Decision Science Letters 12 (2023) 69–88

Contents lists available at GrowingScience

Decision Science Letters

homepage: www.GrowingScience.com/dsl

A novel memory-based simulated annealing algorithm to solve multi-line facility layout problem

Kamran Zolfi[a], Javid Jouzdani[b*] and Hadi Shirouyehzad[c]

[a]*Department of Industrial Engineering, Lenjan Branch, Islamic Azad University, Isfahan, Iran*
[b]*Department of Industrial Engineering, Golpayegan University of Technology, Golpayegan, Iran*
[c]*Department of Industrial Engineering, Najafabad Branch, Islamic Azad University, Najafabad, Iran*

C H R O N I C L E

Article history:
Received August 22, 2022
Received in revised format: September 30, 2022
Accepted October 19, 2022
Available online October 19, 2022

Keywords:
Multi-line Facility Layout Problem
Memory-based Simulated Annealing
Layout

A B S T R A C T

In this paper, a memory-based simulated annealing algorithm called the Dual Memory Simulated Annealing Algorithm (DMSA) is presented to solve multi-line facility layout problems. The objective is to minimize the total material handling cost. Two memory buffers and a restart mechanism are considered. Two benchmark problems were selected from the literature review papers and solved using the standard simulated annealing (SA) algorithm and the DMSA. The obtained results show that solutions provided by the DMSA algorithm are cost-effective compared to the standard SA algorithm and the other algorithms used for solving these test cases. Moreover, to further evaluate the performance of the DMSA algorithm in large scale problems, eleven test cases were selected from the benchmark library of the quadratic assignment problem (QAP). According to the results, the performance of the algorithm in finding solutions to complex problems is exemplary.

1. Introduction

Many industries and service providers face different forms of facility layout problem (FLP) including but not confined to designing the layout for hospitals, schools, landing fields, stock, circuit boards, and backboard wiring (Feng & Che, 2018). A facility may be a machinery tool, a workstation, a warehouse, a department, or a manufacturing unit, etc. (Besbes et al., 2019). A FLP is about determining the most effective department arrangement inside a facility while considering certain goals and limitations (Hasda, 2017). There is no accurate and common definition of layout problems due to the vaiety of published articles. For example, in Lee and Lee (2002), the FLP is introduced as the arrangement of multiple facilities with unequal areas in a general space such that they are placed along the length and width of a factory in a way that minimizes the total costs of material handling and slack area.

According to McKendall Jr et al. (2006), an effective arrangement inside a facility leads to a good workflow among resources. An effective establishment helps other operations that rely on workflow have a better performance. For example, in a manufacturing factory, an effective arrangement can be defined as the flow of materials among machinery in a way that a correct amount of raw materials is supplied to machinery at the precise time in such a way that worker safety is observed and the accumulation of materials in the manufacturing flow is prevented. Moreover, an effective arrangement reduces the cost of material handling. According to Kheirkhah and Bidgoli (2016), given that in past studies the cost of material handling constitutes about 20 to 50 percent of the total operating cost and 15 to 70 percent of the total manufacturing cost, therefore, the minimization of material cost is one of the most common objective functions in FLP.

* Corresponding author.
E-mail address: jouzdani@gut.ac.ir (J. Jouzdani)

doi: 10.5267/dsl.2022.10.005

In the QAP, it is assumed that the areas of the facilities are equal, and the distance between them is considered equal to the distance between the center of facilities (C. Chen & Tiong, 2019). In most papers concerning the FLP, centers of departments are assumed to be the distance between two departments. Still, other cases can also be found in the literature review, such as considering loading and unloading points (Xiao et al., 2019).

Since FLP is NP-hard, different models and algorithms have been presented in the recent decade (Neghabi et al., 2014), and several exact and approximate methods have been used by researchers to solve it (Kothari & Ghosh, 2014). Since exact methods require huge memory capacity and computations, these methods are not efficient to solve large-scale problem instances, therefore, heuristic and meta-heuristic approaches have been used increasingly (J. Guan & Lin, 2016). Methods used to solve FLP mainly include exact methods (Ahmadi & Jokar, 2016; Hammad et al., 2016; Neghabi & Ghassemi Tari, 2015), heuristic methods (Armour & Buffa, 1963; Drezner, 1987; Kumar et al., 1995) and meta-heuristic methods such as genetic algorithm (GA) (Aiello et al., 2012; Paes et al., 2017), tabu search algorithm (McKendall Jr & Liu, 2012; Samarghandi & Eshghi, 2010), simulated annealing algorithm (SA) (Matai, 2015; Şahin, 2011), particle swarm optimization (Asl & Wong, 2017; Hu & Yang, 2019), and ant colony (Baykasoglu et al., 2006; Kulturel-Konak & Konak, 2011b).

