Pesquisa Operacional (2022) 42: e261044 p.1-25
doi: 10.1590/0101-7438.2022.042.00261044
© 2022 Brazilian Operations Research Society
Printed version ISSN 0101-7438 / Online version ISSN 1678-5142
www.scielo.br/pope
ARTICLES

A MATHEMATICAL OPTIMIZATION APPROACH BASED ON LINEARIZED MIP MODELS FOR SOLVING FACILITY LAYOUT PROBLEMS

Evelyn Michelle Henrique Braga[1,2*] and Luiz Leduino de Salles Neto[3]

Received February 16, 2022 / Accepted April 7, 2022

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 general, the objective function of the facility layout problem (FLP) is to reduce the total material handling cost. Although over six decades have been passed since the first work on FLP modeling was published, research on many aspects of this problem is still in an early stage and needs to be further explored, which motivated this study. In this paper, the unequal area of rectangular blocks with fixed dimensions and input/output points are considered for FLPs. Four new mixed-integer programming (MIP) models based on previous research formulations are developed. Then, a mathematical optimization approach based on the linearization of the models is applied. An algorithm that solves the linearized MIP model by CPLEX setting a time limit for the solution obtained excellent results for different test problems when compared to those reported in the literature.

Keywords: facilities planning and design, unequal area facility layout problem, mixed integer programming.

1 INTRODUCTION

In an increasingly competitive global market, guaranteeing the optimization of results inside a company is essential. Maximizing productivity is one of the main objectives of the industrial production sector. To reach this potential, the most adequate physical arrangement for each situation should be analyzed to speed up the production processes. In the industrial context, the execution of a product goes through several stages that can be characterized by facilities, such as

*Corresponding author

[1]ICT, Universidade Federal de São Paulo (UNIFESP), São José dos Campos, SP, Brazil – E-mail: evelyn.braga@unifesp.br – http://orcid.org/0000-0003-1318-4593

[2]Instituto Tecnológico de Aeronáutica (ITA), São José dos Campos/SP, Brazil

[3]ICT, Universidade Federal de São Paulo (UNIFESP), São José dos Campos, SP, Brazil – E-mail: luiz.leduino@unifesp.br – http://orcid.org/0000-0001-8938-5370

installations, machines, equipment, sectors, or departments. In times past, facilities planning was primarily considered to be a science, but today, it is a strategy (Tompkins et al., 2010).

An arrangement or layout can be defined as the relative positioning of the various facilities within an enclosure. In manufacturing perspective, the facility layout design instructs how to arrange the physical layout of manufacturing facility systems to provide the best support for production (Leno et al., 2018). Organizing the layout can improve the use of available space, reduce the material handling distances, services, and people, increase the production, and lower indirect costs and manufacturing time, since the disposition of the facilities directly influences the time spent on activities and processes. Hence the Facility Layout Problem (FLP) is one of the main issues in the Management of Production and Industrial Engineering literature, attracting the attention of many researchers in the field of statical and dynamic layouts (Hosseini-Nasab et al., 2018).

Production systems and services must be operated with optimized planning and well-done operational practices to reach their potential. Moreover, defining a good installation layout ensures that the whole system acts in a more efficient way (Hosseini-Nasab et al., 2018). This physical arrangement question is very important and can impact the viability of the long-term manufacturing process; thus, it should be contemplated in the initial phase of the project. A poorly designed layout will result in reduced productivity, more work in progress, more manufacturing time, cluttered material handling, and so on (Pillai et al., 2011).

Inside the operations research, layout optimization, considering the more diverse situations of process and demand, is sought to minimize the production time and use the existing space in the most efficient possible way. The general problem designs the positioning of the factory installations, intending to determine the more efficient disposition in agreement with some criteria or objectives, under certain constraints (García-Hernández et al., 2013).

Conventionally, the objective function of the layout of the installations problem consists of the reduction of the Total Material Handling Cost (TMHC), an expense without aggregate value for the materials flows and the distances between facilities. It is estimated that efficient planning installations can reduce the TMHC by up to 30% (Jung et al., 2017), thus decreasing the total operational cost of the project. In contrast, research shows that more than 35% of the efficiency of the system could be lost by applying incorrect localization designs (Izadinia & Eshghi, 2016).

Although the facility layout problem has been widely addressed in the literature, several peculiarities deserve attention, such as the facilities in the form of rectangular blocks of unequal areas and input and output locations of processes in the blocks. The study of these issues is justified, as they exacerbate the complexity of the challenge within the reality found in the production sector.

This work aims to study different models for optimization of layouts found in the literature and, from them, develop new mathematical models for specific situations, as well as implement the models in recognized programming language. The paper is organized into five sections. Section 2 explores concepts related to FLPs. Section 3 presents the mathematical models developed for the problem. Section 4 describes the resolution approach used. Section 5 shows the results obtained and a comparison with those found in the literature. Finally, Section 6 brings the conclusions and some directions for future research.

2 LITERATURE REVIEW

2.1 General Facility Layout Problem

When the flow of materials between installations is a parameter that does not change with time, the problem is known as a static facility layout problem and, in a simple form, can be formulated as a Quadratic Assignment Problem (QAP) (Emami & Nookabadi, 2013). The classical optimization model, introduced by Koopmans & Beckmann (1957), is defined as follows. The notation used in the formulation are shown in Table 1.

Table 1 – Notation used in the general FLP model.

Indices and input data	
N	total number of facilities/locations
i e j	index for facilities = (1,...,N)
r e s	index for locations = (1,...,N)
d_{rs}	distance between locations r and s
f_{ij}	material flow between facilities i e j
Decision variables	
x_{ir}	1 if facility i is located in r, 0 cc.

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

subject to

$$\sum_{r=1}^{N} x_{ir} = 1 \qquad \forall i \tag{2}$$

$$\sum_{i=1}^{N} x_{ir} = 1 \qquad \forall r \tag{3}$$

$$x_{ir} \in \{0,1\} \qquad \forall i, r \tag{4}$$

In the QAP formulation for layouts, every facility has the same area, and the floor space is divided into N locations with the same size, where each installation is assigned to exactly one location and vice versa (Konak et al., 2006). The objective is to minimize the summation of Material Handling Costs (MHCs) between each pair of installations, the MHC being a product of the flow of materials between two facilities by the distance between them. The constraints ensure that each facility will occupy a single location (2) and that each location will be occupied by a single facility (3).

In this basic model, the distances are measured between the centroids of the blocks and can be Euclidean (diagonal) or rectilinear (sum of horizontal and vertical), according to Figure 1. The rectilinear distances are also called rectangular or Manhattan distances (Zheng, 2014).

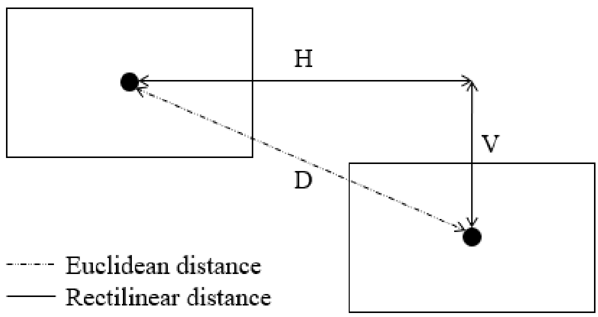

Figure 1 – Representation of Euclidean ($d = D$) and rectilinear ($d = H + V$) distances.

Although the diagonal represents the actual distance between the centroids, the rectilinear distance is more used because it better represents a real situation of displacement that would occur through the aisles between facilities.

2.2 Unequal Area Facility Layout Problem

In a real situation, it is unlikely that the several facilities of a company will have all the same area. One particularly interesting FLP, due to its direct application in real cases, is known as the Unequal Area Facility Layout Problem (UAFLP) (García-Hernández et al., 2019).

