

An Algorithm for Solving Generalized Single Row Facility Layout Problem

Mana Meskar, Kourosh Eshghi*

Department of Industrial Engineering, Sharif University of Technology, Tehran, Iran Email: *eshghi@shari.edu

How to cite this paper: Meskar, M. and Eshghi, K. (2020) An Algorithm for Solving Generalized Single Row Facility Layout Problem. *American Journal of Operations Research*, **10**, 299-320. https://doi.org/10.4236/ajor.2020.106017

Received: July 13, 2020 Accepted: November 24, 2020 Published: November 27, 2020

Copyright © 2020 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0). http://creativecommons.org/licenses/by/4.0/ Abstract

Layout design problem is to determine a suitable arrangement for the departments so that the total costs associated with the flow among departments become least. Single Row Facility Layout Problem, SRFLP, is one of the layout problems that have many practical applications. This problem and its specific scenarios are often used to model many of the raised issues in the field of facility location. SRFLP is an arrangement of n departments with a specified length in a straight line so that the sum of the weighted distances between the pairs of departments is minimized. This problem is NP-hard. In this paper, first, a lower bound for a special case of SRFLP is presented. Then, a general case of SRFLP is presented in which some new and real assumptions are added to generate more practical model. Then a lower bound, as well as an algorithm, is proposed for solving the model. Experimental results on some instances in literature show the efficiency of our algorithm.

Keywords

Single Row Facility Layout Problem, Facility Location, Plant Layout, Optimization, Cuckoo Optimization Algorithm

1. Introduction

Layout design problem is to determine a suitable arrangement for the departments so that the total cost associated with the flow among departments becomes least. One of common layout problems is Single Row Facility Layout Problem (SRFLP). SRFLP was widely studied until the mid-nineties. Then, about the first years of the last decade, again an extensive research began in this field. Interestingly, research on many aspects of this problem is still open. SRFLP is a highly regarded problem in the field of facility layout (Kothari & Ghosh, 2012) [1]. SRFLP can be defined as follows: The main goal is to determine the arrangement of rectangular departments with the same height along a straight line such that the total cost of relations between departments is minimized. The cost of the relationship between a pair of departments is proportional to the flow rate and the distance between the two departments center. The number of departments in this problem is known as SRFLP's size (Simmons, 1969) [2].

It is proved that SRFLP is an NP-complete problem (Suresh & Sahu, 1993) [3]. So it is almost impossible to find optimal solution in large-sized instances in a reasonable time. Thus, efforts to obtain near-optimal solutions or remodeling the problem has attracted lots of attentions. SRFLP's model is known as ABSMODEL and was introduced by Heragu and Kusiak (1991) [4].

The next section of this paper is a short literature review of SRFLP and its solution procedures. A special case of the model and its lower bound are discussed in the Section 3. In Section 4, a Cuckoo Optimization Algorithm for solving an extension of SRFLP is proposed. The experimental results are reported in Section 5. Finally, discussion, conclusion and some suggestions for future research are presented in the following sections.

2. Literature Review

SRFLP was first presented by Simmons entitled "one-dimensional space allocation problem". (Simmons, 1969) [2]. It is associated with large classes of problems. For these problems, many exact and heuristic approaches have been proposed. Some of these problems are related to each other. Some of these problems including SAP¹ and GLOP² have been mentioned in (Kothari & Ghosh, 2012) [1]. In addition to these issues, the Quadratic Assignment Problem (QAP) is very similar to SRFLP. The objective function of QAP can be stated as:

$$\min_{p \in \pi} \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} b_{p(i)p(j)}$$
(1)

where π is the set of all permutations of $\{1, 2, \dots, n\}$ and $A = (a_{ij}) \in \mathcal{R}^{n \times n}$,

 $B = (b_{ij}) \in \mathbb{R}^{n \times n}$. This is called Koopmans and Beckmann form of QAP (Koopmans & Beckmann, 1957) [5]. In a typical example of QAP, there are *n* departments and also *n* certain locations. There is a given amount of flow between each pair of departments *i* and *j*, namely a_{ij} in Equation (1). For each pair of departments *i* and *j*, $b_{p(i)p(j)}$ represents the distance between them. The communication cost of each pair of departments is equal to the product of flow and distance between them. The goal is to assign each department to a location so that the total cost of communications between departments is minimal (Picard & Queyranne, 1981) [6]. The main difference of QAP and SRFLP is that distances between locations are known in QAP, but in SRFLP is unknown due to uncertainty about the locations of the departments.

The graph bandwidth minimization is also a special case of the single-row layout problem. This is a graph in which each edge has been assigned a number

¹Space Allocation Problem.

²Generalized Linear Ordering Problem.

as a weight of the edge and there is a specific mathematical function that assigns each vertex a certain value. The weight of each edge is the same as the amount of the flow between departments. In this case, the absolute value of the difference between two vertices labels is in accordance with the related departments distance. This problem is listed as one of the famous problems in graph theory and various algorithms have been developed to solve it.

2.1. SRFLP's Mathematical Models

Several different models for SRFLP has been proposed in previous studies. The first mathematical model was developed by Picard and Queyranne, (1981) [6]. In their paper, for the first time, it was quite clearly stated that any solution of SRFLP is, in fact, a permutation of the departments set. They used

 $\pi = (\pi(1), \pi(2), \dots, \pi(n))$ as a permutation of departments indices. The inverse of this permutation: $\pi^{-1}(i)$ is the position of department *i* in the permutation π and $B(i, j, \pi)$ is the set of indices of departments between *i* and *j*. The main goal of the model is to find the location of each department somehow to minimize the total weighted flow among departments calculated as follows:

Minimize
$$\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} c_{ij} d_{ij}$$

 $d_{ij} = \frac{l_i}{2} + \sum_{k \in B(i,j,\pi)}^{n} L_k + \frac{l_j}{2}$

The notations used in this formulation are presented as follows:

 c_{ii} Flow between the two departments *i* and *j*

 d_{ii} Distance between the centers of two departments *i* and *j*

 $B(i, j, \pi)$ The set of indices of departments between *i* and *j*

 l_i The length of department *i*

The above constraint is illustrated in Figure 1.

Love & Wong (1976) [7] presented the first binary mixed integer program to solve SRFLP. Later, a quadratic model was presented by Kumar, Hadjinicola, and Lin (1995) [8]. A new MIP model for SRFLP was also presented by Amaral (2006) [9]. In this model, additional variables α_{ij} were used. If department *i* is located left of the department *j*, then α_{ij} is one. Otherwise, it is zero.

Another model of SRFLP was presented by Amaral (2009) [10]. In this model a three-dimensional variable ζ_{ijk} was defined. The variable ζ_{ijk} is one if department k lies between departments i and j and otherwise its value is zero. A complete review of presented models in the literature have been provided by Kothari and Ghosh (2012d) [1] and Keller and Buscher (2015) [11].



Figure 1. The basic constraint of SRFLP model.

2.2. Solution Procedure

The exact methods for solving SRFLP, especially when n is big, are difficult and require a lot of computer memory. These methods are not able to solve largescale problems in a reasonable time. Exact method proposed so far includes branch and bound strategies (Simmons, 1969) [2], branch and cut strategies (André RS Amaral & Letchford, 2013) [12], cutting plane (André RS Amaral, 2009) [10], (Yen, 2008) [13], dynamic programming (Picard & Queyranne, 1981) [6] and semi-definite programming (Anjos, Kennings, & Vannelli, 2005; Anjos & Vannelli, 2008; Yen, 2008) [13] [14] [15]. The most efficient codes developed in literature was based on MIP model provided in 2008 that was able to solve SRFLP up to size 18 (Andre RS Amaral, 2008) [16]. In 2013, a significant improvement on the original codes was provided by Amaral and Letchford (2013) [12]. By proposing an algorithm based on branch and cut, solving time was also improved. In this method, problems up to size 30 departments were solved in reasonable time. Also, this algorithm presented a better upper and lower bounds on the main node. The algorithm was applied to three SRFLP with 110 departments and found that the gap between the lower and upper bounds on the running time of less than 2.5 days, was 3% to 4%.

