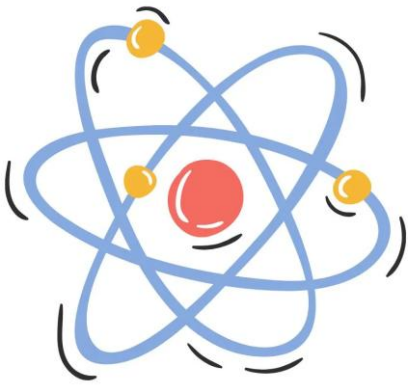

ISSN: 1672 - 6553

JOURNAL OF DYNAMICS AND CONTROL

VOLUME 10 ISSUE 01: P263-279

A STUDY ON SOLVING THE FACILITY LAYOUT PROBLEM USING SIMULATED ANNEALING

Pawan Kumar S. S.[1], Irappa Basappa Hunagund[2], U.N. Kempaiah[3]

[1]PhD Scholar, Department of Mechanical Engineering, University Visvesvarayya College of Engineering, Bangalore- 560001, Karnataka, India.
[2]Professor, Department of Mechanical Engineering, Government Engineering College Haveri, Haveri - 581110, Karnataka, India.
[3]Professor, Department of Mechanical Engineering, University Visvesvarayya College of Engineering, Bangalore- 560001, Karnataka, India.

JOURNAL OF DYNAMICS AND CONTROL
https://doi.org/10.71058/jodac.v10i01016

A STUDY ON SOLVING THE FACILITY LAYOUT PROBLEM USING SIMULATED ANNEALING

Pawan Kumar S. S.[1*], Irappa Basappa Hunagund[2], U.N. Kempaiah[3]

[1]PhD Scholar, Department of Mechanical Engineering, University Visvesvarayya College of Engineering, Bangalore- 560001, Karnataka, India.
[2]Professor, Department of Mechanical Engineering, Government Engineering College Haveri, Haveri - 581110, Karnataka, India.
[3]Professor, Department of Mechanical Engineering, University Visvesvarayya College of Engineering, Bangalore- 560001, Karnataka, India.
sspawan09@gmail.com, iranna346@yahoo.com , unkemp@rediffmail.com

Abstract: *This paper proposes a Simulated Annealing approach for solving equal-area facility layout problems. The technique employs a swap-based neighbourhood moves and Metropolis acceptance criteria to balance exploration and exploitation of the solution space. The algorithm was implemented in Python and evaluated on standard Quadratic Assignment Problem instances from QAPLIB, and its robustness was evaluated by solving the literature problem. Experimental results demonstrate that the proposed algorithm produced an optimal solution with minimal CPU time. Using a suitable Machine Learning concept, exploit the data of a large solution space and extract the minimum material handling cost.*

Keywords: Simulated Annealing; facility layout; metaheuristic; QAP; Optimisation

*Corresponding Author

https://doi.org/10.71058/jodac.v10i01016

1. Introduction

The Facility Layout Problem (FLP) refers to the arrangement of departments or machines in various locations in a job shop, making resources easily available for various operational sequences to perform efficient work. Parameters such as plant location, operation sequence, material consumption rate, availability of resources, and active labour involvement are some factors involved in the design of the layout. The principal objective of any layout design is to maximise work efficiency, minimise material waste at the lowest cost and with the shortest time. The industries should adopt a layout where they can easily discharge material to every possible location, without any interruption from external factors. Achieving an effective operational environment and well-structured production and service system is difficult, and it must be adopted initially, or else industries should rearrange their job shop location, which leads to an increase in infrastructure cost. More of these factors significantly influence the long-term viability and productivity of the overall system. The challenge in FLPs is material handling cost (MHC), which accounts for 20–50% of total operating expenses. Strategic and efficient layout planning has the potential to minimise MHC by 10–30% and thereby enhance the overall productivity of the system. [1], [2], [3].

The Facility Layout Problem is considered as the Quadratic Assignment Problem (QAP), which is also an NP-hard problem [1]. It is difficult to find an optimal solution as the problem size increases. Over the previous decades, an enormous amount of research has been conducted on the layout problems in various industrial structures. The scope of the FLP is seen across manufacturing, arrangement of electronic components, circuit board layout design, design of hospital layout, and hydraulic turbine design, etc., [4], [5].

The facility layout problem is categorised based on the approach or method to be adopted for solution, depending on the area and shape and also on problem formulation. The paper follows the discrete space, equal area static layout problem (SFLP), where the location of departments is composed of integers, and departments are labelled alphabetically. The assumption that all equal areas are represented in Figure 1 does not typically concern the actual shape of the layout.

D (1)	H (2)	C (3)
A (4)	E (5)	F (6)
G (7)	B (8)	I (9)

Figure 1. 3X3 layout arrangement

For illustrative purposes, a grid of 3X3 with the arrangement of the 9 machines in 9 locations is shown. The grid represents the machines or departments in alphabetical letters

https://doi.org/10.71058/jodac.v10i01016

A to I, whereas locations are shown in numbers 1 to 9 in brackets. It is understood that Machine D is in location 1, and Machine I is located in the 9th position of the layout. Here, both machines are allocated a constant specific location. To generate the optimal solution, the machine distance and material flow between the machines are the factors used to calculate the material handling cost. Simulations are run multiple times with different machine arrangements to different locations using a swapping mechanism, as FLP is a stochastic combinatorial optimisation problem where the solution space grows factorial times with the problem size. Finding an optimal solution is exponentially more difficult as the problem size increases [6]. For example, the above layout has 9 departments assigned to 9 locations, which will generate 9! (3,62,880) possible solutions. As the solution space increases, finding an optimum solution in a large solution space is complex in a pool of several possible solutions within a possible CPU time. This article proposed a Simulated Annealing algorithm to extract the optimal solution from the pool of solutions.

