F O U N D A T I O N S O F C O M P U T I N G A N D D E C I S I O N S C I E N C E S
Vol. 47 (2022) No. 2
ISSN 0867-6356
DOI: 10.2478/fcds-2022-0006 e-ISSN 2300-3405

Development of an Adaptive Genetic Algorithm to Optimize the Problem Of Unequal Facility Location

Hendrik Setia Budi[1], Marischa Elveny[2], Pavel Zhuravlev[3], *Abduladheem Turki Jalil**[4], Samaher Al-Janabi[5], Ayad F. Alkaim[5], Marwan Mahmood Saleh[6], Rustem Adamovich Shichiyakh[7], Sutarto[8]

Abstract. The problem of unequal facility location involves determining the location of a set of production equipment whose dimensions are different, as well as the interrelationships between each of them. This paper presents an efficient method for optimizing the problem of unequal facility layouts. In this method, the genetic algorithm is improved and developed into an adaptive genetic algorithm. In this algorithm, the mutation operator is applied only when the similarity of chromosomes in each population reaches a certain level. This intelligence prevents jumps in situations where they are not needed and reduces computational time. In order to measure the performance of the proposed algorithm, its performance is compared with the performance of conventional genetic algorithms and refrigeration simulators. Computational results show that the adaptive genetic algorithm is able to achieve higher-quality solutions.

---

[1] Department of Oral Biology, Faculty of Dental Medicine, Universitas Airlangga, Surabaya 60132, Indonesia.
[2] DS & CI Research Group, Universitas Sumatera Utara, Medan, Indonesia, address Dr. Mansur Road No.9. Kampus USU. 20155. Indonesia
[3] Plekhanov Russian University of Economics, Stremyanny lane, 36, Moscow, , 117997, Russian Federation, Pavel ZHURAVLEV, Doctor of Economics, Professor, ORCID ID: https://orcid.org/0000-0002-6061-9935
[4] Medical Laboratories Techniques Department , Al-Mustaqbal University College, Babylon, Hilla, 51001, Iraq ., Email: abedalazeem799@gmail.com , ORCID : https://orcid.org/0000-0001-8403-7465
[5] College of science for women/ University of Babylon/Iraq
[6] University of Anbar
[7] Kuban State Agrarian University named after I.T. Trubilin, Krasnodar, Russia, Address: Russian Federation, Krasnodar region, 350044, Krasnodar, Kalinina street 13. Department of Management, Candidate of Economic Sciences, Associate Professor, ORCID ID: http://orcid.org/0000-0002-5159-4350.
[8] Mathematics Education, Faculty of Science, Engineering and Applied, Universitas Pendidikan Mandalika, Indonesia
[*] SDG1(Zero Hunger) Research Cluster Landmark University Nigeria, π SDG6(Clean Energy) Research Cluster Landmark University, Nigeria

Keywords: Unequal Facility location, Interactions, Adaptive Genetic Algorithm, Mutation Operator Intelligence, healthy lifestyle

1. Introduction

A facility layout problem involves determining the location of each facility, which can be a production unit, an office building, or machinery inside a plant. Choosing an appropriate location for facilities has considerable effects on production costs, work in progress, production time, and productivity of manufacturing systems [1,2]. Overall, 20-50% of the entire production costs inside a manufacturing plant are related to transportation costs. In this respect, [3] mentioned that lack of optimal layout of different facilities leads to huge material-handling costs, thereby increasing the costs of the system. Therefore, researchers have focused on developing mathematical models and optimization algorithms to deal with the facility layout problem, which is often modeled as a location problem where the goal is to place equal-size facilities into pre-specified locations [4-6]. However, the problem will be less like a real-world issue when similar size facilities are considered. Accordingly, facilities are allowed to have different dimensions in the present study. Meanwhile, unequal facilities shift the problem from a location issue into a formulated mathematical model. To date, several studies have been performed on the facility layout problem. For instance, [7] proposed a single-row layout problem (SRFLP) of rectangular facilities with varying dimensions using a tabu search algorithm with an adaptive memory for variation and intensification methods to find solutions in the neighborhood of good solutions and solutions that have not yet been found. Another research, [4] developed a modeling technique through which real-time facility location in plants could be easily carried out in another research. These scholars adopted an Analytical Hierarchy Process–Genetic Algorithm (AHP–GA) based optimization scheme for automatic layout planning [8-10].

Xu and Song [10] suggested a new method for dynamic temporary construction facilities with unequal-area departments. They considered transportation costs between facilities as fuzzy parameters and regarded facilities in two-dimensional forms. Subsequently, the multi-objective position-based adaptive particle swarm optimization (p-based MOPSO) was developed to optimize the problem. Ultimately, the performance of the proposed algorithm was assessed and proven using a case study. Wang et al. [11] introduced a double-row facility layout problem by combining refrigeration simulation algorithm and mathematical programming. In the end, experiments showed that the methodology could obtain the optimal solutions for small-size issues and determine a real method for problems with real sizes. Ulutas and Islier [8] researched a footwear plant to solve a dynamic facility layout problem, aiming to minimize the total material handling and layout/re-layout costs while considering several duty cycles.