Given that the SRFLP's exact algorithms are of high costs, researchers use heuristic methods to solve large size SRFLP. These heuristic algorithms are much faster than exact algorithms, but there is no guarantee that the optimal solution to the problem is found. Different algorithms presented in the literature are all trying to find near optimal solution in the least amount of time, and they differ in the type and structure of algorithm, such as being construction or improvement heuristic, the initial steps, search method of the feasible region, searching in a discrete or continuous space and stopping criteria.

All research on this topic was in a deterministic space until that Azadeh *et al.* (2011) [17] considered the input data uncertain and presented a fuzzy multivariate approach for facility layout design with ambiguity. Rubio-Sánchez, *et al.* (2016) [18] proposed an algorithm based on greedy randomized adaptive search procedure combined with path relinking to have a highly efficient search and stated that the proposed method outperformed all state-of-the-art methods until 2016. The exact and heuristic solutions that have been proposed to solve this problem are summarized in **Table 1**.

3. Single Row Equidistant Facility Layout Problem

Consider the SRFLP and suppose that all departments' length is the same, for example with constant length *I* and the flow between each two departments is an arbitrary number. This problem is called Single Row Equidistant Facility Layout Problem, SREFLP (Hungerländer, 2014) [58]. Without loss of generality, it is assumed that the department in the leftmost location is labeled by index 1 and the last department with the rightmost location is labeled with index n (number of departments). According to the new numbering, distance matrix of departments is shown in the below matrix.

| Table 1. Su | aggested solution | is and algorithms | s for solving SRFLP. |
|-------------|-------------------|--------------------|-----------------------|
| 14010 1.00 | issested solution | is and argorithmic | , ioi solving old Li. |

| | Method | Papers | | |
|----------------------|---|--|--|--|
| | Branch and bound | (Simmons, 1969) [2], (Heragu & Kusiak, 1991) [4], (Andre RS Amaral, 2006) [9] | | |
| Exact solutions | Branch and cut strategies | (André RS Amaral, 2009) [10], (André RS Amaral & Letchford, 2013) [12] | | |
| | Dynamic programming | (Picard & Queyranne, 1981) [6] | | |
| | Semidefinite programming | (Anjos & Vannelli, 2008) [15], (Yen, 2008) [13], (Anjos <i>et al.</i> , 2005) [14], (Anjos & Yen, 2009) [19], (Anjos & Vannelli, 2006a, 2006b) [20] [21] | | |
| | Cutting plane | (Anjos & Vannelli, 2008) [15], (Yen, 2008) [13], (Sanjeevi & Kianfar, 2010) [22] | | |
| | Genetic Algorithm (GA) | (Kothari & Ghosh, 2012a) [23], (Ozcelik, 2012) [24], (Datta, Amaral, & Figueira, 2011) [25], (Ponnambalam & Ramkumar, 2001) [26], (Kothari & Ghosh, 2014a) [27], (Ku, Hu, & Wang, 2011) [28], (Utamima & Ou-Yang, 2012) [29] | | |
| | Bat algorithm | (Büyüksaatçı, 2015) [30] | | |
| | Ant Colony Optimization (ACO) | (Guan & Lin, 2016) [31], (Solimanpur, Vrat, & Shankar, 2005) [32], (Clauss, Bernt, & Middendorf, 2013) [33] | | |
| | Tabu search | (Samarghandi & Eshghi, 2009) [34], (Samarghandi & Eshghi, 2010) [35], (Kothari & Ghosh, 2013) [36], (Kothari & Ghosh, 2012e) [1], (Ahonen, de Alvarenga, & Amaral, 2014) [37], (Yu, Zuo, & Murray, 2014) [38], (Lenin, Siva Kumar, Ravindran, & Islam, 2014) [39] | | |
| | Cuckoo Optimization (CO) | (Maadi, Javidnia, & Ghasemi, 2016) [40] | | |
| | Cross-entropy approach | (Ning & Li, 2018) [41] | | |
| | Forest Optimization (FO) | (Maadi <i>et al.</i> , 2016) [40] | | |
| | Particle Swarm Optimization (PSO) | (Samarghandi, Taabayan, & Jahantigh, 2010) [42], (Ou-Yang & Utamima, 2013) [43], (Utamima, 2012) [29] | | |
| Heuristic methods | Charged System Search (CSS) | (Kaveh & Safari, 2014) [44] | | |
| | Neighborhood search | (Kothari & Ghosh, 2012b) [45] | | |
| | Scatter search | (Kothari & Ghosh, 2014b) [45] | | |
| | Local search | (André RS Amaral, 2008) [46], (Ghosh, 2011) [47], (Palubeckis, 2015a) [48] | | |
| | Path relinking | (Kothari & Ghosh, 2012c) [49], (Rubio-Sánchez et al., 2016) [18] | | |
| | Hybrid algorithms | (Ou-Yang & Utamima, 2013) [43], (Azadeh, Roozbahani, & Moghaddam, 2013) [17], (Teo & Ponnambalam, 2008) [50] | | |
| | Imperialist Competitive Algorithm (ICA) | (Lian, Zhang, Gao, & Shao, 2006) [51] | | |
| | Hybrid Simulated Annealing | (Kamali, 2015) [52] | | |
| | Clonal Selection Algorithm (CSA) | (Ulutas, 2013) [53], (Hosseini-Nasab & Emami, 2012) [54] | | |
| | Bacterial Foraging Algorithm (BFA) | (Ulutas, 2013) [53] | | |
| | Simulated Annealing (SA) | (Palubeckis, 2017) [55], (Kamali, 2015) [52], (Ahonen <i>et al</i> , 2014) [37], (Ku <i>et al</i> , 2011) [28], (Palubeckis, 2015b) [56] | | |
| | Artificial bee colony algorithm | (Amaral, 2019) [57] | | |

| | [_ | 1 | 2 | 3 | | $n-1^{-1}$ |
|----------------|--------------|-----|-----|---|---|------------|
| | 1 | - | 1 | 2 | | n-2 |
| | 2 | 1 | _ | 1 | | n-3 |
| $D = l \times$ | 3 | 2 | 1 | _ | 1 | ÷ |
| | ÷ | ÷ | ÷ | 1 | _ | 1 |
| | <i>n</i> – 1 | n-2 | n-3 | | 1 | _ |

Since the distance matrix is calculated based on the center to center, adjacent departments have the distance *l*. The distance between every other one departments is equal to 2*l* and so on. In this case, the problem will be finding the indices corresponding to the departments. Clearly, the optimal solution or the final layout of departments is not related to the length of departments. Note that the distance matrix between departments with a primary index (unchanged) will be a permutation of the above matrix.

Now suppose that the matrix product of the flow matrix and the distance matrix is calculated. The sum of the values of the diagonal of the resulted matrix is equal to the objective function value. Let E be the matrix product of these two matrices ($E = C \times D$). The sum of diagonal values of matrix E is calculated as follows: (note that C is a symmetric matrix and $C_{ii} = 0$).

$$\sum_{i=1}^{n} e_{ii} = \sum_{i=1}^{n} \sum_{k=1}^{n} C_{ik} D_{ik} = 2 * \sum_{i=1}^{n-1} \sum_{k=i+1}^{n} C_{ik} D_{ik}$$
(2)

According to the SRFLP's definition, the right amount of Equation (2) is twice the amount of the objective function. The sum of the values of the diagonal of a matrix \boldsymbol{A} is called the matrix trace and denoted by $\operatorname{tr}(\boldsymbol{A})$, so we have $\operatorname{tr}(\boldsymbol{E}) = \operatorname{tr}(\boldsymbol{C} \times \boldsymbol{D}) = 2Z$.