The paper contains five sections. Section 1 gives the introduction, Section 2 summarises The literature review, Section 3, explains the proposed Simulated Annealing algorithm and pseudo code for the proposed algorithm, Section 4 contains numerical experiments conducted on QAPLIB problem instances and literature problems, and Section 5 concludes the present work.

2. Literature review

The Quadratic Assignment Problem (QAP) was established by Koopman and Beckmann in (1957) [7], and formulated for the static facility layout problem with discrete areas. In its classical form, the QAP assigns *'m'* departments or machines to *'m'* locations, intending to minimise total material handling cost. MHC cost is modelled as the sum of the products of the material flow between facilities and rectilinear distances between their assigned locations, resulting in a "quadratic" nature [8]. Researchers have recognised QAP as the most challenging combinatorial optimisation and NP-hard problem, which indicates that, as the problem size increases, the computational effort required to obtain an exact optimal solution increases exponentially with the size of the problem [9]. Hence, solving large-scale problems is often computationally infeasible within a reasonable timeframe. Consequently, researchers frequently employ metaheuristic and heuristic approaches to obtain high-quality near-optimal solutions. Meta-heuristics are an algorithmic framework designed to find a near-optimal solution within a reasonable time. Meta-heuristics use stochastic optimisation techniques that provide powerful search capabilities for complex combinatorial problems.

In this section, a brief review of the literature on various articles related to heuristic and meta-heuristic approaches. Genetic algorithm (GA) is an optimisation technique, which imitates natural selection and biological evolution introduced by John Holland in 1972 [10]. The parent layout adopts mainly crossover and mutation processes to generate the optimal solution [2], [11], [12], [13], [14],. Simulated Annealing (SA) adopts the metallurgical process to extract the optimum solution [15],[16]. Tabu Search (TS) is a method for finding the solution to the combinatorial optimisation problem using a near local search technique [17]. Ant Colony Optimisation (ACO) is a computational technique used to mimic real ants using pheromone trails to find an optimal path for complex problems [18]. Graph theory uses a graphical representation of the block layout and adopts greedy construction to obtain the optimum solution [19] used for solving FLPs have been developed.

Simulated annealing (SA) was first introduced by Kirkpatrick, Gelatt, and Vecchi in 1983

https://doi.org/10.71058/jodac.v10i01016

[20], inspired by the annealing process in metallurgy, where metals are heated to high temperatures and cooled slowly to reach a stable state. It uses a Metropolis criterion to accept worse solutions probabilistically, making exploration of solutions more likely and escaping from local optima. SA has been proven to be effective for large combinatorial optimisation problems like DFLP, MOFLP(Multi-Objective facility layout problem) [6] and Robust layout design [3] Problems like hospital [21], [22], [23], cellular manufacturing system, Flexible manufacturing system [24], and footwear industries problems [25] are solved with real-time applications. In the present work also an efficient SA algorithm is also proposed to solve the FLPs.

3. Proposed Simulated Annealing algorithm

Simulated Annealing (SA), inspired by the Annealing Process in metallurgy, where the material is heated to above its recrystallisation temperature and below its melting point, which allows the atoms to have enough energy to move and rearrange. The process allows the material to hold temperature for a certain period, where its microstructure and internal stress are relieved, and a new defect-free grain structure molecule will be formed. After a certain holding period at a certain temperature, materials cool down at a controlled rate. Through this process, the materials get specific characteristics like hardness and ductility, which are increased by relieving internal stress and enhancing their material properties.

Similarly algorithm works in the same manner as the annealing process. The algorithm starts with a high temperature to explore the solution space and then cools down slow cooling rate, which allows the algorithm to develop the best cost with an optimum solution. The algorithm uses Metropolis criteria to hold the temperature and accept the worst solution to escape from the local optima. SA probabilistically accepts the non-improving solutions, and it allows for exploring a large solution space, making it well-suited for non-linear, multi-modal facility layout problems. The core idea of SA is to escape local optima and find the best solution in the solution space [26].

The study proposed the Simulated Annealing (SA) model and applied it to the static equal-area Facility Layout Problem. The problem consists of a set of equal area machines/facilities (labelled as A–I), and they are allocated to specific locations in a 3 × 3 grid, as shown in Figure 1. From equation 1, Z(x) the total material handling cost is calculated by multiplying the flow between facilities i and j by the rectilinear distance between facilities i and j. The Mathematical model for QAP has been taken from Hunagnud et al (2021) [1].

$$\text{Minimise } Z(x) = \sum_{i=1}^{m} \sum_{j=1}^{m} (F_{ij}\, C_{ij}) d_{ij} \quad (1)$$

Where,

$$d_{ij} = |X_i - X_j| + |Y_i - Y_j| \quad (2)$$

F_{ij}= material flow between facility i and j, C_{ij} is cost of material moving from facility i to j (assume unit cost), m indicate size of the layout, d_{ij} is the distance between the facility i and facility j, to optimise in objective function the distance between facility i and j will be calculated through Manhattan distance (rectilinear distance) equation 2, where X and Y are centre coordinate for the facility i and j.

https://doi.org/10.71058/jodac.v10i01016

3.1. Initial Parameter Setting

The proposed algorithm contains parameters called initial temperature (T_0), cooling rate (g), termination temperature (T_f), epoch length (N) and acceptance criteria, which play a crucial role in guiding the search process and ensuring proper convergence of the algorithm. In the present work, the above parameter settings are analysed and proposed in the SA algorithm.

