Computational methods in engineering sciences, vol. 1, no. 4, Winter 2024 Serial no. 4
DOI: 10.22034/cmes.2024.2024013.1021

Solving the Double Row Layout Problem via Barnacles Mating Optimizer

Masoud Seidi [1,*] and Mohammad Zandieh [1]

[1] Faculty of Engineering, Ilam University, Ilam, Iran
* Corresponding Author

Received: 31/12/2023, Revised: 20/01/2024, Accepted: 04/02/2024.

Abstract
One of the most important problems in facility planning is designing the layout of facilities, which has a lot of variety. Most of the facility layout problems are of the NP-hard type, so the method of solving them has a great impact on the quality of the solutions. The meta-heuristic methods of solving the layout problem were investigated in the literature. The most widely used solution method is the Genetic Algorithm. In this research, Barnacles Mating Optimizer (BMO) algorithm was used for the first time to solve a double row layout problem. A practical computer program was designed in which two BMO and GA were used to solve the problem. After solving a common problem by both algorithms, the results showed the superiority of BMO algorithm. The improvement value of the objective function in the DRLP, using BMO, is equal to 48%. As the initial population and the number of iterations increase in BMO, the improvement of the objective function increases. The designed computer program has the ability to solve problems of different sizes.

Keywords
Facility layout, Meta heuristic, BMO, Double row.

1. Introduction

The facility layout problem (FLP) is the placement of facilities in a production system, with the aim of determining the most suitable arrangement in order to achieve some goals or satisfy some criteria under constraints such as shape, size, orientation, and pick-up/drop-off point of the facilities. Based on the type of material handling system (MHS), we will have five basic layouts: single row layout, double row layout, multi row layout, cluster layout and loop layout(Keller and Buscher 2015). Five basic layouts are shown in Figure 1. In single row layout problems (SRLP), the order of placing a certain number of rectangular facilities next to each other and along a line is checked so that the cost of the entire material handling system, which is the sum of the flows and center-to-center distances between all pairs of facilities, is minimized. This type of arrangement can be done in several shapes, such as a straight line, a semicircle, or a U shape (Drira, Pierreval et al. 2007, Samarghandi and Eshghi 2010). Double Row Layout Problem (DRLP) is to determine the arrangement of a number of rectangular facilities with different dimensions on both sides of a line corridor with the aim of minimizing the total cost of moving materials between facilities. Chung and Tanchoco formulated DRLP as a mixed integer programming (MIP) model(Chung and Tanchoco 2010). Besbes et al. developed this model(Besbes, Zolghadri et al. 2020, Besbes, Zolghadri et al. 2021). In this research, the model developed by Besbes et al. has been used. MRLP is related to determining the arrangement of a number of rectangular facilities with different sizes on more than two rows. In this type of problems, each facility can be assigned to each of the given rows(Herrán, Colmenar et al. 2021). The purpose of loop layout problem (LLP) is to find an allocation of n facilities to n predetermined candidate locations in a closed loop network, so that the total cost of the material handling is minimized (Cheng, Gent et al. 1996, Cheng and Gen 1998). The cluster layouts problems (CLP) are related to the determination of the arrangement without considering the limitation of the rows in the problems of single-row, double row, multi-row and loop layouts. In CLP, it is very important that the facilities do not overlap with each other(Niroomand, Hadi-Vencheh et al. 2015, Abdali et al. 2023). In this research, a DRLP from the literature review is stated and its solution will be investigated. In the following, the solution methods are discussed.

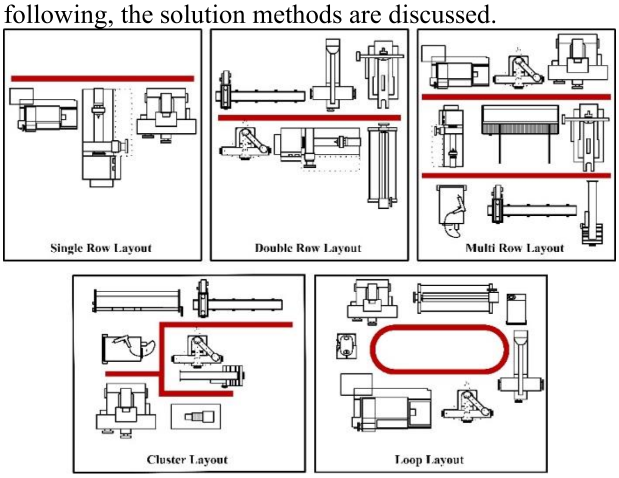


Figure 1. Layout configurations depending on MHS

Researchers have provided different categories of FLP. Various aspects of these categories have been investigated. These aspects are organized as layout drivers: enabling technologies (simulation, computer-integrated manufacturing, expert systems, automation and automatic computation, industry 4.0, etc.), tools and techniques (exact methods, heuristic methods, meta-heuristic method, etc.), manufacturing systems (flexible manufacturing system, agile manufacturing system, prefabricated manufacturing system, etc.) and support factors (safety and risk, quality, ergonomics, etc.)(Al-Zubaidi, Fantoni et al. 2021).

Six decades have passed since Koopmans and Beckman published their seminal paper on FLP modeling. Since then, researchers have made innovations in modeling and solving these models. However, research on many different aspects of FLP, including solution methods, is still at an early stage(Hosseini-Nasab, Fereidouni et al. 2018). Therefore, the use of new optimization methods to solve the FLP is one of the attractive fields for research. Solving the FLP is one of the important issues in facility planning, which has a significant impact on the effectiveness and efficiency of the solutions. Most of the FLP are NP-hard(Drira, Pierreval et al. 2007). Optimization problems that have attracted the attention of meta-heuristic approaches have a lot of variances, from single to multi-objective, continuous to discrete, bounded to unbounded. Solving these problems is not an easy task due to their complex behavior. Meta-heuristic algorithms provide a practical and suitable solution for many of problems. Meta-heuristic algorithms are designed to achieve approximate/optimal solutions in practical execution times for NP-hard optimization problems(Dokeroglu, Sevinc et al. 2019). Due to the NP-hard nature of the layout problem, the meta-heuristic methods used in solving such problems were reviewed and the results are given in Table 1. In Figure 2, it has been tried to show the different dimensions of the layout problem in this research according to previous studies.

Table 1. Summary of literature review of meta-heuristic methods in solving FLP.

Genetic Algorithm
(Tate and Smith 1995), (Kochhar, Foster et al. 1998), (Li and Love 2000), (Lee and Lee 2002), (Lee, Han et al. 2003), (El-Baz 2004), (Wang, Hu et al. 2005), (Sirinaovakul and Limudomsuk 2007), (Diego-Mas, Santamarina-Siurana et al. 2009), (Wong, Fung et al. 2010), (Ripon, Glette et al. 2010), (Datta, Amaral et al. 2011), (Aiello, La Scalia et al. 2012), (García-Hernández, Pierreval et al. 2013), (Kulturel-Konak and Konak 2013), (Mazinani, Abedzadeh et al. 2013), (Pourvaziri and Naderi 2014), (Gonçalves and Resende 2015), (Hasda, Bhattacharjya et al. 2017), (Paes, Pessoa et al. 2017), (Palomo-Romero, Salas-Morera et al. 2017), (Safarzadeh and Koosha 2017), (Phanden, Demir et al. 2018), (Lin and Yingjie 2019), (Wei, Yuan et al. 2019) ,(Dong and Bian 2020), (Besbes, Zolghadri et al. 2020), (Erfani, Ebrahimnejad et al. 2020), (Liu, Liu et al. 2020), (Besbes, Zolghadri et al. 2021), (Besbes, Zolghadri et al. 2021), (Besbes, Zolghadri et al. 2021), (Suh and Choi 2022), (Vineetha and Shiyas 2023)
Simulated Annealing
(Meller and Bozer 1996), (Chwif, Barretto et al. 1998), (McKendall Jr, Shang et al. 2006), (Chae and Peters 2006), (Singh and Sharma 2008), (Şahin and Türkbey 2009), (Şahin, Ertoğral et al. 2010), (Şahin 2011), (Ku, Hu et al. 2011), (Matai 2015), (Kulturel-Konak and Konak 2015), (Grobelny and Michalski 2017), (Palubeckis 2015), (Palubeckis 2017), (Allahyari and Azab 2018), (Tayal and Singh 2018), (Chen and Tiong 2019), (Forghani, Fatemi Ghomi et al. 2020), (Lamba, Kumar et al. 2020), (Khajemahalle, Emami et al. 2021), (Lin, Shen et al. 2023)
Tabu Search
(Chiang and Kouvelis 1996), (Liang and Chao 2008), (Scholz, Petrick et al. 2009), (McKendall Jr and Liu 2012), (Kulturel-Konak 2012), (Kothari and Ghosh 2013), (Bozorgi, Abedzadeh et al. 2015)
Particle Swarm Optimization
(Paul, Asokan et al. 2006), (Önüt, Tuzkaya et al. 2008), (Rezazadeh, Ghazanfari et al. 2009), (Samarghandi, Taabayan et al. 2010), (Ohmori, Yoshimoto et al. 2010), (Kulturel-Konak and Konak 2011), (Jolai, Tavakkoli-Moghaddam et al. 2012), (Hosseini-Nasab and Emami 2013), (Chraibi, Kharraja et al. 2016), (Derakhshan Asl and Wong 2017), (Liu, Zhang et al. 2018), (Guan, Zhang et al. 2019)
Ant Colony Optimization
(Baykasoglu, Dereli et al. 2006), (McKendall and Shang 2006), (Hani, Amodeo et al. 2006), (Hani, Amodeo et al. 2007), (Lam, Ning et al. 2007), (Ramkumar, Ponnambalam et al. 2009), (Komarudin and Wong 2010), (Kulturel-Konak and Konak 2011), (Yu-Hsin Chen 2013), (Guan and Lin 2016), (Liu and Liu 2019), (Arnaout, ElKhoury et al. 2020), (Zouein and Kattan 2022)
Bee Colony Algorithm
(Cheng and Lien 2012), (Yahya and Saka 2014), (Lien and Cheng 2014), (Saravanan and Arulkumar 2015), (Samanta, Philip et al. 2018), (Nguyen 2021)
Cuckoo Search Algorithm
(Rehman, Ali et al. 2016), (Kang, Kim et al. 2018)