When departments have different areas, the problem can no longer be solved by assigning departments to n distinct centroids localizations and distances become variables. The additional complications of the requisites of unequal areas, with continuous department positions that can be anywhere in a rectangular area and varying areas (width and height), make the FLP extremely difficult to solve (Castillo & Westerlund, 2005).

The UAFLP is studied in many aspects. To begin with, the form of the facilities can be described as regular (e.g., rectangular blocks) or irregular (usually polygons). The dimensions of a facility can be represented by fixed measurements of height and width (rigid block) or by a fixed area (with an aspect ratio). According to Hosseini-Nasab et al. (2018), for the configuration, FLPs are classified into seven categories distinguished by the shape of their material handling path, namely: single-row, multi-row, double-row, parallel-row, loop, open-field, and multifloor layouts. Still according to them, most of the authors in the field of layouts consider in their studies facilities with regular form, fixed dimensions, and open-field system. This leads to the belief that such circumstances have good relevance in the practical context.

The unequal area block placement problem was first reported by Armour & Buffa (1963) more than 50 years ago. Over time, several authors have studied static UAFLP, e.g.: Bazaraa (1975) divided the total area into small units formulating the problem as a quadratic set covering the problem and solving it with a Branch and Bound (B&B) approach; Tate & Smith (1995) presented a restrictive version of the slicing tree (ST) formulation known as flexible bay structure (FBS) and solved the problem by genetic research; and Konak et al. (2006) created a new formulation for the Mixed-Integer Programming (MIP) model for FBS.

In the last decade, several authors have proposed resolution methods to address UAFLPs: Kulturel-Konak (2012) used linear programming and tabu search (TS) for the FBS situation providing significant reduction in MHC although the relaxed-FBS representation was more restrictive compared to the general representation used in many exact formulations; Chang & Ku (2013) represented a ST and developed a heuristic solution that found optimal or near-optimal solutions for most problems as long as they had fewer than 20 departments; Gonçalves & Resende (2015) solved the problem by using a biased random-key genetic algorithm (BRKGA) generating high-quality solutions in relatively small computing times with the unconstrained version of the approach, but not so good with the constrained one; Palomo-Romero et al. (2017) made an island model genetic algorithm as an alternative approach, which helped to avoid premature convergence and excessive execution time; and García-Hernández et al. (2019) used a coral reefs optimization (CRO) algorithm that showed excellent performance when considering exclusively FBS representation, but not so good when considering both ST structure and FBS.

Furthermore, these authors also found excellent solutions with their approaches: Liu & Liu (2019) applied the multi-objective ant colony optimization (ACO) algorithm; Moradi & Shadrokh (2019) considered bi-objective function and solved the problem with SA; García-Hernández et al. (2020) created a novel island model based on CRO; and Liu et al. (2020) developed a heuristic that combines Pareto optimization and niche technology for a multi-objective problem.

2.3 Facility Layout Problem with Input/Output Points

The constraints considered for FLPs are of the most diverse, but among these, those that define input and output points of the blocks are of great importance. When it comes to facilities, the flow of materials, products, and/or people is expected to arrive by a certain location and leave by a certain location (which may or may not be the same location). In the traditional FLPs models, the input and output locations of the facilities are not considered, but are determined after the obtention of a block layout (Kim & Kim, 2000).

In literature, these facilities are called by different names such as input/output points (I/O); pick-up/drop-off or delivery stations (P/D); and load/unload locations. In articles written in recent years, priority has been given to using the term I/O points, thus, they will be referred by this.

The I/O points can be fixed (predetermined) or variable with constraints, depending on the type of facility being addressed in the problem. When the facilities, for example, are machines or equipment that have defined input and output locations, i.e., that cannot be altered, it is interesting to consider fixed points. On the other hand, in a situation with departments, whose room doors could be positioned with more flexibility in the project, it is possible to consider variable points in the model (in this case, with the constraint of being located on the edges of the blocks).

O'brien & Abdel Barr (1980) were the first to consider I/O points in the input data required by their procedure. Since then, many authors have proposed method of resolution for the I/O points situation in the FLP: Das (1993) suggested a four-step heuristic that combines variable partitioning and integer programming methods for flexible manufacturing system (FMS) with fixed I/O points; Welgama & Gibson (1993) used a construction algorithm considering fixed I/O points; Rajasekharan et al. (1998) found good results for the problem through GA considering the formulation of Das (1993); Kim & Kim (2000) treated blocks of fixed dimensions and developed a MIP model that they solved with a two-phase (construction and improvement) heuristic algorithm; Dunker et al. (2003) proposed a coevolutionary algorithm for a MIP model; Rajagopalan et al. (2004) approached Lagrangian relaxation for AGV flowpath projects; Deb & Bhattacharyya (2005) used random search techniques and solutions with GA, SA and hybrid algorithm; and Hu et al. (2007) developed a GA for the situation of free I/O points at the edges.

Other examples of methodologies aimed at I/O points can be cited: Xiao et al. (2013) suggested a two-phase heuristic with interconnected zone and SA besides a reduced MIP to further improve the solution with better results than those provided by existing algorithms; Leno et al. (2018) created a MIP model and an elitist strategy hybrid GA-SA algorithm (ESHGA) that generated good layouts compared to reported result; Park & Seo (2019) suggested a two-phase constructive heuristic that produced good results within a much shorter time than previous research.

3 MATHEMATICAL MODELS

When some or all variables of an optimization problem belong to the set of integers, it is represented using an Integer Programming model. When all the variables are integers, the model is denominated Pure Integer Programming; otherwise, it is denominated Mixed-Integer Programming (Hillier & Lieberman, 2013). Though producing solutions for integer programs may seem easy, the solution is an NP-hard problem.

One of the first Mixed-Integer Programming formulations to solve problems of the continuous layout was introduced by Montreuil (1991). Afterward, several authors formulated Facility Layout Problems in continuous surfaces as Mixed-Integer Programming models.

For this work, four Mixed-Integer Programming models were formulated for the Unequal Area Facility Layout Problems with Input and Output locations with the following characteristics:

- Rectilinear Distance Metric (as seen in Figure 1);
- Regular facility shape (rectangular), with fixed dimensions of height and width and subject to an open-field system; and
- Fixed and variable input and output locations.

The models proposed by Hu et al. (2007) and Leno et al. (2018) were used as a basis for the MIP models developed for this research.

Although the four models developed in this work are similar, each represents a particular situation. The difference between them is in the location of the input and output points, as explained below:

1. Model 1 (UAFLPfixed) – Fixed I/O locations (at any point on the block, be it in the centroid, at some other point on the inside, or in the edges);
2. Model 2 (UAFLP1) – Variable I/O locations restricted to the middle points of the edges;
3. Model 3 (UAFLP2) - Variable I/O locations restricted to the corner points;
4. Model 4 (UAFLP3) Variable I/O locations restricted to any point on the edges.

Figure 2 illustrates the three possibilities for variable I/O points.

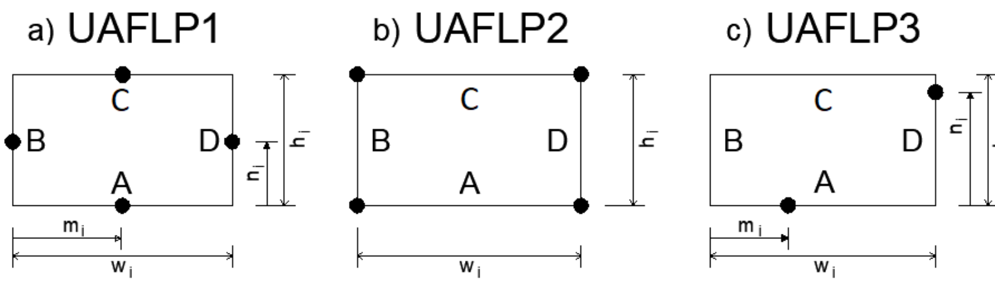

Figure 2 – Departments with candidate locations for I/O points: a) four possibilities located on middle of the edges of the block; b) four options located on the corners of the block; and c) can assume infinite positions around on the boundaries of the block.

