto chave necessário para que uma empresa se torne competitiva o bastante para entrar no mercado ou para que amplie sua capacidade de fornecimento de produtos, sendo que a abertura de novas unidades de fabricação geram emprego e renda para a população no seu entorno. Além disso, reduzir o MHC que está associado ao deslocamento de materiais e pessoas entre diferentes pontos resulta em um sistema de transporte interno mais eficiente, podendo diminuir: a emissão de carbono gerada, por exemplo, por carrinhos, empilhadeiras e guindastes; o desgaste destes equipamentos aumentando sua vida útil; e, também, gastos de energia que nem sempre provém de fontes renováveis.

6.4 Contribuição Científica

Conclui-se que esta tese se mostra uma excelente contribuição para o tema de layouts, tanto no que se refere a modelagem matemática nos formatos não-linear e linear, quanto ao uso de métodos exatos, heurísticos e híbridos para resolver o problema. Em especial, o uso de programação matemática aplicada aos modelos linearizados apresentou resultados notáveis nos problemas de facilidades de dimensões fixas, sendo o método que une heurísticas e programação matemática também uma boa alternativa para problemas maiores.

Outro ponto importante desta pesquisa que vale ressaltar é a análise das situações

de restrições geométricas e corredores entre facilidades, que fazem muito sentido quando pensamos em situações reais e podem fazer toda a diferença nas resoluções, porém foram pouco exploradas na literatura até então. As soluções encontradas com o método que associa busca local e agregação de rankings para problemas com restrições geométricas foram interessantes e abrem caminho para novos estudos com esse enfoque.

Os objetivos propostos para esta tese foram atingidos e pode-se dizer que, certamente, todo o processo de pesquisa e elaboração deste trabalho não só contribuíram como foram indispensáveis para a formação desta aluna em seu doutorado na área de pesquisa operacional.

Referências

AIELLO, G.; ENEA, M. Fuzzy approach to the robust facility layout in uncertain production environments. **International Journal of Production Research**, v. 39, n. 18, p. 4089–4101, 2001.

AIELLO, G.; ENEA, M.; GALANTE, G. An integrated approach to the facilities and material handling system design. **International Journal of Production Research**, Taylor & Francis, v. 40, n. 15, p. 4007–4017, 2002.

ALI, A.; MEILĂ, M. Experiments with kemeny ranking: What works when? **Mathematical Social Sciences**, Elsevier, v. 64, n. 1, p. 28–40, 2012.

ANJOS, M. F.; VIEIRA, M. V. An improved two-stage optimization-based framework for unequal-areas facility layout. **Optimization letters**, Springer, v. 10, n. 7, p. 1379–1392, 2016.

ARMOUR, G. C.; BUFFA, E. S. A heuristic algorithm and simulation approach to relative location of facilities. **Management Science**, Informs, v. 9, n. 2, p. 294–309, 1963.

ASL, A. D.; WONG, K. Y. Solving unequal-area static and dynamic facility layout problems using modified particle swarm optimization. **Journal of Intelligent Manufacturing**, 2017.

AZADEH, S.; HAGHIGHI, S. M.; ASADZADEH, S. M. A novel algorithm for layout optimization of injection process with random demands and sequence dependent setup times. **Journal of Manufacturing Systems**, v. 33, p. 287–302, 2014.

AZADIVAR, F.; WANG, J. Facility layout optimization using simulation and genetic algorithms. **International Journal of Production Research**, v. 38, n. 17, p. 4369–4383, 2000.

BAZARAA, M. S. Computerized layout design: a branch and bound approach. **AIIE transactions**, Taylor & Francis, v. 7, n. 4, p. 432–438, 1975.

BOCK, S.; HOBERG, K. Detailed layout planning for irregularly-shaped machines with transportation path design. **European Journal of Operational Research**, Elsevier, v. 177, n. 2, p. 693–718, 2007.

BORTOLETE, J. C.; BUENO, L. F.; BUTKERAITES, R.; CHAVES, A. A.; COLLAçO, G.; MAGUETA, M.; PELOGIA, F. J. R.; NETO, L. L. S.; SANTOS, T. S.; SILVA, T. S.; SOBRAL, F. N. C.; YANASSE, H. H. A support tool for planning

classrooms considering social distancing between students. **Computational and Applied Mathematics**, v. 41, n. 22, 2022.

BRAGA, E. M. H.; SALLES-NETO, L. L. Modelagem e resolução do problema de layout de facilidades de Áreas desiguais com locais de entrada e saída. **Anais do Simpósio Brasileiro de Pesquisa Operacional, João Pessoa**, Campinas, Galoá, 2021.

BRAGA, E. M. H.; SALLES-NETO, L. L. A mathematical optimization approach based on linearized mip models for solving facility layout problems. **Pesquisa Operacional [online]**, v. 42, 2022.

BRAGLIA, M.; SIMONE, Z.; ZAVANELLA, L. Layout design in dynamic environments: Strategies and quantitative indices. **International Journal of Production Research**, v. 41, n. 5, p. 995–1016, 2003.

BRANSKI, R. M. **Arranjos Físicos ou Layout**: Arranjo físico por processo ou funcional. [S.l.], 2008.

BRIMBERG, J.; WESOLOWSKY, G. O. Optimizing facility location with euclidean and rectilinear distances. In: FLOUDAS, C.; PARDALOS, P. (Ed.). **Encyclopedia of Optimization**. 2nd. ed. USA: Springer, 2008. p. 2869–2873.

CASTILLO, I.; WESTERLUND, T. An ε-accurate model for optimal unequal-area block layout design. **Computers & Operations Research**, Elsevier, v. 32, n. 3, p. 429–447, 2005.

CHAN, W. M.; CHAN, C. Y.; KWONG, C. K. Development of the main algorithm for a cellular manufacturing machine layout. **International Journal of Production Research**, v. 42, n. 1, p. 51–65, 2004.

CHANG, M.-S.; KU, T.-C. A slicing tree representation and qcp-model-based heuristic algorithm for the unequal-area block facility layout problem. **Mathematical Problems in Engineering**, Hindawi, v. 2013, 2013.

COIT, D. W.; SMITH, A. E.; TATE, D. M. Adaptive penalty methods for genetic optimization of constrained combinatorial problems. **INFORMS Journal on Computing**, INFORMS, v. 8, n. 2, p. 173–182, 1996.

CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L.; STEIN, C. **Introduction to algorithms**. 2nd. ed. Massachusetts, USA: MIT press, 2009.

DAS, S. A facility layout method for flexible manufacturing systems. **The International Journal of Production Research**, Taylor & Francis, v. 31, n. 2, p. 279–297, 1993.

DEB, S.; BHATTACHARYYA, B. Facilities layout planning based on fuzzy multiple criteria decision-making methodology. **International Journal of Production Research**, Taylor & Francis, v. 41, n. 18, p. 4487–4504, 2003.

DEB, S.; BHATTACHARYYA, B. Solution of facility layout problems with pickup/drop-off locations using random search techniques. **International journal of production research**, Taylor & Francis, v. 43, n. 22, p. 4787–4812, 2005.

DIJKSTRA, E. W. A note on two problems in connexion with graphs. **Numerische mathematik**, v. 1, n. 1, p. 269–271, 1959.

DUNKER, T.; RADONS, G.; WESTKÄMPER, E. A coevolutionary algorithm for a facility layout problem. **International Journal of Production Research**, Taylor & Francis, v. 41, n. 15, p. 3479–3500, 2003.

