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

Evelyn Michelle Henrique Braga [1,2], **Luiz Leduino de Salles Neto** [1]
[1]Instituto de Ciência e Tecnologia - Universidade Federal de São Paulo
Av. Cesare Mansueto Giulio Lattes, nº 1.201, São José dos Campos/SP
[2]Instituto Tecnológico de Aeronáutica
Praça Marechal Eduardo Gomes, nº 50, São José dos Campos/SP
evelyn.braga@unifesp.br, luiz.leduino@unifesp.br

RESUMO

Uma das estratégias utilizadas para otimizar processos de produção é a definição do melhor arranjo físico. Nesse sentido, é realizado um estudo do posicionamento relativo dos diversos equipamentos, áreas ou atividades funcionais dentro da empresa. A disposição adequada das instalações resultará em um menor tempo de processo e maior rendimento dos fatores de produção. O presente artigo trata do Problema de Layout de Facilidades considerando blocos retangulares com dimensões fixas de áreas desiguais e locais de entrada e saída nos blocos. Nesse contexto, foram desenvolvidos novos modelos matemáticos de programação inteira mista com base em pesquisas anteriores, posteriormente linearizados. Um método de solução exata foi aplicado obtendo excelentes resultados para diferentes instâncias em comparação com aqueles encontrados na literatura.

PALAVRAS CHAVE. Problema de layout de facilidades de áreas desiguais. Locais de entrada e saída. Programação inteira mista.

Tópicos (PO na Indústria, Programação Matemática, Logística e Transportes)

ABSTRACT

One of the strategies used to optimize production processes is to define the best layout. For this, the relative positioning of the various equipment, areas, or functional activities inside the company is studied. Proper arrangement of facilities will result in shorter process times and higher productivity. In this paper, the unequal area of rectangular blocks with fixed dimensions and input/output points are considered for Facility Layout Problems. New mathematical models of mixed-integer programming based on previous research formulations are developed. An exact solution method is applied, which obtained excellent results for different test problems by comparing to those reported in the literature.

KEYWORDS. Unequal area facility layout problem. Input and output points. Mixed integer programming.

Topics (OP in Industry, Mathematical Programming, Logistics and Transport)

1. Introdução

Em um mercado global cada vez mais competitivo, maximizar a produtividade é fundamental. Uma das análises realizadas para atingir esse potencial é a do arranjo físico mais adequado de forma a agilizar os processos produtivos. No contexto industrial, a execução de um produto passa por diversas etapas que podem ser caracterizadas por facilidades, como por exemplo, instalações, máquinas, equipamentos, setores ou departamentos. Um arranjo, ou layout, pode ser definido como o posicionamento relativo das diversas facilidades. Organizando o layout é possível melhorar a utilização do espaço disponível, diminuir as distâncias de movimentação de materiais, serviços e pessoas, incrementar a produção e reduzir os custos indiretos e o tempo de manufatura. Por essas razões, o Problema de Layout de Facilidades - FLP (do inglês, *Facility Layout Problem*) 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].

Convencionalmente, a função objetivo do problema de layout de instalações consiste na redução do Custo de Manuseio de Material Total - TMHC (do inglês, *Total Material Handling Cost*), um custo sem valor agregado que relaciona os fluxos de materiais e as distâncias entre as facilidades. A aplicação de designs de localização incorretos impacta diretamente na medida do TMHC. Estima-se que o planejamento eficiente das instalações pode reduzir o TMHC em até 30% [Tompkins et al., 2010], diminuindo assim o custo operacional total do projeto.

Embora o FLP já tenha sido amplamente abordado na literatura, existem várias peculiaridades a serem exploradas que merecem atenção como facilidades na forma de blocos de áreas desiguais e locais de entrada e saída dos processos nos blocos. O estudo dessas problemáticas se justifica, pois tornam o desafio mais complexo e dentro da realidade encontrada no setor de produção.

Quando o fluxo de materiais entre as instalações é um parâmetro que não muda com o tempo, o problema é conhecido como FLP Estático e, na forma mais simples, com blocos de mesmo tamanho, pode ser formulado como um Problema de Atribuição Quadrática [Emami e Nookabadi, 2013]. Porém quando os departamentos têm áreas diferentes, o problema não pode mais ser resolvido atribuindo-se n departamentos a n localizações distintas e as distâncias passam a ser variáveis. As complicações adicionais dos requisitos de áreas desiguais, com áreas variadas (largura e altura) e posições de departamento contínuas que podem estar em qualquer lugar de uma área retangular, tornam o FLP extremamente difícil de resolver [Castillo e Westerlund, 2005]. O Problema de Layout de Facilidades de Áreas Desiguais - UAFLP (do inglês, *Unequal Area Facility Layout Problem*) foi primeiramente reportado por Armour e Buffa [1963] mais de 50 anos atrás. Com o passar do tempo, vários foram os autores que propuseram métodos de resolução para o UAFLP, como por exemplo Van Camp et al. [1992], Konak et al. [2006], Kulturel-Konak [2012] e, mais recentemente, García-Hernández et al. [2019] e Liu et al. [2020].