As the models have an identical objective function and several constraints, they were divided into two parts, one common and one unique and presented in five topics. The first topic provides exactly the objective function and the equations that repeat for all the models and can be considered the first part of any of the four models individually. The other subsequent topics provide the second part of each model separately because, at this point, the equations are different and characterize specific conditions of the location of the input and output stations. The notations, valid for all models, are shown in the Table 2.

Table 2 – Notation used in the Models 1, 2, 3 and 4.

Indices, input data and markers	
N	total number of facilities
i e j	index for facilities = (1,...,N)
f_{ij}	material flow between facilities i and j
W	width of the floor space
H	height of the floor space
w	original width of facility i
h	original height of facility i
Ix_i, Iy_i	original coordinates x and y of the input of facility i
Ox_i, Oy_i	original coordinates x and y of the output of facility i
A, B, C, D	sides of facilities (bottom, left, top and right)
Decision variables	
d_{ij}	rectilinear distance between the output of facility i and the input of facility j
wr_i	real width of facility i (considering the rotation)
hr_i	real height of facility i (considering the rotation)
x_i, y_i	coordinates x and y of lower left corner of facility i
xs_i, ys_i	coordinates x and y of upper right corner of facility i
x_i^I, y_i^I	coordinates x and y of the input of facility i
x_i^O, y_i^O	coordinates x and y of the output of facility i
m_i^I, n_i^I	perimeter position in directions x and y of the input of facility i
m_i^O, n_i^O	perimeter position in directions x and y of the output of facility i
l_{ij}	(relative position) 1 if facility i is placed totally to the left of j, 0 cc.
b_{ij}	(relative position) 1 if facility i is placed totally below j, 0 cc.
u_i, v_i	(0,0) if facility i is in its original orientation; (1,0) if facility i is rotated 90° clockwise; (0,1) if facility i is rotated 180° clockwise; (1,1) if facility i is rotated 270° clockwise
p_i^I, q_i^I	defines on which side the input of facility i is located: (0,0) if on side A; (1,0) if on side B; (0,1) if on side C; (1,1) if on side D
p_i^O, q_i^O	defines on which side the output of facility i is located: (0,0) if on side A; (1,0) if on side B; (0,1) if on side C; (1,1) if on side D

First part of Models 1, 2, 3 and 4

$$\min \sum_{i=1}^{N} \sum_{j=1}^{N} f_{ij} d_{ij} \tag{5}$$

subject to

$$wr_i = (1-u_i)w_i + u_i h_i \qquad \forall i \tag{6}$$

$$hr_i = (1-u_i)h_i + u_i w_i \qquad \forall i \tag{7}$$

$$xs_i = x_i + wr_i \qquad \forall i \tag{8}$$

$$ys_i = y_i + hr_i \qquad \forall i \tag{9}$$

$$l_{ij} + l_{ji} + b_{ij} + b_{ji} \geq 1 \qquad \forall i,j; i<j \tag{10}$$

$$xs_i \leq l_{ij}x_j + W(1-l_{ij}) \qquad \forall i,j \tag{11}$$

$$ys_i \leq b_{ij}y_j + H(1-b_{ij}) \qquad \forall i,j \tag{12}$$

$$d_{ij} = \left|x_i^O - x_j^I\right| + \left|y_i^O - y_j^I\right| \qquad \forall i,j \tag{13}$$

$$d_{ij}, wr_i, hr_i, x_i, y_i, xs_i, ys_i, x_i^{I(O)}, y_i^{I(O)}, m_i^{I(O)}, n_i^{I(O)} \geq 0 \qquad \forall i \tag{14}$$

$$u_i, v_i, p_i^{I(O)}, q_i^{I(O)} \in \{0,1\} \qquad \forall i \tag{15}$$

$$l_{ij}, b_{ij} \in \{0,1\} \qquad \forall i,j \tag{16}$$

Second part of Model 1 (UAFLPfixed)

$$m_i^{I(O)} = n_i^{I(O)} = p_i^{I(O)} = q_i^{I(O)} = 0 \qquad \forall i \tag{17}$$

$$\begin{aligned} x_i^{I(O)} = x_i &+ (1-u_i)(1-v_i)I(O)x_i + u_i(1-v_i)I(O)y_i \\ &+ (1-u_i)v_i(w_i - I(O)x_i) + u_i v_i(h_i - I(O)y_i) \qquad \forall i \end{aligned} \tag{18}$$

$$\begin{aligned} y_i^{I(O)} = y_i &+ (1-u_i)(1-v_i)I(O)y_i + u_i(1-v_i)(w_i - I(O)x_i) \\ &+ (1-u_i)v_i(h_i - I(O)y_i) + u_i v_i I(O)x_i \qquad \forall i \end{aligned} \tag{19}$$

Second part of Model 2 (UAFLP1)

$$m_i^{I(O)} = \frac{wr_i}{2} \qquad \forall i \tag{20}$$

$$n_i^{I(O)} = \frac{hr_i}{2} \qquad \forall i \tag{21}$$

$$x_i^{I(O)} = x_i + (1-p_i^{I(O)})m_i^{I(O)} + p_i^{I(O)}q_i^{I(O)}wr_i \qquad \forall i \tag{22}$$

$$y_i^{I(O)} = y_i + p_i^{I(O)}n_i^{I(O)} + (1-p_i^{I(O)})q_i^{I(O)}hr_i \qquad \forall i \tag{23}$$

Second part of Model 3 (UAFLP2)

$$m_i^{I(O)} = 0 \qquad \forall i \tag{24}$$

$$n_i^{I(O)} = 0 \qquad \forall i \tag{25}$$

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

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

Second part of Model 4 (UAFLP3)

$$m_i^{I(O)} \leq wr_i \qquad \forall i \tag{28}$$

$$n_i^{I(O)} \leq hr_i \qquad \forall i \tag{29}$$

$$x_i^{I(O)} = x_i + (1 - p_i^{I(O)})m_i^{I(O)} + p_i^{I(O)}q_i^{I(O)}wr_i \qquad \forall i \tag{30}$$

$$y_i^{I(O)} = y_i + p_i^{I(O)}n_i^{I(O)} + (1 - p_i^{I(O)})q_i^{I(O)}hr_i \qquad \forall i \tag{31}$$

The objective function of the models (expression 5) seeks to minimize the TMHC which is equal to the sum of the material flows by the distances between the output of one facility and the input of another. The constraints (6) and (7) define the configuration of the facilities. If $u_i = 0$, then the facility continues in its original configuration; if $u_i = 1$, then the block is rotated 90°, so the real width (i.e., the width of the block in the layout defined by the model) becomes the original height and the real height becomes the original width. Figure 3 illustrates both situations.

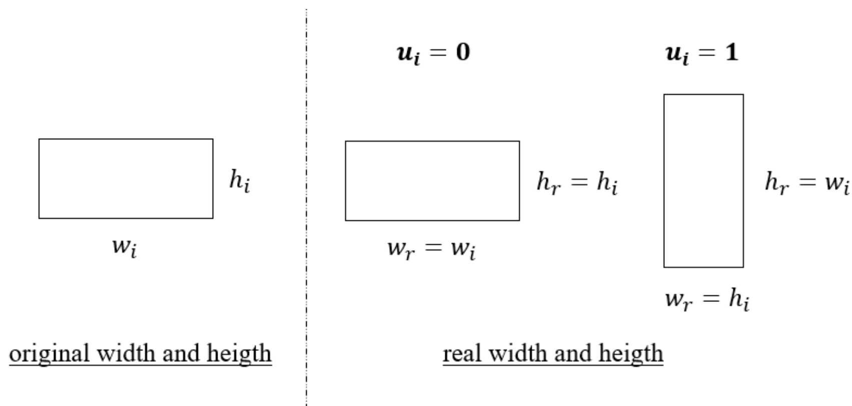

Figure 3 – Block orientations.