DUNKER, T.; RADONS, G.; WESTKÄMPER, E. Combining evolutionary computation and dynamic programming for solving a dynamic facility layout problem. **European Journal of Operational Research**, Elsevier, v. 165, n. 1, p. 55–69, 2005.

EMAMI, S.; NOOKABADI, A. S. Managing a new multi-objective model for the dynamic facility layout problem. **International Journal of Advanced Manufacturing Technology**, v. 68, n. 9, p. 2215–2228, 2013.

FAZLELAH, F. Z.; POURNADER, M.; GHARAKHANI, M.; SADJADI, S. J. A robust approach to design a single facility layout plan in dynamic manufacturing environments using a permutation-based genetic algorithm. **Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture**, v. 230, n. 12, p. 2264–2274, 2016.

FORGHANI, K.; MOHAMMADI, M.; GHEZAVATI, V. Designing robust layout in cellular manufacturing systems with uncertain demands. **International Journal of Industrial Engineering Computations**, v. 4, n. 2, p. 215–226, 2013.

FRIEDRICH, C.; KLAUSNITZER, A.; LASCH, R. Integrated slicing tree approach for solving the facility layout problem with input and output locations based on contour distance. **European Journal of Operational Research**, Elsevier, v. 270, n. 3, p. 837–851, 2018.

FURTADO, J. C.; LORENA, L. A. N. Otimização de leiaute usando busca tabu. **Gestão & Produção**, SciELO Brasil, v. 4, n. 1, p. 88–108, 1997.

GARCÍA-HERNÁNDEZ, L.; PIERREVAL, H.; SALAS-MORERA, L.; ARAUZO-AZOFRA, A. Handling qualitative aspects in unequal area facility layout problem: An interactive genetic algorithm. **Applied Soft Computing**, v. 13, n. 4, p. 1718–1727, 2013.

GARCÍA-HERNÁNDEZ, L.; SALAS-MORERA, L.; CARMONA-MUÑOZ, C.; GARCÍA-HERNÁNDEZ, J. A.; SALCEDO-SANZ, S. A novel island model based on coral reefs optimization algorithm for solving the unequal area facility layout problem. **Engineering Applications of Artificial Intelligence**, v. 89, p. 103445, 2020.

GARCÍA-HERNÁNDEZ, L.; SALAS-MORERA, L.; GARCÍA-HERNÁNDEZ, J. A.; SALCEDO-SANZ, S.; OLIVEIRA, J. V. de. Applying the coral reefs optimization algorithm for solving unequal area facility layout problems. **Expert Systems With Applications**, v. 138, p. 112819, 2019.

GAU, K.-Y.; MELLER, R. An iterative facility layout algorithm. **International Journal of Production Research**, Taylor & Francis, v. 37, n. 16, p. 3739–3758, 1999.

GLOVER, F.; WOOLSEY, E. Converting the 0-1 polynomial programming problem to a 0-1 linear program. **Operations research**, INFORMS, v. 22, n. 1, p. 180–182, 1974.

GONÇALVES, J. F.; RESENDE, M. G. A biased random-key genetic algorithm for the unequal area facility layout problem. **European Journal of Operational Research**, Elsevier, v. 246, n. 1, p. 86–107, 2015.

HOSSEINI-NASAB, H.; FEREIDOUNI, S.; GHOMI, S. M. T. F.; FAKHRZAD, M. B. Classification of facility layout problems: a review study. **International Journal of Advanced Manufacturing Technology**, v. 94, n. 1-4, p. 957–977, 2018.

HOU, S.; WEN, H.; FENG, S.; WANG, H.; ; LI, Z. Application of layered coding genetic algorithm in optimization of unequal area production facilities layout. **Computational Intelligence and Neuroscience**, Hindawi, v. 2019, 2019.

HU, G.; CHEN, Y.; ZHOU, Z.; FANG, H. A genetic algorithm for the inter-cell layout and material handling system design. **The International Journal of Advanced Manufacturing Technology**, Springer, v. 34, n. 11-12, p. 1153–1163, 2007.

INÁCIO, W.; ERTHAL, M. Dimensionamento da carga térmica de resfriamento de ambientes como estratégia para melhoria da eficiência energética em instituições de ensino. **Revista Vértices**, v. 19, p. 211–236, 12 2017.

IRAPPA, B. H.; MADHUSUDANAN, P. V. Development of a heuristic for layout formation and design of robust layout under dynamic demand. In: INTERNATIONAL CONFERENCE ON DIGITAL FACTORY. Coimbatore, India, 2008. p. 1398–1405.

IZADINIA, N.; ESHGHI, K. A robust mathematical model and aco solution for multi-floor discrete layout problem with uncertain locations and demands. **Computers Industrial Engineering**, v. 96, p. 237–248, 2016.

JOLAI, F.; TAVAKKOLI-MOGHADDAM, R.; TAGHIPOUR, M. A multi-objective particle swarm optimisation algorithm for unequal sized dynamic facility layout problem with pickup/drop-off locations. **International Journal of Production Research**, Taylor & Francis, v. 50, n. 15, p. 4279–4293, 2012.

JUNG, K.; CHOI, S.; KULVATUNYOU, B.; CHO, H.; MORRIS, K. C. A reference activity model for smart factory design and improvement. **Production planning & control**, Taylor & Francis, v. 28, n. 2, p. 108–122, 2017.

KANG, S.; CHAE, J. Harmony search for the layout design of an unequal area facility. **Expert Systems with Applications**, Elsevier, v. 79, p. 269–281, 2017.

KELACHANKUTTU, H.; BATTA, R.; NAGI, R. Contour line construction for a new rectangular facility in an existing layout with rectangular departments. **European Journal of Operational Research**, Elsevier, v. 180, n. 1, p. 149–162, 2007.

KIM, J.-G.; KIM, Y.-D. Layout planning for facilities with fixed shapes and input and output points. **International Journal of Production Research**, Taylor & Francis, v. 38, n. 18, p. 4635–4653, 2000.

KONAK, A.; KULTUREL-KONAK, S.; NORMAN, B. A.; SMITH, A. E. A new mixed integer programming formulation for facility layout design using flexible bays. **Operations Research Letters**, v. 34, n. 6, p. 660–672, 2006.

KOOPMANS, T. C.; BECKMANN, M. Assignment problems and the location of economic activities. **Econometrica: journal of the Econometric Society**, JSTOR, p. 53–76, 1957.

KOUVELIS, P.; KUAWARWALA, A. A.; GUTIERREZ, G. J. Algorithms for robust single and multiple period layout planning for manufacturing systems. **European Journal of Operational Research**, v. 63, p. 287–303, 1992.

KULTUREL-KONAK, S. Approaches to uncertainties in facility layout problems: Perspectives at the beginning of the 21st century. **Journal of Intelligent Manufacturing**, v. 18, p. 273–284, 2007.

KULTUREL-KONAK, S. A linear programming embedded probabilistic tabu search for the unequal-area facility layout problem with flexible bays. **European Journal of Operational Research**, Elsevier, v. 223, n. 3, p. 614–625, 2012.

KULTUREL-KONAK, S. The zone-based dynamic facility layout problem. **INFOR: Information Systems and Operational Research**, Taylor & Francis, v. 57, n. 1, p. 1–31, 2019.

KULTUREL-KONAK, S.; SMITH, A. E.; NORMAN, B. A. Layout optimization considering production uncertainty and routing flexibility. **International Journal of Production Research**, v. 42, n. 21, p. 4475–4493, 2004.

LEE, H. Y.; KANG, S.; CHAE, J. Mutation effects in a genetic algorithm for a facility layout problem in qap form. **International Journal of Advanced Logistics**, v. 4, n. 3, p. 170–179, 2015.