Computational methods in engineering sciences, vol. 1, no. 4, Winter 2024 Serial no. 4
DOI: 10.22034/cmes.2024.2024013.1021

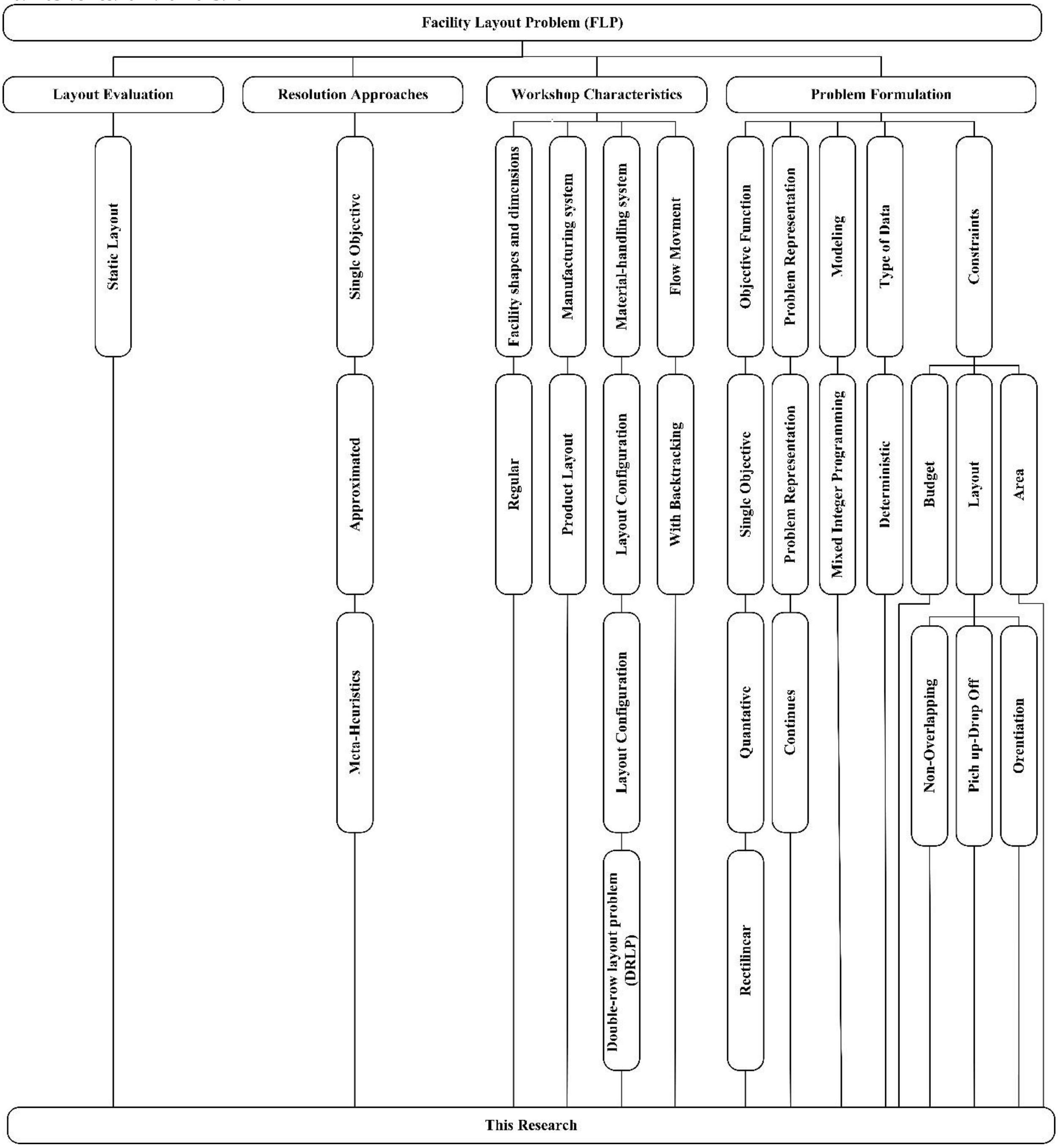


Figure 2. Problem aspects

Meta-heuristic algorithms can be classified into four groups: evolutionary computation, swarm intelligence, inspired human and inspired physics(Besbes, Zolghadri et al. 2021). Evolutionary computation algorithms mimic the evolutionary process in nature and follow the principle that children inherit characteristics from parents. The set of candidate solutions is iteratively improved until the final conditions are met. Therefore, during iterations or generations, the probability of achieving better results near the global optimum increases, even if there is no guarantee of achieving a very accurate approximation of the global optimum. Among the popular algorithms in this category, we can mention genetic algorithm.

As can be seen in Table 1, genetic algorithm has been used more than other algorithms in solving the layout problem. In this research, for the first time, a new evolutionary algorithm called Barnacles Mating Optimizer (BMO)(Sulaiman, Mustaffa et al. 2020) has been used to solve the DRFLP. Then the answers obtained from BMO have been compared with the answers obtained from GA(Besbes, Zolghadri et al. 2021). The results show improvement in BMO's solutions. It should be mentioned that BMO and GA is coded as an application program in MATLAB and can be used in problems of different dimensions.

2. Statement of the two-dimensional facility layout problem

The purpose of the arrangement-facilities problem is to determine how to place different machines next to each other in such a way that the cost function is minimized. The place where these machines are placed has a rectangular shape with fixed length (L) and width (W). Aisle and obstacles are included in this problem. For each machine, input and output points and four basic

DOI: 10.22034/cmes.2024.2024013.1021

orientations (See Figure 3) are considered. By considering the pick-up (input) and drop-off (output) of machines, the rotation of facilities can affect the cost of moving materials. Machines should not overlap each other. Facilities should not be placed on aisle and obstacles. The parameters, decision variables, objective function and constraints are listed below. Problem parameters are shown in Figure 4.

Parameters:

N:	Number of machines	L_{oi}:	Length of obstacle i
L:	Length of the workshop	w_{oi}:	Width of obstacle i
W:	Width of the workshop	x_{oi}:	The x-coordinate of obstacle i
M:	Number of obstacles	y_{oi}:	The y-coordinate of obstacle i
L_i:	Length of machine i in original orientation	y_{upper}:	Vertical coordinates of the upper side of aisle
W_i:	Width of machine i in original orientation	y_{lower}:	Vertical dimension of the lower side of aisle
(x_{io}^i, y_{io}^i):	Coordinates of the pick-up of machine i in relation to its centroid	f_{ij}:	Material flow from machin i to machin j
(x_{oo}^i, y_{oo}^i):	Coordinates of the drop-off of machine i in relation to its centroid.	c_{ij}:	Unit cost for transportation between two machines

See Figure 4.

Decision variables:

d_{ij}	Distance between two machines	(x_{out}^i, y_{out}^i):	Coordinates of the drop-off of machine i in each orientation
r:	Orientation of machines (r=1,2,3,4)	Z_{ij}^x:	=1 if machine i is strictly to the right of machine j 0 otherwise
θ:	Angle of machine	Z_{ij}^y:	=1 if machine i is strictly above machine j 0 otherwise
(x_i, y_i):	Coordinates of the centre of machine i	Z_{iv}^x:	=1 if machine i is strictly to the right of obstacle v 0 otherwise
(x_{in}^i, y_{in}^i):	Coordinates of the pick-up of machine i in each orientation	Z_{iv}^y:	=1 if machine i is strictly above obstacle v 0 otherwise

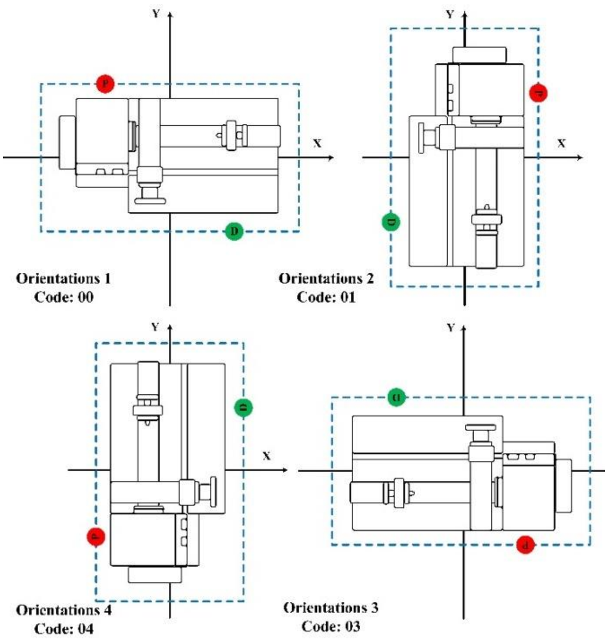


Figure 3. Orientations of a machine.