If the facility has a regular area with a rectangular shape, the FLP can be classified into one-row, multi-row, open field, loop, and multi-floor layouts according to the material flow path between facilities (Hu & Yang, 2019). One method to represent the layout of irregular facilities with unequal areas is the space-filling curve method, where departments are joined in a cascaded mode with no disconnections (C. Guan et al., 2019; Wang et al., 2005).

The single-row FLP (SRFLP) includes several rectangular facilities arranged on a line. The multi-row FLP (MRFLP) also tries to arrange departments on the plant floor so that the material handling cost is minimized (Neghabi et al., 2014). Precise methods using mathematical methods seek the best solution and generally employ a mathematical programming formulation (Vázquez-Román et al., 2019). Unlike heuristics, metaheuristics are a general form of the search process in the solution space regardless of the considered problem, without guaranteeing finding the optimal solution. However, they can provide desirable solutions in less computational time in comparison to full space search methods (de Sousa Junior et al., 2020). The meta-heuristic algorithm used here is the simulated annealing algorithm, which is one of the well-known local search algorithms, developed in the early 1980s (Pan et al., 2019). This algorithm is a stochastic memoryless approach that uses a completely random rule in each iteration and doesn't withdraw the information obtained among searching. Thus, the quality of the final solution may be reduced (Rabbouch et al., 2019). Regarding these weaknesses and by using an auxiliary memory, the proposed algorithm tries to use the search information and improve the quality of the final result.

In this research, the FLP has been analyzed in the form of a multi-line facility layout (MLFLP). The objective is to minimize the material handling cost, which has been used in most research conducted on the FLP. Also, a memory-based SA algorithm is utilized as a near-optimal solution to solve the test cases mentioned in related papers.

The contribution of this research can be stated as follows:

- The standard SA algorithm was extended, and a new algorithm, namely DMSA, was proposed in which two auxiliary memories and a restart mechanism were used.
- Two MLFLP benchmark test cases were solved via the DMSA algorithm, and the solutions obtained were better than the results of other algorithms in previous research.
- Eleven QAP benchmark problems were selected to assess the efficiency of the proposed algorithm in large-scale problems. The results of implementing these problems by the proposed method were compared with the results of the standard SA algorithm and three other approaches, showing that the proposed algorithm outperformed other methods.

In the next section, past studies conducted about FLPs will be reviewed. The Problem definition is described in section 3. The proposed algorithm will be evaluated in section 4 and examined using numerical examples in section 5. Finally, section 6 offers conclusions.

2. Literature review

Facility layout is a problem that focuses on the way departments are arranged in the work area. Since a suitable layout design leads to increased operational performance, effective use of space and reduction of handling, it is a strategic problem and has a major effect on the performance of a production system. For this reason, companies usually aim to evaluate current layout designs and try to design an appropriate layout for their facilities (Domschke & Krispin, 1997). According to Balakrishnan et al. (2003), an FLP falls under the category of NP-hard problems. For example, in a dynamic facility layout problem (DFLP), if N is the number of facilities and T is the planning horizon, the number of layouts that must be examined to reach the optimal solution is $(N!)^T$. Hence, it is practically impossible to find an optimal solution for problems with large dimensions via existing commercial software. Accordingly, it is recommended to employ heuristic algorithms to solve this problem.