LENO, I. J.; SANKAR, S. S.; PONNAMBALAM, S. An elitist strategy genetic algorithm using simulated annealing algorithm as local search for facility layout design. **The International Journal of Advanced Manufacturing Technology**, Springer, v. 84, n. 5-8, p. 787–799, 2016.

LENO, I. J.; SANKAR, S. S.; PONNAMBALAM, S. Mip model and elitist strategy hybrid ga–sa algorithm for layout design. **Journal of Intelligent Manufacturing**, Springer, v. 29, n. 2, p. 369–387, 2018.

LIU, Q.; MELLER, R. D. A sequence-pair representation and mip-model-based heuristic for the facility layout problem with rectangular departments. **IIE transactions**, Taylor & Francis, v. 39, n. 4, p. 377–394, 2007.

LOIOLA, E. M.; ABREU, N. M. M. de; NETTO, P. O. B. Uma revisão comentada das abordagens do problema quadrático de alocação. **Pesquisa Operacional**, v. 24, n. 1, p. 73–109, 2004.

MAURI, G. R.; LORENA, L. A. N. Decomposições lagrangeanas para o problema de programação quadrática binária irrestrita. **Pesquisa Operacional**, SciELO Brasil, v. 29, n. 1, p. 111–127, 2009.

MAZINANI, M.; ABEDZADEH, M.; MOHEBALI, N. Dynamic facility layout problem based on flexible bay structure and solving by genetic algorithm. **The International Journal of Advanced Manufacturing Technology**, v. 65, p. 929–943, 2013.

MELO, J. L. R. **Por que o layout fabril faz a diferença no ambiente produtivo?** Florianópolis/SC, Brasil, 2017. Disponível em: <https://certi.org.br/blog/layout-fabril/>. Acesso em: 05 out. 2019.

MOAREFDOOST, M. M.; SNYDER, L. V.; ALNAJJAB, B. Layouts for ocean wave energy farms: Models, properties, and optimization. **Omega**, v. 66, p. 185–194, 2017. New Research Frontiers in Sustainability.

MONTREUIL, B. A modelling framework for integrating layout design and flow network design. In: **Material handling'90**. Germany: Springer, 1991. p. 95–115.

MOSLEMIPOUR, G.; LEE, T. S.; LOONG, Y. T. Performance analysis of intelligent robust facility layout design. **Chinese Journal of Mechanical Engineering**, v. 30, p. 407–418, 2017.

NEGHABI, H.; ESHGHI, K.; SALMANI, M. H. A new model for robust facility layout problem. **Information Sciences**, v. 278, p. 498–509, 2014.

NEMATIAN, J. A robust single row facility layout problem with fuzzy random variables. **The International Journal of Advanced Manufacturing Technology**, v. 72, p. 255–267, 2014.

NILSON, C. **3 Passos para definir um layout ideal na produção da marcenaria**. São Leopoldo/RS, Brasil, 2017. Disponível em: <https://www.gabster.com.br/marcenaria-/producao/3-passos-para-definir-um-layout-ideal-na-marcenaria/>. Acesso em: 02 mar. 2020.

NUGENT, C. E.; VOLLMANN, T. E.; RUML, J. An experimental comparison of techniques for the assignment of facilities to locations. **Operations research**, INFORMS, v. 16, n. 1, p. 150–173, 1968.

O'BRIEN, C.; BARR, S. A. An interactive approach to computer aided facility layout. **International Journal of Production Research**, Taylor & Francis, v. 18, n. 2, p. 201–211, 1980.

PALOMO-ROMERO, J. M.; SALAS-MORERA, L.; GARCÍA-HERNÁNDEZ, L. An island model genetic algorithm for unequal area facility layout problems. **Expert Systems with Applications**, Elsevier, v. 68, p. 151–162, 2017.

PARK, H.; SEO, Y. An efficient algorithm for unequal area facilities layout planning with input and output points. **INFOR: Information Systems and Operational Research**, v. 57, p. 56–74, 2019.

PENG, Y.; ZENG, T.; FAN, L.; HAN, Y.; XIA, B. An improved genetic algorithm based robust approach for stochastic dynamic facility layout problem. **Discrete Dynamics in Nature and Society**, v. 2018, 2018.

PILLAI, V. M.; HUNAGUND, I. B.; KRISHNAN, K. K. Design of robust layout for dynamic plant layout problems. **Computers Industrial Engineering**, v. 61, n. 3, p. 813–823, 2011.

POURVAZIRI, H.; NADERIB, B. A hybrid multi-population genetic algorithm for the dynamic facility layout problem. **Applied Soft Computing**, v. 24, p. 457–469, 2014.

QIN, L.; ZHAO, Y. The facility layout problem in a logistics park based on accessibility and transport diversity. In: **LISS 2020**. [S.l.]: Springer, 2021. p. 159–173.

RAGOV, B. **Como funciona a cozinha de um fast-food?** São Paulo/SP, Brasil, 2018. Disponível em: <https://super.abril.com.br/mundo-estranho/como-funciona-a-cozinha-de-um-fast-food/>. Acesso em: 17 jul. 2018.

RAHIMI, H.; JAHANIRAD, H. An evolutionary approach to implement logic circuits on three dimensional fpgas. **Expert Systems with Applications**, v. 174, p. 114780, 2021.

RAJASEKHARAN, M.; PETERS, B. A.; YANG, T. A genetic algorithm for facility layout design in flexible manufacturing systems. **International journal of Production research**, Taylor & Francis, v. 36, n. 1, p. 95–110, 1998.

ROSENBLATT, M. J.; LEE, H. L. A robustness approach to facilities design. **International Journal of Production Research**, v. 25, n. 4, p. 479–486, 1987.

SALIMPOUR, S.; POURVAZIRI, H.; AZAB, A. Semi-robust layout design for cellular manufacturing in a dynamic environment. **Computers Operations Research**, v. 133, p. 105367, 2021.

SEE, P. C.; WONG, K. Y. Application of ant colony optimisation algorithms in solving facility layout problems formulated as quadratic assignment problems: a review. **International Journal of Industrial and Systems Engineering**, v. 3, n. 6, p. 644–672, 2008.

SHERALI, H. D.; FRATICELLI, B. M.; MELLER, R. D. Enhanced model formulations for optimal facility layout. **Operations Research**, INFORMS, v. 51, n. 4, p. 629–644, 2003.

SMITH, A. E.; NORMAN, B. A. Evolutionary design of facilities considering production uncertainty. In: PARMEE, I. C. (Ed.). **Evolutionary design and manufacture**. London, England: Springer, 2000. p. 175–186.

SOOLAKI, M.; IZADI, A. A robust optimisation model for manufacturing cell design problem under uncertainty. **International Journal of Services and Operations Management**, v. 15, n. 2, p. 238–258, 2013.

TAM, K. Y. Genetic algorithms, function optimization, and facility layout design. **European journal of operational research**, Elsevier, v. 63, n. 2, p. 322–346, 1992.

TATE, D. M.; SMITH, A. E. Unequal-area facility layout by genetic search. **IIE transactions**, Taylor & Francis, v. 27, n. 4, p. 465–472, 1995.

TOMPKINS, J. A.; WHITE, J. A.; BOZER, Y. A.; TANCHOCO, J. M. A. **Facilities planning**. [S.l.]: John Wiley & Sons, 2010.

URBAN, T. L. Solution procedures for the dynamic facility layout problem. **Annals of Operations Research**, v. 76, p. 323–342, 1998.