The equalities (8) and (9) delimit the blocks using the coordinates x_i, y_i of the lower left corner and xs_i, ys_i of the upper right corner. The inequalities (10), (11), and (12) guarantee that there will be no overlapping of parts because a block will always be, at least, or totally to the left, or totally to the right, or totally above, or totally below some other block. Furthermore, (11) and (12) delimit the blocks to the total space available for the layout, both in width and height. In Leno et al. (2018), constraint (10) is presented as an equation, which generates undesired results for the model, so in this work it was written as an inequality as in the model formulation of Hu et al. (2007).

The measure adopted for the proposed models was the Rectilinear Distance Metric (RDM). Although, in the case of considering Euclidean distance, the constraint (13) could be exchanged by the following one:

$$d_{ij} = \sqrt{\left[x_i^O - x_j^I\right]^2 + \left[y_i^O - y_j^I\right]^2} \qquad \forall i,j \tag{32}$$

Another option would be to consider Perimeter Distance Metric (PDM), or Contour Distance (a concept very well presented by Leno et al. (2018) and reproduced in the sequence), which seeks to better represent a real situation in which it is not possible to transport materials through departments, but only through aisles or empty areas. To calculate this distance, a graph that represents the viable flow paths must be constructed. To do this, first, a grid is constructed in which the horizontal and vertical lines cross all the points of the corners as well as input and output points of the blocks. The intersection of each horizontal and vertical line defines the grid points. The points that appear inside the departments are removed from the original grid point set and the rest forms the V set of vertices or nodes for the graph. The grid lines that touched removed points are also extinguished; the others form the E set of edges for the graph (connecting all adjacent points). To each edge, a weight corresponding to the length of the edge in question (which is the actual distance) is assigned. Given that graph $G = (V,E)$, the shortest distance d_{ij} from the output station of the i facility to the input station of the j facility (and between all pairs of departments) is calculated using the Dijkstra algorithm created to solve shortest-paths problems. The algorithm developed by Dijkstra (1959) is widely known in the field of computer science and is well explained in Cormen et al. (2009), which served as the basis for the programming in this part of our work. Figure 4 provides the representation of a grid (graph generator) for a layout with three departments and shows the shortest path between two points.

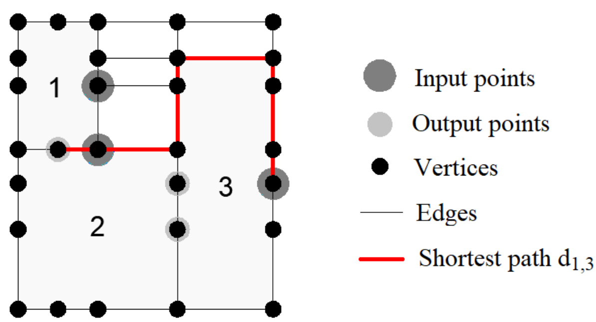

Figure 4 – Grid with feasible flow paths for a problem with three facilities and contour distance $d_{1,3}$ between the output of block 1 and the input of block 3 highlighted.

In (14), (15), and (16), the types of variables are defined, whether positive, continuous, or binary.

In Model 1, the I/O points are fixed, which justifies equation (17), because the variables m, n, p, and q are not necessary. Constraints (18) and (19) define the coordinates of the I/O points of the blocks based on how these were rotated to form the final layout (90°, 180°, 270° or not rotated). If the I/O points of the problem are fixed in the centroids of the blocks, these constraints could be replaced by the following:

$$x_i^{I(O)} = x_i + \frac{wr_i}{2} \qquad \forall i \tag{33}$$

$$y_i^{I(O)} = y_i + \frac{hr_i}{2} \qquad \forall i \tag{34}$$

In Model 2, the I/O points must be allocated to the midpoints of the block edges. Equations (20), (21), (22), and (23) guarantee the positioning in one of the four possible locations. In Model 3, equations (24), (25), (26), and (27) establish that the I/O points will be allocated to one of the four corners of the corresponding blocks. Constraints (22), (23), (26) and (27) were modeled differently from the one presented in Leno et al. (2018).

Finally, in Model 4, the constraints (28), (29), (30), and (31) ensure that the I/O points will be positioned somewhere on the boundaries of the blocks. It is interesting to note that Model 4 has two equations similar to Model 2. The constraints that define $m_i^{I(O)}$ and $n_i^{I(O)}$ are what differentiates the models, because while Model 2 has equalities that force these variables to be equal to half of the real dimensions, Model 4 has inequalities that provide freedom for the value of $m_i^{I(O)}$ and $n_i^{I(O)}$ as long as they are smaller than the real dimensions. In the model presented in Hu et al. (2007), the equations that define $x_i^{I(O)}$ and $y_i^{I(O)}$ are different from (30) and (31) and similar to equations (18) and (19) considered in this work for the fixed UAFLP. The UAFLP3 case is not addressed in Leno et al. (2018).

4 PROPOSED APPROACH

In order to solve the Unequal Area Facility Layout Problem with Input/Output Points, we use a mathematical optimization approach based on the linearization of the models.

Linearization is a technique used to obtain a linear model, from the equivalent nonlinear model, to provide equal or approximate solutions to those of the original problem. There are programs specifically developed to solve linear (at most quadratic) problems using exact method algorithms such as simplex, interior-point barrier and branch-and-bound. As, depending on the case, they may stand out from programs for nonlinear problems, one of the intentions of linearization is precisely to adapt the modeling to the application of these solvers in the search for better and/or faster results. Considering this, the Mixed-Integer Programming models previously presented were linearized and thus transformed into Mixed-Integer Linear Optimization (MILO) models.

As a reference, the linearization technique of Glover & Woolsey (1974) presented by Mauri & Lorena (2009) in the application for Binary Quadratic Problems was consulted. Adapting the technique to the Facility Layout models, the quadratic terms $l_{ij}x_i$, $b_{ij}y_i$, u_iv_i, $p_i^{I(O)}m_i^{I(O)}$, $p_i^{I(O)}n_i^{I(O)}$, $p_i^{I(O)}wr_i$ and $q_i^{I(O)}hr_i$, and cubic terms $p_i^{I(O)}q_i^{I(O)}wr_i$ and $p_i^{I(O)}q_i^{I(O)}hr_i$ (which are nothing more than multiplications of variables that appear in the original models) have been replaced by the continuous variables lx_{ij}, by_{ij}, uv_i, $p^{I(O)}m_i^{I(O)}$, $p^{I(O)}n_i^{I(O)}$, $p^{I(O)}wr_i$, $q^{I(O)}hr_i$, $p^{I(O)}q^{I(O)}wr_i$ and $p^{I(O)}q^{I(O)}hr_i$, and by constraints that ensure equality between each term and its corresponding variable.

For instance, consider the inequality (11) from the nonlinear model where $xs_i \leq l_{ij}x_j + W(1-l_{ij})$ $\forall i,j; i<j$. In order to linearize the quadratic term preserving the rules of the original model, the inequality can be replaced by these constraints:

$$xs_i \leq lx_{ij} + W(1-l_{ij}) \qquad \forall i,j; i<j \tag{35}$$

$$lx_{ij} - Wl_{ij} \leq 0 \qquad \forall i,j \tag{36}$$

$$lx_{ij} - x_j \leq 0 \qquad \forall i,j \tag{37}$$

$$W(l_{ij}-1) + x_j - lx_{ij} \leq 0 \qquad \forall i,j \tag{38}$$

Constraints (36), (37) and (38) ensure that when $l_{ij} = 0$ or $x_j = 0$, then $lx_{ij} = 0$, and when $l_{ij} = 1$ and $x_j > 0$, then $lx_{ij} = x_j$. Therefore, lx_{ij} in (35) is guaranteed to be equal the product $l_{ij}x_j$.