It is clear that the new flow matrix can be generated by arranging the rows and columns of the first flow matrix. Thus, this scenario of SRFLP can be described as finding the best permutation matrix which leads us to the smallest value of the trace of the matrix product of new distance matrix and the first flow matrix. In mathematical terms, the problem can be stated as

min tr (XCX^TD) = min tr (CX^TDX) Here **X** is a permutation matrix which is unknown and should be find. In literature, this problem is called Quadric Assignment Problem (QAP) (Commander, 2005) [59].

SREFLP Lower Bound

Rendl and Wolkowicz, (1992) [60] used a trace form of QAP and presented a lower bound for it. Suppose that C is symmetric. If flow matrix between departments is not symmetric, for each element C_{ij} the average value of C_{ij} and C_{ji} can be replaced. Then a symmetric matrix would be resulted. It does not have any effect on the optimal solution.

Because of positive values, all eigenvalues of this symmetric matrix are real numbers. The matrixes C and D can be written in a diagonal format. Suppose that $\lambda_1, \lambda_2, \dots, \lambda_n$ and $\mu_1, \mu_2, \dots, \mu_n$ are eigenvalues of matrix C and Matrix D

arranged in descending and ascending order, respectively. Λ and Σ are defined as follows:

$$\mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & \cdots & 0\\ \vdots & \ddots & \vdots\\ 0 & \cdots & \lambda_n \end{bmatrix}, \quad \mathbf{\Sigma} = \begin{bmatrix} \mu_1 & \cdots & 0\\ \vdots & \ddots & \vdots\\ 0 & \cdots & \mu_n \end{bmatrix}$$

Let $C = U\Lambda U^{T}$ and $D = V\Sigma V^{T}$, where U and V are corresponding to the eigenvalues of Λ and Σ respectively.

By a theory from Rendl and Wolkowicz, (1992) [60], we will simply have for an arbitrary and orthogonal P (not exactly permutation matrix P), the following relation:

$$\operatorname{Min}_{\boldsymbol{P}\boldsymbol{P}^{\mathrm{T}}=\boldsymbol{I}}\operatorname{Trace}\left(\boldsymbol{C}\boldsymbol{P}^{\mathrm{T}}\boldsymbol{D}\boldsymbol{P}\right) = \sum_{i,j} \mu_{i}\lambda_{j}$$

Hence the MinTrace $(CP^{T}DP)$ for an arbitrary and orthogonal P equals to $\sum_{i,j} \mu_i \lambda_j$ which ocurrs in $P = U^{T}V$ (Rendl & Wolkowicz, 1992) [60]. If $P = U^{T}V$ is a permutation matrix, it would be the optimal solution for the problem, otherwise $\sum_{i,j} \mu_i \lambda_j$ can be considered as a lower bound for the original problem.

Later on, it is proved in Lemma 1 that if *CD* equals to *DC*, *U* and *V* can be selected in such a way that $P = U^{T}V$ is a permutation matrix. With the mentioned distance matrix, in a case that *C* is a coefficient of *D*, *CD* equals to *DC*.

If *CD* is not equal to *DC*, $P = U^{T}V$ is not a permutation matrix, and, therefore, it can be considered as a lower bound for the optimal solution. In this case |CD - DC| can be considered as an index for measuring the gap of the lower bound.

Suppose C = aD where D is a coefficient of C, therefore, U = V. For optimum permutation matrix: $P = U^T V = U^T U = I$. Thus, the exact solution of this version of SRFLP is achieved by replacing identity matrix for permutation matrix P. Note that, as it was mentioned in previous section, we have supposed that C is symmetric matrix and if it is not symmetric, for each element C_{ij} the average value of C_{ij} and C_{ji} should be replaced. For every QAP scenario when C (after changing C to become symmetric) is a coefficient of D, the exact solution is known.

Example 1: Consider 5 departments with the same 4 meters length. Consider the flow matrix between departments as *C*.

| | [| 3 | 5 | 4 | 10 |
|------------|---|---|---|---|-----|
| | 1 | _ | 1 | 4 | 7 |
| <i>C</i> = | 3 | 3 | _ | 1 | 7 |
| | 8 | 4 | 3 | - | 3 |
| | 6 | 5 | 1 | 1 | _] |

Because the lengths of all departments are the same, the distance matrix is not related to the matrices arrangement. The distance matrix D is:

| | | 4 | 8 | 12 | 16 | |
|------------|----|----|---|----|----|--|
| | 4 | _ | 4 | 8 | 12 | |
| D = | 8 | 4 | - | 4 | 8 | |
| | 12 | 8 | 4 | _ | 4 | |
| | 16 | 12 | 8 | 4 | _ | |

Here, C is not symmetric and for each element C_{ij} , the average value of C_{ij} and C_{ii} can be replaced. After doing the calculations, C changes to:

$$\boldsymbol{C} = \begin{bmatrix} -2 & 4 & 6 & 8 \\ 2 & -2 & 4 & 6 \\ 4 & 2 & -2 & 4 \\ 6 & 4 & 2 & -2 \\ 8 & 6 & 4 & 2 & - \end{bmatrix}$$

As it can be seen, *C* is a coefficient of *D* with $\alpha = 0.5$ and the best solution of this problem is shown in Figure 2.

Lemma 1. If $P = U^T V$ is a permutation matrix, it can be concluded that *CD* equals to *DC*.

Proof. U and V corresponding to the eigenvalues of Λ and Σ , respectively. Hence, U and V are orthogonal matrices and $U^{T} = U^{-1}$ and $V^{T} = V^{-1}$. Suppose that $P = U^{T}V$ is a permutation matrix. Hence:

$$VP^{-1} = VP^{T} = V(U^{T}V)^{T} = VV^{T}U = U$$

 $PV^{T} = U^{T}VV^{T} = U^{T}$

Let $X = P\Sigma P^{-1}$. For matrix product of *C* and *D*, it can be stated that:

$$\boldsymbol{C}\boldsymbol{D} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{T}} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{P}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{T}} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{P}\boldsymbol{\Sigma}\left(\boldsymbol{P}^{-1}\boldsymbol{P}\right)\boldsymbol{V}^{\mathrm{T}} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{X}\boldsymbol{P}\boldsymbol{V}^{\mathrm{T}} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{X}\boldsymbol{U}^{\mathrm{T}}$$

For matrix product of *D* and *C*, it can be stated that:

$$DC = V\Sigma V^{\mathsf{T}} U\Lambda U^{\mathsf{T}} = V\Sigma P^{\mathsf{T}} \Lambda U^{\mathsf{T}} = V\Sigma P^{-1} \Lambda U^{\mathsf{T}}$$
$$= V (P^{-1}P) \Sigma P^{-1} \Lambda U^{\mathsf{T}} = VP^{-1} X \Lambda U^{\mathsf{T}} = UX \Lambda U^{\mathsf{T}}.$$

Because P is a permutation matrix, it can be concluded that X is a diagonal matrix. Also, Λ is a diagonal matrix. In product of matrices, two adjacent diagonal matrices can be replaced by each other. Therefore, if P is a permutation matrix, then *CD* equals to *DC*.

4. Generalized Single Row Facility Layout Problem

The problem presented in this study is Generalized Single Row Facility Layout Problem (or GSRFLP). In SRFLP, each solution shows the sequence of locations of departments without considering cost factor. However, in real-world, the construction cost of a department can be different when the location is changed.



Figure 2. The solution of a specific scenario of SRFLP considered in Example 1.