VITAYASAK, S.; PONGCHAROEN, P.; HICKS, C. A tool for solving stochastic dynamic facility layout problems with stochastic demand using either a genetic algorithm or modified backtracking search algorithm. **International Journal of Production Economics**, v. 190, p. 146–157, 2017.

VITAYASAK, S.; PONGCHAROEN, P.; HICKS, C. Robust machine layout design under dynamic environment: Dynamic customer demand and machine maintenance. **Expert Systems with Applications: X**, v. 3, p. 100015, 2019.

WANG, S.; ZUO, X.; LIU, X.; ZHAO, X.; LI, J. Solving dynamic double row layout problem via combining simulated annealing and mathematical programming. **Applied Soft Computing**, v. 37, p. 303–310, 2015.

WELGAMA, P.; GIBSON, P. A construction algorithm for the machine layout problem with fixed pick-up and drop-off points. **The International Journal of Production Research**, Taylor & Francis, v. 31, n. 11, p. 2575–2589, 1993.

WU, Y.; APPLETON, E. The optimisation of block layout and aisle structure by a genetic algorithm. **Computers & Industrial Engineering**, Elsevier, v. 41, n. 4, p. 371–387, 2002.

XIAO, X.; HU, Y.; WANG, W.; REN, W. A robust optimization approach for unequal-area dynamic facility layout with demand uncertainty. In: CONFERENCE ON MANUFACTURING SYSTEMS. Ljubljana, Slovenia, 2019. v. 81, p. 594–599.

XIAO, Y.; SEO, Y.; SEO, M. A two-step heuristic algorithm for layout design of unequal-sized facilities with input/output points. **International Journal of Production Research**, Taylor & Francis, v. 51, n. 14, p. 4200–4222, 2013.

XIE, W.; SAHINIDIS, N. V. A branch-and-bound algorithm for the continuous facility layout problem. **Computers & Chemical Engineering**, Elsevier, v. 32, n. 4-5, p. 1016–1028, 2008.

YANG, T.; PETERS, B. A. Flexible machine layout design for dynamic and uncertain production environments. **European Journal of Operational Research**, v. 108, n. 1, p. 49–64, 1998.

ZHA, S.; GUO, Y.; HUANG, S.; WANG, F.; HUANG, X. Robust facility layout design under uncertain product demands. **Manufacturing Systems 4.0 – Proceedings of the 50th CIRP Conference on Manufacturing Systems**, v. 63, p. 354–359, 2017.

ZHENG, X.-J. A connectivity graph generation approach for manhattan path calculation in detailed facility layout. **Applied Mathematics and Computation**, Elsevier, v. 237, p. 238–251, 2014.

ZUO, X.; MURRAY, C.; SMITH, A. Sharing clearances to improve machine layout. **International Journal of Production Research**, Taylor Francis, v. 54, n. 14, p. 4272–4285, 2016.

Apêndice A - Instâncias da Literatura

A.1 Instâncias de Tam (1992)/Nugent *et al.* (1968)

No artigo de Tam (1992), as instâncias referentes às áreas, razões de aspecto e dimensões do espaço disponível são apresentadas nas páginas 334 a 336; no artigo de Nugent *et al.* (1968), as instâncias referentes aos fluxos são apresentadas nas páginas 168 a 170.

Nº de facilidades (N)	Largura do espaço disponível (W)	Altura do espaço disponível (H)
12	40	25
15	44	25
20	40	35
30	45	40

TABELA A.1 – Número total de facilidades e dimensões do espaço disponível (Tabela válida para Tam92N12, Tam92N15, Tam92N20 e Tam92N30).

Facilidade (i)	Área (a_i)	*Lower bound* (lb_i)	*Upper bound* (ub_i)
1	100	0.7	1
2	80	1	1
3	50	0.7	1.3
4	60	0.5	0.8
5	120	0.9	1
6	40	0.6	1
7	20	0.7	1.4
8	40	1	1
9	150	0.8	1.1
10	120	0.5	1.5
11	50	0.7	1.1
12	10	0.8	1.2
13	20	0.95	1.5
14	30	0.75	1.25
15	50	0.9	1.1
16	20	0.8	1.5
17	40	0.4	1.4
18	20	0.9	1.9
19	80	1	1
20	100	0.95	1.15
21	40	0.5	1.5
22	50	1	1.1
23	80	0.6	1
24	10	0.9	1
25	40	0.8	1.1
26	10	0.5	1.2
27	40	0.8	1
28	10	0.5	1.3
29	80	0.9	1.05
30	40	0.9	1.1

TABELA A.2 – Área e limites da razão de aspecto das facilidades (Tabela válida para Tam92N12, Tam92N15, Tam92N20 e Tam92N30).

A.1.1 Instância Tam92N12

Facilidade fixa (i)	(xf_i)	(yf_i)	(xsf_i)	(ysf_i)
1	0	17	5	25
2	32	0	40	15

TABELA A.3 – Posição das facilidades fixas no espaço inicial (Tam92N12).

i	1	2	3	4	5	6	7	8	9	10	11	12
1	0	1	2	3	1	2	3	4	2	3	4	5
2	5	0	1	2	2	1	2	3	3	2	3	4
3	2	3	0	1	3	2	1	2	4	3	2	3
4	4	0	0	0	4	3	2	1	5	4	3	2
5	1	2	0	5	0	1	2	3	1	2	3	4
6	0	2	0	2	10	0	1	2	2	1	2	3
7	0	2	0	2	0	5	0	1	3	2	1	2
8	6	0	5	10	0	1	10	0	4	3	2	1
9	2	4	5	0	0	1	5	0	0	1	2	3
10	1	5	2	0	5	5	2	0	0	0	1	2
11	1	0	2	5	1	4	3	5	10	5	0	1
12	1	0	2	5	1	0	3	0	10	0	2	0

TABELA A.4 – Matriz de fluxos entre facilidades (Tam92N12).

A.1.2 Instância Tam92N15

Facilidade fixa (i)	(xf_i)	(yf_i)	(xsf_i)	(ysf_i)
1	0	0	10	5
2	0	20	10	25
3	30	10	40	16

TABELA A.5 – Posição das facilidades fixas no espaço inicial (Tam92N15).

i	1	2	3	4	5	6	7	8	9	10	11	12
1	0	1	2	3	1	2	3	4	2	3	4	5
2	5	0	1	2	2	1	2	3	3	2	3	4
3	2	3	0	1	3	2	1	2	4	3	2	3
4	4	0	0	0	4	3	2	1	5	4	3	2
5	1	2	0	5	0	1	2	3	1	2	3	4
6	0	2	0	2	10	0	1	2	2	1	2	3
7	0	2	0	2	0	5	0	1	3	2	1	2
8	6	0	5	10	0	1	10	0	4	3	2	1
9	2	4	5	0	0	1	5	0	0	1	2	3
10	1	5	2	0	5	5	2	0	0	0	1	2
11	1	0	2	5	1	4	3	5	10	5	0	1
12	1	0	2	5	1	0	3	0	10	0	2	0

TABELA A.6 – Matriz de fluxos entre facilidades (Tam92N15).

A.1.3 Instância Tam92N20

Facilidade fixa (i)	(xf_i)	(yf_i)	(xsf_i)	(ysf_i)
1	0	0	10	5
2	0	30	10	35
3	30	30	40	35
4	20	20	30	25

TABELA A.7 – Posição das facilidades fixas no espaço inicial (Tam92N20).