A clonal selection-based algorithm was proposed to solve the real-life dynamic facility layout problem. In the end, numerical results were obtained, and the proposed method was reported to have a more appropriate performance than other techniques [12-17]. Neghabi and Tari [5] presented a new adjacency and closeness rating phenomenon for constructing the optimal facility layout design. In this approach, a special point is given to the plan based on the adjacency of facilities. Moreover, farness was considered as the safety purpose concurrently. A mathematical model was developed, and its performance was assessed using computational experiments. The computational results proved the efficiency of the proposed model is simultaneously considering economic and safety criteria and creating various layout designs.

In another study, Guan and Lin [2] proposed a hybrid algorithm based on variable neighborhood search and ant colony optimization. Three neighborhood structures were utilized in the proposed algorithm to enhance the exploitation ability. Meanwhile, new gain techniques are developed to reduce the mathematical calculations of the objective function values. On the other hand, a new method was used to update the formulas in the ant colony algorithm. These scholars applied standard problems in the literature to evaluate the algorithm and proved its superiority compared to previous techniques [18-22]. Paes et al. [6] introduced a meta-heuristic algorithm and GA to solve unequal facility layout problems. The proposed algorithms were compared to the methods existing in the literature, and the computational results showed that the hybrid GA was able to achieve higher quality solutions in less time. Shavarani et al. [12] a hierarchical facility location problem for online delivery systems. The problem was related to the distribution of products of Amazon in San Francisco, and the shortest path algorithm was used to solve the problem. In another study, Shan et al. [15] addressed the problem of competitive facility location for chain stores and presented a mathematical model in this regard. This model attempted to achieve the most suitable price to increase market share in competitive conditions. A heuristic algorithm was proposed to solve the model.

Guo and Kluse [13] presented a comprehensive framework for locating solar panels, where investment, location, and transportation costs were minimized. Fu et al. [14] optimized facility location and capacity determination in a supply chain network. In this regard, they used a simulation-optimization approach, and the model was tested in two certain and fuzzy uncertain modes. Saif & Delage [16] studied a distributionally robust version of the classical capacitated facility location problem with a distributional ambiguity set. They proposed two single-stage and double-stage mathematical models. In addition, two algorithms based on column generation were developed for solving the problem exactly, and the results were evaluated and analyzed using various numerical problems.

Overall, due to the significance of the unequal facility layout issue and given the statements above, this current study mainly attempts to design and propose an efficient method for optimizing the unequal facility layout issues. The genetic algorithm is boosted and developed into an adaptive genetic algorithm to fulfill that aim. The adaptive genetic algorithm can acquire higher and more proper quality solutions based on the results obtained.

The remainder of the article is constructed as follows: the second section describes mathematical equations and problem modeling, and the third section presents the algorithm. The fourth section shows the results of the proposed algorithm following the adjustment of the algorithm's parameters, and the fifth section concludes and makes suggestions for future studies.

2. Statement of the Problem

The main objective of the model is to determine the location of facilities in a way that the facility transportation costs are minimized. Given the fact that the layout of unequal facilities is considered in the research, the problem is formulated in the form of a linear mathematical model. The model's primary goal is to locate facilities so that the transportation costs between facilities are minimized. To this end, the location of the center of each facility, as well as its horizontal or vertical position, must be determined. The main limitation of study is that, given the limited space, the layout of all facilities must be carried out in a way that no overlap occurs among them. Each of the facilities is available in any part of the space. A model is proposed in the next section to find a solution for the problem. In addition, the model premises are presented below:

- The rectangular-square facilities have different sizes.
- The required length, width, and area of each facility are pre-specified.
- The total available area is limited.
- The amount of transportation between facilities is definite and pre-specified.
- The distance between the Euclidean facilities is considered center-to-center.

2.1. Problem Formulation

The decision variables, symbols, and parameters of the model are presented in the below:

Indexes	
i,j:	Facility index. At least N facilities exist.
Parameters	
$F_{i,j}$	The material flow between the i-th and j-th facilities
Sh_i	The smaller length of the i-th facility
Lng_i	The larger length of the i-th facility
L	The entire length of the area available
W	The entire width of the area available
M	A big number
Dependent decision variables	
$x\text{-}p_{ij}$, $y\text{-}p_{ij}$	The horizontal and vertical distance between the i-th facilitation center and the j-th facilitation center
$Left_{ij}$	1, if the i-th facility is placed on the left side of the j-th facility; otherwise, 0. 1, if the i-th facility is placed on the lower side of the j-th facility; otherwise, 0.
Independent decision variables	
(x_i, y_i)	Coordination of the i-th facilitation center
H_i	If the i-th facility is placed horizontally

Ultimately, the facility layout problem is mathematically formulated, as follows.