On the other hand, the adjacency of two departments can be problematic. For example, the risk of fire or toxic gases produced in a department can be increased based on safety reason. In some situations, the minimum or maximum distance of two certain departments is pre-determined. Furthermore, in some cases, it may be necessary that two departments are located within a certain distance due to the shared resources allocate. The main difference of GSRFLP with SRFLP is consideration of departments' construction cost and safety factor. In this case, the cost of department construction is entered as a parameter through a matrix. The exact definition of the proposed GSRFLP is as follows: The goal is to determine the arrangement of rectangular departments with almost the same height along a straight line such that the total cost is minimized. Every two separate departments are related to each other. The relation between two departments is to move materials from one department to another and is calculated according to the distance between two departments. The cost of the relation between two departments is dependent on the distance between the two departments and the flow rate between them. The flow between two departments is defined as the number of times the materials between two departments is transferred.

In real-world cases, sometimes due to safety concerns, two departments should not be adjacent. Thus security factor is defined between each pair of departments. This parameter is imported as an input data through a matrix. Safety parameter b_{ij} is defined between two departments *i* and *j*. It is multiplied by the relation cost between two departments *i* and *j*. If adjacency of two departments doesn't have any problem this parameter value equals to one, otherwise, it is less than one. The construction cost of departments is a matrix and considered as an input data. Also, the minimum or maximum distance of each two departments is pre-determined if necessary. The total cost is the sum of construction and transportation costs between departments.

Given that GSRFLP is more complex than SRFLP and since SRFLP is NPcomplete (Suresh & Sahu, 1993) [3], GSRFLP is also NP-Complete. Thus, no polynomialy exact algorithm has presented yet to solve the problem in a reasonable time for large-sized problems. In this study, an algorithm is also presented for solving a large size GSRFLP in a reasonable time.

Constraints of the GSRFLP Model

There are some general constraints in the model such as the distance between departments, no overlapping of departments, and placement of the departments in the given space, derived from Love & Wong (1976) [7] as follows:

$$L_{ij} - R_{ij} = x_j - x_i + \frac{1}{2} \left(L_i - L_j \right) \quad i, j = 1, 2, \cdots, n; i \neq j$$
(3)

$$x_i - x_j + M\left(\alpha_{ij}\right) \ge L_i \qquad i, j = 1, 2, \cdots, n; i \neq j$$
(4)

$$x_i - x_j + M\left(1 - \alpha_{ij}\right) \ge L_j \quad i, j = 1, 2, \cdots, n; i \ne j$$
(5)

$$\alpha_{ij} = (0,1) \quad i, j = 1, 2, \cdots, n$$
 (6)

$$L_i \le x_i \le Y \quad i, j = 1, 2, \cdots, n \tag{7}$$

In **Figure 3**, the distance between the two departments is shown as a function of the departments' endpoints.

Minimum or maximum distance constraints: If necessary, the restriction related to the minimum or maximum distance between the departments can be written through the following inequalities:

$$R_{ij} - L_{ij} \ge Smin_{ij} + \frac{1}{2} (L_i + L_j) \quad i, j = 1, 2, \cdots, n; i \neq j$$
(8)

$$R_{ij} - L_{ij} \le Smax_{ij} + \frac{1}{2}(L_i + L_j) \quad i, j = 1, 2, \cdots, n; i \neq j$$
(9)

The construction costs constraints: The matrix CO is the construction cost matrix. The parameter co_{iy} is the construction cost of department *i*, if the department *i* locates on the point *y*. The variable δ_{iy} is defined as a binary variable which equals one, if the department *i* locates on the point *y*. Therefore, the following logical expression must be modeled:

$$\delta_{iy} = 1 \leftrightarrow x_i = y \tag{10}$$

The parameters *UB* and *LB* are defined as upper and lower limits of $x_i - y$, respectively.

$$UB: \min\{u \mid u \ge x_i - y, i = 1, 2, \cdots, n; y = 1, \cdots, Y\}$$
(11)

$$LB: \max\{lb \mid lb \le x_i - y, i = 1, 2, \dots, n; y = 1, \dots, Y\}$$
(12)

Two dummy variables, δ_{iy1} and δ_{iy2} , are used for modelling the expression (10).

$$x_i - y \le (UB)(1 - \delta_{1iy})$$
 $i = 1, 2, \dots, n; y = 1, \dots, Y$ (13)

$$x_i - y \ge (LB - \varepsilon) (\delta_{1iy}) + \varepsilon \quad i = 1, 2, \cdots, n; \ y = 1, \cdots, Y$$
(14)

$$x_i - y \ge LB(1 - \delta_{2iy})$$
 $i = 1, 2, \dots, n; \ y = 1, \dots, Y$ (15)

$$x_i - y \le (UB + \varepsilon) (\delta_{2iy}) - \varepsilon \quad i = 1, 2, \cdots, n; \ y = 1, \cdots, Y$$
(16)

$$\delta_{iy} = \delta_{1iy} + \delta_{2iy} - 1 \quad i = 1, 2, \dots, n; \ y = 1, \dots, Y$$
(17)

$$\delta_{iv}, \delta_{1iv}, \delta_{2iv} = (0,1) \tag{18}$$



Figure 3. Modeling the distance between departments.

The objective function is shown in the Equation (19):

$$\text{Minimize } \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} c_{ij} b_{ij} \left(R_{ij} + L_{ij} \right) + \sum_{i=1}^{n} \sum_{y=1}^{Y} \delta_{iy} co_{iy}$$
(19)

Since c_{ij} and b_{ij} are defined as parameters of the model, the product of two matrices *C* and *B*, can be replaced by a new flow matrix.

5. Proposed Cuckoo Optimization Algorithm for Solving SRFLP

A review of SRFLP's literature shows that many heuristic algorithms have been developed for finding its near-optimal solutions (Büyüksaatçı, 2015; Datta *et al.*, 2011; Kothari & Ghosh, 2012e, 2013, 2014a; Kumar *et al.*, 1995; Ozcelik, 2012; Palubeckis, 2015b; Samarghandi & Eshghi, 2010; Samarghandi *et al.*, 2010; Solimanpur *et al.*, 2005) [1] [24] [25] [27] [30] [32] [34] [36] [38] [42] [47]. The Cuckoo Optimization Algorithm (COA) is a new metaheuristic algorithm with high performance in some NP-hard problems (Rajabioun, 2011) [61]. Here, we apply an algorithm based on COA for SRFLP.

Cuckoo search algorithm was first introduced by Yang and Deb (2009) [62]. The COA was inspired by the life of a bird, called cuckoo. Characteristics of some species of this bird in egg laving in the nests of other bird spices and breeding was the basic motivation for the development of this new evolutionary optimization algorithm. The host may recognize some cuckoo eggs and kill them or it may abandon the nest and build a new nest elsewhere. The rest eggs that are not identified by the host, are fed by it. After hatching from eggs and growing up, the mature cuckoos follow the lifestyle of their mothers and lay eggs in other birds' nests. Also, grown up cuckoo societies immigrate toward the regions with better access to foods (Rajabioun, 2011) [61]. The mechanism of egg laying and immigration in COA help find local and global optimal solution of the problem. After multiple iterations of the algorithm, the cuckoos inhabit in the place with the best access to foods. This place is the best founded solution of the problem. For applying COA for SRFLP, it is necessary that the concepts of eggs, nest, step, objective function and search space be specified in details. Following, these concepts are described.

5.1. Definitions

Here, there are some basic definitions used in COA.

Nest: The number of nests is constant and equals to the number of initial population. In SRFLP, each nest is a sequence of departments.

Eggs: In the proposed algorithm, it is assumed that in each nest (solution), cuckoo puts one or more eggs. An egg is a new solution. In other words, in SRFLP an egg is equivalent to a permutation of the departments. The aim is to replace not good solutions by better ones.