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	1	2	3	4	1	2	3	4	5	2	3	4	5	6	3	4	5	6	7
2	0	0	1	2	3	2	1	2	3	4	3	2	3	4	5	4	3	4	5	6
3	5	3	0	1	2	3	2	1	2	3	4	3	2	3	4	5	4	3	4	5
4	0	10	2	0	1	4	3	2	1	2	5	4	3	2	3	6	5	4	3	4
5	5	5	0	1	0	5	4	3	2	1	6	5	4	3	2	7	6	5	4	3
6	2	1	5	0	5	0	1	2	3	4	1	2	3	4	5	2	3	4	5	6
7	10	5	2	5	6	5	0	1	2	3	2	1	2	3	4	3	2	3	4	5
8	3	1	4	2	5	2	0	0	1	2	3	2	1	2	3	4	3	2	3	4
9	1	2	4	1	2	1	0	1	0	1	4	3	2	1	2	5	4	3	2	3
10	5	4	5	0	5	6	0	1	2	0	5	4	3	2	1	6	5	4	3	2
11	5	2	0	10	2	0	5	10	0	5	0	1	2	3	4	1	2	3	4	5
12	5	5	0	2	0	0	10	10	3	5	5	0	1	2	3	2	1	2	3	4
13	0	0	0	2	5	10	2	2	5	0	2	2	0	1	2	3	2	1	2	3
14	0	10	5	0	1	0	2	0	5	5	5	10	2	0	1	4	3	2	1	2
15	5	10	1	2	1	2	5	10	0	1	1	5	2	5	0	5	4	3	2	1
16	4	3	0	1	1	0	1	2	5	0	10	0	1	5	3	0	1	2	3	4
17	4	0	0	5	5	1	2	5	0	0	0	1	0	1	0	0	0	1	2	3
18	0	5	5	2	2	0	1	2	0	5	2	1	0	5	5	0	5	0	1	2
19	0	10	0	5	5	1	0	2	0	5	2	2	0	5	10	2	2	1	0	1
20	1	5	0	5	1	5	10	10	2	2	5	5	5	0	10	0	0	1	6	0

TABELA A.8 – Matriz de fluxos entre facilidades (Tam92N20).

A.1.4 Instância Tam92N30

Facilidade fixa (i)	(xf_i)	(yf_i)	(xsf_i)	(ysf_i)
1	0	0	10	10
2	40	0	45	10
3	35	35	45	40

TABELA A.9 – Posição das facilidades fixas no espaço inicial (Tam92N30).

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	0	1	2	3	4	5	1	2	3	4	5	6	2	3	4	5	6	7	3	4	5	6	7	8	4	5	6	7	8	9
2	3	0	1	2	3	4	2	1	2	3	4	5	3	2	3	4	5	6	4	3	4	5	6	7	5	4	5	6	7	8
3	2	4	0	1	2	3	3	2	1	2	3	4	4	3	2	3	4	5	5	4	3	4	5	6	6	5	4	5	6	7
4	0	0	3	0	1	2	4	3	2	1	2	3	5	4	3	2	3	4	6	5	4	3	4	5	7	6	5	4	5	6
5	0	10	4	0	0	1	5	4	3	2	1	2	6	5	4	3	2	3	7	6	5	4	3	4	8	7	6	5	4	5
6	2	4	0	0	5	0	6	5	4	3	2	1	7	6	5	4	3	2	8	7	6	5	4	3	9	8	7	6	5	4
7	10	0	5	0	2	1	0	1	2	3	4	5	1	2	3	4	5	6	2	3	4	5	6	7	3	4	5	6	7	8
8	5	0	5	2	0	2	10	0	1	2	3	4	2	1	2	3	4	5	3	2	3	4	5	6	4	3	4	5	6	7
9	0	2	5	2	0	2	10	1	0	1	2	3	3	2	1	2	3	4	4	3	2	3	4	5	5	4	3	4	5	6
10	5	2	1	0	0	1	5	3	10	0	1	2	4	3	2	1	2	3	5	4	3	2	3	4	6	5	4	3	4	5
11	2	1	4	6	0	4	10	5	2	5	0	1	5	4	3	2	1	2	6	5	4	3	2	3	7	6	5	4	3	4
12	5	0	1	0	2	10	10	0	1	5	0	0	6	5	4	3	2	1	7	6	5	4	3	4	8	7	6	5	4	3
13	0	5	0	2	0	10	6	0	5	6	0	5	0	1	2	3	4	5	1	2	3	4	5	6	2	3	4	5	6	7
14	0	0	4	5	0	2	0	0	2	0	1	5	2	0	1	2	3	4	2	1	2	3	4	5	3	2	1	2	3	4
15	2	0	0	2	0	5	0	2	0	1	2	2	0	2	0	1	2	3	3	2	1	2	3	4	4	3	2	3	4	5
16	0	0	4	5	0	5	10	4	3	5	1	0	4	1	4	0	1	2	4	3	2	1	2	3	5	4	3	2	3	4
17	5	0	0	1	2	0	2	5	0	5	0	0	2	0	5	0	0	1	5	4	3	2	1	2	6	5	4	3	2	3
18	6	2	6	1	1	5	1	2	2	0	2	0	2	0	1	3	2	0	6	5	4	3	2	1	7	6	5	4	3	2
19	3	0	3	1	0	0	10	10	0	5	0	0	1	3	0	0	2	5	0	1	2	3	4	5	1	2	3	4	5	6
20	0	1	2	1	0	0	1	6	0	2	0	2	0	10	1	2	0	1	0	0	1	2	3	4	2	1	2	3	4	5
21	1	6	5	2	2	0	5	0	4	3	0	0	6	0	0	2	0	2	5	5	0	1	2	3	3	2	1	2	3	4
22	10	1	5	2	0	10	5	5	0	5	6	4	2	0	5	0	0	10	5	2	4	0	1	2	4	3	2	1	2	3
23	0	0	2	4	5	0	2	5	5	0	6	5	1	4	0	2	6	10	1	1	0	5	0	1	5	4	3	2	1	2
24	10	1	1	0	1	0	3	2	2	5	0	10	5	2	2	0	5	4	0	3	1	0	0	0	6	5	4	3	2	1
25	2	2	0	2	0	0	5	5	0	2	4	1	5	0	0	5	3	0	5	1	0	4	4	5	0	1	2	3	4	5
26	1	2	0	0	2	4	0	0	5	10	5	0	0	0	0	0	5	0	2	5	0	4	4	5	1	0	1	2	3	4
27	1	5	3	2	1	0	2	5	2	10	3	0	0	4	5	5	0	5	1	6	0	5	1	0	0	0	0	1	2	3
28	1	1	1	2	0	10	0	5	2	1	2	0	1	2	1	2	0	0	2	5	5	0	0	1	10	0	0	0	1	2
29	0	10	0	5	2	1	1	0	5	5	2	0	5	5	1	5	5	0	10	5	0	2	2	0	1	0	0	0	0	1
30	1	5	2	5	1	1	3	2	2	2	10	1	5	5	0	10	1	0	10	3	0	5	2	0	0	0	10	2	2	0

TABELA A.10 – Matriz de fluxos entre facilidades (Tam92N30).

A.2 Instâncias de Das (1993)

No artigo de Das (1993), as instâncias são apresentadas nas páginas 290 e 291.

A.2.1 Instância Das93N4

$N = 4$	$W = 38$	$H = 38$	Ponto de entrada		Ponto de saída	
Facilidade (i)	Largura (w_i)	Comprimento (h_i)	(Ix_i)	(Iy_i)	(Ox_i)	(Oy_i)
1	18	10	9	7,5	9	7,5
2	9	5	5,5	2,5	5,5	2,5
3	10	10	5	6,2	5	6,2
4	20	15	13,2	7,5	13,2	7,5

TABELA A.11 – Características das facilidades (Das93N4).

i	1	2	3	4
1	0	20	30	15
2	0	0	5	38
3	0	0	0	12
4	0	0	0	0

TABELA A.12 – Matriz de fluxos entre facilidades (Das93N4).