Initial Temperature T_0: The initial Temperature is set to be high to facilitate exploration of the solution space, which allows the algorithm to accept all kinds of solutions at the early stages. The initial temperature depends on the problem size 'n', flow matrix F_{ij}, and area of the facility layout. In this work, the initial temperature is determined through the trial-and-error method, which mainly depends on the problem size. In the present work, for problem sizes up to 12, the initial temperature is set to 1000, and for problem sizes above 12, the initial temperature is set from 5000 to 7000. Meanwhile, in the operation, temperature decreases gradually with the cooling rate, which makes accepting worse solutions in a probabilistic manner. This parameter encourages exploitation around the solution space and guides convergence toward a near-optimal layout configuration.

Cooling Rate (g): Typically, the cooling rate ranges between 0.8 and 0.99 [3]. It contributes to SA by balancing exploration (searching new, unvisited areas of the solution space) and exploitation (intensifying the search in promising regions). It influences the algorithm's ability to find an optimal or near-optimal solution. The cooling rate has a direct, nonlinear relationship with temperature, resulting in a slow cooling schedule. The search effort at each stage influences the algorithm's ability to escape local optima. A larger 'g' value means temperature decreases slowly and exploration increases. Slower cooling rate, more search effort at each iteration and help to escape local optima.

Epoch length (N): Number of iterations at each Temperature (T_0), The feasible changes in configuration at each level of temperature are computed with the formula, $N = n^{2,}$ where 'n' is the problem size or number of facilities [27] [28].

Termination temperature (T_f): The termination temperature determined by the probabilistic configuration using equation 3 [15, 28]. where P_f is the probability of accepting a worse configuration, P_c indicates the acceptance probability at the beginning, which is taken as 95% [28], and CT indicate the total temperature decreasing cycle. The value for P_f is taken as very low $1x10^{-15}$ [15, 28] because at the end, the solution is converged and the algorithm will terminate by accepting a lower bad solution, which makes the algorithm end up with the best configuration. Termination temperature T_f serves as a stopping criterion for the algorithm, and it will be computed using equation 4 [15, 28].

$$CT = 1 + \frac{(\log(\log(Pc) / \log(Pf))}{\log(g)} \quad (3)$$

$$T_f = T_o * g^{CT} \quad (4)$$

Acceptance criteria: the layout configuration is accepted or rejected based on the Metropolis criterion. This criteria include two cases:

https://doi.org/10.71058/jodac.v10i01016

1. The net result of the neighbourhood solution is less than the objective function value: the configuration is accepted without any condition.
2. If the net result of the neighbourhood solution is higher than the objective function value, the chance of accepting a worse solution is increased under the condition. A new layout with a higher cost (i.e., a worse solution) may still be accepted based on a probabilistic criterion defined by the Boltzmann function:

$$P = \exp\left(-\frac{\Delta}{T_0}\right) \tag{5}$$

Where, $\Delta = \text{sum}_{\text{new}} - \text{sum}_{\text{init}}$ represents the difference between the new and current layout costs, and T_0 denotes the current temperature [27].

The above SA parameters, assigned with proper values, ensure that the algorithm converges to a high-quality and near-optimal solution. Ultimately, proper parameter setting is determined through experimental calibration, which enables the algorithm to achieve the best configuration solution.

3.2 Solution Encoding

To encode the facility layout problem, departments or machines are arranged in a grid format, which is represented in Figure 1. The layout arrangement is encoded in a single array, where the index array represents the location number and the alphabet in the array represents the machines or facilities.

The proposed algorithm was implemented on two-level loop structures, the outer loop M iterates over temperature levels, and for each temperature, the inner loop iterates over N random neighbour trails. The program terminates upon completion of M X N loops, or it can terminate at an early stage. The outer loop begins with assigning a high initial temperature (T_0). The assumption is made that the facility layout problem consists of departments or machines located in equal areas and equal distances between each other.

The Solution Encoding process is witnessed in the inner loop (N) process, where both the flow of material and the distance between each department are arranged in matrix format. The program adopts the Manhattan distance formula to calculate the distance between departments. Further, the program converts the block of layout into a list or sequence of departments as shown in Figure 2.

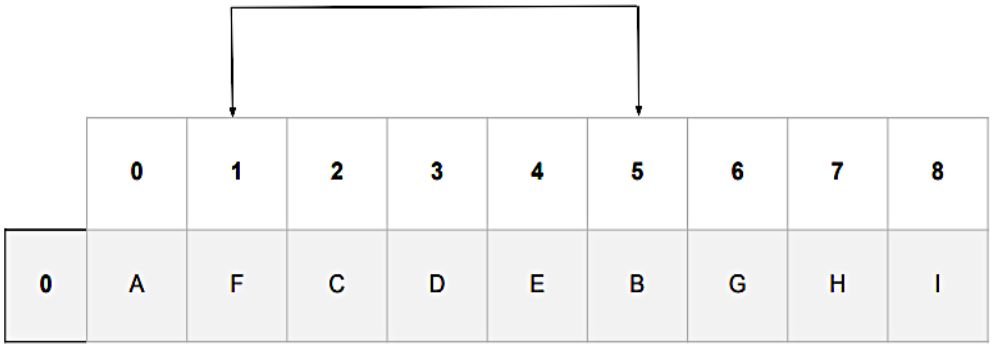

Figure 2. List of departments before swapping

https://doi.org/10.71058/jodac.v10i01016

3.3 Neighbourhood Movement

The swapping operation is used to adopt the neighbouring department. The movement of departments occurs randomly, as illustrated in Figure 3. The swapping operation consists of switching departments from one location to another, and calculating the material handling cost with respect to every swap operation [29]. This swap operation occurs in every inner loop of the algorithm and generates the new layout until it reaches the best cost.

In the proposed study, the programming adopts the swapping operation for the movement of departments. In this operation, departments A-I are randomly selected through their positions. The whole list of a department is labelled as the best label, and the cost for the present layout is calculated and labelled as the best cost. Then, after a random swapping operation, i.e. switching of two departments randomly. a new layout, swapping of location, i=1 of machine "B" with index j=5 of machine "F" and then the algorithm calculates the neighbourhood solution cost for the new layout.