For the constraint (13), a common technique to linearize modules was used. Thus, the modules of the differences between x_i^O and x_j^I and between y_i^O and y_j^I were converted to equivalent constraints by creating the variables dx_{ij} and dy_{ij}, whose sum results in d_{ij}.

The linearized models are presented next in four topics, in a format close to that adopted previously for the MIP models, with the difference that the second part of Models 2 and 4 was condensed as being very similar. The notations are the same as those in Table 2, along with the additional notations created in Table 3.

Table 3 – Additional notations used in Models 1, 2, 3 and 4 linearized.

Decision variables	
dx_{ij}	auxiliary variable that replaces the nonlinear binomial $\left\lvert x_i^O - x_j^I \right\rvert$
dy_{ij}	auxiliary variable that replaces the nonlinear binomial $\left\lvert y_i^O - y_j^I \right\rvert$
lx_{ij}	auxiliary variable that replaces the nonlinear term $l_{ij}x_i$
by_{ij}	auxiliary variable that replaces the nonlinear term $b_{ij}y_i$
uv_i	auxiliary variable that replaces the nonlinear term u_iv_i
$p^{I(O)}m_i^{I(O)}$	auxiliary variable that replaces the nonlinear term $p_i^{I(O)}m_i^{I(O)}$
$p^{I(O)}n_i^{I(O)}$	auxiliary variable that replaces the nonlinear term $p_i^{I(O)}n_i^{I(O)}$
$p^{I(O)}wr_i$	auxiliary variable that replaces the nonlinear term $p_i^{I(O)}wr_i$
$q^{I(O)}hr_i$	auxiliary variable that replaces the nonlinear term $q_i^{I(O)}hr_i$
$p^{I(O)}q^{I(O)}wr_i$	auxiliary variable that replaces the nonlinear term $p_i^{I(O)}q_i^{I(O)}wr_i$
$p^{I(O)}q^{I(O)}hr_i$	auxiliary variable that replaces the nonlinear term $p_i^{I(O)}q_i^{I(O)}hr_i$

First part of Models 1, 2, 3 and 4 linearized

$$\min \sum_{i=1}^{N} \sum_{j=1}^{N} f_{ij}d_{ij} \tag{39}$$

subject to

$$wr_i = (1-u_i)w_i + u_i h_i \qquad \forall i \tag{40}$$

$$hr_i = (1-u_i)h_i + u_i w_i \qquad \forall i \tag{41}$$

$$xs_i = x_i + wr_i \qquad \forall i \tag{42}$$

$$ys_i = y_i + hr_i \qquad \forall i \tag{43}$$

$$l_{ij} + l_{ji} + b_{ij} + b_{ji} \geq 1 \qquad \forall i,j; i<j \tag{44}$$

$$xs_i \leq lx_{ij} + W(1-l_{ij}) \qquad \forall i,j \tag{45}$$

$$ys_i \leq by_{ij} + H(1-b_{ij}) \qquad \forall i,j \tag{46}$$

$$lx_{ij} - Wl_{ij} \leq 0 \qquad \forall i,j \tag{47}$$

$$lx_{ij} - x_j \leq 0 \qquad \forall i,j \tag{48}$$

$$W(l_{ij}-1) + x_j - lx_{ij} \leq 0 \qquad \forall i,j \tag{49}$$

$$by_{ij} - Hb_{ij} \leq 0 \qquad \forall i,j \tag{50}$$

$$by_{ij} - y_j \leq 0 \qquad \forall i,j \tag{51}$$

$$H(b_{ij}-1) + y_j - by_{ij} \leq 0 \qquad \forall i,j \tag{52}$$

$$d_{ij} = dx_{ij} + dy_{ij} \qquad \forall i,j \tag{53}$$

$$dx_{ij} \geq x_i^O - x_j^I \qquad \forall i,j \tag{54}$$

$$dx_{ij} \geq x_j^I - x_i^O \qquad \forall i,j \tag{55}$$

$$dy_{ij} \geq y_i^O - y_j^I \qquad \forall i,j \tag{56}$$

$$dy_{ij} \geq y_j^I - y_i^O \qquad \forall i,j \tag{57}$$

$$\begin{aligned} d_{ij}, wr_i, hr_i, x_i, y_i, xs_i, ys_i, x_i^{I(O)}, y_i^{I(O)}, m_i^{I(O)}, n_i^{I(O)}, \\ lx_{ij}, by_{ij}, dx_{ij}, dy_{ij}, uv_i, p^{I(O)}m_i^{I(O)}, p^{I(O)}n_i^{I(O)}, \\ p^{I(O)}q^{I(O)}wr_i, p^{I(O)}q^{I(O)}hr_i, p^{I(O)}wr_i, q^{I(O)}hr_i \geq 0 \qquad \forall i \end{aligned} \tag{58}$$

$$u_i, v_i, p_i^{I(O)}, q_i^{I(O)} \in \{0,1\} \qquad \forall i \tag{59}$$

$$l_{ij}, b_{ij} \in \{0,1\} \qquad \forall i,j \tag{60}$$

Second part of Model 1 linearized (UAFLPfixed)

$$m_i^{I(O)} = n_i^{I(O)} = p_i^{I(O)} = q_i^{I(O)} = 0 \qquad \forall i \tag{61}$$

$$\begin{aligned} x_i^{I(O)} = x_i + uv_i(h_i - I(O)y_i) + (1 - u_i - v_i + uv_i)I(O)x_i \\ +(u_i - uv_i)I(O)y_i + (v_i - uv_i)(w_i - I(O)x_i) \qquad \forall i \end{aligned} \tag{62}$$

$$\begin{aligned} y_i^{I(O)} = y_i + uv_iI(O)x_i + (1 - u_i - v_i + uv_i)I(O)y_i \\ +(u_i - uv_i)(w_i - I(O)x_i) + (v_i - uv_i)(h_i - I(O)y_i) \qquad \forall i \end{aligned} \tag{63}$$

$$uv_i - u_i \leq 0 \qquad \forall i \tag{64}$$

$$uv_i - v_i \leq 0 \qquad \forall i \tag{65}$$

$$u_i + v_i - uv_i \leq 1 \qquad \forall i \tag{66}$$

Second part of Models 2 and 4 linearized (UAFLP1 and 3)

$$m_i^{I(O)} \leq wr_i \ (if\ m_i^{I(O)} = \frac{wr_i}{2},\ we\ have\ UAFLP1) \qquad \forall i \tag{67}$$

$$n_i^{I(O)} \leq hr_i \ (if\ n_i^{I(O)} = \frac{hr_i}{2},\ we\ have\ UAFLP1) \qquad \forall i \tag{68}$$

$$x_i^{I(O)} = x_i + m_i^{I(O)} - p^{I(O)}m_i^{I(O)} + p^{I(O)}q^{I(O)}wr_i \qquad \forall i \tag{69}$$

$$y_i^{I(O)} = y_i + p^{I(O)}n_i^{I(O)} + q^{I(O)}hr_i - p^{I(O)}q^{I(O)}hr_i \qquad \forall i \tag{70}$$

$$p^{I(O)}m_i^{I(O)} - (w_i + h_i)p_i^{I(O)} \leq 0 \qquad \forall i, j \tag{71}$$

$$p^{I(O)}m_i^{I(O)} - m_i^{I(O)} \leq 0 \qquad \forall i, j \tag{72}$$

$$(w_i + h_i)(p_i^{I(O)} - 1) + m_i^{I(O)} - p^{I(O)}m_i^{I(O)} \leq 0 \qquad \forall i, j \tag{73}$$

$$p^{I(O)}n_i^{I(O)} - (w_i + h_i)p_i^{I(O)} \leq 0 \qquad \forall i, j \tag{74}$$