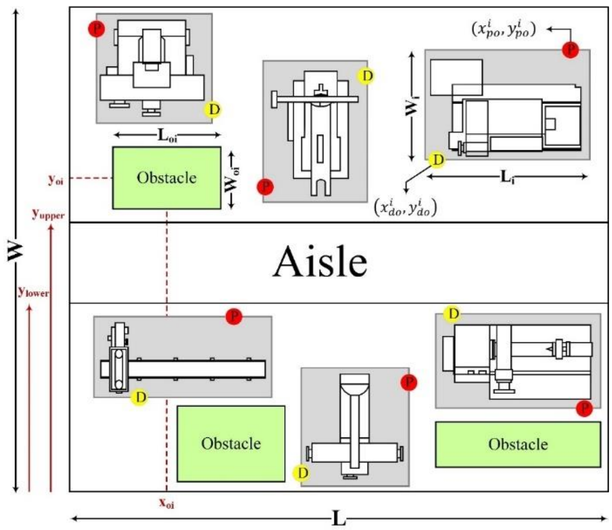


Figure 4. Problem parameters

DOI: 10.22034/cmes.2024.2024013.1021

Objective function

$$MHC = \sum_{i=1}^{N}\sum_{j=1}^{N} f_{ij} * c_{ij} * d_{ij} \qquad (1)$$

Constraints

$$\frac{l_i}{2} \le x_i \le L - \frac{l_i}{2} \qquad (2)$$

$$\frac{w_i}{2} \le y_i \le W - \frac{w_i}{2} \qquad (3)$$

$$\theta_i = {}^{\pi}/_{2} * r_i \qquad (4)$$

$$l_i = |\cos(\theta) * l_i| + |\sin(\theta) * w_i| \qquad (5)$$

$$w_i = |\cos(\theta) * w| + |\sin(\theta) * l_i| \qquad (6)$$

$$x_{in}^{i} = x_i + \left(\cos(\theta) * x_{io}^{i}\right) - \left(\sin(\theta) * y_{io}^{i}\right) \qquad (7)$$

$$y_{in}^{i} = y_i + \left(\cos(\theta) * y_{io}^{i}\right) + \left(\sin(\theta) * x_{io}^{i}\right) \qquad (8)$$

$$x_{out}^{i} = x_i + (\cos(\theta) * x_{oo}^{i}) - (\sin(\theta) * y_{oo}^{i}) \qquad (9)$$

$$y_{out}^{i} = y_i + (\cos(\theta) * y_{oo}^{i}) + (\sin(\theta) * x_{oo}^{i}) \qquad (10)$$

$$\left(x_j - x_i\right) > Z_{ij}^{x}\left(\frac{l_i}{2} - \frac{l_j}{2}\right) \qquad (11)$$

$$\left(x_i - x_j\right) > (1 - Z_{ij}^{x})\left(\frac{l_i}{2} - \frac{l_j}{2}\right) \qquad (12)$$

$$\left(y_j - y_i\right) > Z_{ij}^{y}\left(\frac{w_i}{2} - \frac{w_j}{2}\right) \qquad (13)$$

$$\left(y_i - y_j\right) > (1 - Z_{ij}^{y})\left(\frac{w_i}{2} - \frac{w_j}{2}\right) \qquad (14)$$

$$(x_v - x_{oi}) > Z_{iv}^{x}\left(\frac{l_{oi}}{2} - \frac{l_v}{2}\right) \qquad (15)$$

$$(x_{oi} - x_v) > (1 - Z_{iv}^{x})\left(\frac{l_{oi}}{2} - \frac{l_v}{2}\right) \qquad (16)$$

$$(y_v - y_{oi}) > Z_{iv}^{y}\left(\frac{w_{oi}}{2} - \frac{w_v}{2}\right) \qquad (17)$$

$$(y_{oi} - y_v) > (1 - Z_{iv}^{y})\left(\frac{w_{oi}}{2} - \frac{w_v}{2}\right) \qquad (18)$$

$$\left(\left(y_i + \frac{w_i}{2}\right) - y_{lower}\right)\left(y_{upper} - \left(y_i - \frac{w_i}{2}\right)\right) > 0 \qquad (19)$$

$$\left(\left(x_i + \frac{l_i}{2}\right) - x_{lower}\right)\left(x_{upper} - \left(x_i - \frac{l_i}{2}\right)\right) > 0 \qquad (20)$$

$Z_{ij}^{x}, Z_{ij}^{y}, Z_{iv}^{x}, Z_{iv}^{y}$
$\in$ Error! Bookmark not defined.
$All\ other\ variables \ge 0$

Equation (1) expresses the objective function of the problem. The goal is to minimize the total cost of moving materials between machines. Constraint sets (2) and (3) verify that the machines are placed within the workshop boundaries. Equation (4) determines the orientation of the machine. The set of constraints (5)-(6) provide the length and width of each machine according to its direction. Equations (7)-(10) determine the pick-up and drop-off coordinates of each machine according to its direction. The set of constraints (11)-(14) prevents overlapping of machines. Constraints (15)-(18) prevent overlapping between machines and obstacles. Constraints (19) and (20) confirm that no machines are arranged in the aisle. At the end, the range of different variables is defined. In the next section, Barnacles Mating Optimizer (BMO), which is used to solve the problem, is explained.

3. Barnacles Mating Optimizer

Barnacles Mating Optimizer (BMO) is a new nature-inspired optimization algorithm that is presented to solve optimization problems. BMO mimics the mating behavior of barnacles in nature to solve optimization problems. BMO can be classified as evolutionary algorithms. Barnacles are microorganisms that have existed since the Jurassic period. Barnacles can swim when they are born, and when they reach the adult stage, they attach themselves to objects in the water and develop a shell. After growing up, this animal becomes hard-skinned and usually sticks to different water surfaces, including rocks, bodies submerged in water, water structures, and even the hulls of ships and boats. The most interesting fact about barnacles is their long penis relative to their body (seven to eight times their body length). Variation in penile accessibility may play an important role in determining mating group size and local mate competition. Figure 5, shows the life cycle of barnacles. The larval development of barnacles includes six naupliar instars and a cyprid instar before settling. After settling on a surface, a cyprid will develop into an adult. A includes three main processes: 1) Initialization process, 2) Selection process and 3) Mating process.

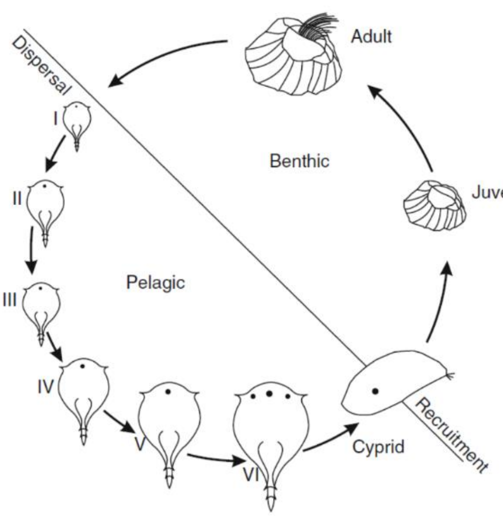


Figure 5. Life cycle of acorn/rock barnacle (Anil, Desai et al. 2012).

3.1. Initialization process

In BMO, the solution is assumed to be barnacles whose population vector can be expressed as:

$$X = \begin{bmatrix} x_1^1 & \cdots & x_1^N \\ \vdots & \ddots & \vdots \\ x_n^1 & \cdots & x_n^N \end{bmatrix} \qquad (21)$$

Where N is the number of decision variables and n is the population or number of barnacles. The decision variables in equation (21) are limited to the upper and lower boundaries of the problem, which should be considered as equations (22) and (23):

$$ub = [ub_1, \ldots, ub_i] \qquad (22)$$

$$lb = [lb_1, \ldots, lb_i] \qquad (23)$$

ub and lb represent the upper and lower bounds of the i^{th} variable. The evaluation of the vector X is done first and the sorting process is performed to place the best solution (the best solution so far) at the top of the vector X.

DOI: 10.22034/cmes.2024.2024013.1021

3.2. Selection process

BMO uses a different approach for mating selection compared to other evolutionary algorithms such as genetic algorithm, differential evolution algorithm, etc., because the selection of two barnacles is based on their penis length(pl). pl mimics the selection process of barnacle behavior. These assumptions are considered:

1) The selection process is random, but limited to the length of the penis pl.

2) Each barnacle may help spread its own sperm as well as receive sperm from another barnacle. 3) Each barnacle can only be fertilized by one barnacle at a time.

4) If at a certain point the selection process selects the same barnacle, at that point no new generation will be produced.

5) If selection is set higher than pl in a given iteration, the sperm casting process occurs.