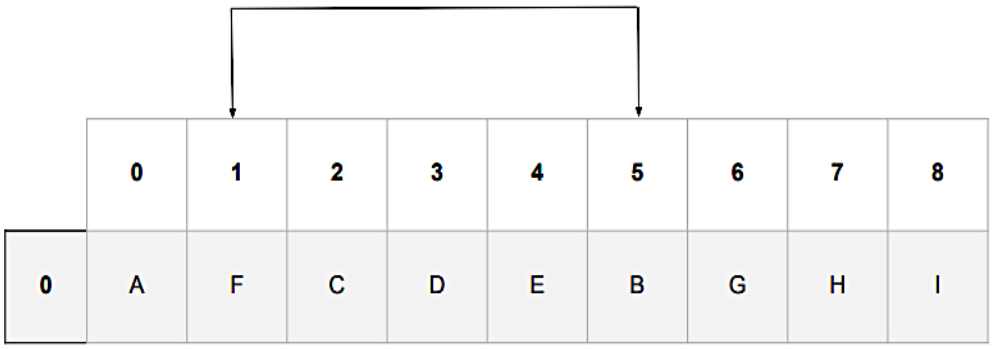

Figure 3. List of departments after swapping

If the cost of the new layout is lower than that of the current layout, the new layout replaces it as both the current and the best layout. Otherwise, the existing layout is retained as the best layout. A new layout is accepted based on the acceptance criteria mentioned in section 3.1. From Equation 5**,** the solution is accepted if $\Delta < 0$, then the new solution yields a lower cost and is directly accepted; if $\Delta < 0$, the probability of accepting a worse solution decreases exponentially with increasing Δ or decreasing T_0. A random number $r \in [0,1]$ is generated and compared with P. If $r < P$, even the worst solution is accepted as the current layout, to escape from local optima. Otherwise, the algorithm rejects it and proceeds to generate a new configuration layout for the next inner iteration.

The iteration process continues till the inner loop satisfies its acceptance criteria (based on the objective function improvement through swapping operations), while the outer loop is controlled by the cooling schedule through the relation $T_f = g^* T_o$, and finally, the algorithm terminates when T_f reaches its final convergence threshold.

3.4 Proposed Simulated Algorithm steps

Step 1. Import Python libraries

i. Importing modules to access their content

https://doi.org/10.71058/jodac.v10i01016

ii. NumPy module supporting large arrays and matrices, along with a collection of high-level mathematical functions
iii. The pandas module offers a powerful data structure and data analysis tools for a one-dimensional array capable of holding any data type
iv. The Matplotlib module creates a static and interactive visualisation, which supports bar and pie charts, histograms, line graphs, colouring, etc.
v. Random module used to select a random element and random shuffling of elements, and returns a random float between 0.0 and 1.0

Step 2. Creating a list of labels and a Manhattan distance matrix

i. Create a label list ('A', 'B', 'C', 'D', 'E', 'F', 'G', ……), consider it has the initial layout and dictionary of labels coordinate {'A': (0,0), 'B': (0,1), 'C': (0,2)……}
ii. Calculate the rectilinear distance, according to the label coordinates, using its formula abs(p1[0]-p2[0]) + abs(p1[1]-p2[1])
iii. Print the rectilinear distance
iv. Input the material flow in a two-dimensional matrix format from the obtained data

Step 3. Parameter initialisation

i. Calculate the cost of the initial labelled layout using equation 1
ii. Set the simulated annealing parameters with assigned values to Initial Temperature (T_0), Cooling Rate (α), Outer loop (M), Inner loop (N)
iii. best cost=[ ], best layout= initial layout, improvement count=[ ]
iv. Iteration history [ ], empty at the beginning to store the iteration history and global iteration [ ] to count all inner trails across the whole run
v. Calculate the cost for the initial layout

Step 4. Annealing Procedure

i. Add termination condition “g”
ii. Initialise the Outer loop (M) for initial temperature T0, No_improvement = False
iii. Initialise the inner loop (N), generate two random column indices for the initial label
    - While loop (random1=random 2)
    - generate the random column index for random 2
    - Implement the swap operation, switching the department of the two random column indices of the initial label
    - Reindex the initial layout label
    - Calculate the fitness function for the NEW and CURRENT_Label
    - Sum_Initial = sum_CURRENT_OBJECTIVE function and Sum_NEW= Sum_NEW_OBJECTIVE
    - Generate Rand_1=probability (0.0-1.0) and calculate Form= exponential (-(Δ) / T0), where Δ= Sum_Initial ~ Sum_NEW
    - Acceptance criteria: if Sum_NEW <= Sum_initial or Rand1<= Form, Initial_label= NEW_label and Sum_Init=Sum_NEW, Improvemnet = TRUE, append Global_Iteration and append Inner_Iteration_History

https://doi.org/10.71058/jodac.v10i01016

iv. Append Min_Cost and append Iteration_History
v. Update IF Sum_Initial <BEST_COST then BEST_COST= Sum_Initial and BEST_Label= CURRENT_Label
vi. NO_Improvement = 0, ELSE append NO_Improvement
vii. Cooling operation by T0 = g * T0
viii. Print the BEST_COST and BEST_LAYOUT

4 Numerical Experiment

4.1 Example 1: The proposed algorithm was developed using the Python programming language (version 3.11.5) in a Jupyter notebook and executed on a MacBook Air M1 processor with 8 GB of RAM. The example problem evaluates the robustness of the proposed algorithm by conducting several experiments using benchmark problems from Nugent et al. and Wilhem and Ward problems, published on the QAPLIB website. Also solved the Mihajlovic et al., 2007 [14] layout problem, which was studied earlier by Mak et al.[2], Chan and Tansri [12], El-Baz [13]. These are the most often used instances in the literature. The details of the problem size and the results are shown in Table 1**.** Supporting the result, Figure 4 shows the optimum layout configuration of Nugent 12, Nugent 14 and Nugent 16, respectively**.** For all the tested cases, the proposed algorithm has achieved an optimal solution for all the instances except size 50.