Armor and Buffa (1963) formulated the unequal area FLP (UA-FLP) by considering the factory floor in the blocked form. In the study, a modified form of the QAP was utilized, in which departments were divided into blocks of the same areas. If the length and width of departments are not assumed constant in the UA-FLP, the shape of departments will be flexible. Kulturel-Konak and Konak (2011a) used the representation of a flexible bay structure to find location and size of departments in the model. Researchers have utilized a variety of representations for the UA-FLP. Meller and Bozer (1996) used an SA algorithm and space-filling curve to address the problem of one-floor and multi-floor facility layout. Asef-Vaziri et al. (2017) used a flexible bay structure, and Scholz et al. (2009) employed the slicing tree method. Li and Mashford (1990) utilized the GA to solve the quadratic assignment problem (QAP). The GA is a heuristic optimization algorithm introduced by Holland (1975), and is used to solve hybrid optimization problems.

In a research by Tam (1992), the GA was employed to solve an FLP with departments with unequal areas. The SA was presented by Kirkpatrick et al. (1983) to solve hybrid optimization problems. This algorithm is a probability method obtained from the physical concept of annealing metals. If metal is cooled too rapidly, a defect may appear in it. The SA algorithm is a random pair-swap heuristic that avoids getting stuck in a local optimum by considering non-optimal swaps. The first research that used SA to solve the QAP was conducted by Burkard and Rendl (1984). Some researchers have used a memory buffer and/or a restart mechanism to optimize the SA algorithm. For example, Zou et al. (2017) utilized a triple-memory buffer to solve the problem of fixed outline floor-planning with soft blocks which periodically stores the best solution in a memory buffer and extracts the oldest solution, and when the number of sequential failures reaches the maximum allowed limit, the algorithm refers to one of the solutions stored in the memory and the temperature is reset adaptively. In a paper by Vincent et al. (2017), to solve the vehicle routing problem (VRP), if the best solution was not optimized after every 10 rounds of temperature reduction, the SA algorithm was reset to the initial temperature and a primary new solution was generated.

The Tabu search (TS) algorithm is another heuristic algorithm introduced by Glover (1989). The first research to use the TS algorithm to solve the QAP was pubished by Skorin-Kapov (1990) in which the algorithm is a pair-swap heuristic that uses a memory (tabu list) to store the number of recent swaps. The traditional decision-making approach used for layout problems is only based on a criterion with one objective function and is based on the minimization of the total handling cost. However, in order to create an appropriate layout, the sole minimization of handling costs is not sufficient, instead, there are several other quantitative and qualitative objectives that affect the creation of a good layout. Therefore, it is logical to select a layout that can consider all important design criteria. Layout problem solving via a multi-objective optimization approach was introduced by Rosenblatt (1979). He considered the two factors of cost and the relationships of the activities of departments for objective functions and used a weighted objective function consisting of both factors to solve the problem. Samarghandi and Eshghi (2010), employed an adaptive memory of solutions for an intensification strategy and a probability method for the selection of bad solutions for a diversification strategy in the TS algorithm.

Solimanpur and Jafari (2008) presented a mathematical mixed-integer planning model for the FLP in two dimensions and used a Branch and Bound (BB) algorithm to obtain the optimal solution of the problem, although the corresponding algorithm to solve large scale problems is inefficient.

Şahin and Türkbey (2009) used the two criteria of cost and closeness rating for the FLP, and utilized the SA algorithm and the Pareto concept to solve the problem. Bashiri and Dehghan (2010) utilized a three-stage approach for solving a multiple-criteria DFLP. In the first stage, they solved a classic layout model and considered the solution as a suitable primary solution. Next, by considering other criteria, they optimized this solution. In the second stage, the Decision-Making Units (DMUs), inputs, and outputs were defined. Any change from any layout of a certain point in time to another layout of another point in time was considered to be a DMU. In the last stage, the problem was modeled and solved, the DFLP was combined with Data Envelopment Analysis (DEA) and the proposed model was solved via the global criterion method. Ultimately, the most effective layout was selected. G. Chen and Rogers (2009) offered a qualitative and quantitative objective function for the FLP. The objective of the qualitative part was to maximize the closeness rating for departments. The quantitative objective function was to minimize distance based on cost.

Hasani and Soltani (2015) proposed a hybrid model of a DFLP and a transportation system design where the transportation method is selected by TOPSIS. In a paper by Rezazadeh et al. (2009), the PSO was employed to solve the DFLP. Barzinpour et al. (2019) utilized the invasive weed optimization (IWO) algorithm to solve the DFLP, and the performance efficiency of the solution in small, medium and large problems were compared. Ulutas and Islier (2015), considering the seasonal demand changes in the footwear industry and using real-life data, presented a clonal selection-based algorithm for the real-life DFLP.