In order to select parents, equations 24 and 25 are used:

$$barnacle\_d = rand\ perm\ (N) \quad (24)$$

$$barnacle\_m = rand\ perm\ (N) \quad (25)$$

barnacle_d and *barnacle_m* are the parents to be mated and N is the population number.

3.3. Mating process

Suppose that the maximum length of the barnacles' penis is seven times larger than its size (pl = 7), so in a given iteration, barnacle number 1 can only mate with one of barnacles number 2 to 7. If barnacle #1 chooses barnacle #8, it is overkill, so the normal mating process does not occur (See Figure 6). Therefore, reproduction is done in two ways: normal (Equation 26) and casting (Equation 27).

$$x_i^{N\_new} = p x_{barnacle\_d}^N - q x_{barnacle\_m}^N \quad (26)$$

$$x_i^{N\_new} = rand\ () x_{barnacle\_m}^N \quad (27)$$

Where:

p: Pseudo random number distributed between [0, 1]

q: $1 - p$

$x_{barnacle\_d}^N$: Dad barnacle

$x_{barnacle\_m}^N$: Mum barnacle

$rand\ ()$: Random number between [0, 1]

The Hardy–Weinberg principle will be used in the offspring generation of BMO(Sulaiman, Mustaffa et al. 2020).

The most important issue in BMO algorithm is how the algorithm works in making decisions about exploitation and exploration processes. Over-exploitation may lead to getting stuck in the local optimum, while over-exploration may lead to not achieving the global optimum. Therefore, pl regulation plays an important role in determining exploitation and exploration processes.

First, the fitness of each of the barnacles is calculated and placed in the *BarnaclesFitness* array, then it is stored in the sorted_fitness variable in ascending order based on the obtained fitness, which is in terms of MHC. The value of pl is set between 1 and 7. As long as the current iterations condition is less than *Maximum_iterations*, this loop continues to work as follows:

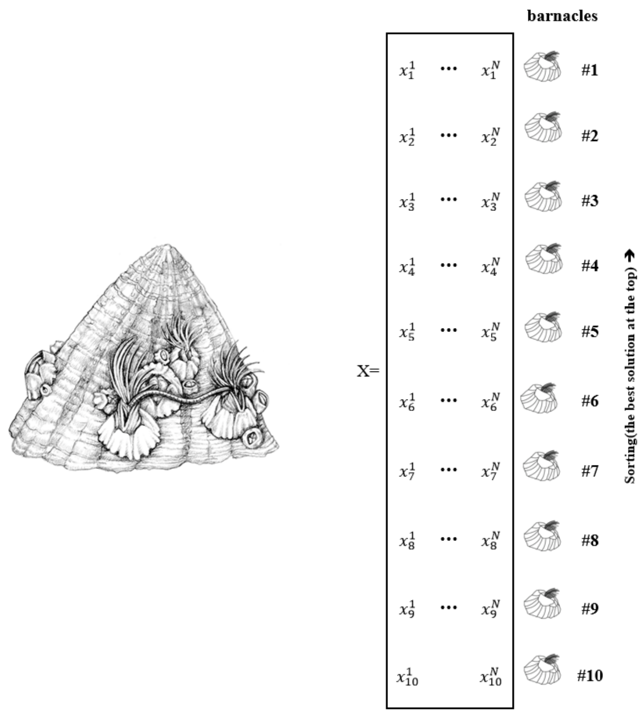


Figure 6) Selection of mating process of BMO

The father and mother barnacles are randomly assigned in *Dad_Barnacle* and *Mom_Barnacle* according to equation 24 and 25.

According to the obtained random values, the position is set in two *tempPosition* variables for the parent barnacle named *Dad_BarnaclePositions* and *Mom_BarnaclePosition*. Distance between the parent barnacle is calculated and set in the *DistanceSelection* variable then the pair of barnacles whose distance is greater than pl is stored in *OverThanPLs*. For the barnacles whose distance is less than pl, first, randomly between 0 and 1, one in the variable p and q is the result of p-1, which expresses the percentage of inheritance of the characteristic of parents.

For each of *Dad_BarnaclePositions(i)* and *Mom_BarnaclePositions(i)*, the variables are multiplied by p to equation 26 for the father's barnacle and q for the mother's barnacle.

According to the father barnacle and the new mother barnacle, a new child is produced, which is stored in *Barnaclesoffspring(i)*.If there is a variable in the *OverThanPLs* array, according to equation 27, the X, Y, Orientation variables are updated and stored in the *Barnaclesoffspring* array.

For each of the newly produced children, the overlap condition is first checked. If there is an overlap, the barnacles are placed again and their fit is calculated, and the *Barnacleoffspring* is stored in *BarnaclesPositions*, and the iterations go one-step forward. Finally, the output is displayed based on the number of best answers. The flowchart and pseudocode used in this research are shown in Figure 7 and Figure 8.

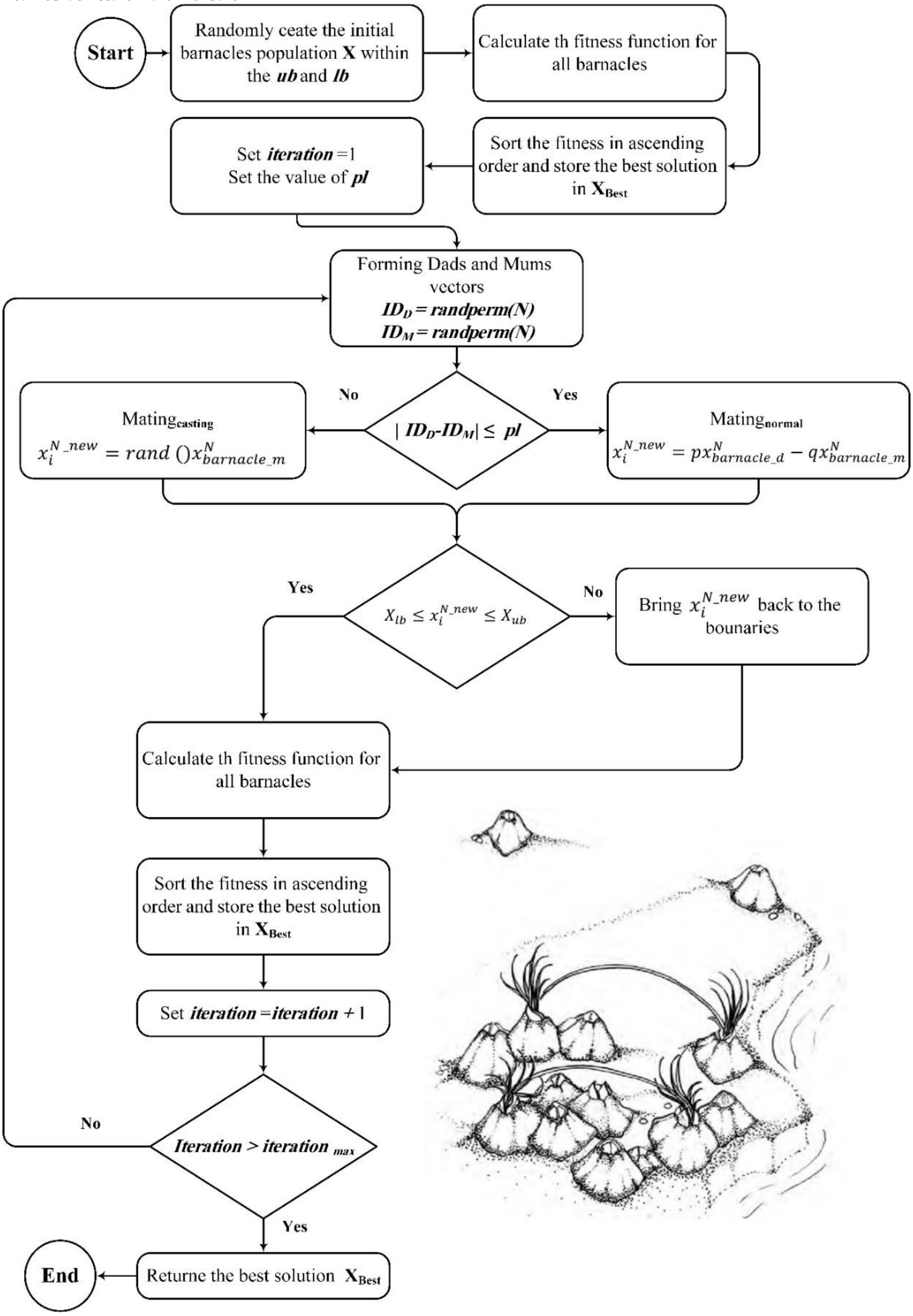


Figure 7. BMO flowchart according to (Selim, Kamel et al. 2022).