Objective function: Every solution in the search space is equivalent to a numeric value for the objective function. So the quality of a possible answer is

proportional to the objective function value. In COA, nests with eggs having a lower value for the objective function, are leading to new generation of population. The cuckoo egg quality is proportional to its ability to create new cuckoo.

Search Space: In COA, the search space represents a potential nest. In the current problem, all possible permutations of departments are possible solutions to the problem. By changing a possible solution, creating another solution is possible. Searching the problem space is achieved by each step being carried out through consecutive changes in the solution called "*laying*".

Laying: The changes in the arrangement are determined by exchanging the locations of two randomly selected departments or two sequentially adjacent departments. To produce new solutions, steps length will be applied in accordance with the theory of Lévy flight (Yang & Deb, 2009) [62]. In the proposed algorithm, the step length is defined as the number of consecutive changes of the arrangement of departments.

5.2. Levy Flight

In nature, some insects, birds and many other animals follow paths that have been modeled using Levy flight. They can find and hunt their bait or food without planning ahead. In fact, they use Levy flight as a pattern for searching foods. A Levy flight is a complex set of random walks in which the step-lengths have a Levy distribution (Roy & Chaudhuri, 2013) [63]. In a Levy flight, most steps have a small length and the walk is within a small area, but longer routes are taken occasionally. In Equation (20), step length is presented in accordance with Levy flight distribution, where *s* is the step length and $Levy(s,\lambda)$ is the probability of having step with length *s*. As it can be seen, by increasing the step length, the probability reduces. As it is shown in Equation (21), constant C is determined in a way that the sum of the probabilities equals to 1. The search in the problem space is done according to the Levy flight theory. As a result, chance of taking a longer step is lower compared to a shorter step.

$$p(s) = CS^{-\lambda} \qquad 1 < \lambda \le 3 \tag{20}$$

$$\sum_{l_{\min}}^{\infty} CS^{-\lambda} = 1$$
 (21)

5.3. Distance Matrix Calculation

Given the importance of algorithm runtime, generating the neighborhoods of a solution is done in a way that would require less time to calculate the new distance matrix. If the neighborhood of a solution is generated by moving two adjacent departments or two random departments, then there is no need to calculate the distance matrix from the beginning, and with some changes in the distance matrix of the solution, the new distance matrix can be obtained.

By changing two adjacent departments i and j, without loss of generality, if we assume that after changing the location, department i is at the right side of department j, new distance of department k with departments i and j is as follows:

if
$$x_k < x_i < x_j$$

 $\begin{cases} new D_{ik} = D_{ik} - L_j \\ new D_{jk} = D_{jk} + L_i \end{cases}$

if $x_i < x_j < x_k$
 $\begin{cases} new D_{ik} = D_{ik} + L_j \\ new D_{jk} = D_{jk} - L_i \end{cases}$

Also, new distance of department k with department k' is as follows:

new
$$D_{kk'} = D_{kk'}$$
 $k, k' \neq i$ and j

By changing two random departments *i* and *j*, it can be stated that after changing the locations, new distance of department $k (k \neq i \text{ and } j)$ with departments *i* and *j* is as follows:

$$\text{if } x_i < x_k < x_j \begin{cases} new \ D_{ik} = D_{jk} + (L_i - L_j)/2 \\ new \ D_{jk} = D_{ik} + (L_j - L_i)/2 \end{cases}$$
$$\text{if } x_k < x_i < x_j \begin{cases} new \ D_{ik} = D_{jk} + (L_i - L_j)/2 \\ new \ D_{jk} = D_{ik} - (L_j - L_i)/2 \end{cases}$$
$$\text{if } x_i < x_j < x_k \begin{cases} new \ D_{ik} = D_{jk} - (L_i - L_j)/2 \\ new \ D_{jk} = D_{jk} - (L_i - L_j)/2 \end{cases}$$

Also, new distance of department k with department k', when only one of k and $k'(k \text{ and } k' \neq i \text{ and } j)$ is between departments i and j is as follows:

$$\begin{array}{ll} \text{if } x_i < x_k < x_j < x_{k'} \text{ or } x_i < x_k' < x_j < x_k & new \ D_{kk'} = D_{kk'} + L_j - L_i \\ \\ \text{if } x_{k'} < x_i < x_k < x_j \text{ or } x_{k'} < x_i < x_k < x_j & new \ D_{kk'} = D_{kk'} + L_i - L_j \end{array}$$

When departments k and k' are between departments i and j, or they are at the right and left side of departments i and j or vice versa, and when they are both at the right or left side of departments i and j, it can be shown that:

new
$$D_{kk'} = D_{kk'}$$
 for $k, k' \neq i$ and j

5.4. Algorithm Steps

This section represents an efficient algorithm, Algorithm 1, based on COA for solving GSRFLP. After initializing the algorithm's parameters, the first step is to generate cuckoos' population. Then, the cuckoos start to search for other birds' nest to lay their eggs. They choose nests with less probability of having their eggs recognized by the host to increase the probability of surviving. So at the second step, each cuckoo lay certain eggs in the host nest. Laying is done by consecutive changes of departments' arrangement. Each change in the arrangement is considered as length of one. Thus a step with length s is equivalent to s consecutive changes of departments' arrangement.

The host bird may recognize some cuckoo eggs and then it will either abandon the nest or kill the cuckoos. If a nest is not appropriate, the cuckoos will be killed by the host by probability of $1 - p_a$. In this step, the quality of the solutions is evaluated and then decision is made by replacing the eggs in the nest. The survived cuckoo eggs will grow up and immigrate to new better nests. They will follow lifestyle of their mothers. These steps are done in algorithm until stopping criteria reached. The steps of the algorithm are shown in Table 2.

5.5. Parameters Initialization

To set parameter for our algorithm, Taguchi method was used. Therefore, first, with some experiments, acceptable intervals for parameters were defined and then, for each parameter, 4 levels were designed. These levels were, for initial population number, 6, 8, 10 and 12, for Levy flight parameter (λ), 1.5, 2, 2.5, and 3, for number of eggs of each cuckoo (E), 1, 2, 3, and 4, and for maximum step length, 1, 2, 3, and 4. L16 was used for the 4-level design. Then, for each sets of parameters, 10 runs run on a sample from Heragu and Kusiak (1991) [4] with 30 departments. The designed Taguchi experiment was analyzed using Minitab 16.2.4. The Results of Taguchi analysis is shown in **Figure 4**. As it can be seen in **Figure 4**, the best initial number of population, Levy flight parameter (λ), number of eggs of each cuckoo (E), and maximum step length, were 10, 2.5, 4, and 3, respectively. Note that, there was a very small difference between value 3 and 4, for Levy flight parameter. But by considering run time, this parameter value was considered 3.

Table 2. Proposed algorithm.

| Algorithm 1, Cuckoo Optimization Algorithm | | | | |
|--|--|--|--|--|
| 1) Initialization includes parameters: initial population number (<i>n</i>), number of eggs of each cuckoo | | | | |
| (<i>E</i>), maximum step length, Levy flight parameter (λ), maximum iteration (max_it) | | | | |
| 2) Generating n initial population. | | | | |
| 3) While maximum number of algorithm repetition is not exceeded, it <= max_it | | | | |
| 4) For each cuckoo i = 1,, <i>n</i> | | | | |
| 5) For $e = 1,, E$ | | | | |
| 6) Search in the problem space according to the Levi flight Theory (Yang and Deb 2009) [62] which | | | | |
| is equivalent to laying an egg | | | | |
| 7) The egg quality is evaluated (<i>fi</i>) | | | | |
| 8) A nest is randomly selected. For example, Nest <i>j</i> | | | | |
| 9) If $f_i < f_j$. | | | | |
| 10) The cuckoo is replaced in the nest. | | | | |
| 11) else | | | | |
| 12) The inappropriate cuckoo (based on the quality) is abandoned with a probability of p_a . | | | | |
| 13) End IF | | | | |
| 14) End For | | | | |
| 15) End For | | | | |
| 16) Best solutions are stored in the memory (solutions with minimum costs). | | | | |
| 17) it = it + 1 | | | | |
| 18) Check stopping criteria | | | | |
| 19) End While | | | | |
| 20) Ranking the answers and determining the best | | | | |



Figure 4. Results of Taguchi analysis for parameter setting.