In a paper by Kulturel-Konak (2017), a hybrid meta-heuristic approach with concepts related to TS and mathematical programming was proposed. In this method, the relative locations of departments and the allocation of those locations to sections is done via the TS algorithm while their exact location is identified through mathematical programming. According to Cravo and Amaral (2019), the SRFLP includes facility layout along a direct line such that the total weight of distances between each facility pair is minimized. In their paper, a greedy metaheuristic problem-solving method was employed to solve samples in sizes ranging up to 1000. Finally, the numerical results showed the efficiency of their proposed algorithm.

Out of 93 instances the algorithm provided 29 instances were with improved results. Kalita and Datta (2018) studied the SRFLP in which some placement or/and a determined ordering constraint were imposed on some facilities. Gai and Ji (2019) regarding the FLP for healthcare services, proposed an integrated approach to solve the problem with two quantitative and qualitative objectives. In the first phase, considering the transportation cost as the objective to be minimized using a mathematical programming model, several alternative facilities layouts were generated. Then, in the second phase, considering the qualitative criterion, the alternatives were ranked using multi-attribute group decision-making. Amar et al. (2018), regarding the environmental constraints and production efficiency requirements, considered the FLP, in which minimization of the CO2 emissions as the environmental factor of the corresponding system was considered.

Turanoğlu and Akkaya (2018) introduced bacterial foraging optimization (BFO) for solving DFLPs. In their study, a new heuristic hybrid algorithm called simulated annealing bacterial foraging optimization (SABFO) was presented for FLP for which the parameter adjustment was done by Taguchi design of experiments. The proposed algorithm was tested on problems used in subject studies and satisfactory results were obtained in a reasonable calculation time. Rubio-Sánchez et al. (2016) considered two common methods namely GRASP and path relinking (PR) which effectively searches high-quality paths for SFLPs. Ultimately, non-parametric tests were employed to identify the differences between the algorithms. Kang et al. (2018), first, proposed a Cuckoo Search Algorithm for the closed loop-based FLP in which the cells could be located inside or outside of the loop. Although the SA algorithm is likely to escape local optima due to selecting poor solutions, it may eventually be trapped at a local optimum at low temperatures. Allahyari and Azab (2018) used a SA algorithm with multiple starts to increase the algorithm's search possibility to solve the UA-FLP. To improve the results of the SA algorithm, Palubeckis (2017) combined it with the variable neighborhood search (VNS) algorithm so that the solutions of the SA algorithm were used as the input of the second algorithm solution. Anjos et al. (2018) modeled a special case of a MRFLP, in which the departments were considered one-dimensional, the space between the two departments was assumed acceptable, and the distances between two departments were considered as the horizontal distance between the departments.

Hu and Yang (2019) modeled the MLFLP in semiconductor fabrication and the PSO algorithm was used to find a near-optimal solution. According to Safarzadeh and Koosha (2017), the MRFLP is a special type of FLPs where facility layout in a number of fixed rows is selected in a way that minimizes material handling costs. Today, based on new needs existing in the area of layouts, MRFLPs have many applications. In their paper, four fundamental hypotheses are considered. The mathematical formula of these hypotheses was written using a nonlinear mixed-integer programming model with fuzzy constraints. They solved the problem via the GA that enables finding the best solution with minimum opportunity costs. J. Liu et al. (2018) focused on FLPs with unequal areas. Their innovation was employing an objective space-division method in the multi-objective PSO (MOPSO) algorithm. Kulturel-Konak (2019) presented a novel mathematical model for the zone-based UA-FLP. The proposed approach combined SA, variable neighborhood search, and mixed-integer programming. The dimensions of departments were considered as decision variables, and the departments are allocated to flexible sections with pre-structured placements. Atta and Mahapatra (2019) proposed a population-based heuristic algorithm by considering a local search to solve the SRFLP. S. Liu et al. (2021) studied the SRFLP with various types of constraints such as positioning and ordering constraints and relations between departments and utilized a fireworks algorithm to solve the problem. Herrán et al. (2021) proposed a variable neighborhood search algorithm to solve a particular case of the MRFLP in which no free space was permitted between two adjacent facilities, and the space between the leftmost department and the left margin should be zero. Uribe et al. (2021) proposed a The Greedy Randomized Adaptive Search Procedure algorithm along with an improved local search for the Multiple-Row Equal FLP based on efficient calculations of the objective function. Further, they employed a probabilistic method to choose the solutions. Lakehal et al. (2021) presented two metaheuristic algorithms based on the biogeography-based optimization for QAP-formulation of FLP. They used a parallel computation method to reduce the algorithm runtime and enable further scanning of the search space. S. Liu et al. (2022) studied Double-Row FLP by considering positioning, ordering, and relation constraints and proposed a differential evolution algorithm to solve the problem on a large scale.