DOI: 10.22034/cmes.2024.2024013.1021

```
BarnaclesFitness = Calculate the fitness of each barnacle by Method ( Fitness )
 sorted_fitness = Sorting to the Lowest MHC by BarnaclesFitness
pl  = Set the value of Pines Length
while (iterations < Maximum_iterations)
        Dad_Barnacle = random select by eq (24)
       Mom_Barnacle = random select by eq (25)
        Put BarnaclesPositions into tempPosition by Dad_Barnacle & Mom_Barnacle
        DistanceSelection = callculate distance between Dad_Barnacle & Mom_Barnacle
        OverThanPLs = find DistanceSelection > pl
         If  DistanceSelection less than pl
                for i to Size(BarnaclesPositions)
                        p = random();q = 1- p; // eq 26
                        update X , Y , Orientation of Dad_BarnaclePositions(i) based on Eq. (26) by P
                        do same for the Mom_BarnaclePosition  but Instead P with q
                        Barnaclesoffspring = Dad_BarnaclePositions(i) + Mom_BarnaclePositions(i)
                end for
         else if any exist in OverThanPLs
                for each OverThanPLs
                        p = random();
                        update X , Y , Orientation of Mom_BarnaclePositions (i) based on Eq. (27) by P
                     Barnaclesoffspring(Mom_Barnacles(i)) =   Mom_BarnaclePositions(i)
                end
         end if
for each Barnaclesoffspring
        if IsOverLapHappend(Barnacleoffspring) equal true // means that barnacle goes out side and must
       bring back
                Barnacleoffspring=CreateCar(MachineNumber,LengthWorkshop,WidthWorkshop,L,W,L
              oC,WoC,XoC,YoC);
        end
        BarnaclesFitness = calculate fitness (Barnacleoffspring);
        BarnaclesPositions = Barnacleoffspring;
end
         iterations = iterations + ١
end while
set bestsolution find first(ShowBestAnswer) in BarnaclesPositions
Draw Map(bestsolution)
```

Figure 8. BMO pseudo-code.

DOI: 10.22034/cmes.2024.2024013.1021

Table 2. Problem data (Besbes, Zolghadri et al. 2021).

Facilities	1	2	3	4	5	6	7	8
1	0	50	45	20	0	19	46	15
2	28	0	13	15	24	27	25	48
3	13	28	0	0	31	12	0	49
4	0	14	20	0	26	47	41	33
5	47	49	42	33	0	48	25	12
6	16	10	27	32	19	0	19	0
7	43	41	47	15	15	30	0	24
8	32	0	17	44	17	23	13	0

Material flow from machine i to machine j (f_{ij})

Facilities	1	2	3	4	5	6	7	8
x_{io}^{i}	1	-2	-2	2	-1	-1	2	-1
y_{io}^{i}	-2	2	2	-2	-2	-3	1	2
x_{oo}^{i}	2	-4	-2	2	2	-2	-2	1
y_{oo}^{i}	-1	-1	-2	1	0	3	1	-2

P and D point of machines in original orientation

Dimensions	1	2	3	4	5	6	7	8
L_i	4	8	4	4	4	4	4	4
W_i	4	4	4	4	4	6	4	4

Machin dimensions in original orientation

Coordinates (x,y)	Width	Length
(12,3)	2	4
(7,18)	4	2

Obstacles data

4. Numerical Results

In this section, in order to check the effectiveness of the barnacle method, we will re-solve a problem raised by Besbes. The problem data is shown in table 2.

4.1. Main pseudo-code

A First, the problem values are initialized and then the program is executed with the existing algorithms and the same inputs. The main pseudo code is shown in Figure 9.

```
Initial values;
Algorithms = {'GA (A*)','BMO'};
For each algorithm in algorithms
        Run (algorithm)
For
```

Figure 9. Main pseudo-code.

As seen in figure 9, two algorithms have been used to solve the problem. BMO method has been used for the first time in solving this problem, and the results will be compared with the GA(A*) algorithm proposed by Besbes for validation(Besbes, Zolghadri et al. 2020).

4.3. Fitness function

First, the overlap between machines, obstacles and aisle is checked. If there is no overlap, the cost of moving materials is calculated through equation 1. In the fitting function of the two-dimensional facility layout problem, the following formula was used when calculating the Theta angle:

$$\theta = (\frac{\pi}{2}) \times Orientation(r_i) \qquad (28)$$

The pseudo-code of the fitting function is as Figure 10.

4.2. Problem space modeling

In order to model machines, asile and obstacles, a two-dimensional array has been used, and initially all the houses of the array are equal to 1, as shown in Figure 11.

Then the machines, aisle and obstacles are added according to the length and width and the central point in the problem space and the space of that region is changed to 0 value. For example, by placing a machine in the problem, the value of one changed to zero (See Figure 12).

According to the equations 5 to 10, the length and width and the input and output points are calculated, and according to the equation 1, MCH is obtained. Before the fitting function is called, first the overlap is checked and then the fit is calculated in such a way that if a chromosome is not included in the problem according to equations 11 to 20, its fit is not calculated. Pseudo-code for building a problem space is as Figure 13.

4.3. Overlap control

First, obstacles and aisle are allocated one by one in the problem space, if a machine is added to the problem space and does not overlap with any of them, it is placed in the problem space, otherwise, an overlap has occurred and this chromosome is invalid. If all machines, obstacles and aisle are placed in the problem space, this chromosome will be valid. Pseudo-code for overlap control is as Figure 14. First, we solve the problem using the barnacle method and solve various problems to adjust the PL parameter. The results are shown in Table 3.

DOI: 10.22034/cmes.2024.2024013.1021

```
IsOverLap = IsOverLapHappend(chromosome);
IF IsOverLapHappend = false then
    For i to Size(chromosome)
             Theta(i) = set by eq 4
            Xin(i) = set by eq 7
            Yin(i) = set by eq 8
            Xout(i) = set by eq 9
            Yout(i) = set by eq 10
            L(i) = set by eq 5
            W(i) =  set by eq 6
           Add car into map and change 1 cell to 0
    // Calculate MHC by Eq  (1)
    For i to Size(chromosome)
             For j to Size(chromosome)
                      Distance = set Calculate distance between Input(i) to Output(j)
                      MHC = MHC +  f(i,j)*C(i,j)*length(Distance);
Else if
            MHC = 0
```

Figure 10. Fitness function pseudo-code.

1	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
1	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
1	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
1	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
1	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
1	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
1	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
1	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
1	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
1	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**

Figure 11. Initialization of the problem space.

1	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
1	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
1	**0**	**0**	**0**	**1**	**1**	**1**	**1**	**1**	**1**
1	**0**	**0**	**0**	**1**	**1**	**1**	**1**	**1**	**1**
1	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
1	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
1	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
1	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
1	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
1	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**

Figure 12. Placing a machine in the problem space.

DOI: 10.22034/cmes.2024.2024013.1021

```
Map = set array with 1
Add obstacle  and replace location into map and replace 1 to 0
Add aisle and replace location into map and replace 1 to 0
For i in Chromosome
    Chromosome(i) = set randomly x, y , Orientation
    If Chromosome(i) CanNot Replace into Map then
       Repeat Again
  Else
     i = i +1
end
```

Figure 13. Problem space pseudo-code.

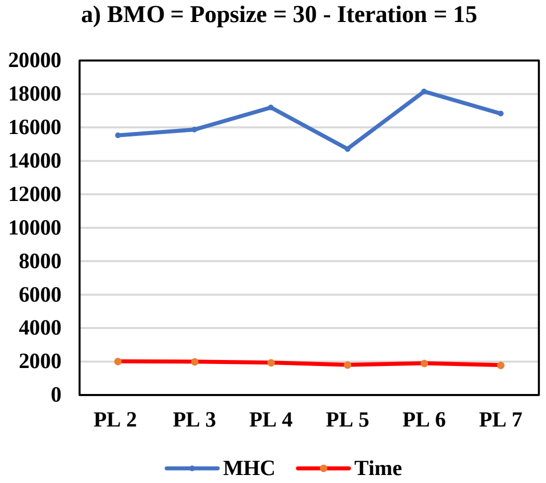
Table 3. Results of BMO with several population and iteration.

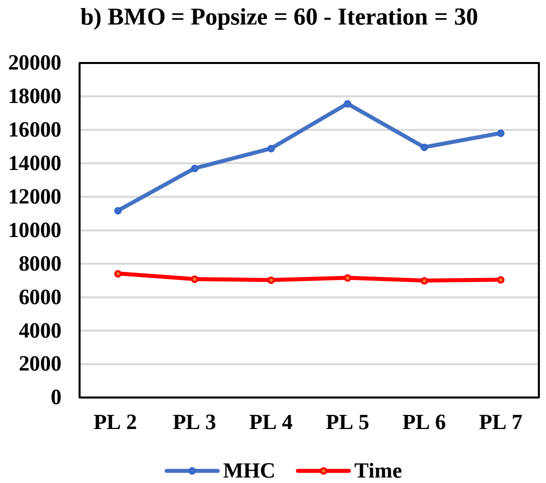
Population	Iteration	MHC	Time	pl	Population	Iteration	MHC	Time	pl
30	15	15533	2018	2	120	60	14250	30285	2
30	15	15875	1999	3	120	60	14397	29218	3
30	15	17195	1947	4	120	60	11298	30627	4
30	15	14717	1801	5	120	60	14577	31104	5
30	15	18151	1895	6	120	60	11903	30773	6
30	15	16828	1791	7	120	60	14997	31810	7
60	30	11174	7417	2	240	120	14719	49849	2
60	30	13703	7073	3	240	120	15110	51854	3
60	30	14888	7030	4	240	120	9626	51001	4
60	30	17572	7152	5	240	120	10126	50929	5
60	30	14961	6995	6	240	120	9576	51554	6
60	30	15815	7051	7	240	120	10453	52007	7

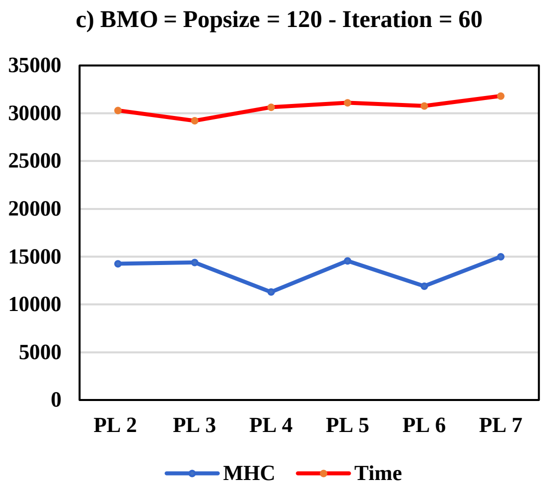
```
Map = Creating an array of "Length Workshop" and "Width Workshop" width
Result = Set to False
For i In Size(obstacle_positionsX)
 Map obstacle into map array
End
For i In Size(chromosome)
        set Theta by Orientation by Eq 4
        Set NewWidth & NewLength by Eq 5 & Eq 6
         if x, y, NewWidth, NewLength in WorkSpace by Eq 11 & Eq 22 then
            Result = set True
         end
end
```

Figure 14. Overlap control pseudo-code.