$min = \sum_{i=1}^{N}\sum_{j=1}^{N} F_{i,j}\,(x-p_{ij}+y-p_{ij})$		(1)
$(x_i+0.5l_i)-(x_i-0.5l_i) \leq M(1-left_{ij})$	$\forall i,j$	(2)
$(y_i+0.5w_i)-(y_i-0.5w_i) \leq M(1-below_{ij})$	$\forall i,j$	(3)
$left_{ij}+left_{ji}+below_{ij}+below_{ji}=1$	$\forall i,j$	(4)
$(x_i+0.5l_i) \leq L$	$\forall i$	(5)
$(x_i-0.5l_i) \geq 0$	$\forall i$	(6)
$(y_i+0.5w_i) \leq W$	$\forall i$	(7)
$(y_i+0.5w_i) \geq 0$	$\forall i$	(8)
$x-p_{ij} \geq x_i-x_j$	$\forall i,j>i$	(9)
$x-p_{ij} \geq x_j-x_i$	$\forall i,j>j$	(10)
$y-p_{ij} \geq y_i-y_j$	$\forall i,j>i$	(11)
$y-p_{ij} \geq y_j-y_i$	$\forall i,j>i$	(12)
$l_i = Lng_i h_i + Sh_i(1-h_i)$	$\forall i$	(13)
$w_i = Lng_i(1-h_i) + Sh_i h_i$	$\forall i$	(14)
$x_i, y_i, l_i, w_i, x-p_{ij}, y-p_{ij} \geq 0$	$\forall i,j$	(15)
$below_{ij}, left_{ij}, h_i \in \{0,1\}$	$\forall i,j$	(16)

The objective function (1) minimizes the total transportation costs between facilities. Constraints (2-4) guarantee the lack of facility overlap, whereas constraints (5-8) ensure that the facility is located inside the general area. Constraints (9 and 10) calculate the transverse distance between two facilities, while constraints (11 and 12) estimate the longitudinal distance between two facilities. Constraints (13) calculate the length of each facility, according to which the facility's length equates the larger size in case of the horizontal location of the facility. Constraints (14) estimate the width of each facility, according to which the facility's width equates the smaller side in case of the vertical location of the facility. Constraints (15 and 16) control the range of decision variables. The static facility layout problem with equal facilities is an NP-hard problem [1]. Since unequal facilities are considered in the problem, it can be stated that the problem is of NP-hard type, which complicates problem-solving. However, this complexity shows the necessity of developing efficient algorithms for such problems [9]. The following section presents a GA with adaptive mutation to solve the layout problem.

2.2 GA with Adaptive Mutation

The basis of the algorithm derived from nature is to use stochastic search for optimization of learning processes and problems. In nature, the combination of proper chromosomes leads to more efficient generations. Meanwhile, mutations often occur in chromosomes that might improve the next generation. The GA can search different areas of a solution space simultaneously. However, the GA method easily converges to local optimization due to chromosome similarity. The best solution to avoid local optimization is using a mutation operator. To improve algorithm performance, it is best to apply the mutation operator when chromosome similarity is too high. As such, the current study develops GA into an adaptive GA (AGA), in which the mutation operator is applied only under certain conditions.

2.3 Chromosomes (Encoding and Decoding)

In GA, each chromosome shows a point in the search space and is recognized as a possible solution for the problem. The chromosomes (solutions) encompass a fixed number of genes (variables). Encoding is usually used to show the chromosomes. In the present study, chromosomes must demonstrate facility location, so a coded chromosome is applied. This chromosome is a 3*N matrix, where N is the number of facilities. All numbers in the matrix are random and between zero and one. The numbers in each column are related to one facility. For instance, Figure 1 shows one chromosome for the problem of the location of 24 facilities in an available space.

0.51	0.93	0.12	0.00	0.58	0.20	0.67	0.02	0.92	0.73	0.93	0.88	0.62	0.39	0.84	0.63	0.68	0.51	0.99	0.26	0.93	0.01	0.20	0.12
0.91	1.00	0.02	1.00	0.01	0.03	0.02	0.27	0.27	0.27	0.01	0.29	0.53	0.38	0.29	0.32	0.91	0.32	0.32	0.01	0.67	0.67	0.91	0.32
0.19	0.03	0.24	0.68	0.38	0.24	0.15	0.19	0.15	0.68	0.24	0.03	0.19	0.38	0.24	0.15	0.24	0.24	0.15	0.15	0.15	0.15	0.38	0.38