Another necessary step to implement the proposed algorithm was defining the initial population.

The initial population was produced as below:

• First population: Initially, a new hypothetical length for each department was calculated based on its actual length and its flow with other departments,

 $NewL_i = \frac{L_i}{\sum_j c_{ij}}$. Then, based on the new hypothetical length, taking into

account a constant value for the flow between departments, the optimal solution of SRFLP with a constant value for the flow between departments, was considered as the initial population.

- Second population: First, two departments with the highest flow, c_{ij} , were selected and placed next to each other. Then, among the other departments, department with the maximum flow with selected departments (sum of flow) was selected and placed in the best location. This step continued until all departments were selected.
- Other solutions required for initial population, were randomly generated.

The last step was to set a suitable stopping criteria for the algorithm. Based on some experiments on the instance of Heragu and Kusiak (1991) [4] with 30 departments, it was considered as the minimum of 3000 algorithm runs or no change in the best answer in 100 consecutive runs.

6. The Computational Results

In this section, the performance of our approach on several sets of GSRFLP instances is reported. The tests have been carried out on a portable computer with Intel(R) Core(TM) i3-2357M CPU with processor speed of 1.3 GHz and memory of two GB of Ram on MATLAB R2016b version 9.1.0.441655.

The algorithm was tested on sets of SRFLP instances by Anjos *et al.* (2005) [14] with some modifications due to the difference between SRFLP and GSRFLP.

This set consists of four groups with instances of size 60, 70, 75 and 80 and each group consists five instances.

Each problem ran ten times and the best solution of runs is considered as the final solution. Information related to these problems including name of the problem, the problem size (number of departments), the best answer found so far, the answer of the proposed algorithm, the running time in seconds for proposed algorithm, the difference between the obtained solution and best solution so far in percentage, is presented as the columns of **Table 3**. It is seen that the solutions for all instances have under 0.1% gap with the optimal value. Also, for 19 instances out of 20 instances, the proposed algorithm is able to find the best solution with approximately 0.0% gap. The execution time seems to be significantly efficient for large instances compared to literature (Kothari & Ghosh, 2014b) [64].

In **Figure 5**, the horizontal axis shows the name of the problem and the vertical axis shows the gap between the best eight random solutions and the first initial population with the best known solution of the problem. Despite this fact that the proposed initial population does not guarantee the suitability of the objective function's value in general, there is not even one case that best random population objective function value has been lower than the initially proposed population's one. On average, the minimum distance of the objective function on a random solution is 42% more than the proposed layout for initial population. In 18 from 20 instances the first initial population had less than 10% gap and in 5 from 20 instances it had less than 5% gap with the best found solution.



Figure 5. Comparison between best answer of proposed initial population with random population.

| | Name | Size | Best solution | Proposed solution | Time (seconds) | Gap% |
|----|-------------|------|--------------------------|-------------------|----------------|------|
| 1 | Anjos-60-01 | 60 | 1477834° | 1,477,834 | 27.66 | 0.0 |
| 2 | Anjos-60-02 | 60 | 841776 ^b | 841,814 | 23.48 | ~0.0 |
| 3 | Anjos-60-03 | 60 | 648,337.5 ^c | 648,651.5 | 45.04 | ~0.0 |
| 4 | Anjos-60-04 | 60 | 398,406 ^b | 398,481 | 15.60 | ~0.0 |
| 5 | Anjos-60-05 | 60 | 318,805 ^c | 318,855 | 31.28 | ~0.0 |
| 6 | Anjos-70-01 | 70 | 1,528,537 ^b | 1,528,760 | 55.96 | ~0.0 |
| 7 | Anjos-70-02 | 70 | 1,441,028 ^c | 1,441,504 | 34.68 | ~0.0 |
| 8 | Anjos-70-03 | 70 | 1,518,993.5° | 1,519,578.5 | 52.48 | ~0.0 |
| 9 | Anjos-70-04 | 70 | 968,796ª | 968,796 | 60.30 | 0.0 |
| 10 | Anjos-70-05 | 70 | 4,218,002.5 ^b | 4,218,017.5 | 44.87 | ~0.0 |
| 11 | Anjos-75-01 | 75 | 2,393,456.5ª | 23,958,845 | 62.64 | 0.1 |
| 12 | Anjos-75-02 | 75 | 4,321,190° | 4,321,381 | 75.73 | ~0.0 |
| 13 | Anjos-75-03 | 75 | 1,248,423 ^b | 1,248,664 | 66.85 | ~0.0 |
| 14 | Anjos-75-04 | 75 | 3,941,816.5 ^b | 3,942,749.5 | 72.41 | ~0.0 |
| 15 | Anjos-75-05 | 75 | 1,791,408 ^c | 1,792,038 | 45.05 | ~0.0 |
| 16 | Anjos-80-01 | 80 | 2,069,097.5° | 2,069,192.5 | 62.57 | ~0.0 |
| 17 | Anjos-80-02 | 80 | 1,921,136 ^b | 1,921,202 | 54.57 | ~0.0 |
| 18 | Anjos-80-03 | 80 | 3,251,368ª | 3,251,435 | 65.37 | ~0.0 |
| 19 | Anjos-80-04 | 80 | 3,746,515 ^c | 3,747,548 | 48.26 | ~0.0 |
| 20 | Anjos-80-05 | 80 | 1,588,885 ^b | 1,588,963 | 65.26 | ~0.0 |

Table 3. Comparison of thesolutions for the benchmark instances of Anjos *et al.* (2005)[14].

^aIndicates the best known solution reported in Datta *et al.* (2011); ^bIndicates the best known solution reported in Lian *et al.* (2011); ^cIndicates the best known solution reported in Samarghandi and Eshghi (2010).

7. Discussion

Single Row Facility Layout Problem, SRFLP, is one of the most interesting problems in the field of facility layout and it has many practical applications. SRFLP is NP-hard problem. In this paper, First, a new version of this problem, called Single Row Equidistant Facility Layout Problem, was introduced and its lower bound was calculated theoretically by proving a lemma. Then, another version of this problem, called Generalized Single Row Facility Layout Problem (or GSRFLP), was also presented and its model was discussed. Moreover, a Cuckoo Optimization Algorithm for solving GSRFLP is proposed. The experimental results indicate that the proposed cuckoo algorithm is capable of solving SRFLP and reached to satisfactory solutions in an acceptable time for small and large size instances. The highest gap between optimal solution and proposed algorithm solution is less than 0.1%. Also, it could improve the execution time for large size problems. The algorithm starts with a suitable population with average of 6.7% gap with optimal solution. In summary, three difficult models, their lower bounds and solution methods are discussed in this paper.

8. Conclusions and Future Research

In this paper, a lower bound, as well as an algorithm, are proposed for solving special classes of SRFLP. Computational experiments were also carried on large size sets of instances.