DOI: 10.22034/cmes.2024.2024013.1021







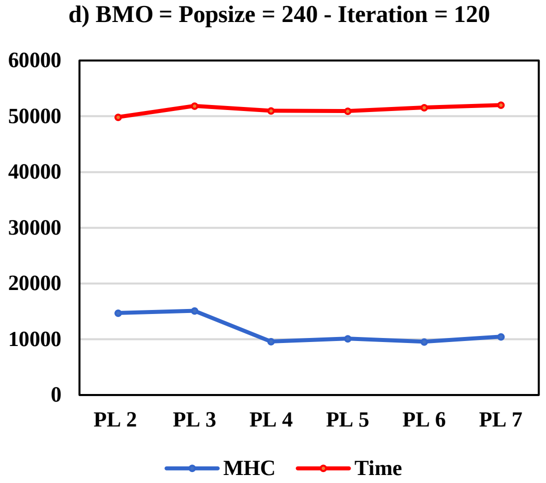


Figure 15. MCH and solution time for different pl.

In Figure 15, the results related to the effect of different pl parameter changes on MHC and the solution time in Barnacle are shown. As can be seen, the influence of pl changes on the solution time is negligible. In population 30 and iteration 15, pl = 5 creates the lowest value of the MHC. According to population 60 and iteration 15, the lowest value of the objective function occurs for pl=2. By increasing the population to 30 and iteration to 15, the best value of pl is equal to 4, which minimizes MHC. In part (d) of figure 15, population equal to 240 and iteration equal to 120 are considered, and as it can be seen, per pl equal to 4, the minimum of the objective function occurs. According to the obtained results, the value of the pl in the computer program was set as 4 and the results were compared with GA (A*) algorithm.

In Figure 16, the output of the computer program for two coded algorithms is shown. As can be seen, the improvement in the value of the objective function is evident in BMO, and BMO creates more suitable solutions than Algorithm GA(A*). In Figure 16, part (a), the answer of the GA(A*) algorithm with population 30 and iteration 30 is shown, and in part (b), the answer of the BMO algorithm with the same population and iteration is shown. BMO has improved 16% of the MHC. In part (c) of Figure 16, the layout of facilities per population 40 and iteration 100 is shown by GA(A*) and in part (d), there is a solution related to BMO. The results show a 36% improvement in the MHC. As the number of population and iterations increase, the performance of BMO is much better than GA(A*) in reducing MHC.

DOI: 10.22034/cmes.2024.2024013.1021

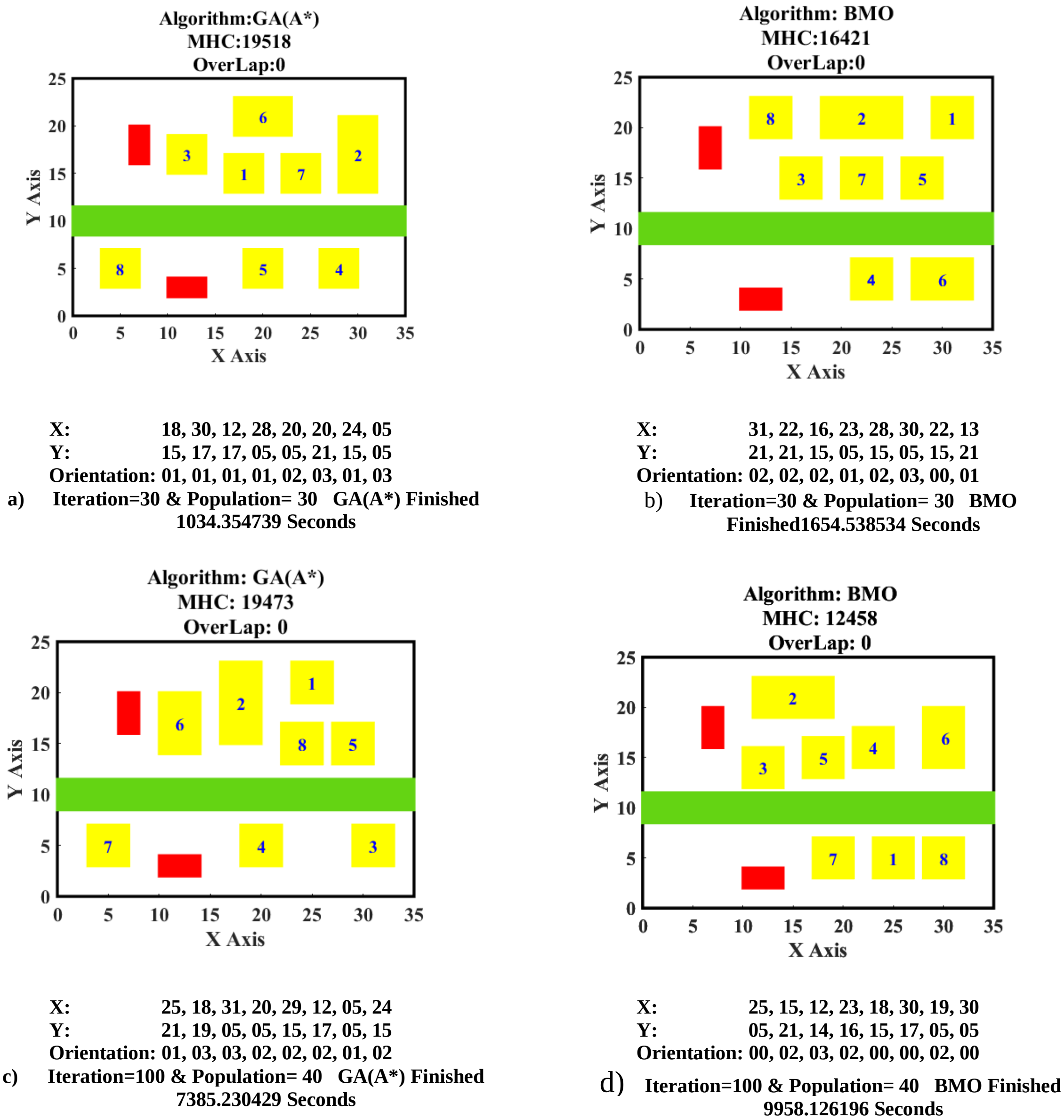


Figure 16. Computer program output.

5. Conclusion

In this research, the two-dimensional facility layout problem was solved using BMO for the first time. A practical computer program was designed that solves the facility layout problem with two meta-heuristic methods: BMO and GA(A*). In order to check the performance of the BMO method, a problem that was solved in the previous research using the GA(A*) was re-solved in this study using the BMO method. In the past research, the combination of GA(A*) parameters that gives the best layout is as follows: Population size: 230, Crossover probability: 0.15, Mutation probability: 0.455, Number of iterations: 120, Selection operator: Roulette wheel selection operator, MHC: 18404. The coded BMO algorithm in this research was able to achieve an MHC value of 9626 with a population of 240 and repetitions of 120. In other words, BMO was able to improve almost 48% of the MHC function. Therefore, the results show that the BMO meta-heuristic method is strongly recommended for solving two-dimensional DRLPs.

References

Abdali A, Monjezi V. Machine Learning-based Flexible Link Robot Control. Computational methods in engineering sciences. 2023 Aug 23;1(2):12-7.

Adrian, A. M., et al. (2015). "A comparative study of GA, PSO and ACO for solving construction site layout optimization." KSCE Journal of Civil Engineering **19**: 520-527.

Aiello, G., et al. (2012). "A multi objective genetic algorithm for the facility layout problem based upon slicing structure encoding." Expert Systems with Applications **39**(12): 10352-10358.

Al-Zubaidi, S. Q. D., et al. (2021). "Analysis of drivers for solving facility layout problems: A Literature review." Journal of Industrial Information Integration **21**: 100187.

Allahyari, M. Z. and A. Azab (2018). "Mathematical modeling and multi-start search simulated annealing for unequal-area facility layout problem." Expert Systems with Applications **91**: 46-62.