3. Problem definition

The MLFLP arranges some facilities on several lines on the plant floor so that the number of facilities is less than the number of locations, and facilities do not overlap each other. Nevertheless, in these problems, some products are usually manufactured in different volumes and different routings (Sadrzadeh, 2012). Fig. 1 presents an instance of a multi-line facility in a 3×4 grid with eight facilities.

	1	6	
5	8	4	3
7		2	

Fig. 1. A sample multi-line facility layout

The total material handling cost or the total cost in short (C), can be defined by equation 1. The model presents the simple MLFLP where each facility is at most in one location.

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

$$\sum_{i=1}^{n} x_{ik} \leq 1 \quad \forall k \in M \tag{2}$$

$$\sum_{k=1}^{n} x_{ik} = 1 \quad \forall i \in N \tag{3}$$

$$d_{ij} = \sum_{k=1}^{M}\sum_{l=1}^{M} dis_{kl} * z_{ijkl} \quad \forall i,j \ \in N \tag{4}$$

$$z_{ikjl} <= x_{ik} \qquad \forall i,j \in N, k,l \in M \tag{5}$$

$$z_{ikjl} <= x_{jl} \qquad \forall i,j \in N, k,l \in M \tag{6}$$

$$x_{ik} + x_{jl} - 1 <= z_{ikjl} \qquad \forall i,j \in N, k,l \in M \tag{7}$$

$$x\ , z\ is\ binary\ variable \tag{8}$$

where i and j are facilities, N is the total number of facilities, $M = r * c$ is the total number of locations, r is the number of rows, c is the number of columns in the grid, k and l are locations, dis_{kl} is the rectilinear distance from location k to location l, f_{ij} is the amount of material flow from the facility i to the facility j, c_{ij} is the cost of moving one unit of materials from the facility i to the facility j, and d_{ij} is the rectilinear distance between the centroid location of the facility i and the facility j. According to constraint (2), there is at most one facility at a single location, and according to constraint (3), every facility is located at a single location. If facility i is in location k, x_{ik} is set to 1; otherwise, it is set to zero. If facility i is in location k, and facility j is in location l, z_{ijkl} is set to 1; otherwise, it is set to zero.

3.1 Dual-Memory Simulated Annealing Algorithm (DMSA)

The simulated annealing algorithm is a probability-based meta-heuristic algorithm introduced by Kirkpatrick et al. (1983). Using the similarity between annealing metal and optimal solution search, this algorithm is used in optimization problems. In physics, annealing is a process where metal is first heated up to a very high temperature and then slowly cooled until it reaches the minimum energy level. In fact, simulated annealing is a replication of this process. At a high temperature, the chance of accepting low-quality neighbors is higher which allows for hill climbing in search. As the temperature gradually drops, the search for finding higher quality solutions increases until the system reaches the equilibrium state which is actually the optimal or sub-optimal point (Allahyari & Azab, 2018). In this method, the algorithm accepts worse solutions with a probability P (Grobelny & Michalski, 2017) where $P = e^{-\Delta/(K_b * T)}$ and K_b is Boltzmann constant and T is the temperature.

One of the shortcomings of the standard SA algorithm is the possibility to trap into the local optimum. For this purpose, the proposed algorithm's capability to escape from the local optimum and move toward the global optimum has increased using two local and global memories and a restart mechanism. A local memory buffer stores solutions generated at a temperature and a global memory buffer stores a portion of the total search space. If no improvement is made in an algorithm implementation process, solutions stored in the global memory buffer will be used as the restarting point. In what follows, the parameters, variables, solution representation, and the steps of the algorithm are discussed.