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 dos blocos. Tratando de facilidades, 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 modelos de FLPs tradicionais, os pontos de entrada e saída - I/O *points* (do inglês, *Input/Output points*) das instalações são ignorados e somente determinados após a obtenção de um layout de bloco [Kim e Kim, 2000]. Quando considerados, os I/O *points* podem ser fixos (predeterminados) ou variáveis com restrições, a depender do tipo de facilidade que está sendo tratada no problema. O'brien e Abdel Barr [1980] foram os primeiros a considerar I/O *points* nos dados básicos de entrada exigidos pelo seu procedimento. Ao longo do tempo, muitos autores propuseram métodos de resolução para

a situação de I/O *points* no FLP, a exemplo de Welgama e Gibson [1993], Rajasekharan et al. [1998], Kim e Kim [2000], Hu et al. [2007], Leno et al. [2018] e Kulturel-Konak [2019].

O presente artigo está organizado em cinco seções. A Seção 2 apresenta os modelos matemáticos desenvolvidos para o problema. A Seção 3 descreve a abordagem de resolução usada. A Seção 4 mostra os resultados obtidos e a comparação com os encontrados na literatura. Por fim, a Seção 5 traz as conclusões e algumas propostas de trabalho futuro.

2. Modelagem Matemática

Para esse trabalho foram formulados quatro modelos de Programação Inteira Mista - MIP *models* (do inglês, *Mixed Integer Programming models*) para o UAFLP com I/O *points* fixos e variáveis, considerando medida de distância retilínea e facilidades regulares do tipo retangulares, de dimensões fixas de altura e largura e sujeitas a um sistema *open-field*. Como base para os MIP *models* desenvolvidos, foram tomadas as propostas de Hu et al. [2007] e Leno et al. [2018].

Embora os quatro modelos desenvolvidos neste trabalho sejam similares, cada um representa uma situação particular. A diferença entre eles está na condição de localização dos I/O *points*, como explicado a seguir:

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

A Figura 1 ilustra as três possibilidades para I/O *points* variáveis.

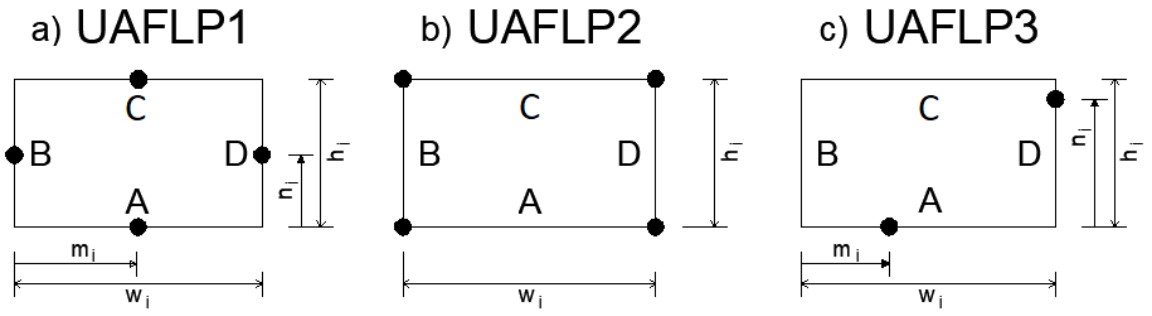

Figura 1: Departamentos com locais candidatos aos I/O *points*: a) quatro possibilidades localizadas no meio das bordas do bloco; b) quatro opções localizadas nos cantos do bloco; e c) podem assumir infinitas posições ao redor do bloco nas bordas.

Como os modelos possuem função objetivo e diversas restrições idênticas entre si, optou-se por dividi-los em duas partes, uma comum e uma particular, e apresentá-los em cinco tópicos. O primeiro tópico traz exatamente a função objetivo e as equações comuns a todos os modelos, ou seja, pode ser considerada como a primeira parte de qualquer um dos quatro modelos individualmente. Os demais tópicos subsequentes trazem a segunda parte de cada um dos modelos separadamente, pois nesse ponto 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 1.