$$p^{I(O)}n_i^{I(O)} - n_i^{I(O)} \leq 0 \qquad \forall i, j \tag{75}$$

$$(w_i + h_i)(p_i^{I(O)} - 1) + n_i^{I(O)} - p^{I(O)}n_i^{I(O)} \leq 0 \qquad \forall i, j \tag{76}$$

$$q^{I(O)}hr_i - (w_i + h_i)q_i^{I(O)} \leq 0 \qquad \forall i, j \tag{77}$$

$$q^{I(O)}hr_i - hr_i \leq 0 \qquad \forall i, j \tag{78}$$

$$(w_i + h_i)(q_i^{I(O)} - 1) + hr_i - q^{I(O)}hr_i \leq 0 \qquad \forall i, j \tag{79}$$

$$p^{I(O)}q^{I(O)}wr_i - (w_i + h_i)p_i^{I(O)} \leq 0 \qquad \forall i, j \tag{80}$$

$$p^{I(O)}q^{I(O)}wr_i - (w_i + h_i)q_i^{I(O)} \leq 0 \qquad \forall i, j \tag{81}$$

$$p^{I(O)}q^{I(O)}wr_i - wr_i \leq 0 \qquad \forall i, j \tag{82}$$

$$(w_i + h_i)(p_i^{I(O)} + q_i^{I(O)} - 2) + wr_i - p^{I(O)}q^{I(O)}wr_i \leq 0 \qquad \forall i, j \tag{83}$$

$$p^{I(O)}q^{I(O)}hr_i - (w_i + h_i)p_i^{I(O)} \leq 0 \qquad \forall i, j \tag{84}$$

$$p^{I(O)}q^{I(O)}hr_i - (w_i + h_i)q_i^{I(O)} \leq 0 \qquad \forall i, j \tag{85}$$

$$p^{I(O)}q^{I(O)}hr_i - hr_i \leq 0 \qquad \forall i, j \tag{86}$$

$$(w_i + h_i)(p_i^{I(O)} + q_i^{I(O)} - 2) + hr_i - p^{I(O)} q^{I(O)} hr_i \leq 0 \qquad \forall i, j \tag{87}$$

Second part of Model 3 linearized (UAFLP2)

$$m_i = 0 \qquad \forall i \tag{88}$$

$$n_i = 0 \qquad \forall i \tag{89}$$

$$x_i^{I(O)} = x_i + p^{I(O)} wr_i \qquad \forall i \tag{90}$$

$$y_i^{I(O)} = y_i + q^{I(O)} hr_i \qquad \forall i \tag{91}$$

$$p^{I(O)} wr_i - (w_i + h_i) p_i^{I(O)} \leq 0 \qquad \forall i, j \tag{92}$$

$$p^{I(O)} wr_i - wr_i \leq 0 \qquad \forall i, j \tag{93}$$

$$(w_i + h_i)(p_i^{I(O)} - 1) + wr_i - p^{I(O)} wr_i \leq 0 \qquad \forall i, j \tag{94}$$

$$q^{I(O)} hr_i - (w_i + h_i) q_i^{I(O)} \leq 0 \qquad \forall i, j \tag{95}$$

$$q^{I(O)} hr_i - hr_i \leq 0 \qquad \forall i, j \tag{96}$$

$$(w_i + h_i)(q_i^{I(O)} - 1) + hr_i - q^{I(O)} hr_i \leq 0 \qquad \forall i, j \tag{97}$$

After linearization, the mathematical models were transposed to a proper programming language and solved using specialized software. The entire phase of implementation and instance testing was performed in the mathematical programming language AMPL using a Notebook with an Intel Core i7 Processor, 8GB RAM and Windows 10 operating system. For resolution, two mathematical software were used: the Baron solver (version 19.7.13), specialized in solving nonlinear optimization problems and the CPLEX solver (version 12.10.0.0), high performance for linear programming, effective for the case of linearized models developed for this work.

The Baron solver was found to be insufficient to provide optimal solutions in a timely manner for the tested instances. Even for minor problems (of $N = 6$) and leaving the program running for some days, the solver failed to achieve a proven optimal result (in one case it achieved a result already known in the literature as optimal but could not prove that it was optimal and thus continued trying to improve the solution indefinitely). For major problems, the results proved to be bad even after the program ran for many hours.

Because of that, instance tests from the implementation of linearized models were prioritized, with the help of the CPLEX solver.

5 COMPUTATIONAL RESULTS AND DISCUSSIONS

Developing, or even improving, mathematical models that portray real-world problem situations as faithfully as possible is extremely important, but so is finding solutions for these models. In this work, as previously explained, a mathematical optimization approach was used to solve optimization problems. The results were obtained for several instances found in the literature. The instances were selected to cover both simpler problems and larger ones that are, consequently, more difficult to solve computationally.

Table 4 shows the ten tested instances, with their respective referenced articles, the position of the I/O points according to the input data, number of facilities (directly related to the size of the problem), and the code used to refer to each instance in this paper.

Table 4 – Features of the instances available in the literature.

Reference	Data type	Size	Code
Das (1993)	fixed I/O points	4	Das93N4
		6	Das93N6
		8	Das93N8
		10	Das93N10
		12	Das93N12
Welgama & Gibson (1993)	fixed I/O points	6	Wel93N6
		12	Wel93N12
Deb & Bhattacharyya (2005)	without previous I/O points	12	Deb05N12
		18	Deb05N18
Dunker et al. (2003)	I/O points in the centroids	62	Dun03N62

As shown in the table, some problems previously define the position of the I/O points in the blocks and others do not present this information. In the latter case, the I/O points can be considered variables, and their location is defined in the solution of the problem.

As previously mentioned, an algorithm was implemented to solve layout problems, which consists in solving the linearized MIP models by CPLEX solver with a time limit of two hours. A time limit was established for standardization purposes and this was defined after some performance tests. In the case of CPLEX, for the largest instance, the program had a memory overflow error when running for a long time, so $t = 2h$ was set.

The analysis of the results was divided into two subsections, the first referring to the initial fixed I/O points situation contemplated by Model 1 and the second referring to the situation of variable I/O points treated in Models 2, 3, and 4. Although some of the results compared in this paper date from a long time, they are still the best found in the literature for each situation.

5.1 Fixed I/O points

From the ten instances presented in Table 4, notice that two do not have initial input and output coordinate data; therefore, in this subsection, the results for eight instances found by Das (1993), Dunker et al. (2003), Rajasekharan et al. (1998), and Welgama & Gibson (1993) were evaluated. The objective function values of the authors were compared with those found by the approach proposed in this work. For that, the distance metric adopted was the same: rectilinear distance. The data compilation is shown in Table 5.

The results found with the linearization of the models and use of CPLEX were the best for almost all instances, the exception being the largest one with 62 facilities.

Table 5 – Comparative results for fixed I/O points.

	Best solution in literature			Our
Instance	Reference	Algorithm method	Solution	approach
Das93N4	RAJ1998	Genetic algorithm	**1393.6**[1]	**1393.6**[1]
Das93N6			2612.7	**2556**[1]
Das93N8	DUN2003	Coevolutionary algorithm	**8778.3**[1]	**8778.3**[1]
Das93N10			15694.5	**15222.9**
Das93N12			37396.1	**36622.8**
Wel93N6	WEL1993	Construction algorithm	421.5	**398.5**[1]
Wel93N12			5903	**5458.5**
Dun03N62	DUN2003	Coevolutionary algorithm	**3939362**	5484080

[1]Optimal solution.

For the minor instances (with $N \leq 8$), our approach was able to return the optimal result of the problem before the time limit. Figure 5 presents representation of the optimal results achieved as well as the time spent solving the problems.