A.2.2 Instância Das93N6

$N = 6$	$W = 40$	$H = 40$	Ponto de entrada		Ponto de saída	
Facilidade (i)	Largura (w_i)	Comprimento (h_i)	(Ix_i)	(Iy_i)	(Ox_i)	(Oy_i)
1	10	8	6,5	4	6,5	4
2	20	15	10	10,5	10	10,5
3	12	10	6	6,2	6	6,2
4	12	8	8,6	4	8,6	4
5	8	8	5,6	4	5,6	4
6	9	6	4,5	4,1	4,5	4,1

TABELA A.13 – Características das facilidades (Das93N6).

i	1	2	3	4	5	6
1	0	50	0	15	16	34
2	0	0	28	11	6	0
3	0	0	0	45	0	10
4	0	0	0	0	0	15
5	0	0	0	0	0	12
6	0	0	0	0	0	0

TABELA A.14 – Matriz de fluxos entre facilidades (Das93N6).

A.2.3 Instância Das93N8

$N = 8$	$W = 50$	$H = 50$	Ponto de entrada		Ponto de saída	
Facilidade (i)	Largura (w_i)	Comprimento (h_i)	(Ix_i)	(Iy_i)	(Ox_i)	(Oy_i)
1	14	10	10	5	10	5
2	10	7	5	4,5	5	4,5
3	10	10	6,5	5	6,5	5
4	20	15	14,2	7,5	14,2	7,5
5	18	13	9	9	9	9
6	8	8	4,3	4	4,3	4
7	9	6	4,5	4,1	4,5	4,1
8	15	11	7,5	6,9	7,5	6,9

TABELA A.15 – Características das facilidades (Das93N8).

i	1	2	3	4	5	6	7	8
1	0	30	12	16	15	0	4	11
2	0	0	6	0	4	3	0	0
3	0	0	0	30	16	85	0	31
4	0	0	0	0	26	60	34	18
5	0	0	0	0	0	30	24	45
6	0	0	0	0	0	0	41	19
7	0	0	0	0	0	0	0	48
8	0	0	0	0	0	0	0	0

TABELA A.16 – Matriz de fluxos entre facilidades (Das93N8).

A.2.4 Instância Das93N10

$N = 10$	$W = 63$	$H = 63$	Ponto de entrada		Ponto de saída	
Facilidade (i)	Largura (w_i)	Comprimento (h_i)	(Ix_i)	(Iy_i)	(Ox_i)	(Oy_i)
1	15	10	7,5	7,2	7,5	7,2
2	12	8	8,5	4	8,5	4
3	15	12	11	6	11	6
4	9	9	6	4,5	6	4,5
5	9	6	4,5	4,1	4,5	4,1
6	25	18	17	9	17	9
7	20	16	15,5	8	15,5	8
8	8	5	4	3,2	4	3,2
9	11	11	7,1	5,5	7,1	5,5
10	19	13	9,5	11	9,5	11

TABELA A.17 – Características das facilidades (Das93N10).

i	1	2	3	4	5	6	7	8	9	10
1	0	10	35	18	25	45	11	0	31	16
2	0	0	0	10	18	0	4	11	30	0
3	0	0	0	16	0	53	0	30	61	8
4	0	0	0	0	18	25	11	8	29	19
5	0	0	0	0	0	63	30	16	0	21
6	0	0	0	0	0	0	0	40	13	14
7	0	0	0	0	0	0	0	0	17	79
8	0	0	0	0	0	0	0	0	0	11
9	0	0	0	0	0	0	0	0	0	35
10	0	0	0	0	0	0	0	0	0	0

TABELA A.18 – Matriz de fluxos entre facilidades (Das93N10).

A.2.5 Instância Das93N12

$N = 12$	$W = 58$	$H = 58$	Ponto de entrada		Ponto de saída	
Facilidade (i)	Largura (w_i)	Comprimento (h_i)	(Ix_i)	(Iy_i)	(Ox_i)	(Oy_i)
1	9	7	4,5	5	4,5	5
2	14	10	7	7,5	7	7,5
3	12	10	8,5	5	8,5	5
4	8	8	5,2	4	5,2	4
5	12	12	8,7	6	8,7	6
6	10	8	5	5,2	5	5,2
7	20	14	14,5	7	14,5	7
8	15	11	10,5	5,5	10,5	5,5
9	12	9	6	5,6	6	5,6
10	10	10	7,1	5	7,1	5
11	11	11	8,3	5,5	8,3	5,5
12	12	8	6	5,7	6	5,7

TABELA A.19 – Características das facilidades (Das93N12).

i	1	2	3	4	5	6	7	8	9	10	11	12
1	0	31	14	38	19	11	15	12	10	18	26	50
2	0	0	60	17	17	40	33	12	17	0	14	17
3	0	0	0	61	70	18	42	13	21	25	35	43
4	0	0	0	0	50	10	12	60	63	18	51	15
5	0	0	0	0	0	18	71	71	60	36	25	21
6	0	0	0	0	0	0	19	17	49	53	25	22
7	0	0	0	0	0	0	0	11	13	26	39	52
8	0	0	0	0	0	0	0	0	13	28	42	17
9	0	0	0	0	0	0	0	0	0	15	14	12
10	0	0	0	0	0	0	0	0	0	0	17	31
11	0	0	0	0	0	0	0	0	0	0	0	19
12	0	0	0	0	0	0	0	0	0	0	0	0

TABELA A.20 – Matriz de fluxos entre facilidades (Das93N12).

A.3 Instâncias de Welgama e Gibson (1993)

No artigo de Welgama e Gibson (1993), as instâncias são apresentadas nas páginas 2584 e 2585.

A.3.1 Instância Wel93N6

$N = 6$	$W = 27$	$H = 27$	Ponto de entrada		Ponto de saída	
Facilidade (i)	Largura (w_i)	Comprimento (h_i)	(Ix_i)	(Iy_i)	(Ox_i)	(Oy_i)
1	10	5	0	2,5	10	2,5
2	5	5	0	2,5	5	2,5
3	20	5	10	0	10	5
4	8	6	4	0	4	0
5	12	4	0	2	6	0
6	9	6	4,5	0	0	3

TABELA A.21 – Características das facilidades (Wel93N6).

i	1	2	3	4	5	6
1	0	1	2	1	2	3
2	5	0	1	2	1	2
3	2	3	0	3	2	1
4	4	0	0	0	1	2
5	1	2	0	5	0	1
6	0	2	0	2	10	0

TABELA A.22 – Matriz de fluxos entre facilidades (Wel93N6).