Conjunto, Índices, Dados de Entrada e Marcadores	
N	número total de facilidades/locais
i e j	índice para facilidades = (1,...,N)
f_{ij}	fluxo de peças entre as facilidades i e j
W,H	largura e altura do espaço disponível
w,h	largura e altura original da facilidade i
Ix_i, Iy_i,Ox_i, Oy_i	coordenadas x e y originais da entrada e da saída da facilidade i
A, B, C, D	lados das facilidades (inferior, esquerdo, superior e direito)
Variáveis de Decisão Contínuas e Binárias	
d_{ij}	distância entre a saída da facilidade i e a entrada de j
wr_i,hr_i	largura e 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,x_i^O, y_i^O	coordenadas x e y da entrada e da saída da facilidade i
m_i^I, n_i^I,m_i^O, n_i^O	posição perimetral nas direções x e y da entrada e da saída da facilidade i
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
u_i, v_i	orientação relativa: é igual a (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; e igual a (1,1) se a facilidade i está rotacionada 270° no sentido horário
p_i^I, q_i^I,p_i^O, q_i^O	posição lateral: é igual a (0,0) se a entrada e a saída da facilidade i estão localizadas no lado A; (1,0) se no lado B; (0,1) se no lado C; e igual a (1,1) se no lado D

Tabela 1: Notações usadas nos Modelos 1, 2, 3 e 4

1ª parte dos Modelos 1, 2, 3 e 4

$$\min \sum_{i=1}^{N} \sum_{j=1}^{N} f_{ij} d_{ij} \tag{1}$$

sujeito a

$$wr_i = (1 - u_i)w_i + u_i h_i \qquad \forall i \tag{2}$$
$$hr_i = (1 - u_i)h_i + u_i w_i \qquad \forall i \tag{3}$$
$$xs_i = x_i + wr_i \qquad \forall i \tag{4}$$
$$ys_i = y_i + hr_i \qquad \forall i \tag{5}$$
$$l_{ij} + l_{ji} + b_{ij} + b_{ji} \geq 1 \qquad \forall i, j; i < j \tag{6}$$
$$xs_i \leq l_{ij} x_j + W(1 - l_{ij}) \qquad \forall i, j \tag{7}$$
$$ys_i \leq b_{ij} y_j + H(1 - b_{ij}) \qquad \forall i, j \tag{8}$$
$$d_{ij} = \left| x_i^O - x_j^I \right| + \left| y_i^O - y_j^I \right| \qquad \forall i, j \tag{9}$$
$$d_{ij}, wr_i, hr_i, x_i, y_i, xs_i, ys_i, x_i^{I(O)}, y_i^{I(O)}, m_i^{I(O)}, n_i^{I(O)} \geq 0 \qquad \forall i \tag{10}$$
$$u_i, v_i, p_i^{I(O)}, q_i^{I(O)} \in \{0, 1\} \qquad \forall i \tag{11}$$
$$l_{ij}, b_{ij} \in \{0, 1\} \qquad \forall i, j \tag{12}$$

2ª parte do Modelo 1 (UAFLPfixo)

$$m_i^{I(O)} = n_i^{I(O)} = p_i^{I(O)} = q_i^{I(O)} = 0 \qquad \forall i \tag{13}$$

$$\begin{aligned} x_i^{I(O)} = x_i + (1-u_i)(1-v_i)I(O)x_i + u_i(1-v_i)I(O)y_i \\ +(1-u_i)v_i(w_i - I(O)x_i) + u_iv_i(h_i - I(O)y_i) \qquad \forall i \end{aligned} \tag{14}$$

$$\begin{aligned} y_i^{I(O)} = y_i + (1-u_i)(1-v_i)I(O)y_i + u_i(1-v_i)(w_i - I(O)x_i) \\ +(1-u_i)v_i(h_i - I(O)y_i) + u_iv_iI(O)x_i \qquad \forall i \end{aligned} \tag{15}$$

2ª parte do Modelo 2 (UAFLP1)

$$m_i^{I(O)} = \frac{wr_i}{2} \qquad \forall i \tag{16}$$

$$n_i^{I(O)} = \frac{hr_i}{2} \qquad \forall i \tag{17}$$

$$x_i^{I(O)} = x_i + (1-p_i^{I(O)})m_i^{I(O)} + p_i^{I(O)}q_i^{I(O)}wr_i \qquad \forall i \tag{18}$$

$$y_i^{I(O)} = y_i + p_i^{I(O)}n_i^{I(O)} + (1-p_i^{I(O)})q_i^{I(O)}hr_i \qquad \forall i \tag{19}$$

2ª parte do Modelo 3 (UAFLP2)

$$m_i = 0 \qquad \forall i \tag{20}$$

$$n_i = 0 \qquad \forall i \tag{21}$$

$$x_i^{I(O)} = x_i + p_i^{I(O)}wr_i \qquad \forall i \tag{22}$$

$$y_i^{I(O)} = y_i + q_i^{I(O)}hr_i \qquad \forall i \tag{23}$$

2ª parte do Modelo 4 (UAFLP3)

$$m_i^{I(O)} \leq wr_i \qquad \forall i \tag{24}$$

$$n_i^{I(O)} \leq hr_i \qquad \forall i \tag{25}$$

$$x_i^{I(O)} = x_i + (1-p_i^{I(O)})m_i^{I(O)} + p_i^{I(O)}q_i^{I(O)}wr_i \qquad \forall i \tag{26}$$

$$y_i^{I(O)} = y_i + p_i^{I(O)}n_i^{I(O)} + (1-p_i^{I(O)})q_i^{I(O)}hr_i \qquad \forall i \tag{27}$$

A função objetivo dos modelos (Expressão 1) busca minimizar o TMHC que é igual a somatória do produto dos fluxos de material (parâmetro) pelas distâncias entre a saída de uma facilidade e a entrada de outra (variável). As restrições (2) e (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 2 ilustra as situações.