Figure 1. Encoded chromosomes display

As mentioned before, the chromosome in Figure 1 is coded, and the numbers existing in the chromosome must be decoded to determine the actual layout. The decoding process starts from the third row. First, all numbers in the third row are rounded up or down. This turns the third row into a binary vector. In this respect, zero and one are interpreted as horizontal and vertical positions of the facility. This row indicates that facility one must be located horizontally, so does facility two and the rest of the facilities. On the other hand, the second and third rows of the encoded chromosome indicate the coordination of the center of facilities. To decode these numbers, we first consider the length and width of the entire space available. Afterward, the range at which the facility is allowed to be located is determined. Finally, these numbers are multiplied into the numbers existing in the second and third rows, and the coordination of the center of facilities is obtained. For example, if the total length of the available area is 121 meters, x of facility one will be greater than 12.5 considering that the length of facility 1 is equal to 25 meters. In addition, since the length of the available space is 121 meters, x of facility 1 is smaller than 121-12.5=108.5. In other words, the x coordination of facility one must be in the 12.5-108.5 range. Ultimately, x coordination of facility 1 is equal to 12 + (108-12.8) * 51=61.6. The same process is used to determine the coordination of the center of other facilities. Figure 2 shows the decoded solution of chromosome in Figure 1. Notably, there is some overlap in the chromosome due to its random production. For instance, facilities 5 and 7 overlap. In order to prevent this situation, the present research suggests a penalty function, through which unjustified chromosomes will be removed by an optimization process and replaced by justified ones.

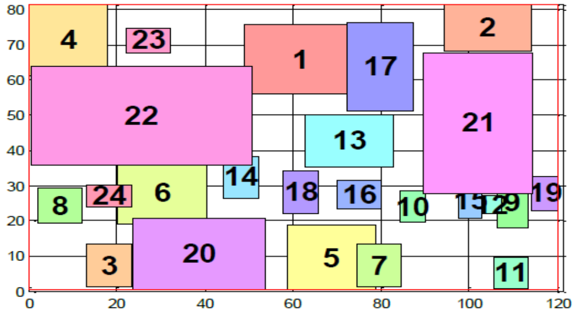

Figure 2. Facility layout presentation

2.4 Intersection Operator

During the operation of the intersection, parts of the chromosomes are accidentally exchanged. This allows children to have a combination of their parents' characteristics. The present study applies two intersection operators; in the first one, parent chromosomes are first chosen based on the roulette wheel, and a random number in the range of {1…N} is generated, where N is the number of facilities. The selected number is the intersection point. Columns 1 to the intersection point in the first parent are directly copied in the first child. In addition, the columns of +1 intersection point up to N in the first parent are directly copied in the first child. The opposite of the mentioned process generates the second child, meaning that columns 1 to the intersection point in the second parent are directly copied in the second child. Moreover, the columns of +1 intersection up to N in the first parent are directly copied in the second child. Ultimately, the chromosome of children is produced in an encoded form and will be decoded as described in the chromosome description section. Two random points are chosen in the second-type intersection operator, and parents' genes are transferred to children based on the selected points. For example, points 3 and 9 are chosen as intersection points in Figure 3.