A.3.2 Instância Wel93N12

$N = 12$	$W = 57$	$H = 57$	Ponto de entrada		Ponto de saída	
Facilidade (i)	Largura (w_i)	Comprimento (h_i)	(Ix_i)	(Iy_i)	(Ox_i)	(Oy_i)
1	10	5	0	2,5	10	2,5
2	2	1	0	0,5	1	0
3	5	5	0	2,5	0	2,5
4	8	6	4	0	0	3
5	20	5	10	5	10	0
6	15	10	7,5	0	7,5	0
7	20	20	0	10	10	20
8	4	1	0	0,5	2	0
9	50	5	0	2,5	50	2,5
10	10	10	5	0	5	0
11	10	6	0	3	0	3
12	20	14	0	7	20	7

TABELA A.23 – Características das facilidades (Wel93N12).

i	1	2	3	4	5	6	7	8	9	10	11	12
1	0	1	2	3	1	2	3	4	2	3	4	5
2	5	0	1	2	2	1	2	3	3	2	3	4
3	2	3	0	1	3	2	1	2	4	3	2	3
4	4	0	0	0	4	3	2	1	5	4	3	2
5	1	2	0	5	0	1	2	3	1	2	3	4
6	0	2	0	2	10	0	1	2	2	1	2	3
7	0	2	0	2	0	5	0	1	3	2	1	2
8	6	0	5	10	0	1	10	0	4	3	2	1
9	2	4	5	0	0	1	5	0	0	1	2	3
10	1	5	2	0	5	5	2	0	0	0	1	2
11	1	0	2	5	1	4	3	5	10	5	0	1
12	1	0	2	5	1	0	3	0	10	0	2	0

TABELA A.24 – Matriz de fluxos entre facilidades (Wel93N12).

A.4 Instâncias de Deb e Bhattacharyya (2005)

No artigo de Deb e Bhattacharyya (2005), as instâncias são apresentadas na página 4801.

Nº de facilidades (N)	Largura do espaço disponível (W)	Altura do espaço disponível (H)
12	220	220
18	254	254

TABELA A.25 – Número total de facilidades e dimensões do espaço disponível (Tabela válida para Deb05N12 e Deb05N18).

			Ponto de entrada		Ponto de saída	
Facilidade (i)	Largura (w_i)	Comprimento (h_i)	(Ix_i)	(Iy_i)	(Ox_i)	(Oy_i)
1	40	20	-	-	-	-
2	8	4	-	-	-	-
3	20	20	-	-	-	-
4	32	24	-	-	-	-
5	80	20	-	-	-	-
6	60	40	-	-	-	-
7	80	80	-	-	-	-
8	16	4	-	-	-	-
9	100	20	-	-	-	-
10	40	40	-	-	-	-
11	40	24	-	-	-	-
12	80	56	-	-	-	-
13	40	40	-	-	-	-
14	20	20	-	-	-	-
15	32	24	-	-	-	-
16	80	20	-	-	-	-
17	20	20	-	-	-	-
18	60	40	-	-	-	-

TABELA A.26 – Características das facilidades (Tabela válida para Deb05N12 e Deb05N18).

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	0	1	2	3	1	2	3	4	2	3	4	5	1	5	2	0	5	4
2	5	0	1	2	2	1	2	3	3	2	3	4	1	0	2	5	1	0
3	2	3	0	1	3	2	1	2	4	3	2	3	4	3	2	1	5	4
4	4	0	0	0	4	3	2	1	5	4	3	2	1	2	2	1	2	3
5	1	2	0	5	0	1	2	3	1	2	3	4	5	1	0	3	0	10
6	0	2	0	2	10	0	1	2	2	1	2	3	0	4	3	2	1	5
7	0	2	0	2	0	5	0	1	3	2	1	2	3	0	1	3	2	1
8	6	0	5	10	0	1	10	0	4	3	2	1	1	3	2	1	2	4
9	2	4	5	0	0	1	5	0	0	1	2	3	1	5	4	3	2	1
10	1	5	2	0	5	4	2	0	0	0	1	2	1	0	2	5	1	0
11	1	0	2	5	1	5	3	5	10	5	0	1	0	4	3	2	1	5
12	1	0	2	5	1	0	3	0	10	0	2	0	5	0	1	2	2	1
13	0	2	0	2	0	5	0	1	3	2	1	2	0	10	0	4	3	2
14	1	2	0	5	0	1	2	3	1	2	3	4	7	0	2	1	2	4
15	5	0	1	2	2	1	2	3	3	2	3	4	2	6	0	8	0	1
16	2	4	5	0	0	1	5	0	0	1	2	3	3	0	5	0	2	9
17	2	3	0	1	3	2	1	2	4	3	2	3	2	1	2	4	0	4
18	0	2	0	2	0	5	0	1	3	2	1	2	4	6	0	3	2	0

TABELA A.27 – Matriz de fluxos entre facilidades (Tabela válida para Deb05N12 e Deb05N18).

A.5 Instância de Dunker *et al.* (2003)

No artigo de Dunker *et al.* (2003), a instância é apresentada na página 3497.

$N = 62$	$W = 176$	$H = 176$	Ponto de entrada		Ponto de saída	
Facilidade (i)	Largura (w_i)	Comprimento (h_i)	(Ix_i)	(Iy_i)	(Ox_i)	(Oy_i)
1	15	14	7,5	7	7,5	7
2	13	10	6,5	5	6,5	5
3	16	14	8	7	8	7
4	20	13	10	6,5	10	6,5
5	16	13	8	6,5	8	6,5
6	21	14	10,5	7	10,5	7
7	19	17	9,5	8,5	9,5	8,5
8	19	14	9,5	7	9,5	7
9	21	21	10,5	10,5	10,5	10,5
10	20	17	10	8,5	10	8,5
11	13	11	6,5	5,5	6,5	5,5
12	14	12	7	6	7	6
13	19	18	9,5	9	9,5	9
14	21	17	10,5	8,5	10,5	8,5
15	21	20	10,5	10	10,5	10
16	21	7	10,5	3,5	10,5	3,5
17	20	19	10	9,5	10	9,5
18	16	9	8	4,5	8	4,5
19	17	11	8,5	5,5	8,5	5,5
20	16	15	8	7,5	8	7,5
21	14	10	7	5	7	5
22	19	16	9,5	8	9,5	8
23	20	15	10	7,5	10	7,5
24	18	9	9	4,5	9	4,5
25	18	14	9	7	9	7
26	14	14	7	7	7	7
27	16	11	8	5,5	8	5,5
28	12	12	6	6	6	6
29	17	13	8,5	6,5	8,5	6,5
30	13	10	6,5	5	6,5	5
31	14	13	7	6,5	7	6,5
32	17	8	8,5	4	8,5	4
33	21	19	10,5	9,5	10,5	9,5

TABELA A.28 – (continuação)

Facilidade (i)	Largura (w_i)	Comprimento (h_i)	(Ix_i)	(Iy_i)	(Ox_i)	(Oy_i)
34	21	10	10,5	5	10,5	5
35	15	10	7,5	5	7,5	5
36	12	9	6	4,5	6	4,5
37	21	17	10,5	8,5	10,5	8,5
38	16	9	8	4,5	8	4,5
39	18	14	9	7	9	7
40	15	9	7,5	4,5	7,5	4,5
41	17	12	8,5	6	8,5	6
42	17	12	8,5	6	8,5	6
43	11	9	5,5	4,5	5,5	4,5
44	20	8	10	4	10	4
45	21	17	10,5	8,5	10,5	8,5
46	20	13	10	6,5	10	6,5
47	19	10	9,5	5	9,5	5
48	20	14	10	7	10	7
49	18	10	9	5	9	5
50	13	8	6,5	4	6,5	4
51	18	11	9	5,5	9	5,5
52	16	10	8	5	8	5
53	20	10	10	5	10	5
54	19	19	9,5	9,5	9,5	9,5
55	21	11	10,5	5,5	10,5	5,5
56	20	7	10	3,5	10	3,5
57	11	7	5,5	3,5	5,5	3,5
58	17	11	8,5	5,5	8,5	5,5
59	21	11	10,5	5,5	10,5	5,5
60	13	7	6,5	3,5	6,5	3,5
61	21	10	10,5	5	10,5	5
62	17	16	8,5	8	8,5	8

TABELA A.28 – Características das facilidades (Dun03N62).