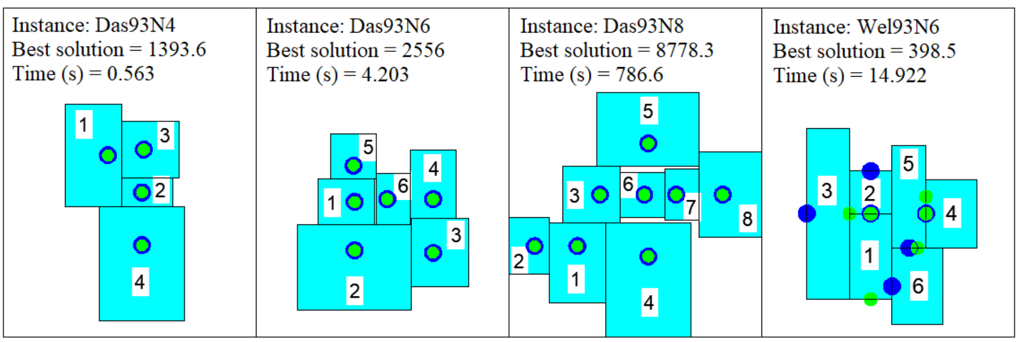

Figure 5 – Optimal layouts generated for fixed I/O points.

5.2 Variable I/O points

For the variable I/O points case, even instances initially proposed with fixed I/O points can be tested, because these coordinates can be disregarded in the problem input data.

The literature results of two articles referring to five instances were evaluated for comparison purposes, which vary as to the constraints applied to the implementation locations of the input and output points of the departments that define the UAFLP1, UAFLP2, and UAFLP3 models (as shown in 2).

The analysis of the results can be seen in the tables below. Table 6 refers to the values found by Deb & Bhattacharyya (2005), who used the GA-SA method of random search associated with a hybrid algorithm that integrates genetic algorithm and simulated annealing. Leno et al. (2018) developed the ESHGA method, a hybrid genetic algorithm based on an elitist strategy that uses simulated annealing as local search, and their results are in Table 7.

Table 6 – Comparative results for variable I/O points - Part I.

Model	Instance	Best solution in literature[1] Metric	Solution	Our approach
UAFLP1	Deb05N12	rectilinear distance	21142	**11504**
	Deb05N18		62902	**37382**
UAFLP3	Deb05N12		19892	**6924**
	Deb05N18		61756	**27092**

[1]Reference: Deb & Bhattacharyya (2005); Method: GA-SA.

Table 7 – Comparative results for variable I/O points - Part II.

Model	Instance	Best solution in literature[1] Metric	Solution	Our approach
UAFLP2	Deb05N12	rectilinear distance	8676	**6804**
	Deb05N18		440124	**27780**
UAFLP1	Wel93N6	contour distance	325	**237**[2]
	Wel93N12		5750	**5288**
	Deb05N12		37894	**12404**
	Deb05N18		72451	**40754**
UAFLP2	Wel93N6		266	**115**[2]
	Wel93N12		5454	**2889**
	Deb05N12		16404	**7028**
	Deb05N18		58324	**28540**
	Dun03N62		4921320	**3972940**

[1]Reference: Leno et al. (2018); Method: ESHGA.

[2]Optimal solution.

As can be seen in the tables, the results achieved with our approach were all better than those found in literature, highlighting their importance.

For the smaller instances (with $N = 6$), our approach was able to return the optimal result of the problem before the time limit. Figure 6 contains the representation of the optimal results achieved, as well as the time spent on solving the problems.

For visual comparison, in Figures 7 and 8, the final layouts generated by Leno et al. (2018) with the ESHGA method and by the methodology presented in this work are represented, for the instances Deb05N12 and Dun03N62, UAFLP2 Model.

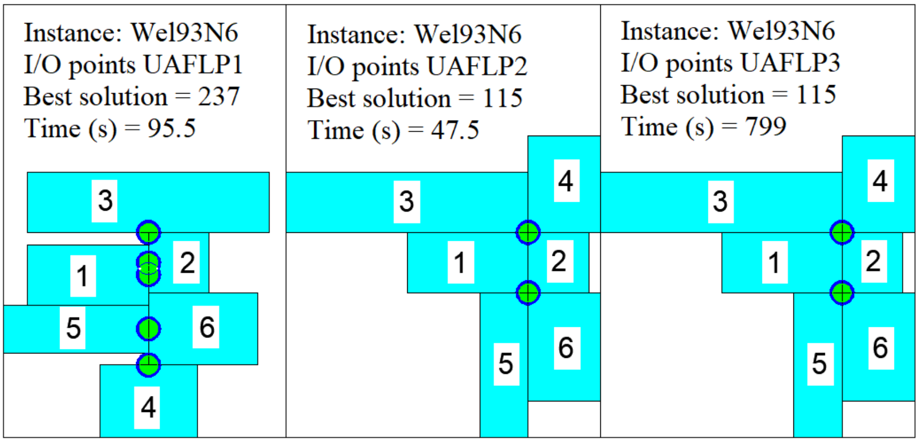

Figure 6 – Optimal layouts generated for variables I/O points.

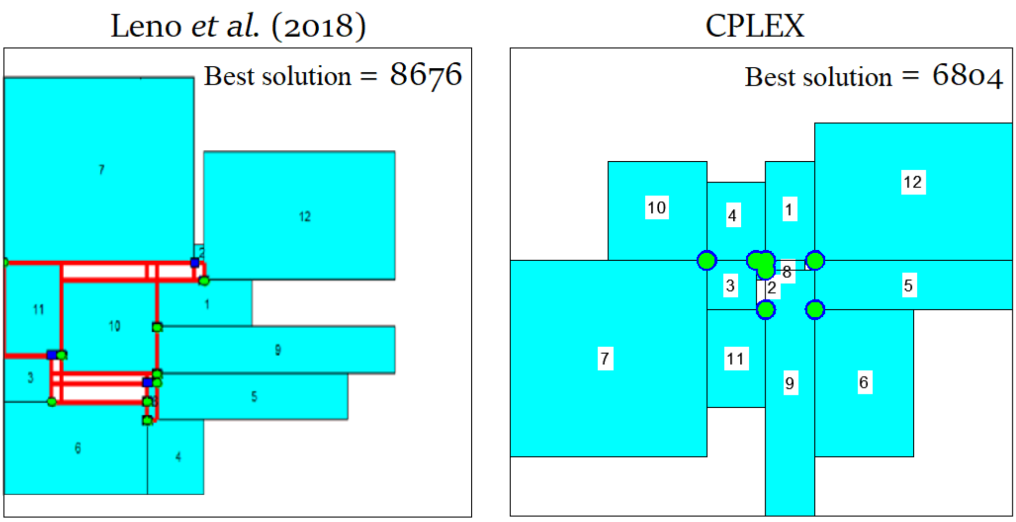

Figure 7 – Layouts generated for the UAFLP2 Model - Instance: Deb05N12.

6 CONCLUSIONS

The unequal area facility layout problem with input and output locations has already been solved in the literature with the use of the most varied methods. In most of the articles consulted during the literature review, the authors use heuristics and metaheuristics to solve the UAFLP problem with fixed or variable input and output locations. Heuristic methods are applied in optimization problems to find good results in a reasonable time, which cannot always be achieved by mathematical optimization approaches. However, with the advancement of solvers over time, the results became more competitive and, in the case of this work, were promising. Even so, for larger instances the challenge remains, as seen in the case of N = 62 which had a worse result with CPLEX compared to a metaheuristic developed much earlier by Dunker et al. (2003).