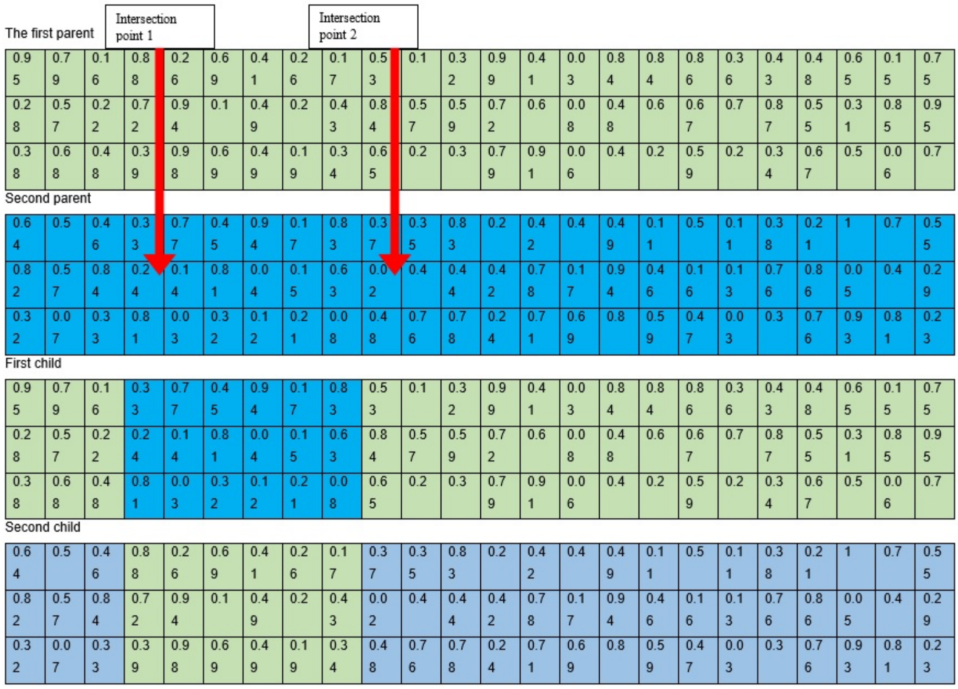

Figure 3: Multi-point integration

Figure 3. Example of GA operators

2.5 Adaptive Mutation Operator

In GA, the mutation is done randomly with low possible, and elements are changed in the chromosome. The mutation operator changes the chromosome and prevents premature algorithm convergence. Therefore, a mutation in populations must be smart and rational as much as possible. In order to improve GA performance, an AGA is used, where the mutation operator is only applied when there is a certain level of chromosome similarity in each population. The following equation obtains the chromosome similarity coefficient:

$$SC_{ab} = \frac{\sum_{i=1}^{N} \partial(X_{ija}, X_{ijb})}{N} \qquad (17)$$

Where X_{ijb} and X_{ija} are the numbers existing in the i-th column and j-th row in a and b chromosomes.

$$\partial(X_{ija}, X_{ijb}) = \begin{cases} 1 & if\ X_{ija} = X_{ijb} \\ 0 & otherwise \end{cases} \qquad (18)$$

The mean coefficient of similarity between chromosomes of a population is determined by Equation (19):

$$\overline{SC} = \frac{\sum_{a=1}^{N-1}\sum_{b=a+1}^{N} SC_{ab}}{\binom{N}{2}} \qquad (19)$$

Where N is the number of works. Therefore, the mutation operator can only be applied on chromosomes when the value of $\overline{SC}$ exceeds a specific threshold. The size of the threshold is determined based on preliminary experiments and the trial and error method. The similarity coefficient is assessed in each iteration, and the mutation operator will be applied when chromosomes become too similar. The mutation operator will be applied on chromosomes, provided that the prerequisite is established. In this research, we use two types of mutation operators; in the first one, two random numbers are selected for each row, and genes are displaced (Figure 4). Two random numbers are selected for each row in the second mutation operator, and the genes between the numbers are reversed.

Parent

0.95	0.79	0.16	0.88	0.26	0.69	0.41	0.26	0.17	0.53	0.1	0.32	0.99	0.41	0.03	0.84	0.84	0.86	0.36	0.43	0.48	0.65	0.15	0.75
0.28	0.57	0.22	0.72	0.94	0.1	0.49	0.2	0.43	0.84	0.57	0.59	0.72	0.6	0.08	0.48	0.6	0.67	0.7	0.87	0.55	0.31	0.85	0.95
0.38	0.68	0.48	0.39	0.98	0.69	0.49	0.19	0.34	0.65	0.2	0.3	0.79	0.91	0.06	0.4	0.2	0.59	0.2	0.34	0.67	0.5	0.06	0.7

Mutated child

Mutated child

0.95	0.79	0.16	0.41	0.26	0.69	0.88	0.26	0.17	0.53	0.1	0.32	0.99	0.41	0.03	0.84	0.84	0.86	0.36	0.43	0.48	0.65	0.15	0.75
0.28	0.57	0.22	0.72	0.94	0.1	0.49	0.2	0.43	0.84	0.57	0.67	0.72	0.6	0.08	0.48	0.6	0.59	0.7	0.87	0.55	0.31	0.85	0.95
0.68	0.38	0.48	0.39	0.98	0.69	0.49	0.19	0.34	0.65	0.2	0.3	0.79	0.91	0.06	0.4	0.2	0.59	0.2	0.34	0.67	0.5	0.06	0.7

Figure 4. The first-type mutation operator

2.6 Algorithm Stopping Criterion

Since the GA is based on test and production, the problem's solution is unclear, and we cannot determine the optimal solution to define the stopping criterion for finding a solution in the population. Therefore, the criterion chooses the repeated number, which will be determined based on parameter tuning.

3. Computational Results

This section evaluates the performance of the proposed algorithms for solving a facility layout problem compared to GA and simulated annealing (SA). First, the optimal values of input parameters of the algorithms are determined using the Taguchi method. Afterward, several sample problems are randomly generated to assess the performance of algorithms. The algorithm that achieves the best layout statistically is known as the strongest algorithm. In order to evaluate the performance of the algorithms using statistical tests, numerical examples are randomly generated and on a personal computer. The generated problems are classified into three different classes based on the number of facilities. The first-third classes include problems with 1, 20, and 24 facilities. In each group, ten random issues are generated and used to test the performance of the algorithms. In all sample problems produced, the length and width values of the facilities are produced uniformly and between 20 and 30 meters. Moreover, the material flow between the facilities has a uniform distribution and is between 20 and 70. Table 1 and Figure 5 show the results obtained from implementing 30 sample problems with the desired metaheuristic algorithms.