3.2 Parameters and Variables

The parameters and variables of the proposed algorithm are explained in Table 1 and Table 2, respectively.

Table 1
The parameters of the proposed algorithm

Parameter	Description
$alpha$	Temperature reduction coefficient
$localIteration$	The number of local searches at every temperature
T_0	The initial temperature
T_f	The final temperature
$maxLocalStagnation$	The maximum allowed number of times that temperature decreases and no changes happen in the best local solution
$maxGlobalStagnation$	The maximum allowed number of times that temperature decreases and no changes happen in the best global solution
$globalMBRatio$	A percentage of the global memory buffer that is allocated to global solutions
$restartMBRatio$	A percentage of the global memory buffer that can be used for restarting
$bufferSize$	Shows the size of local and global buffers
K_b	Boltzmann constant

Table 2
The variables of the proposed algorithm

Variable	Description
$localBuffer$	The local memory buffer where the solutions of a temperature degree are stored
$globalBuffer$	It is the global memory buffer
$bestSol$	The best local solution. It is the best solution that exists for the current solution during search and is passed to the next neighbor
$GlobalbestSol$	The best global solution. It is the best solution that we found in the entire search and is considered as the best solution obtained using the algorithm.
$localStagnation$	The number of times that the temperature has reduced but no change occurred in the best local solution.
$globalStagnation$	The number of times that the temperature has reduced but no change occurred in the best global solution
i	It represents the solution storage index in the local memory buffer
$temperature$	The current temperature
Sol	The current solution

3.3 Solution Representation

One of the most important steps in developing the solution representation. In The proposed algorithm, each solution is represented by a structure with the following components in Table 3.

Table 3
The components of the solution structure

Component	Description
$cost$	The total material handling cost related to solution layout
$temperature$	The temperature where the solution was acquired
$bestSol$	The best-known local solution
$layout$	The solution layout

For layout encoding, a one-dimensional array is utilized that is acquired by using a space-filling curve of the corresponding layout. As an example, Fig. 2 shows an example of layout encoding with 13 facilities related, and Fig. 3 depicts the corresponding solution representation. The empty locations are shown with numbers higher than the number of the facilities which in the corresponding example include number 14, 15, and 16.

7	2	3	
1	6	8	4
	5	12	10
9	11	13	

Fig. 2. An example of multi-line facility layout problem

7	2	3	14	4	8	6	1	15	5	12	10	16	13	11	9

Fig. 3. The encoded solution for the layout shown in Fig. 2

3.4 Steps of the Algorithm

The steps and the pseudo-code of the algorithm are described in what follows.

The first step: parameter and variable initialization

In this step, the parameters, i.e. T_f, T_0, $localIteration$, $alpha$, $maxGlobalStagnation$, $maxLocalStagnation$, $bufferSize$, $restartMBRatio$, and $globalMBRatio$, are initialized. $localBuffer$ and $globalBuffer$ are considered as an array of the solution structure with the size of $bufferSize$. The variables of $localStagnation$ and $globalStagnation$ are initialized and set to be zero. The first index of the memory buffer is considered as the initial index, and the current temperature is set to T_0. Value of $bufferSize$ is equal to or greater than value of $localIteration$.

The second step: creating the initial solution

A solution, sol, is created as a random permutation of facilities with empty locations. The $bestSol$ component of sol is considered to be sol and the temperature component of sol is considered to be current temperature. The best local solution and global solution are set to sol, and all the elements of the local and global memory buffers are initialized with this solution.

The third step: creating a neighbor solution

A new solution $newSol$ is created in the neighborhood of the current solution. The best local solution related to $newSol$ is set with the best local solution and its related temperature is also considered as the current temperature. New solutions are generated through the following two types of neighborhood generation.

1) Random Swap Neighborhood: Two facilities are randomly selected and their locations are swapped. In Fig.4, an example of the pair-swap of facility 7 and 3 is shown.