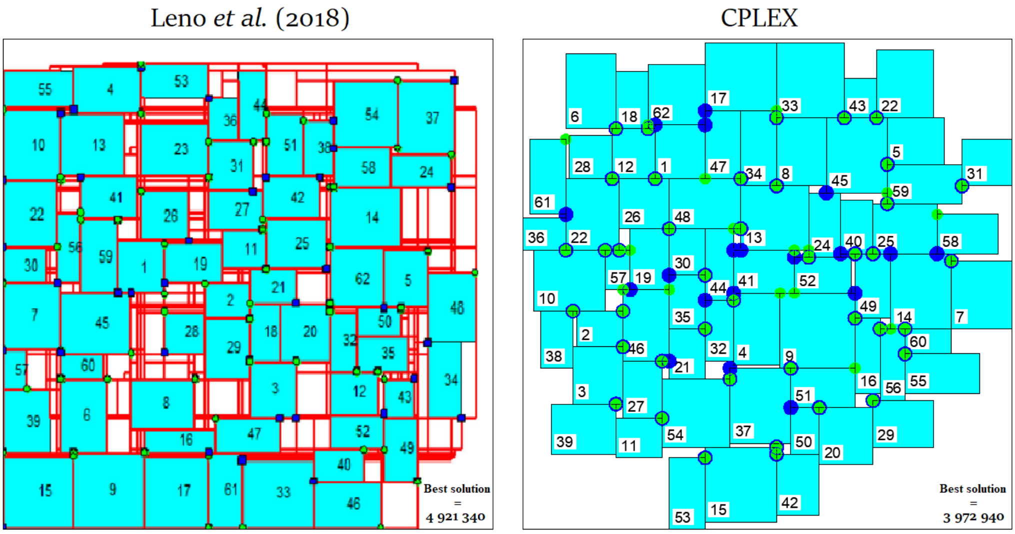

Figure 8 – Layouts generated for the UAFLP2 Model - Instance: Dun03N62.

Therefore, this paper has proved to be an excellent contribution to the subject of layouts, both in terms of modeling in non-linear and linear formats and using mathematical software to solve the problem. More specifically, the use of CPLEX applied to linear models showed remarkable results.

An interesting future line of work could be to analyze, using mathematical optimization techniques, other particular cases of the extensive field of layout optimization, as facilities in the form of rectangular blocks of fixed areas but variable dimensions respecting an aspect ratio. Furthermore, this research could be expanded taking into consideration the possibility of some additional remarks, e.g., the inclusion of pre-fixed facilities, that already start with the problem in places that cannot be modified during the generation of the solution. Future work could also focus on flexible bay structures or slicing tree representations. Finally, another possible research direction could be to treat the Dynamic Facility Layout Problem (DFLP) considering unequal areas and I/O points.

Acknowledgements

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

References

ARMOUR GC & BUFFA ES. 1963. A heuristic algorithm and simulation approach to relative location of facilities. *Management Science*, **9**(2): 294–309.

BAZARAA MS. 1975. Computerized layout design: a branch and bound approach. *AIIE transactions*, **7**(4): 432–438.

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

CHANG MS & KU TC. 2013. A slicing tree representation and QCP-model-based heuristic algorithm for the unequal-area block facility layout problem. *Mathematical Problems in Engineering*, **2013**.

CORMEN TH, LEISERSON CE, RIVEST RL & STEIN C. 2009. *Introduction to algorithms*. 2nd ed.. Massachusetts, USA: MIT press.

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

DEB S & 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.

DIJKSTRA EW. 1959. A note on two problems in connexion with graphs. *Numerische mathematik*, **1**(1): 269–271.

DUNKER T, RADONS G & WESTKÄMPER E. 2003. A coevolutionary algorithm for a facility layout problem. *International Journal of Production Research*, **41**(15): 3479–3500.

EMAMI S & NOOKABADI AS. 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, PIERREVAL H, SALAS-MORERA L & ARAUZO-AZOFRA A. 2013. Handling qualitative aspects in Unequal Area Facility Layout Problem: An Interactive Genetic Algorithm. *Applied Soft Computing*, **13**(4): 1718–1727.

GARCÍA-HERNÁNDEZ L, SALAS-MORERA L, CARMONA-MUÑOZ C, GARCÍA-HERNÁNDEZ JA & SALCEDO-SANZ S. 2020. A novel Island Model based on Coral Reefs Optimization algorithm for solving the unequal area facility layout problem. *Engineering Applications of Artificial Intelligence*, **89**: 103445.

GARCÍA-HERNÁNDEZ L, SALAS-MORERA L, GARCÍA-HERNÁNDEZ JA, SALCEDO-SANZ S & DE OLIVEIRA JV. 2019. Applying the coral reefs optimization algorithm for solving unequal area facility layout problems. *Expert Systems With Applications*, **138**: 112819.

GLOVER F & WOOLSEY E. 1974. Converting the 0-1 polynomial programming problem to a 0-1 linear program. *Operations research*, **22**(1): 180–182.

GONÇALVES JF & RESENDE MG. 2015. A biased random-key genetic algorithm for the unequal area facility layout problem. *European Journal of Operational Research*, **246**(1): 86–107.

HILLIER FS & LIEBERMAN GJ. 2013. *Introdução à pesquisa operacional*. 9th ed.. Brasil: McGraw Hill Brasil.

HOSSEINI-NASAB H, FEREIDOUNI S, GHOMI SMTF & FAKHRZAD MB. 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 & 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.

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

JUNG K, CHOI S, KULVATUNYOU B, CHO H & MORRIS KC. 2017. A reference activity model for smart factory design and improvement. *Production planning & control*, **28**(2): 108–122.

KIM JG & KIM YD. 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 BA & SMITH AE. 2006. A new mixed integer programming formulation for facility layout design using flexible bays. *Operations Research Letters*, **34**(6): 660–672.

KOOPMANS TC & BECKMANN M. 1957. Assignment problems and the location of economic activities. *Econometrica: journal of the Econometric Society*, pp. 53–76.

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.

LENO IJ, SANKAR SS & 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. 2019. Applying multi-objective ant colony optimization algorithm for solving the unequal area facility layout problems. *Applied Soft Computing*, **74**: 167–189.

LIU J, LIU J, YAN X & 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.

MAURI GR & LORENA LAN. 2009. Decomposições Lagrangeanas para o problema de programação quadrática binária irrestrita. *Pesquisa Operacional*, **29**(1): 111–127.

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

MORADI N & SHADROKH S. 2019. Optimization of Bi-objective Un-equal Area Facility Layout Problem with Grid System Approach by Simulated Annealing. *International Journal of Applied Optimization Studies*, **2**(04): 09–22.

O'BRIEN C & ABDEL BARR S. 1980. An interactive approach to computer aided facility layout. *International Journal of Production Research*, **18**(2): 201–211.

PALOMO-ROMERO JM, SALAS-MORERA L & GARCÍA-HERNÁNDEZ L. 2017. An island model genetic algorithm for unequal area facility layout problems. *Expert Systems with Applications*, **68**: 151–162.

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

PILLAI VM, HUNAGUND IB & KRISHNAN KK. 2011. Design of robust layout for Dynamic Plant Layout Problems. *Computers & Industrial Engineering*, **61**(3): 813–823.

RAJAGOPALAN S, HERAGU SS & TAYLOR GD. 2004. A Lagrangian relaxation approach to solving the integrated pick-up/drop-off point and AGV flowpath design problem. *Applied Mathematical Modelling*, **28**(8): 735–750.

RAJASEKHARAN M, PETERS BA & YANG T. 1998. A genetic algorithm for facility layout design in flexible manufacturing systems. *International journal of Production research*, **36**(1): 95–110.

TATE DM & SMITH AE. 1995. Unequal-area facility layout by genetic search. *IIE transactions*, **27**(4): 465–472.

TOMPKINS JA, WHITE JA, BOZER YA & TANCHOCO JMA. 2010. *Facilities planning*. John Wiley & Sons.

WELGAMA P & 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.

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

ZHENG XJ. 2014. A connectivity graph generation approach for Manhattan path calculation in detailed facility layout. *Applied Mathematics and Computation*, **237**: 238–251.

How to cite

BRAGA EMH & SALLES NETO LL. 2022. A mathematical optimization approach based on linearized MIP models for solving facility layout problems. *Pesquisa Operacional*, **42**: e261044. doi: 10.1590/0101-7438.2022.042.00261044.