As equações (4) e (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 (6), (7) e (8) garantem que não ocorrerá sobreposição de peças, pois um bloco sempre estará, pelo menos, ou totalmente à esquerda, ou totalmente à direita, ou totalmente acima ou totalmente abaixo de algum outro bloco. Além disso, (7) e (8) também delimitam os blocos ao espaço total disponível para o layout, tanto em

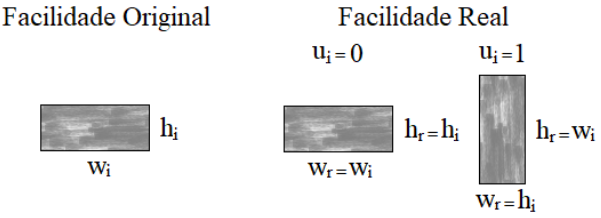

Figura 2: Orientação dos blocos.

largura quanto em altura. No artigo de Leno et al. [2018], a restrição (6) é apresentada como uma equação, o que gera resultados indesejados para o modelo, portanto neste trabalho ela foi escrita como inequação tal como consta no modelo de Hu et al. [2007].

Nos modelos propostos, o tipo de medida adotada foi a Métrica de Distância Retilínea. Porém, no caso de considerar distância Euclidiana, bastaria trocar a restrição (9) pela seguinte:

$$d_{ij} = \sqrt{\left[x_i^O - x_j^I\right]^2 + \left[y_i^O - y_j^I\right]^2} \qquad \forall i, j \tag{28}$$

Outra opção seria considerar a Métrica de Distância Perimetral, ou Distância de Contorno (conceito muito bem apresentado por Leno et al. [2018]), que procura representar melhor uma situação real na qual não é possível transportar materiais passando por dentro de departamentos, atravessando-os, mas, sim, somente por corredores ou áreas vazias. 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 departamentos) é calculada usando o algoritmo de Dijkstra criado para resolver problemas de caminho mais curto. A Figura 3 traz a representação de uma malha (geradora do grafo) para um layout com três departamentos e mostra o caminho mínimo entre dois pontos.

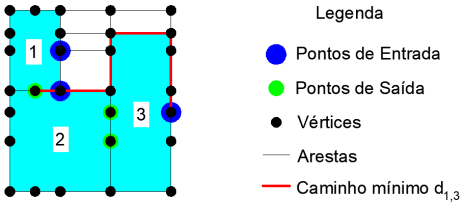

Figura 3: *Grid* com caminhos de fluxo viáveis para um problema com três facilidades e marcação da distância de contorno $d_{1,3}$ entre a saída do bloco 1 e a entrada do bloco 3.

Em (10), (11) e (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 (13), pois as variáveis m, n, p e q não são necessárias. As restrições (14) e (15) definem as coordenadas dos I/O *points* dos blocos com base em como esses foram rotacionados para a formação 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 poderiam ser substituídas pelas seguintes:

$$x_i^{I(O)} = x_i + \frac{wr_i}{2} \qquad \forall i \tag{29}$$

$$y_i^{I(O)} = y_i + \frac{hr_i}{2} \qquad \forall i \tag{30}$$

No Modelo 2, os I/O *points* devem ser alocados aos pontos médios das bordas dos blocos. As equações (16), (17), (18) e (19) garantem o posicionamento em um dos quatro locais possíveis. No Modelo 3, as equações (20), (21), (22) e (23) estabelecem que os I/O *points* estarão alocados em uma das quatro quinas dos blocos correspondentes. As restrições (18), (19), (22) e (23) foram modeladas de maneira diferente da apresentada por Leno et al. [2018]. Por fim, no Modelo 4, as restrições (24), (25), (26) e (27) garantem que os I/O *points* estarão posicionados em algum local nas bordas dos blocos. No modelo de Hu et al. [2007], as equações que definem $x_i^{I(O)}$ e $y_i^{I(O)}$ são diferentes e se assemelham às equações (14) e (15) consideradas neste trabalho para o UAFLPfixo. O caso UAFLP3 não é tratado no artigo de Leno et al. [2018].