As future work, the lower bound of SRFLP can be extended to the other types of row layout problems including double row layout problem and GSRFLP. Another additional future work is to study solving larger instances of SRFLP and other layout problems by exact algorithms. Finally, special type of SRFLP with equal length of departments (SREFLP) can be an interesting topic for future research. Given known distance matrix in SREFLP, there would be cases with predetermined solutions.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- Kothari, R., and Ghosh, D. (2012) Tabu Search for the Single Row Facility Layout Problem in FMS Using a 3-opt Neighborhood. Technical Report, Institute of Management, Indian.
- Simmons, D.M. (1969) One-Dimensional Space Allocation: An Ordering Algorithm. *Operations Research*, 17, 812-826. <u>https://doi.org/10.1287/opre.17.5.812</u>
- [3] Suresh, G. and Sahu, S. (1993) Multiobjective Facility Layout Using Simulated Annealing. *International Journal of Production Economics*, **32**, 239-254. <u>https://doi.org/10.1016/0925-5273(93)90071-R</u>
- [4] Heragu, S.S. and Kusiak, A. (1991) Efficient Models for the Facility Layout Problem. *European Journal of Operational Research*, 53, 1-13. <u>https://doi.org/10.1016/0377-2217(91)90088-D</u>
- Koopmans, T.C. and Beckmann, M. (1957) Assignment Problems and the Location of Economic Activities. *Journal of the Econometric Society*, 25, 53-76. <u>https://doi.org/10.2307/1907742</u>
- [6] Picard, J.C. and Queyranne, M. (1981) On the One-Dimensional Space Allocation Problem. *Operations Research*, 29, 371-391. <u>https://doi.org/10.1287/opre.29.2.371</u>
- [7] Love, R. and Wong, J. (1976) On Solving a One-Dimensional Space Allocation Problem with Integer Programming. *INFOR: Information Systems and Operational Research*, 14, 139-143. <u>https://doi.org/10.1080/03155986.1976.11731633</u>
- [8] Kumar, K.R., Hadjinicola, G.C. and Lin, T.L. (1995) A Heuristic Procedure for the Single-Row Facility Layout Problem. *European Journal of Operational Research*, 87, 65-73. <u>https://doi.org/10.1016/0377-2217(94)00062-H</u>

- [9] Amaral, A.R. (2006) On the Exact Solution of a Facility Layout Problem. *European Journal of Operational Research*, **173**, 508-518. https://doi.org/10.1016/j.ejor.2004.12.021
- [10] Amaral, A.R. (2009) A New lower Bound for the Single Row Facility Layout Problem. *Discrete Applied Mathematics*, 157, 183-190. https://doi.org/10.1016/j.dam.2008.06.002
- [11] Keller, B. and Buscher, U. (2015) Single Row Layout Models. European Journal of Operational Research, 245, 629-644. <u>https://doi.org/10.1016/j.ejor.2015.03.016</u>
- [12] Amaral, A.R. and Letchford, A.N. (2013) A Polyhedral Approach to the Single Row Facility Layout Problem. *Mathematical programming*, **141**, 453-477. <u>https://doi.org/10.1007/s10107-012-0533-z</u>
- [13] Yen, G. (2008) Cutting-Plane Separation Strategies for Semidefinite Programming Models to Solve Single-Row Facility Layout Problems. University of Waterloo, Waterloo.
- [14] Anjos, M.F., Kennings, A. and Vannelli, A. (2005) A Semidefinite Optimization Approach for the Single-Row Layout Problem with Unequal Dimensions. *Discrete Optimization*, 2, 113-122. <u>https://doi.org/10.1016/j.disopt.2005.03.001</u>
- [15] Anjos, M.F. and Vannelli, A. (2008) Computing Globally Optimal Solutions for Single-Row Layout Problems Using Semidefinite Programming and Cutting Planes. *INFORMS Journal on Computing*, 20, 611-617. https://doi.org/10.1287/ijoc.1080.0270
- [16] Amaral, A.R. (2008) An Exact Approach to the One-Dimensional Facility Layout Problem. *Operations Research*, 56, 1026-1033. <u>https://doi.org/10.1287/opre.1080.0548</u>
- [17] Azadeh, A., Roozbahani, M.N. and Moghaddam, M. (2013) Optimisation of Complex and Large-Sized Single-Row Facility Layout Problems with a Unique Hybrid Meta-Heuristic Framework. *International Journal of Operational Research*, 16, 38-67. <u>https://doi.org/10.1504/IIOR.2013.050539</u>
- [18] Rubio-Sánchez, M., Gallego, M., Gortázar, F. and Duarte, A. (2016) GRASP with Path Relinking for the Single Row Facility Layout Problem. *Knowledge-Based Systems*, **106**, 1-13. <u>https://doi.org/10.1016/j.knosys.2016.05.030</u>
- [19] Anjos, M.F. and Yen, G. (2009) Provably Near-Optimal Solutions for Very Large Single-Row Facility Layout Problems. *Optimization Methods and Software*, 24, 805-817. <u>https://doi.org/10.1080/10556780902917735</u>
- [20] Anjos, M.F. and Vannelli, A. (2006) Globally Optimal Solutions for Large Single-Row Facility Layout Problems. University of Waterloo, Waterloo.
- [21] Anjos, M.F. and Vannelli, A. (2006) On the Computational Performance of a Semidefinite Programming Approach to Single Row Layout Problems. In: Hans-Dietrich, H., KopferJörn, H. and Schönberger, J., Eds., *Operations Research Proceedings* 2005, Bremen, 277-282. <u>https://doi.org/10.1007/3-540-32539-5_44</u>
- [22] Sanjeevi, S. and Kianfar, K. (2010) A Polyhedral Study of Triplet Formulation for Single Row Facility Layout Problem. *Discrete Applied Mathematics*, **158**, 1861-1867. <u>https://doi.org/10.1016/j.dam.2010.07.005</u>
- [23] Kothari, R. and Ghosh, D. (2012) A Competitive Genetic Algorithm for Single Row Facility Layout. Technical Report, Indian Institute of Management, India.
- [24] Ozcelik, F. (2012) A Hybrid Genetic Algorithm for the Single Row Layout Problem. *International Journal of Production Research*, 50, 5872-5886. <u>https://doi.org/10.1080/00207543.2011.636386</u>

- [25] Datta, D., Amaral, A.R. and Figueira, J.R. (2011) Single Row Facility Layout Problem Using a Permutation-Based Genetic Algorithm. *European Journal of Operational Research*, 213, 388-394. <u>https://doi.org/10.1016/j.ejor.2011.03.034</u>
- [26] Ponnambalam, S. and Ramkumar, V. (2001) A Genetic Algorithm for the Design of a Single-Row Layout in Automated Manufacturing Systems. *The International Journal of Advanced Manufacturing Technology*, 18, 512-519.
- [27] Kothari, R. and Ghosh, D. (2014) An Efficient Genetic Algorithm for Single Row Facility Layout. *Optimization Letters*, 8, 679-690. <u>https://doi.org/10.1007/s11590-012-0605-2</u>
- [28] Ku, M.Y., Hu, M.H. and Wang, M.J. (2011) Simulated Annealing Based Parallel Genetic Algorithm for Facility Layout Problem. *International Journal of Production Research*, 49, 1801-1812. <u>https://doi.org/10.1080/00207541003645789</u>
- [29] Utamima, A. and Ou-Yang, C. (2012) Solving Single Row Facility Layout Problem using Extended Artificial Chromosome Genetic Algorithm. *Journal of Technology*, 27, 189-194.
- [30] Büyüksaatçı, S. (2015) Bat Algorithm Application for the Single Row Facility Layout Problem. In: Yang, X.S., Ed., *Recent Advances in Swarm Intelligence and Evolutionary Computation*, Springer, Berlin, 101-120. https://doi.org/10.1007/978-3-319-13826-8_6
- [31] Guan, J. and Lin, G. (2016) Hybridizing Variable Neighborhood Search with Ant Colony Optimization for Solving the Single Row Facility Layout Problem. *European Journal of Operational Research*, 248, 899-909. https://doi.org/10.1016/j.ejor.2015.08.014
- [32] Solimanpur, M., Vrat, P. and Shankar, R. (2005) An Ant Algorithm for the Single Row Layout Problem in Flexible Manufacturing Systems. *Computers & Operations Research*, **32**, 583-598. <u>https://doi.org/10.1016/j.cor.2003.08.005</u>
- [33] Clauss, M., Bernt, M. and Middendorf, M. (2013) A Common Interval Guided ACO Algorithm for Permutation Problems. 2013 *IEEE Symposium on Swarm Intelligence* (*SIS*), Singapore, 16-19 April 2013, 64-71.
- [34] Samarghandi, H. and Eshghi, K. (2009) An Efficient Tabu Algorithm for Solving the Single Row Facility Layout Problem. *European Journal of Operational Research*, 205, 98-105.
- [35] Samarghandi, H. and Eshghi, K. (2010) An Efficient Tabu Algorithm for the Single Row Facility Layout Problem. *European Journal of Operational Research*, 205, 98-105. <u>https://doi.org/10.1016/j.ejor.2009.11.034</u>
- [36] Kothari, R. and Ghosh, D. (2013) Tabu Search for the Single Row Facility Layout Problem Using Exhaustive 2-opt and Insertion Neighborhoods. *European Journal* of Operational Research, 224, 93-100. <u>https://doi.org/10.1016/j.ejor.2012.07.037</u>
- [37] Ahonen, H., Alvarenga, A.G. and Amaral, A. (2014) Simulated Annealing and Tabu Search Approaches for the Corridor Allocation Problem. *European Journal of Operational Research*, 232, 221-233. https://doi.org/10.1016/j.ejor.2013.07.010
- [38] Yu, M., Zuo, X. and Murray, C.C. (2014) A Tabu Search Heuristic for the Single Row Layout Problem with Shared Clearances. *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, Beijing, 6-11 July 2014, 819-825.
- [39] Lenin, N., Kumar, S., Ravindran, D. and Islam, M.N. (2014) A Tabu Search for Multi-Objective Single Row Facility Layout Problem. *Journal of Advanced Manufacturing Systems*, 13, 17-40. <u>https://doi.org/10.1142/S0219686714500024</u>
- [40] Maadi, M., Javidnia, M. and Ghasemi, M. (2016) Applications of Two New Algo-