Table 1. Results of comparison of metaheuristic algorithms on 30 sample problems

Test problem	SA		GA		IGA	
	Objective value	CPU time	Objective value	CPU time	Objective value	CPU time
1	2676.3	2.73	2594.8	4.69	2339.55	6.73
2	2715.05	3.15	3336.11	5.53	2435.66	7.17
3	2991.23	3.73	3872.98	5.97	2680.14	7.21
4	3159.41	4.00	4563.34	6.99	3477.26	7.29
5	3345.41	4.46	5486.90	7.78	4474.92	8.44
6	3674.26	5.08	6136.66	9.33	5203.67	8.74
7	3838.36	5.54	6588.76	9.40	6251.83	9.67
8	4288.08	5.67	7896.41	10.48	7897.18	9.94
9	4502.70	5.78	8688.90	10.74	9885.39	10.31
10	4905.83	6.57	8847.72	11.52	10129.65	12.30
11	5333.95	6.72	10960.80	12.67	11503.03	13.64
12	6403.21	7.45	11795.06	14.03	13000.71	16.28
13	6632.90	7.71	11903.15	14.26	13385.68	19.19
14	6668.29	8.65	14862.72	14.28	16495.74	22.14
15	7099.98	9.65	17492.23	16.05	19955.22	22.43
16	8274.93	9.74	18425.55	16.74	21066.75	26.85
17	22025.83	10.97	21795.33	18.02	10155.05	29.35
18	27761.52	12.21	23897.73	18.81	10501.62	29.47
19	34586.40	14.09	24273.05	19.16	12773.15	30.29
20	39355.69	14.63	27991.74	20.75	13281.68	30.77
21	43236.27	17.55	31061.02	21.97	16271.26	32.29
22	54018.09	17.55	35852.74	23.78	16429.16	37.77
23	57117.39	20.30	45275.39	24.02	17943.59	38.45
24	60508.85	22.57	45513.08	24.81	22741.41	40.14
25	66017.66	25.94	55388.99	29.44	26456.97	47.90
26	69115.22	28.71	55514.54	33.26	27849.32	53.44
27	79909.27	32.51	65770.94	35.07	31052.89	62.27
28	88587.27	34.21	74427.60	36.44	35068.75	63.40
29	106617.39	38.45	76414.67	38.71	38289.81	71.96
30	135009.47	44.78	82030.88	40.88	49593.33	77.39

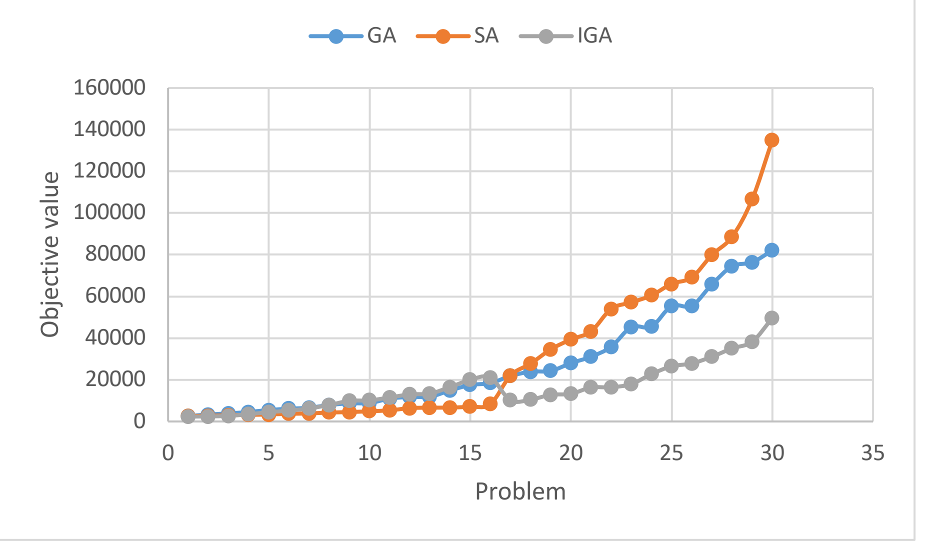

Figure 5. Comparison of the performance of algorithms proposed in different dimensions in solution quality criterion