3. Abordagem Proposta

Para resolver o UAFLP com I/O *points*, os modelos matemáticos foram transpostos para linguagem de programação adequada e resolvidos através de softwares especializados em métodos exatos. Toda a fase de implementação e testes de instâncias da literatura foi realizada em linguagem de programação matemática AMPL usando um Notebook com Processador Intel Core i7, Memória RAM de 8GB e Sistema operacional Windows 10. Para resolução, foram utilizados dois softwares matemáticos: o Solver Baron (version 19.7.13), especializado em resolver problemas de otimização não linear e o Solver CPLEX (version 12.10.0.0), de alto desempenho para programação linear, eficaz para o caso dos modelos linearizados desenvolvidos para este trabalho.

O solver Baron 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 solucionador não conseguiu chegar a um resultado comprovadamente ótimo (em um dos casos ele alcançou um resultado já conhecido da literatura, mas não conseguiu provar que era o ótimo e, assim, continuou tentando melhorar a solução indefinidamente). Para problemas maiores, os resultados se mostraram ruins mesmo após o programa rodar por muitas horas.

Por conta disso, foram priorizados os testes de instâncias a partir da implementação dos modelos linearizados (devido à limitação de espaço, esses foram suprimidos do artigo e podem ser solicitados aos autores), com auxílio do solver CPLEX. Foi usada uma técnica de linearização para problemas quadráticos binários nos termos não lineares (produtos de variáveis).

4. Resultados e Discussões

Desenvolver, ou mesmo aprimorar, 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. Para este trabalho, foram obtidos resultados para diversas instâncias encontradas na literatura. As instâncias foram selecionadas de forma a abranger tanto problemas mais simples quanto os de maior tamanho que são, consequentemente, mais difíceis de resolver computacionalmente.

A Tabela 2 apresenta as dez instâncias testadas, 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 neste artigo.

Como consta na tabela, 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).

Conforme dito na seção anterior, foi implementado um programa para solucionar os problemas de layout com resolução através de CPLEX que se utiliza de métodos exatos (será referenciado como Solver CPLEX). Foi estabelecido um limite de tempo para fins de padronização e esse foi definido após alguns testes de desempenho. No caso do CPLEX, para a instância maior o programa apresentava erro de estouro de memória quando rodava por muito tempo, por isso fixou-se $t = 2h$.

Referência	Tipo de dados	Nº de facilidades	Código
Das [1993]	I/O *points* fixos	4	Das93N4
		6	Das93N6
		8	Das93N8
		10	Das93N10
		12	Das93N12
Welgama e Gibson [1993]	I/O *points* fixos	6	Wel93N6
		12	Wel93N12
Deb e Bhattacharyya [2005]	sem I/O *points* prévios	12	Deb05N12
		18	Deb05N18
Dunker et al. [2003]	I/O *points* nos centroides	62	Dun03N62

Tabela 2: Identificação das instâncias retiradas da literatura.

A análise dos resultados foi divida em duas subseções, a primeira referente à situação de I/O *points* inicialmente fixos contemplada pelo Modelo 1 e a segunda referente à situação de I/O *points* variáveis tratada nos Modelos 2, 3 e 4. Apesar de alguns resultados comparados neste artigo datarem de longo tempo, ainda são os melhores encontrados na literatura para cada situação.

4.1. I/O *points* Fixos

Das dez instâncias apresentadas na Tabela 2, consta-se 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]. Os valores de função objetivo dos autores foram comparados com os encontrados pelo *solver* utilizado neste trabalho e, para isso, a medida de distância adotada foi a mesma: distância retilínea. A compilação de dados consta na Tabela 3 que segue:

Instância	Melhor resultado da literatura				Solver
	Referência	Método	Tempo (s)	Solução	CPLEX
Das93N4	Rajasekharan et al. [1998]	Algoritmo genético	66	**1.393,6***	**1.393,6***
Das93N6			273	2.612,7	**2.556***
Das93N8	Dunker et al. [2003]	Algoritmo coevolucionário	550	**8.778,3***	**8.778,3***
Das93N10			sem dados	15.694,5	**15.222,9**
Das93N12			sem dados	37.396,1	**36.622,8**
Wel93N6	Welgama e Gibson [1993]	Algoritmo construtivo	21	421,5	**398,5***
Wel93N12			190	5.903	**5.458,5**
Dun03N62	Dunker et al. [2003]	Algoritmo coevolucionário	5214	**3.939.362**	5.484.080

*Solução ótima

Tabela 3: Resultados para I/O *points* fixos.

Como é possível observar pela tabela, os resultados encontrados com a linearização dos modelos e uso do CPLEX foram os melhores para quase todas as instâncias, sendo a exceção a maior de todas com 62 facilidades.