Anil, A. C., et al. (2012). Barnacles and Their Significance in Biofouling. Operational and Environmental Consequences of Large Industrial Cooling Water Systems. S. Rajagopal, H. A. Jenner and V. P. Venugopalan. Boston, MA, Springer US**:** 65-93.

Arnaout, J.-P., et al. (2020). "Solving the multiple level warehouse layout problem using ant colony optimization." Operational Research **20**(1): 473-490.

Baykasoglu, A., et al. (2006). "An ant colony algorithm for solving budget constrained and unconstrained dynamic facility layout problems." Omega **34**(4): 385-396.

Besbes, M., et al. (2021). "A method to solve 2D Facility Layout Problem with equipment inputs/outputs constraints using meta-heuristics algorithms." Procedia CIRP **104**: 1698-1703.

Besbes, M., et al. (2020). "A methodology for solving facility layout problem considering barriers: genetic algorithm coupled with A* search." Journal of Intelligent Manufacturing **31**(3): 615-640.

Besbes, M., et al. (2021). "3D facility layout problem." Journal of Intelligent Manufacturing **32**(4): 1065-1090.

Bozorgi, N., et al. (2015). "Tabu search heuristic for efficiency of dynamic facility layout problem." The International Journal of Advanced Manufacturing Technology **77**: 689-703.

Chae, J. and B. A. Peters (2006). "A simulated annealing algorithm based on a closed loop layout for facility layout design in flexible manufacturing systems." International Journal of Production Research **44**(13): 2561-2572.

Chen, C. and L. K. Tiong (2019). "Using queuing theory and simulated annealing to design the facility layout in an AGV-based modular manufacturing system." International Journal of Production Research **57**(17): 5538-5555.

Cheng, M.-Y. and L.-C. Lien (2012). "A hybrid AI-based particle bee algorithm for facility layout optimization." Engineering with Computers **28**(1): 57-69.

Cheng, R. and M. Gen (1998). "Loop layout design problem in flexible manufacturing systems using genetic algorithms." Computers & Industrial Engineering **34**(1): 53-61.

Cheng, R., et al. (1996). "Genetic algorithms for designing loop layout manufacturing systems." Computers & Industrial Engineering **31**(3): 587-591.

Chiang, W.-C. and P. Kouvelis (1996). "An improved tabu search heuristic for solving facility layout design problems." International Journal of Production Research **34**(9): 2565-2585.

Chraibi, A., et al. (2016). "A particle swarm algorithm for solving the multi-objective operating theater layout problem." IFAC-PapersOnLine **49**(12): 1169-1174.

Chung, J. and J. M. A. Tanchoco (2010). "The double row layout problem." International Journal of Production Research **48**(3): 709-727.

Chwif, L., et al. (1998). "A solution to the facility layout problem using simulated annealing." Computers in Industry **36**(1-2): 125-132.

Datta, D., et al. (2011). "Single row facility layout problem using a permutation-based genetic algorithm." European Journal of Operational Research **213**(2): 388-394.

Derakhshan Asl, A. and K. Y. Wong (2017). "Solving unequal-area static and dynamic facility layout problems using modified particle swarm optimization." Journal of Intelligent Manufacturing **28**: 1317-1336.

Diego-Mas, J. A., et al. (2009). "Solving facility layout problems with strict geometric constraints using a two-phase genetic algorithm." International Journal of Production Research **47**(6): 1679-1693.

Dokeroglu, T., et al. (2019). "A survey on new generation metaheuristic algorithms." Computers & Industrial Engineering **137**: 106040.

Dong, Z.-r. and X.-y. Bian (2020). "Ship Pipe Route Design Using Improved A* Algorithm and Genetic Algorithm." IEEE Access **8**: 153273-153296.

Drira, A., et al. (2007). "Facility layout problems: A survey." Annual reviews in control **31**(2): 255-267.

El-Baz, M. A. (2004). "A genetic algorithm for facility layout problems of different manufacturing environments." Computers & Industrial Engineering **47**(2): 233-246.

Erfani, B., et al. (2020). "An integrated dynamic facility layout and job shop scheduling problem: A hybrid NSGA-II and local search algorithm." Journal of Industrial & Management Optimization **16**(4).

Forghani, K., et al. (2020). "Solving an integrated cell formation and group layout problem using a simulated annealing enhanced by linear programming." Soft Computing **24**(15): 11621-11639.

García-Hernández, L., et al. (2013). "Handling qualitative aspects in unequal area facility layout problem: an interactive genetic algorithm." Applied Soft Computing **13**(4): 1718-1727.

Gonçalves, J. F. and M. G. C. Resende (2015). "A biased random-key genetic algorithm for the unequal area facility layout problem." European Journal of Operational Research **246**(1): 86-107.

Grobelny, J. and R. Michalski (2017). "A novel version of simulated annealing based on linguistic patterns for solving facility layout problems." Knowledge-Based Systems **124**: 55-69.

Guan, C., et al. (2019). "Multi-objective particle swarm optimization for multi-workshop facility layout problem." Journal of Manufacturing Systems **53**: 32-48.

Guan, J. and G. Lin (2016). "Hybridizing variable neighborhood search with ant colony optimization for solving the single row facility layout problem." European Journal of Operational Research **248**(3): 899-909.

Hani, Y., et al. (2006). "A HYBRID ANT COLONY OPTIMIZATION ALGORITHM FOR THE LAYOUT PROBLEM." IFAC Proceedings Volumes **39**(3): 425-430.

Hani, Y., et al. (2007). "Ant colony optimization for solving an industrial layout problem." European Journal of Operational Research **183**(2): 633-642.

Hasda, R. K., et al. (2017). "Modified genetic algorithms for solving facility layout problems." International Journal on Interactive Design and Manufacturing (IJIDeM) **11**(3): 713-725.

Herrán, A., et al. (2021). "An efficient variable neighborhood search for the space-free multi-row facility layout problem." European Journal of Operational Research **295**(3): 893-907.

Hosseini-Nasab, H. and L. Emami (2013). "A hybrid particle swarm optimisation for dynamic facility layout problem." International Journal of Production Research **51**(14): 4325-4335.

Hosseini-Nasab, H., et al. (2018). "Classification of facility layout problems: a review study." The International Journal of Advanced Manufacturing Technology **94**(1): 957-977.

Jolai, F., et al. (2012). "A multi-objective particle swarm optimisation algorithm for unequal sized dynamic facility layout problem with pickup/drop-off locations." International Journal of Production Research **50**(15): 4279-4293.

Kang, S., et al. (2018). "A closed loop based facility layout design using a cuckoo search algorithm." Expert Systems with Applications **93**: 322-335.

Keller, B. and U. Buscher (2015). "Single row layout models." European Journal of Operational Research **245**(3): 629-644.

Khajemahalle, L., et al. (2021). "A hybrid nested partitions and simulated annealing algorithm for dynamic facility layout problem: a robust optimization approach." INFOR: Information Systems and Operational Research **59**(1): 74-101.

Kochhar, J. S., et al. (1998). "HOPE: A genetic algorithm for the unequal area facility layout problem." Computers & Operations Research **25**(7): 583-594.

Komarudin and K. Y. Wong (2010). "Applying Ant System for solving Unequal Area Facility Layout Problems." European Journal of Operational Research **202**(3): 730-746.

Kothari, R. and D. Ghosh (2013). "Tabu search for the single row facility layout problem using exhaustive 2-opt and insertion neighborhoods." European Journal of Operational Research **224**(1): 93-100.

Ku, M.-Y., et al. (2011). "Simulated annealing based parallel genetic algorithm for facility layout problem." International Journal of Production Research **49**(6): 1801-1812.

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

Kulturel-Konak, S. and A. Konak (2011). "A new relaxed flexible bay structure representation and particle swarm optimization for the unequal area facility layout problem." Engineering Optimization **43**(12): 1263-1287.

Kulturel-Konak, S. and A. Konak (2011). "Unequal area flexible bay facility layout using ant colony optimisation." International Journal of Production Research **49**(7): 1877-1902.

Kulturel-Konak, S. and A. Konak (2013). "Linear Programming Based Genetic Algorithm for the Unequal Area Facility Layout Problem." International Journal of Production Research **51**(14): 4302-4324.

Kulturel-Konak, S. and A. Konak (2015). "A large-scale hybrid simulated annealing algorithm for cyclic facility layout problems." Engineering Optimization **47**(7): 963-978.

Lam, K. C., et al. (2007). "The application of the ant colony optimization algorithm to the construction site layout planning problem." Construction Management and Economics **25**(4): 359-374.

Lamba, K., et al. (2020). "Sustainable dynamic cellular facility layout: a solution approach using simulated annealing-based meta-heuristic." Annals of Operations Research **290**(1): 5-26.

Lee, K.-Y., et al. (2003). "An improved genetic algorithm for facility layout problems having inner structure walls and passages." Computers & Operations Research **30**(1): 117-138.

Lee, Y. H. and M. H. Lee (2002). "A shape-based block layout approach to facility layout problems using hybrid genetic algorithm." Computers & Industrial Engineering **42**(2): 237-248.

Li, H. and P. E. D. Love (2000). "Genetic search for solving construction site-level unequal-area facility layout problems." Automation in Construction **9**(2): 217-226.