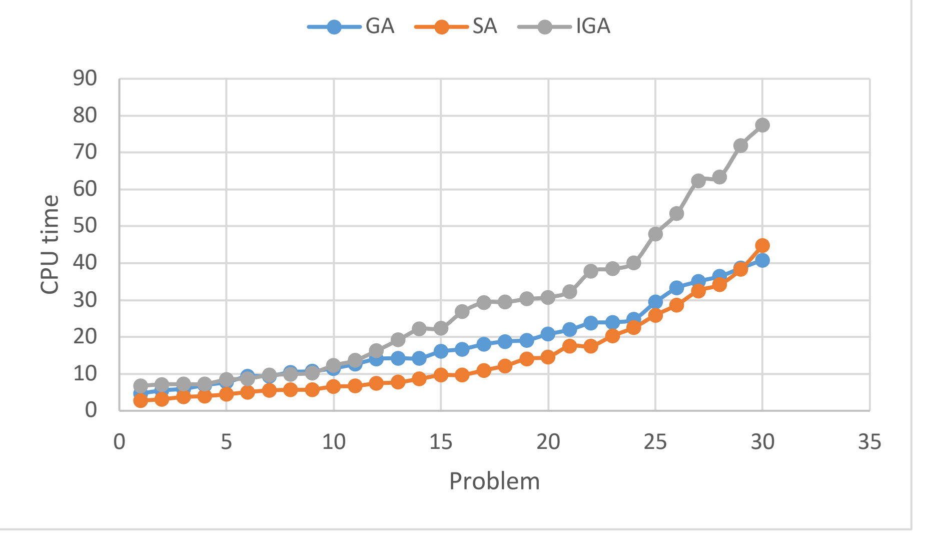

Figure 6. Comparison of the performance of algorithms proposed based on computational time criterion

Figures 5 and 6 show the performance of the algorithms in the solution quality index and computational time. Regarding the solution quality index, the GA and AGA have an extremely better performance than SA and can attain a more efficient solution. However, the GA has the weakest performance in the time index and acted poorly in terms of time. The variance analysis test is used to evaluate the algorithms in this study statistically. First, the results obtained from the algorithms are normalized by the ratio of performance to deviation (RPD), which is calculated using the equation below:

$$RPD_{ij} = \frac{Alg_{sol}(ij) - min_{sol}(j)}{min_{sol}(j)} \tag{20}$$

Where i shows the number of the algorithm, j is the number of the problem, $min_{sol}(j)$ is the best solution obtained in the j-th problem and $Alg_{sol}(ij)$ is the solution obtained from the i-th algorithm for the j-th problem. The results related to the assessment of the RPD index are shown in Figure 6, according to which the AGA has the best performance regarding the solution quality criterion and acted better than the other two algorithms. In addition, GA has a better performance compared to SA. According to Figure 6, SA and AGA have proper performance in terms of the computational time criterion. Meanwhile, the GA has the weakest performance, compared to the other two algorithms, in this regard. The poor performance of the GA can be related to the application of intersection and mutation operators in all iterations on all chromosomes. The AGA has greatly improved this weakness by eliminating the mutation operation in many iterations.

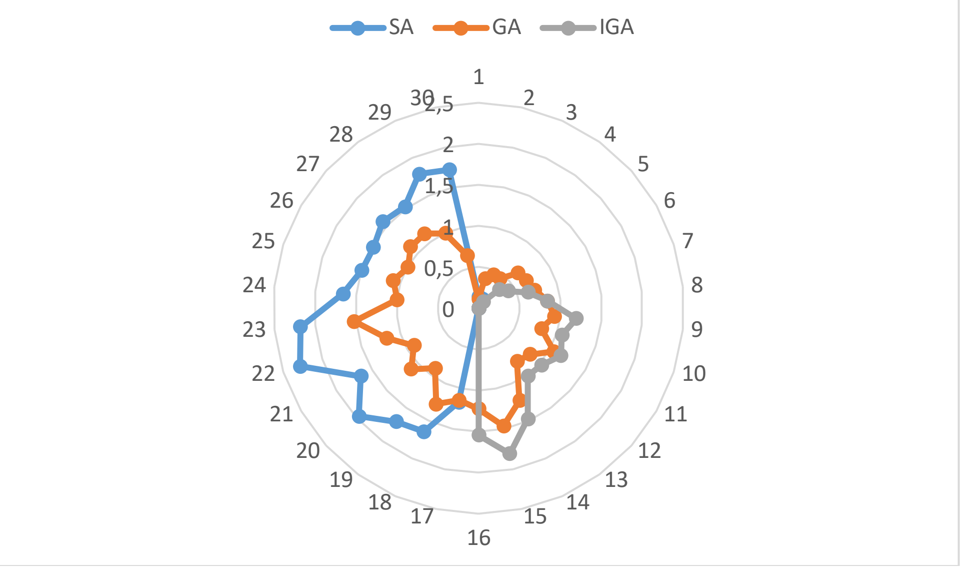

Figure 7. Comparison of meta-heuristic algorithms based on RPD index

4. Conclusion and Recommendations

According to the present study results, population-based algorithms are generally more appropriate for solving layout problems. However, these algorithms need high computational times, which leads to a weakness in this respect. The results indicated the better performance of the SA in the computational time criterion compared to the GA. Therefore, we need an algorithm that has a performance similar to that of the GA, can achieve quality solutions, and is computationally acceptable. To this end, the GA was updated by upgrading the mutation operator, which led to the introduction of the AGA. According to computational results, the AGA achieved the best solutions in acceptable computational time. By considering unequal facilities, the present study solved a problem similar to a real-world issue. However, many other assumptions could be dealt with to get the problem closer to the real world. For instance, a definite flow matrix can be assumed, and flow between departments can be regarded as a random parameter with a specific probability distribution. It is suggested that further studies be carried out to optimize the layout of non-single floor spaces while considering constraints such as budget limitations.