Para as instâncias menores (com $N \leq 8$), o CPLEX foi capaz de retornar o resultado ótimo do problema antes do tempo limite. Na Figura 4 temos a representação dos resultados ótimos alcançados, assim como o tempo gasto na solução dos problemas.

4.2. 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.

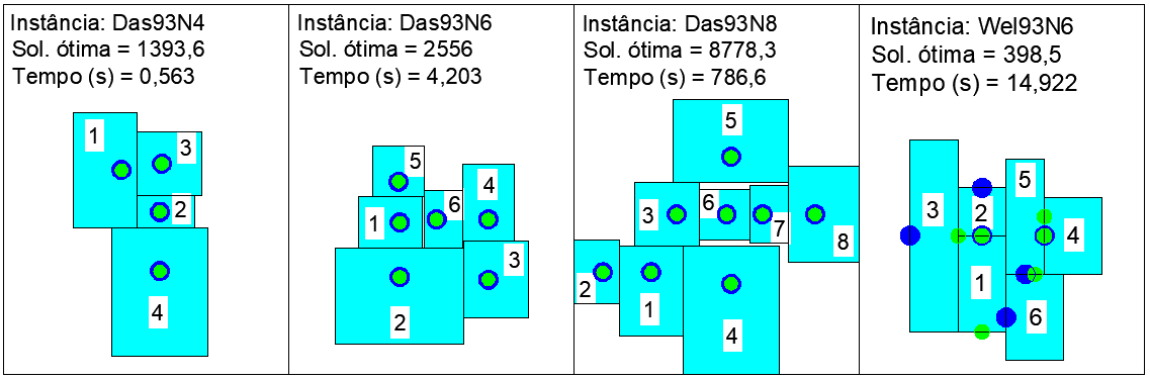

Figura 4: Layouts ótimos gerados para I/O *points* fixos.

Foram avaliados resultados da literatura de dois artigos referentes a cinco instâncias para efeito de comparação, que variam quanto às restrições aplicadas aos locais de implantação dos pontos de entrada e saída dos departamentos que definem os modelos UAFLP1, UAFLP2 e UAFLP3 (conforme Figura 1).

A análise dos resultados pode ser abaixo. A Tabela 4 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; já os autores Leno et al. [2018] desenvolveram o método ESHGA, um algoritmo genético híbrido baseado em estratégia elitista que usa recozimento simulado como busca local (resultados estão na Tabela 5).

Modelo	Instância	Melhor resultado da literatura**			Solver
		Medida	Tempo (s)	Solução	CPLEX
UAFLP1	Deb05N12	distância retilínea	4132	21.142	**11.504**
	Deb05N18		7338	62.902	**37.382**
UAFLP3	Deb05N12		3952	19.892	**6.924**
	Deb05N18		6433	61.756	**27.092**

**Referência: Deb e Bhattacharyya [2005]; Método: GA-SA

Tabela 4: Resultados comparativos para I/O *points* variáveis - Parte I.

Como pode ser notado nas tabelas, os resultados conquistados com o CPLEX foram todos melhores que os encontrados na literatura, ressaltando a importância destes.

Para as instâncias menores (com $N = 6$), o CPLEX foi capaz de retornar o resultado ótimo do problema antes do tempo limite. Na Figura 5 temos a representação dos resultados ótimos alcançados, assim como o tempo gasto na solução dos problemas.

Para efeito de comparação visual, a seguir, nas Figuras 6 e 7, estão representados os layouts finais gerados por Leno et al. [2018] com o método ESHGA e pela metodologia do presente trabalho, para as instâncias Deb05N12 e Dun03N62, Modelo UAFLP2.

5. Conclusões

O problema de layout de facilidades de áreas desiguais com locais de entrada e saída já foi resolvido na literatura com o uso dos mais variados métodos. Na maioria dos artigos consultados, constatou-se que os autores se utilizam de heurísticas e metaheurísticas para resolver o problema.

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

Modelo	Instância	Melhor resultado da literatura**			Solver
		Medida	Tempo (s)	Solução	CPLEX
UAFLP2	Deb05N12	distância retilínea	6000	8.676	**6.804**
	Deb05N18		20400	44.012	**27.780**
UAFLP1	Wel93N6	distância de contorno	936	325	**237***
	Wel93N12		1140	5.750	**5.288**
	Deb05N12		6000	37.894	**12.404**
	Deb05N18		20400	72.451	**40.754**
UAFLP2	Wel93N6		936	266	**115***
	Wel93N12		1140	5.454	**2.889**
	Deb05N12		6000	16.404	**7.028**
	Deb05N18		20400	58.324	**28.540**
	Dun03N62		603500	4.921.320	**3.972.940**

*Solução ótima
**Referência: Leno et al. [2018]; Método: ESHGA

Tabela 5: Resultados comparativos para I/O *points* variáveis - Parte II.