rithms of Cuckoo Optimization and Forest Optimization for Solving Single Row Facility Layout Problem (SRFLP). *Journal of AI and Data Mining*, **4**, 35-48. <u>https://doi.org/10.5829/idosi.JAIDM.2016.04.01.05</u>

- [41] Ning, X. and Li, P. (2018) A Cross-Entropy Approach to the Single Row Facility Layout Problem. *International Journal of Production Research*, 56, 3781-3794. <u>https://doi.org/10.1080/00207543.2017.1399221</u>
- [42] Samarghandi, H., Taabayan, P. and Jahantigh, F.F. (2010) A Particle Swarm Optimization for the Single Row Facility Layout Problem. *Computers & Industrial En*gineering, 58, 529-534. <u>https://doi.org/10.1016/j.cie.2009.11.015</u>
- [43] Ou-Yang, C. and Utamima, A. (2013) Hybrid Estimation of Distribution Algorithm for Solving Single Row Facility Layout Problem. *Computers & Industrial Engineering*, 66, 95-103. <u>https://doi.org/10.1016/j.cie.2013.05.018</u>
- [44] Kaveh, A. and Safari, H. (2014) Charged System Search Adopted for Solution of Traveling Salesman Problem: An Application to Single-Row Facility Layout Problem. *International Journal of Civil Engineering*, 12, 363-370.
- [45] Kothari, R., and Ghosh, D. (2012) A Lin-Kernighan Heurisitic for Single Row Facility Layout. Technical Report, Indian Institute of Management, India.
- [46] Amaral, A.R. (2008) Enhanced Local Search Applied to the Single-Row Facility Layout Problem. *Brazilian Symposium on Operations Research*, 1638-1647.
- [47] Ghosh, D. (2011) An Exponential Neighbourhood Local Search Algorithm for the Single Row Facility Location Problem. Technical Report, Indian Institute of Management, India.
- [48] Palubeckis, G. (2015) Fast Local Search for Single Row Facility Layout. European Journal of Operational Research, 246, 800-814. https://doi.org/10.1016/j.ejor.2015.05.055
- [49] Kothari, R. and Ghosh, D. (2012) Path Relinking for Single Row Facility Layout. Technical Report, Indian Institute of Management, India.
- [50] Teo, Y.T. and Ponnambalam, S. (2008) A Hybrid ACO/PSO Heuristic to Solve Single Row Layout Problem. *IEEE International Conference on Automation Science and Engineering*, Hong Kong, 23-26 August 2008, 597-602.
- [51] Lian, K., Zhang, C., Gao, L. and Shao, X. (2011) Single Row Facility Layout Problem Using an Imperialist Competitive Algorithm. *Proceedings of the* 41*st International Conference on Computers & Industrial Engineering*, Los Angeles, 23-25 October 2011, 578-586.
- [52] Kamali, H.R. (2015) A Hybrid Simulated Annealing algorithm for Single Row Facility Layout Problem. *International Journal of Industrial Engineering & Production Research*, 26, 243-253.
- [53] Ulutas, B.H. (2013) Assessing the Performance of Two Bioinspired Algorithms to Solve Single-Row Layout Problem. *International Journal of Manufacturing Engineering*, 2013, Article ID: 265904. <u>https://doi.org/10.1155/2013/265904</u>
- [54] Hosseini-Nasab, H. and Emami, L. (2012) A Hybrid Clonal Selection for the Single Row Facility Layout Problem with Unequal Dimensions. *iBusiness*, 4, 216-221. <u>https://doi.org/10.4236/ib.2012.43027</u>
- [55] Palubeckis, G. (2017) Single Row Facility Layout Using Multi-Start Simulated Annealing. *Computers & Industrial Engineering*, **103**, 1-16. <u>https://doi.org/10.1016/j.cie.2016.09.026</u>
- [56] Palubeckis, G. (2015) Fast Simulated Annealing for Single-Row Equidistant Facility Layout. *Applied Mathematics and Computation*, 263, 287-301.

https://doi.org/10.1016/j.amc.2015.04.073

- [57] Amaral, A.R. (2019) A Mixed-Integer Programming Formulation for the Double Row Layout of Machines in Manufacturing Systems. *International Journal of Production Research*, 57, 34-47. <u>https://doi.org/10.1080/00207543.2018.1457811</u>
- [58] Hungerländer, P. (2014) Single-Row Equidistant Facility Layout as a Special Case of Single-Row Facility Layout. *International Journal of Production Research*, **52**, 1257-1268. <u>https://doi.org/10.1080/00207543.2013.828163</u>
- [59] Commander, C.W. (2005) A Survey of the Quadratic Assignment Problem, with Applications. *Morehead Electronic Journal of Applicable Mathematics*, **4**, 1-15.
- [60] Rendl, F. and Wolkowicz, H. (1992) Applications of Parametric Programming and Eigenvalue Maximization to the Quadratic Assignment Problem. *Mathematical Programming*, 53, 63-78. <u>https://doi.org/10.1007/BF01585694</u>
- [61] Rajabioun, R. (2011) Cuckoo Optimization Algorithm. *Applied Soft Computing*, 11, 5508-5518. <u>https://doi.org/10.1016/j.asoc.2011.05.008</u>
- [62] Yang, X.S. and Deb, S. (2009) Cuckoo Search via Lévy Flights. 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC 2009), 210-214.
- [63] Roy, S. and Chaudhuri, S.S. (2013) Cuckoo Search Algorithm Using Lévy Flight: A Review. *International Journal of Modern Education and Computer Science*, 5, 10-15. <u>https://doi.org/10.5815/ijmecs.2013.12.02</u>
- [64] Kothari, R. and Ghosh, D. (2014) A Scatter Search Algorithm for the Single Row Facility Layout Problem. *Journal of Heuristics*, 20, 125-142. https://doi.org/10.1007/s10732-013-9234-x