Table 1. Performance Comparison on Nugent Instances

Nugent	Optimal Value	Obtained Value	CPU Time (Sec)	Layout Type
12	578	578	1.628	4 × 3
14	1014	1014	1.608	5 × 3
15	1150	1150	2.573	5 × 3
16	1610	1610	2.556	5 × 4
17	1732	1732	2.573	5 × 4
18	1930	1930	5.043	5 × 4
20	2570	2570	5.133	5 × 4
21	2438	2438	5.132	7 × 3
22	3596	3596	5.157	11 × 2
24	3488	3488	5.086	6 × 4
25	3744	3744	10.180	5 × 5
27	5234	5234	10.306	9 × 3
28	5166	5166	29.881	7 × 4
30	6124	6124	30.211	6 × 5

The dummy departments are used in some layouts: 14 (1 dummy), 16 (4 dummies), 17 (3 dummies) and 18 (2 dummies).

Table 2. Performance Comparison on Will and Ward Instance

https://doi.org/10.71058/jodac.v10i01016

Wilhelm and ward	Optimal value	Obtained value	Layout type
50	48816	48824	50 x 50

Table 2 contains the Wilhem and Ward facility layout work for a 50 size, where the optimal value obtained is 48816 and from the proposed algorithm, the archived 48824 value is obtained. The solution for problem size 50 from Wilhem and Ward was performed on Google Colab Notebook with multiple runs of iterations. The proposed algorithm achieves a very near-optimal solution with 0.016 % variation compared to the optimum solution, and in lower CPU time.

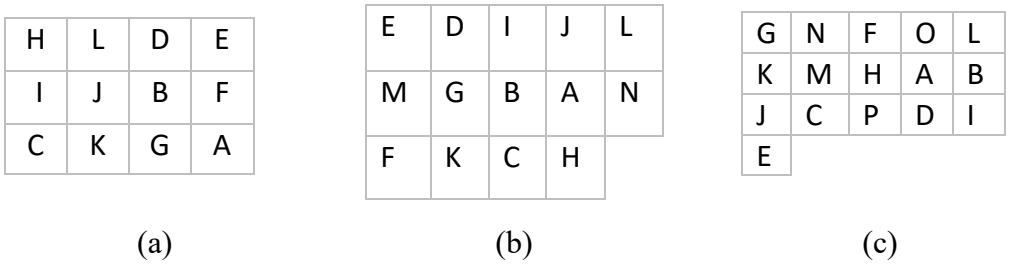

Figure 4. Block of layout of QAPLIB (a) Nugent 12, (b) Nugent 14 and (c) Nugent 16

4.2 Example 2: Furthermore, to examine the performance of the proposed algorithm, a comprehensive experimental study was conducted using a published literature problem from Mihajlovic et al., 2007 [14], and the author used a genetic algorithm to solve the problem. This problem was studied earlier by Mak et al.[2], Chan and Tansri [12], and El-Baz[13]. Mihajlovic et al., 2007 [14], is a well-known optimisation work in manufacturing and operations research. This layout problem contains nine machines, each machine arranged in the layout within a specific location, in a way that the overall material flow must be optimised to a minimum cost. Tables 2 and Table 3 contain the flow of the material and the material handling cost between the machines.

Table 2. Flow of materials between machines

From/To	A	B	C	D	E	F	G	H	I
A		100	3	0	6	35	190	14	12
B			6	8	109	75	1	1	104
C				0	0	17	100	1	31
D					100	1	247	178	1
E						1	10	1	79
F							0	1	0
G								0	0
H									12
I									

Table 3. Material Handling Cost (MHC)

https://doi.org/10.71058/jodac.v10i01016

From/To	A	B	C	D	E	F	G	H	I
A		1	2	3	3	4	2	6	7
B			12	4	7	5	8	6	5
C				5	9	1	1	1	1
D					1	1	1	4	6
E						1	1	1	1
F							1	4	6
G								7	1
H									1
I									

The problem yields a minimum cost of 4818, and the proposed algorithm, with a lower number of experimental trials, reaches the optimum value. Metropolis criterion enhanced the algorithm's ability to escape local optima for the non-improving solutions via probabilistic acceptance, ensuring a robust convergence toward the global best solution.

Table 4 contains the best value of the material handling cost (BEST), and the best trials (#) required to achieve the optimum result. The simulation runs 19 times as in the published work, and is compared with the proposed algorithm result. The comparison observed that the proposed algorithm could achieve an optimal value in the 60th iteration. Which has a less number of trials compared to the published work of Mak et al.[2] , Chan and Tansri [12], and El-Baz [14]. In the 10th and 17th trails the obtained solution had slightly increased as compared to the optimal value (i.e 0.91% and 1.1% inferior to the optimal value).