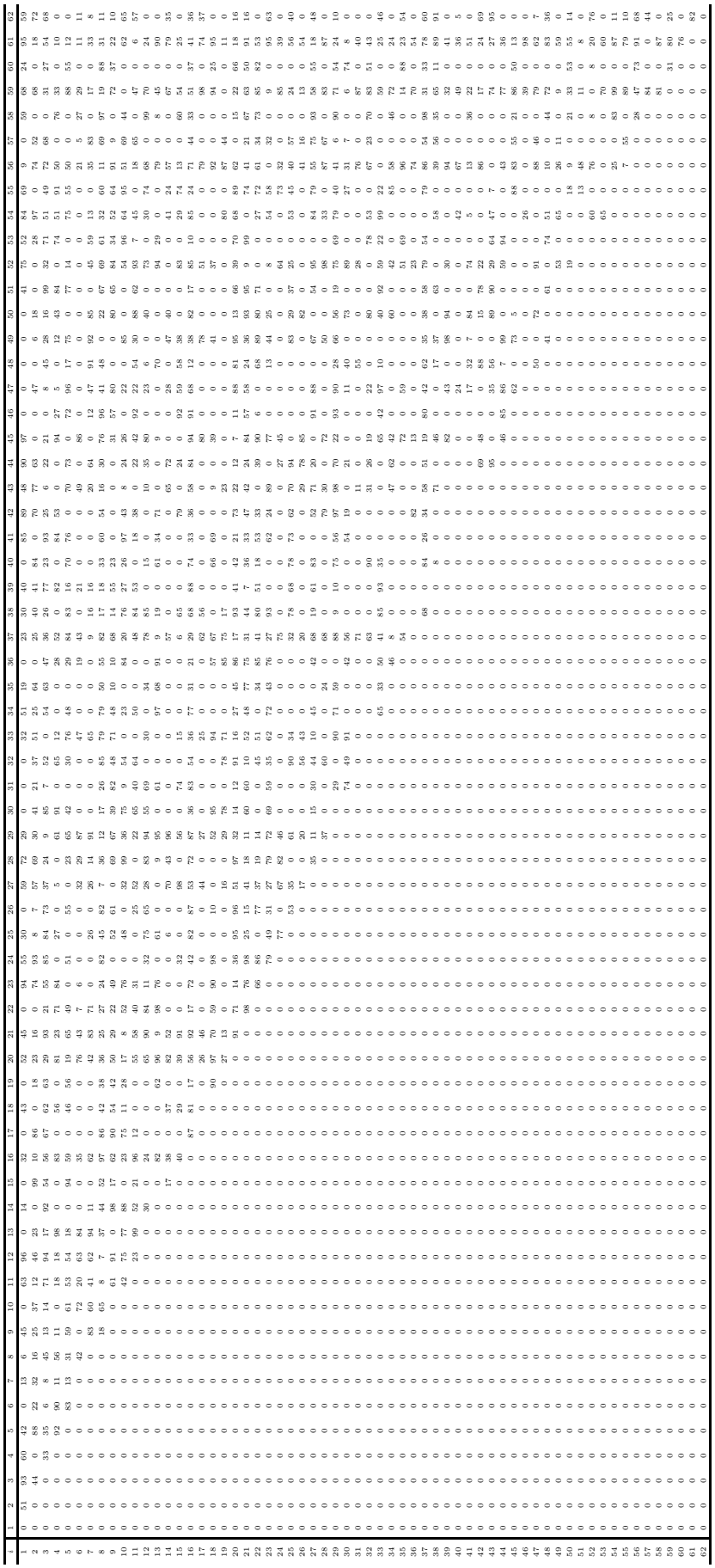

TABELA A.29 – Matriz de fluxos entre facilidades (Dun03N62).

FOLHA DE REGISTRO DO DOCUMENTO

1. CLASSIFICAÇÃO/TIPO	2. DATA	3. DOCUMENTO Nº	4. Nº DE PÁGINAS
TD	26 de agosto de 2022	DCTA/ITA/DM-001/2022	119

5. TÍTULO E SUBTÍTULO:

Modelagem e Resolução do Problema de Layout de Facilidades Robusto de Áreas Desiguais com Locais de Entrada e Saída

6. AUTORA(ES):

Evelyn Michelle Henrique Braga

7. INSTITUIÇÃO(ÕES)/ÓRGÃO(S) INTERNO(S)/DIVISÃO(ÕES):

Instituto Tecnológico de Aeronáutica – ITA / Universidade Federal de São Paulo – UNIFESP

8. PALAVRAS-CHAVE SUGERIDAS PELA AUTORA:

Planejamento e projeto de facilidades; Problema de layout de facilidades de áreas desiguais; Programação inteira mista

9. PALAVRAS-CHAVE RESULTANTES DE INDEXAÇÃO:

Planejamento e projeto de facilidades; Problema de layout de facilidades de áreas desiguais; Programação inteira mista

10. APRESENTAÇÃO: (**X**) **Nacional** ( ) **Internacional**

ITA/UNIFESP, São José dos Campos. Curso de Doutorado. Programa de Pós-Graduação em Pesquisa Operacional. Área de Engenharia de Produção/Pesquisa Operacional. Orientador: Prof. Dr. Luiz Leduino de Salles Neto. Defesa em 26/08/2022. Publicada em 26/08/2022.

11. RESUMO:

Uma das estratégias utilizadas para otimizar processos de produção é a definição do melhor arranjo físico. Neste sentido, muitas empresas realizam um estudo do posicionamento relativo dos seus diversos equipamentos, áreas ou atividades funcionais. A disposição adequada das instalações pode resultar em um menor tempo de processo e maior rendimento dos fatores de produção. Em geral, o objetivo do problema de layout de facilidades é reduzir o custo de manuseio de material, que pode ser representado por uma função que relaciona os fluxos de materiais e as distâncias entre as facilidades. O problema de layout de facilidades dinâmico considera um horizonte de planejamento multi-período no qual os fluxos de materiais entre pares de facilidades podem mudar com o tempo e pode ser resolvido com uma abordagem flexível ou robusta. Uma planta flexível é aquela capaz de aceitar alterações no posicionamento das instalações ao longo do tempo para acompanhar as mudanças de demanda desde que compense os custos de realocação, enquanto que uma planta robusta é uma solução de layout único que pode não ser excelente para nenhum dos cenários individualmente, mas busca ser a melhor quando se avalia o conjunto destes. Considerando um ambiente dinâmico de demanda, o presente trabalho propôs-se a realizar uma análise robusta, ao invés de flexível, por considerar que custos de rearranjo e interrupção da produção são altos, além de que uma abordagem adaptativa pode ser demasiadamente inconveniente na rotina de uma indústria. Dentro do problema de layout existem pontos pouco explorados que foram tratados nesta pesquisa, a saber, blocos de áreas desiguais, locais de entrada e saída nos blocos, facilidades fixas ou espaços ocupados e distância de folga em torno das facilidades. Nesse contexto foram desenvolvidos e validados novos modelos matemáticos de programação inteira mista para o problema de layout de facilidades, a partir da análise de modelos propostos na literatura. Como alternativa para a implementação em programas de modelagem e uso de *softwares* matemáticos, também foi efetuada a linearização desses modelos. A fim de oferecer soluções para os problemas modelados foram desenvolvidos e aplicados alguns métodos de resolução envolvendo programação matemática e heurísticas que obtiveram bons resultados para diferentes instâncias da literatura.

12. GRAU DE SIGILO:

(**X**) **OSTENSIVO** ( ) **RESERVADO** ( ) **SECRETO**