Liang, L. Y. and W. C. Chao (2008). "The strategies of tabu search technique for facility layout optimization." Automation in Construction **17**(6): 657-669.

Lien, L.-C. and M.-Y. Cheng (2014). "Particle bee algorithm for tower crane layout with material quantity supply and demand optimization." Automation in Construction **45**: 25-32.

Lin, J., et al. (2023). "Learning-based simulated annealing algorithm for unequal area facility layout problem." Soft Computing.

Lin, Z. and Z. Yingjie (2019). Solving the Facility Layout Problem with Genetic Algorithm. 2019 IEEE 6th International Conference on Industrial Engineering and Applications (ICIEA).

Liu, J. and J. Liu (2019). "Applying multi-objective ant colony optimization algorithm for solving the unequal area facility layout problems." Applied Soft Computing **74**: 167-189.

Liu, J., et al. (2020). "Configuration space evolutionary algorithm for multi-objective unequal-area facility layout problems with flexible bays." Applied Soft Computing **89**: 106052.

Liu, J., et al. (2018). "Multi-objective particle swarm optimization algorithm based on objective space division for the unequal-area facility layout problem." Expert Systems with Applications **102**: 179-192.

Matai, R. (2015). "Solving multi objective facility layout problem by modified simulated annealing." Applied Mathematics and Computation **261**: 302-311.

Mazinani, M., et al. (2013). "Dynamic facility layout problem based on flexible bay structure and solving by genetic algorithm." The International Journal of Advanced Manufacturing Technology **65**(5): 929-943.

McKendall, A. R. and J. Shang (2006). "Hybrid ant systems for the dynamic facility layout problem." Computers & Operations Research **33**(3): 790-803.

McKendall Jr, A. R. and W.-H. Liu (2012). "New Tabu search heuristics for the dynamic facility layout problem." International Journal of Production Research **50**(3): 867-878.

McKendall Jr, A. R., et al. (2006). "Simulated annealing heuristics for the dynamic facility layout problem." Computers & Operations Research **33**(8): 2431-2444.

Meller, R. D. and Y. A. Bozer (1996). "A new simulated annealing algorithm for facility layout problem." International Journal of Production Research **34**(6).

Nguyen, P. T. (2021). "Construction site layout planning and safety management using fuzzy-based bee colony optimization model." Neural Computing and Applications **33**(11): 5821-5842.

Niroomand, S., et al. (2015). "Modified migrating birds optimization algorithm for closed loop layout with exact distances in flexible manufacturing systems." Expert Systems with Applications **42**(19): 6586-6597.

Ohmori, S., et al. (2010). Solving facility layout problem via particle swarm optimization. 2010 Third International Joint Conference on Computational Science and Optimization, IEEE.

Önüt, S., et al. (2008). "A particle swarm optimization algorithm for the multiple-level warehouse layout design problem." Computers & Industrial Engineering **54**(4): 783-799.

Paes, F. G., et al. (2017). "A hybrid genetic algorithm with decomposition phases for the Unequal Area Facility Layout Problem." European Journal of Operational Research **256**(3): 742-756.

Palomo-Romero, J. M., et al. (2017). "An island model genetic algorithm for unequal area facility layout problems." Expert Syst. Appl. **68**: 151-162.

Palubeckis, G. (2015). "Fast simulated annealing for single-row equidistant facility layout." Applied Mathematics and Computation **263**: 287-301.

Palubeckis, G. (2017). "Single row facility layout using multi-start simulated annealing." Computers & Industrial Engineering **103**: 1-16.

Paul, R. C., et al. (2006). "A solution to the facility layout problem having passages and inner structure walls using particle swarm optimization." The International Journal of Advanced Manufacturing Technology **29**: 766-771.

Phanden, R. K., et al. (2018). Application of genetic algorithm and variable neighborhood search to solve the facility layout planning problem in job shop production system. 2018 7th International Conference on Industrial Technology and Management (ICITM).

Pourvaziri, H. and B. Naderi (2014). "A hybrid multi-population genetic algorithm for the dynamic facility layout problem." Applied Soft Computing **24**: 457-469.

Ramkumar, A. S., et al. (2009). "A population-based hybrid ant system for quadratic assignment formulations in facility layout design." The International Journal of Advanced Manufacturing Technology **44**(5): 548-558.

Rehman, S., et al. (2016). "Wind Farm Layout Design Using Cuckoo Search Algorithms." Applied Artificial Intelligence **30**(10): 899-922.

Rezazadeh, H., et al. (2009). "An extended discrete particle swarm optimization algorithm for the dynamic facility layout problem." Journal of Zhejiang University-Science A **10**: 520-529.

Ripon, K. S. N., et al. (2010). An adaptive local search based genetic algorithm for solving multi-objective facility layout problem. Neural Information Processing. Theory and Algorithms: 17th International Conference, ICONIP 2010, Sydney, Australia, November 22-25, 2010, Proceedings, Part I 17, Springer.

Safarzadeh, S. and H. Koosha (2017). "Solving an extended multi-row facility layout problem with fuzzy clearances using GA." Applied Soft Computing **61**: 819-831.

Şahin, R. (2011). "A simulated annealing algorithm for solving the bi-objective facility layout problem." Expert Systems with Applications **38**(4): 4460-4465.

Şahin, R., et al. (2010). "A simulated annealing heuristic for the dynamic layout problem with budget constraint." Computers & Industrial Engineering **59**(2): 308-313.

Şahin, R. and O. Türkbey (2009). "A new hybrid tabu-simulated annealing heuristic for the dynamic facility layout problem." International Journal of Production Research **47**(24): 6855-6873.

Samanta, S., et al. (2018). "Bi-objective dependent location quadratic assignment problem: Formulation and solution using a modified artificial bee colony algorithm." Computers & Industrial Engineering **121**: 8-26.

Samarghandi, H. and K. Eshghi (2010). "An efficient tabu algorithm for the single row facility layout problem." European Journal of Operational Research **205**(1): 98-105.

Samarghandi, H., et al. (2010). "A particle swarm optimization for the single row facility layout problem." Computers & Industrial Engineering **58**(4): 529-534.

Saravanan, M. and P. V. Arulkumar (2015). "An artificial bee colony algorithm for design and optimize the fixed area layout problems." The International Journal of Advanced Manufacturing Technology **78**(9): 2079-2095.

Scholz, D., et al. (2009). "STaTS: a slicing tree and tabu search based heuristic for the unequal area facility layout problem." European Journal of Operational Research **197**(1): 166-178.

Selim, A., et al. (2022). "Optimal allocation of distributed generation with the presence of photovoltaic and battery energy storage system using improved barnacles mating optimizer." Energy Science & Engineering **10**(8): 2970-3000.

Singh, S. and R. Sharma (2008). "Two-level modified simulated annealing based approach for solving facility layout problem." International Journal of Production Research **46**(13): 3563-3582.

Sirinaovakul, B. and T. Limudomsuk (2007). "Maximum weight matching and genetic algorithm for fixed-shape facility layout problem." International Journal of Production Research **45**(12): 2655-2672.

Suh, Y. J. and J. Y. Choi (2022). "Efficient Fab facility layout with spine structure using genetic algorithm under various material-handling considerations." International Journal of Production Research **60**(9): 2816-2829.

Sulaiman, M. H., et al. (2020). "Barnacles Mating Optimizer: A new bio-inspired algorithm for solving engineering optimization problems." Engineering Applications of Artificial Intelligence **87**: 103330.

Tate, D. M. and A. E. Smith (1995). "Unequal-area facility layout by genetic search." IIE transactions **27**(4): 465-472.

Tayal, A. and S. P. Singh (2018). "Integrating big data analytic and hybrid firefly-chaotic simulated annealing approach for facility layout problem." Annals of Operations Research **270**(1): 489-514.

DOI: 10.22034/cmes.2024.2024013.1021
Vineetha, G. and C. Shiyas (2023). "MACHINE LEARNING-ENHANCED GENETIC ALGORITHM FOR ROBUST LAYOUT DESIGN IN DYNAMIC FACILITY LAYOUT PROBLEMS: Implementation of Dynamic Facility Layout Problems." International Journal of Industrial Engineering: Theory, Applications and Practice **30**(6).

Wang, M.-J., et al. (2005). "A solution to the unequal area facilities layout problem by genetic algorithm." Computers in Industry **56**(2): 207-220.

Wei, X., et al. (2019). "Optimizing facility layout planning for reconfigurable manufacturing system based on chaos genetic algorithm." Production & Manufacturing Research **7**(1): 109-124.

Wong, C. K., et al. (2010). "Comparison of Using Mixed-Integer Programming and Genetic Algorithms for Construction Site Facility Layout Planning." Journal of Construction Engineering and Management **136**(10): 1116-1128.

Yahya, M. and M. Saka (2014). "Construction site layout planning using multi-objective artificial bee colony algorithm with Levy flights." Automation in Construction **38**: 14-29.

Yu-Hsin Chen, G. (2013). "A new data structure of solution representation in hybrid ant colony optimization for large dynamic facility layout problems." International Journal of Production Economics **142**(2): 362-371.

Zouein, P. P. and S. Kattan (2022). "An improved construction approach using ant colony optimization for solving the dynamic facility layout problem." Journal of the Operational Research Society **73**(7): 1517-1531.