Table 4. Experimental result

Exp no.	Trail No.	**Proposed Algorithm**	No. of trials	Mihajlovic et al.	No. of trials	M. Adel ElBaz	Mak et al.	PMX (Chan and Tansri)
	#	Best	#	Best	#	Best	Best	Best
1	60	4818	4050	5119	200	5039	5233	4939
2	177	4818	8595	5150	400	4818	5040	5036
3	301	4818	180	4872	1000	4818	4818	4938
4	358	4818	405	4818	2000	4818	4818	4818
5	511	4818	270	4818	5000	4818	4818	4818
6	531	4818	360	4818	400	4872	5225	4938
7	561	4818	2160	4939	800	4818	4927	4992
8	661	4818	1125	4990	2000	4818	4818	4818
9	696	4818	765	4818	4000	4818	4818	4818
10	696	4872	1485	4818	800	4818	5225	4938
11	706	4818	3105	4818	1600	4818	4927	4992
12	724	4818	990	4818	4000	4818	4818	4818
13	748	4818	2160	4818	8000	4818	4818	4818
14	902	4818	3105	4818	2000	4818	5225	4938
15	935	4818	225	4818	4000	4818	4818	4927
16	941	4818	2160	4818	10000	4818	4818	4818

https://doi.org/10.71058/jodac.v10i01016

17	964	4862	3015	4818	4000	4818	4818	4938
18	1173	4818	3240	4818	8000	4818	4818	4862
19	1236	4818	3600	4818	5000	4818	4818	4818
Sum	12881		40995		63200			

From the above experimental result, the sum of all 19 trials of the proposed algorithm is 12,881, and Mihajlovic et al., 2007 [14] is 40995, and Elbaz's [13] is 63200. Which is a very significant improvement against existing published work. The cumulative sum of the proposed algorithm result is 68.57% lower than the cumulative value obtained from the published genetic algorithm approach of Mihajlovic et al., 2007 [14] and 79.6% lower than the cumulative sum of Elbaz's [13] proposed algorithm. Furthermore, the proposed algorithm in 19 conducted experiments could achieve an optimum value on an average of 677.94 iterations per experiment, and took an average computational time of 2.07 seconds per experiment. This statistical behaviour expresses the high efficiency of the algorithm to achieve an optimal value in fewer iterations with minimum computational time. Figure 5 shows the top four optimised layouts generated using the proposed algorithm.

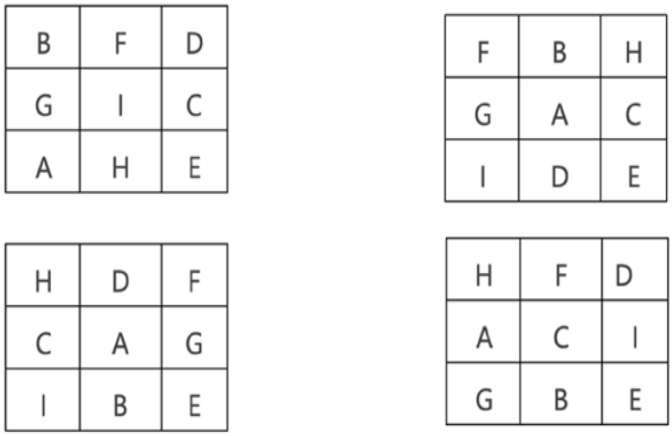

Figure 5. Optimal layout configuration for the top top-most four trials

4.2.1 Data Extraction:

From the above SA programming, it is observed that outer iteration "M" runs for initial layout cost times and inner iteration runs N= n**2 times (n = size of the layout). The program generates M * N times the total number iteration.

In this numerical example, the obtained initial layout cost is 7662, and the size of the layout is 9 in the used problems. This gives a total number of 6,20,622 solutions for each iteration. It is difficult to generate and extract the best cost and best layout from a large pool of solutions. Using the final temperature(T_f) termination condition, the solution space is reduced to 10,368. This reduction of the solution space will improve the efficiency in finding the optimal solution and speed up computer responsiveness. An Exploratory Data Analysis (EDA) have been conducted using the Python open source library pandas diagnose the solution space. The analysis extracts various important statistical data, including

https://doi.org/10.71058/jodac.v10i01016

maximum cost, minimum cost, mean and standard deviation of the cost. Table 5 lists the data inspection caried for the above numerical problem.

Table 5. Data profiling

sample(10)	First top 10 sample data (cost, layout and iteration count)
describe(include='all')	Statistics Summary (getting quick statistical profile)
['Cost'].value_counts()	Frequency analysis (finding outlier of cost column)
shape	Meta data (checking the scale and volume of data)

From the above statistical data, it is found that the minimum number iteration required for achieving the optimal value for all 19 trials. Figure 6 shows the graph of iteration vs cost for the 8th experiment in Table 4. The x-axis denotes the iteration count ranging from 0 to 12000, and the y-axis represents the material handling cost. In the initial phase, the graph shows significant cost fluctuations, reflecting the exploratory nature of the algorithm and is ready to accept both better and worse solutions to escape a local optimum solution. As the iterations progress into the middle phase, the fluctuations begin to decrease, indicating a cooling process wherein the algorithm becomes increasingly selective in accepting suboptimal solutions and at last, the cost values stabilise, suggesting convergence toward a near-optimal solution, with minimal or no further improvement observed.