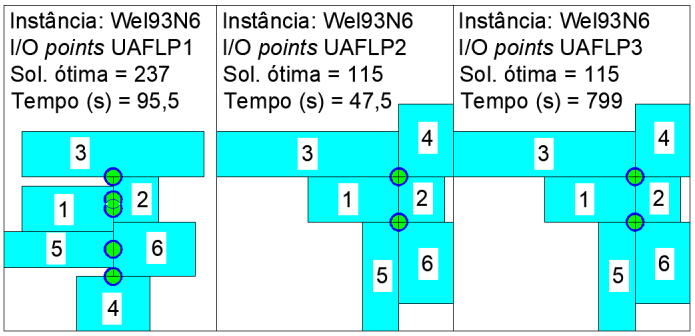

Figura 5: Layouts ótimos gerados para I/O *points* variáveis.

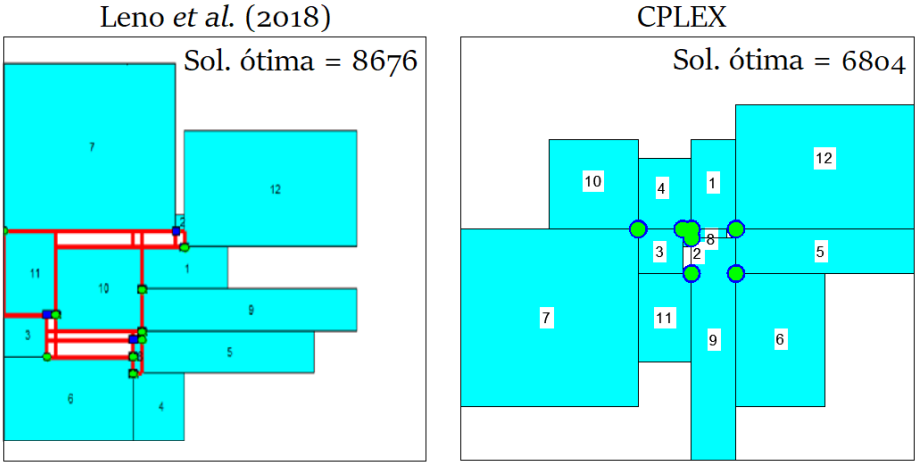

Figura 6: Layouts gerados para o Modelo UAFLP2 - Instância: Deb05N12.

métodos exatos. Porém, com o avanço dos solucionadores no decorrer do tempo, os resultados ficaram mais competitivos e, no caso deste trabalho, foram promissores. Ainda assim, para instâncias

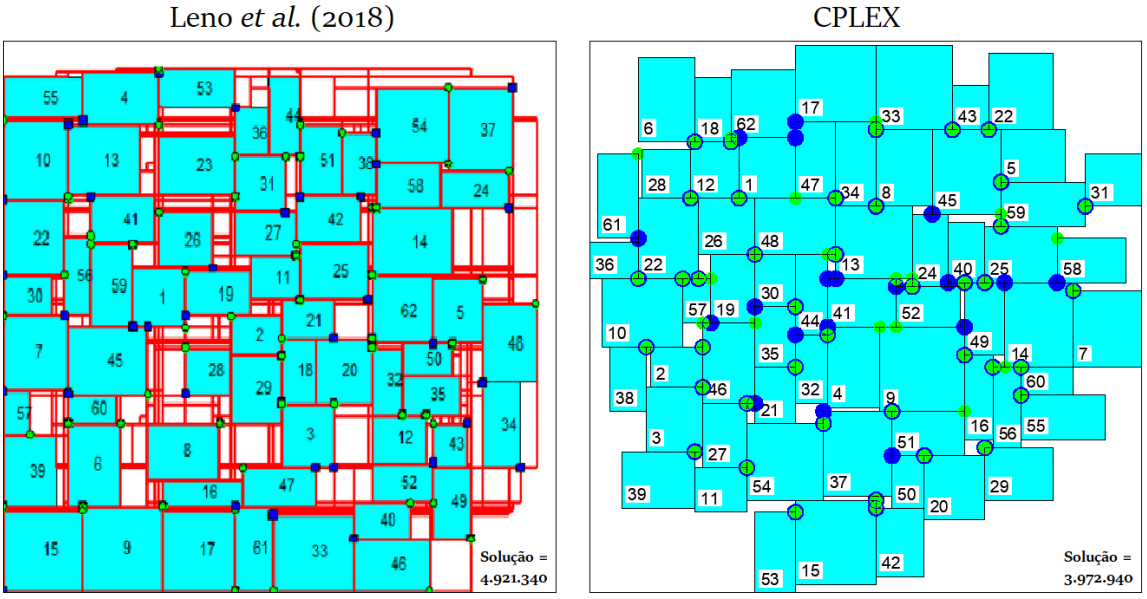

Figura 7: Layouts gerados para o Modelo UAFLP2 - Instância: Dun03N62.

maiores o desafio permanece, como constatado no caso de $N = 62$ que teve um resultado pior com o CPLEX na comparação com uma metaheurística desenvolvida muito antes por Dunker et al. [2003].