References

[1] Garey, M. R., Johnson, D. S. (1999). Computers and intractability: A guide to the theory of NP-completeness. New York: WH Freeman.

[2] Guan J., Lin G. (2011). Hybridizing variable neighborhood search with ant colony optimization for solving the single row facility layout problem, European Journal of Operational Research TEA (C), 499–9.9.

[3] McKendall, A.R., Shang, J. (2009). Hybrid ant systems for the dynamic facility layout problems, Computers and Operations Research **33**: 790-803.

[4] Nee A.Y.C., Jiang S. (2013). A novel facility layout planning and optimization methodology, CIRP Annals-Manufacturing Technology **62**, 483-486.

[5] Neghabi, H., Tari, F.G. (2016). A new concept of adjacency for concurrent consideration of economic and safety aspects in design of facility layout problems, Journal of Loss Prevention in the Process Industries **40**, 603-614.

[6] Paes, F.G., Pessoa, A.A., Vidal, T. (2017). A hybrid genetic algorithm with decomposition phases for the Unequal Area Facility Layout Problem, European Journal of Operational Research **256**(3), 742-756.

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

[8] Ulutas B., Islier A. (2010). Dynamic facility layout problem in footwear industry, Journal of Manufacturing Systems **36**, 55-61.

[9] Urban, T.L. (1993). A heuristic for the dynamic facility layout problem. IIE Transactions **25**(4): 57-63.

[10] Xu, J., Song, X. (2010). Multi-objective dynamic layout problem for temporary construction 4 facilities with | unequal-area departments under fuzzy random, Knowledge-Based Systems, **81**: 30-45.

[11] Wang, S., Zuoa, X., Liua, X., Zhaoc, X., Li, J. (2015). Solving dynamic double row layout problem via combining simulated annealing and mathematical programming, Applied Soft Computing **37**, 303-310.

[12] Shavarani, S. M., Nejad, M. G. Rismanchian, F., & Izbirak, G. (2018). Application of hierarchical facility location problem for optimization of a drone delivery system: a case study of Amazon prime air in the city of San Francisco. *The International Journal of Advanced Manufacturing Technology*, **95**(9), 3141-3153.

[13] Guo, Q., & Kluse, C. (2020). A framework of photovoltaics recycling facility location optimization. *Sustainable Production and Consumption*, **23**, 105-110.

[14] Fu, Y., Wu, D., Wang, Y., & Wang, H. (2020). Facility location and capacity planning considering policy preference and uncertain demand under the One Belt One Road initiative. *Transportation Research Part A: Policy and Practice*, **138**, 172-186.

[15] Shan, W., Yan, Q., Chen, C., Zhang, M., Yao, B., & Fu, X. (2019). Optimization of competitive facility location for chain stores. *Annals of Operations Research*, **273**(1-2), 187-205.

[16] Saif, A., & Delage, E. (2021). Data-driven distributionally robust capacitated facility location problem. *European Journal of Operational Research*, **291**(3), 995-1007.

[17] Fakhrzad, M. B., Amir M. G., and Farzaneh B. (2015). "A mathematical model for P-hub median location problem to multiple assignments between non-hub to hub nodes under fuzzy environment." *JOURNAL OF MANAGEMENT AND ACCOUNTING STUDIES* **3**, 2, 61-67.

[18] Ahmadi-Javid A, Ardestani-Jaafari A. (2021). The unequal area facility layout problem with shortest single-loop AGV path: how material handling method matters. *International Journal of Production Research*, **59**(8), 2352-74.

[19] Zouein PP, Kattan S. (2021). An improved construction approach using ant colony optimization for solving the dynamic facility layout problem. *Journal of the Operational Research Society*. **29**, 1-5.

[20] Bhuiyan, T. H., Harun, S., & Azeem, A. (2021). Development of an optimisation model for unequal-area facility layout problems. *International Journal of Industrial and Systems Engineering,* **37**(1), 27-45.

[21] Liu, J., Liu, S., Liu, Z., & Li, B. (2020). Configuration space evolutionary algorithm for multi-objective unequal-area facility layout problems with flexible bays. *Applied Soft Computing,* **89**, 106052.

[22] García-Hernández, L., Salas-Morera, L., Carmona-Muñoz, C., Garcia-Hernandez, J. A., & Salcedo-Sanz, S. (2020). A novel island model based on coral reefs optimization algorithm for solving the unequal area facility layout problem. *Engineering Applications of Artificial Intelligence*, **89**, 103445.

Received 16.05.2022, Accepted 30.03.2022