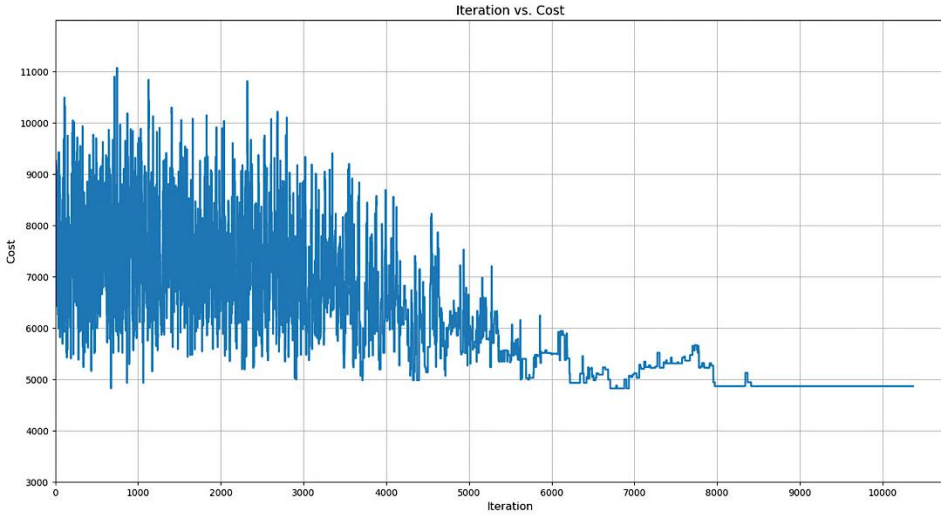

Figure 6. Iteration vs. Cost graph

The convergence plot generated for the above graph is shown in Figure 7, which indicates the optimisation progression for the numerical problem. At each iteration, the algorithm found a layout of lower cost (y-axis) than before. After many iterations (x-axis), the 661 has the first optimum value, and the 6708 above the algorithm did not find a better solution

https://doi.org/10.71058/jodac.v10i01016

than the current best cost. The flat line indicates the algorithm converged and cannot find a better solution anymore.

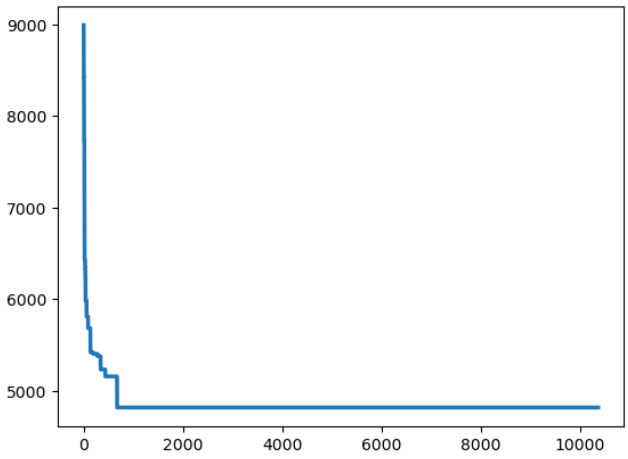

Figure 7. Convergence plot

Figure 8 represents the comparison of the initial and best layout after the optimisation process using the simulated annealing algorithm. The initial configuration as MHC of 7664 (left side) and the optimised configuration achieve the cost of 4818 (right side).The result highlights the algorithm ability to balance the exploration and exploitation through probabilistic acceptance of inferior solution in early iterations and gradual refinement as temperature decreases.

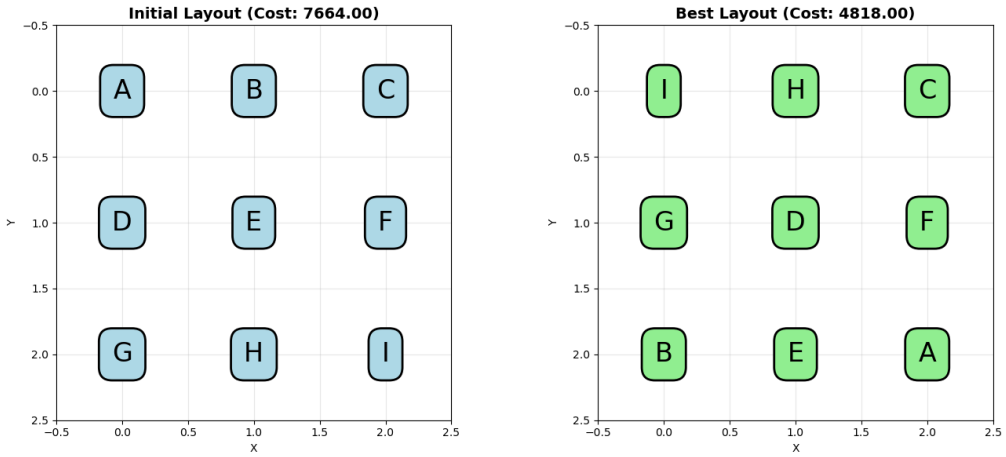

Figure 8. Layout Arrangement

Overall comparison confirms the proposed SA algorithm is a robust metaheuristic for solving the facility layout problem and capable of producing high quality layout with significantly lower material handling cost than the initial configuration. It demonstrate the significant improvement of material handling efficiency.

https://doi.org/10.71058/jodac.v10i01016

4. Conclusion

The paper proposed the simulated annealing algorithm to solve the facility layout problems. The SA algorithm addresses static equal-area layout problems by assuming equal distances between departments. The algorithm is examined with the various benchmark problems and published work by Mihajlovic et al. (2007) [14]. The statistical result achieved by the algorithm illustrates proposed algorithm has the capacity to extract the solution from a large solution space and achieve the optimal solution in a given minimum CPU time, and this result is better compared with the published work in the literature.

Reference

[1] *Irappa Basappa Hunagund, V. Madhusudanan Pillai, and Kempaiah U.N., "A survey on discrete space and continuous space facility layout problems," J. Facil. Manag., vol. 20, no. 2,* ***(2021),*** *pp. 1472–5967.*

[2] *K. L. Mak, Y. S. Wong, and F.T.S Chan, "A Genetic Algorithm for Facility Layout Problems," Comput. Integr. Manuf. Syststem, vol. 11, no. 1–2,* ***(1998)*** *pp. 113–128.*

[3] *V. Madhusudhan Pillia, Irappa Basappa Hunagund, and Krishna K. Krishnan, "Design of Robust Layout for Dynamic Plant Layout Problems," Comput. Ind. Eng., vol. 61, no. 3,* ***(2011),*** *pp. 813–823.*

[4] *Amine Drira, Henri Pierreval, and Sonia Hajri-Gabouj, "Facility layout problems: A survey," Annu. Rev. Control, vol. 31,* ***(2007),*** *pp. 255–267.*