O presente artigo buscou contribuir para o tema de layouts explorando situações específicas, tanto no que se refere a modelagem matemática, quanto ao uso de solucionadores exatos para resolver o problema. Em especial, o uso de CPLEX aplicado aos modelos linearizados apresentou relevantes resultados, incentivando esse tipo de abordagem para problemas dessa natureza.

O campo de estudo de otimização de layouts é extenso e merece ser estudado em sua totalidade e complexidade. Pensando nisso, futuramente a pretensão é de desenvolver novos modelos para outros casos particulares, como o de blocos de dimensões variáveis e o de demanda dinâmica.

Agradecimentos

Este estudo foi realizado com o apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

Referências

Armour, G. C. e Buffa, E. S. (1963). A heuristic algorithm and simulation approach to relative location of facilities. *Management Science*, 9(2):294–309.

Castillo, I. e Westerlund, T. (2005). An ε-accurate model for optimal unequal-area block layout design. *Computers & Operations Research*, 32(3):429–447.

Das, S. (1993). A facility layout method for flexible manufacturing systems. *The International Journal of Production Research*, 31(2):279–297.

Deb, S. e Bhattacharyya, B. (2005). Solution of facility layout problems with pickup/drop-off locations using random search techniques. *International journal of production research*, 43(22): 4787–4812.

Dunker, T., Radons, G., e Westkämper, E. (2003). A coevolutionary algorithm for a facility layout problem. *International Journal of Production Research*, 41(15):3479–3500.

Emami, S. e Nookabadi, A. S. (2013). Managing a new multi-objective model for the dynamic facility layout problem. *International Journal of Advanced Manufacturing Technology*, 68(9): 2215–2228.

García-Hernández, L., Salas-Morera, L., García-Hernández, J. A., Salcedo-Sanz, S., e de Oliveira, J. V. (2019). Applying the coral reefs optimization algorithm for solving unequal area facility layout problems. *Expert Systems With Applications*, 138:112819.

Hosseini-Nasab, H., Fereidouni, S., Ghomi, S. M. T. F., e Fakhrzad, M. B. (2018). Classification of facility layout problems: a review study. *International Journal of Advanced Manufacturing Technology*, 94(1-4):957–977.

Hu, G., Chen, Y., Zhou, Z., e Fang, H. (2007). A genetic algorithm for the inter-cell layout and material handling system design. *The International Journal of Advanced Manufacturing Technology*, 34(11-12):1153–1163.

Kim, J.-G. e Kim, Y.-D. (2000). Layout planning for facilities with fixed shapes and input and output points. *International Journal of Production Research*, 38(18):4635–4653.

Konak, A., Kulturel-Konak, S., Norman, B. A., e Smith, A. E. (2006). A new mixed integer programming formulation for facility layout design using flexible bays. *Operations Research Letters*, 34(6):660–672.

Kulturel-Konak, S. (2012). A linear programming embedded probabilistic tabu search for the unequal-area facility layout problem with flexible bays. *European Journal of Operational Research*, 223(3):614–625.

Kulturel-Konak, S. (2019). The zone-based dynamic facility layout problem. *INFOR: Information Systems and Operational Research*, 57(1):1–31.

Leno, I. J., Sankar, S. S., e Ponnambalam, S. (2018). Mip model and elitist strategy hybrid ga–sa algorithm for layout design. *Journal of Intelligent Manufacturing*, 29(2):369–387.

Liu, J., Liu, J., Yan, X., e Peng, B. (2020). A heuristic algorithm combining pareto optimization and niche technology for multi-objective unequal area facility layout problem. *Engineering Applications of Artificial Intelligence*, 89:103453.

O'brien, C. e Abdel Barr, S. (1980). An interactive approach to computer aided facility layout. *International Journal of Production Research*, 18(2):201–211.

Rajasekharan, M., Peters, B. A., e Yang, T. (1998). A genetic algorithm for facility layout design in flexible manufacturing systems. *International journal of Production research*, 36(1):95–110.

Tompkins, J. A., White, J. A., Bozer, Y. A., e Tanchoco, J. M. A. (2010). *Facilities planning*. John Wiley & Sons.

Van Camp, D. J., Carter, M. W., e Vannelli, A. (1992). A nonlinear optimization approach for solving facility layout problems. *European Journal of Operational Research*, 57(2):174–189.

Welgama, P. e Gibson, P. (1993). A construction algorithm for the machine layout problem with fixed pick-up and drop-off points. *The International Journal of Production Research*, 31(11): 2575–2589.