[5] *S. P. Singh and R. R. K. Sharma, "A review of different approaches to the facility layout problems", Int. J. Adv. Manuf. Technol., vol. 30,* ***(2006)****, pp. 425–433,*

[6] *Ramazan Şahin and Orhan Türkbey, "A simulated annealing algorithm to find approximate Pareto optimal solutions for the multi-objective facility layout problem," Int. J. Adv. Manuf. Technol., vol. 41,* ***(2009),*** *pp. 1003–1018.*

[7] *Koopmans, T. C., & Beckmann, M, "Assignment problems and the location of economic activities", Econometrica, Vol. 25, no. 1,* ***(1957),*** *pp 53–76.*

[8] *Burkard, R. E., Çela, E., Pardalos, P. M., & Pitsoulis, L. S, "The quadratic assignment problem", In Handbook of Combinatorial Optimisation, pp. 1713–1809,* ***(1998)****.*

[9] *Wen-Chyuan Chiang, Panagiotis Kouvelis, and Timothy L. Urban, "Incorporating Workflow Interference in Facility Layout Design: The Quartic Assignment Problem," Manag. Sci., vol. 48, no. 4,* ***(2002)****, pp. 584–590.*

[10] *John Holland "Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence". MIT Press,* ***(1992).***

[11] *Amir Sadrzadeh, "A genetic algorithm with the heuristic procedure to solve the multi-line layout problem," Comput. Ind. Eng., vol. 62, no. 4,* ***(2012)****, pp. 1055–1064.*

https://doi.org/10.71058/jodac.v10i01016

[12] *K. C. Chan and H. Tansri, "A study of genetic crossover operation on the facilites layout problems," Comput. Ind. Eng., vol. 26, no. 3,* ***(1994),*** *pp. 537–550,*

[13] *Elbaz M.Adel," A genetic algorithm for facility layout problems of different manufacturing environments",Comput.Ind. Eng., Vol.47,* ***(2004)****, pp.233-246.*

[14] *Mihajlovic, Z. Zivkovic, and N.Strbac, "Using genetic algorithms to resolve facility layout problem," Serbian J. Manag., vol. 2, no. 1,* ***(2007),*** *pp. 35–46.*

[15] *Min. Dong, Chang Wua, and Forest Hou, "Shortest path based simulated annealing algorithm for dynamic facility layout problem under dynamic business environment," Expert Syst. Appl., vol. 36, no. 8,* ***(2009),*** *pp. 11221–11232.*

[16] *Adil Baykasogylu and Nabil N.Z. Gindy, "A simulated annealing algorithm for dynamic layout problem", Comput & Oper Res., vol.28,* ***(2001)****, pp.1403-1426.*

[17] *Ramazan Şahin & Orhan Türkbey, "A new hybrid tabu-simulated annealing heuristic for the dynamic facility layout problem", Int J. of Prod Res, vol. 47, no. 24,* ***(2009),*** *pp. 6855-6873.*

[18] *Komarudin and Kuan Yew Wong, "Applying Ant System for solving Unequal Area Facility Layout Problems," Eur. J. Oper. Res., vol. 202, no. 3,* ***(2010),*** *pp. 730–746.*

[19] *Janny Leung, "A new graph-theoretic heuristic for facility layout," Manag. Sci., vol. 38, no. 4,* ***(1992)****, pp. 594–605.*

[20] *S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimisation by Simulated Annealing," Science, vol. 220, no. 4598,* ***(1983),*** *pp. 671-680.*

[21] *Alwalid N. Elshafei, "Hospital Layout as a Quadratic Assignment Problem," Oper. Res., vol. 28, no. 1,* ***(2016)****, pp. 167–179.*

[22] *Jiazhen Huo, Jing Liu, and Hond Gao, "An NSGA-II Algorithm with Adaptive Local Search for a New Double-Row Model Solution to a Multi-Floor Hospital Facility Layout Problem," Appl. Sci., vol. 11, no. 4,* ***(2021),*** *pp. 2–22.*

[23] *Vahit Tongur, Mehmet Hacibeyoglu, and Erkan Ulker, "Solving a big-scaled hospital facility layout problem with meta-heuristics algorithms," Eng. Sci. Technol. Int. J., vol. 23, no. 4,* ***(2020),*** *pp. 951–959.*

[24] *Ghorbanali Moslemipour and T. S. Lee, "Intelligent design of a dynamic machine layout in uncertain environment of flexible manufacturing systems," J. Intell. Manuf., vol. 23,* ***(2012)****, pp. 1849–1860.*

[25] *Berna Ulutas and A. Attila Islier, "Dynamic facility layout problem in footwear industry," J. Manuf. Syst., vol. 36,* ***(2015),*** *pp. 55–61.*

[26] *William H. Press, Saul A. Teukolsky, and William T. Vetterling, "Numerical Recipes The Art of Scientific Computing", Third Edition. Cambridge University Press,* ***(2007).***

[27] *Park, M.W. and Kim, Y.D, "A systematic procedure for setting parameters in simulated annealing algorithms", Comp and Operat Res, vol. 25, no. 3,* ***(1998),*** *pp. 207-217.*

[28] *Irappa Basappa Hunagund, V. Madhusudanan Pillai, and Kempaiah U.N., "Design of robust layout for unequal area dynamic facility layout problem*

https://doi.org/10.71058/jodac.v10i01016

with flexible bay structure " J. Facil. Manag., vol. 20, no. 2, ***(2020)****, pp. 1472–5967.*

[29] *Irappa Basappa Hunagund, V. Madhusudanan Pillai, and Kempaiah U.N., "A simulated annealing algorithm for unequal area dynamic facility layout problems with flexible bay structure" Int J of Ind Eng Comp., vol. 09,* ***(2018),*** *pp. 307-330.*