Current Solution	4	8	2	**7**	3	10	11	15	9	**5**	13	1	12	14	6	16
Neighbor Solution	4	8	2	**5**	3	10	11	15	9	**7**	13	1	12	14	6	16

4	8	2	**7**
15	11	10	3
9	**5**	13	1
16	6	14	12

4	8	2	**5**
15	11	10	3
9	**7**	13	1
16	6	14	12

Fig. 4. An example of random swap neighborhood generation

2) Adjacent Swap Neighborhood: A facility is randomly selected from index 1 to m-1 (m = the number of array length) and its element is swapped with the next facility. An example of adjacent swap neighborhood is shown in Fig. 5.

Current Solution	4	8	2	**7**	**3**	10	11	15	9	5	13	1	12	14	6	16
Neighbor Solution	4	8	2	**3**	**7**	10	11	15	9	5	13	1	12	14	6	16

4	8	2	**7**
15	11	10	**3**
9	5	13	1
16	6	14	12

4	8	2	**3**
15	11	10	**7**
9	7	13	1
16	6	14	12

Fig. 5. An example of adjacent swap neighborhood generation

The fourth step: selecting a new solution

If the new solution is better than the current solution, it will replace the current solution. If the new solution is worse than the current solution, the new solution is selected with a probability of $e^{-\Delta C/T}$ where ΔC is the difference between values of the objective function for the two neighboring solutions, and T is the current temperature. If the replaced solution is better than the best local solution, the best local solution will also be updated and $localStagnation$ is set to zero. If the replaced solution is better than the best global solution, the best global solution will also be updated and $globalStagnation$ will be also set to zero. The new solution is placed in the index i of the local buffer and the i index will be increased by one unit.

The fifth step: local search

In this step, a local search consisted of the third to fourth steps are repeated until the number of repetitions reach $localIteration$.

The sixth step: Updating the global memory buffer

The global memory buffer is updated in such a way that, at first, the local memory buffer is sorted in terms of the total cost and replaces the worst solutions of the local memory up to the amount of $globalMBRatio$ from the best global memory buffer solutions. Then the solutions with duplicate cost values are deleted along with sorting the local memory in terms of the cost function. The mechanism for deleting duplicate solutions is in form of taking unique from the mentioned list, hence the first solution is preserved from the beginning of the list and the other duplicate solutions are deleted. This allows the algorithm to search for new spaces at lower temperatures. The resulted buffer replaces the global memory buffer. The global memory buffer increases the intensification strategy of the SA algorithm so that some of the high-quality solutions are maintained and the neighborhood of these solutions is further explored at a lower temperature. The scheme of the updating process for the global memory buffer is shown in Fig. 6.

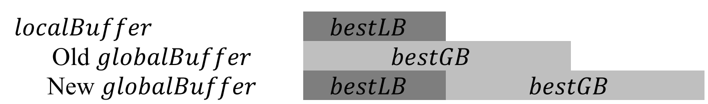

Fig. 6. Update scheme for global memory buffer

In this scheme, $bestLB$ and $bestGB$ represent the best solutions in the local buffer and the best solutions in the global buffer, respectively.

```
Initialize Parameters: T0, Tf, alpha,Kb, localIteration, maxLocalStagnation, maxGlobalStagnation, globalMBRatio, restartMBRatio, bufferSize
Initialize Variables: localStagnation = 0, globalStagnation = 0, temperature = T0
Initialize Solutions: sol, bestSol = sol, globalBestSol = sol,
Initialize Buffers:     localBuffer, globalBuffer
while temperature >= Tf
  for i = 1 to localIteration
        Generate a neighbor solution for Sol to obtain a new solution, newSol; calculate delta = newSol.cost - sol.cost;
        newSol.temperature = temperature;sol.temperature = temperature;
        newSol.bestSol = bestSol;
      if delta <= 0
        sol= newSol;
      Else
    Generate probability r
        if r < exp(-delta/(Kb*temperature));
          sol = newSol;
        end
      end
      if cost(sol) < cost(bestSol)
        bestSol = sol;
        localStagnation = 0;
      end
      if cost(sol) < cost(globalBestSol)
        globalBestSol = sol;
        globalStagnation = 0;
      end
      localBuffer(i) = sol;
  end
  UpdateGlobalBuffer(localBuffer, globalBuffer, globalMBRatio);
  if localStagnation > maxLocalStagnation
        and globalStagnation < maxGlobalStagnation
      sol = BufferSelection(globalBuffer, restartMBRatio);
      localStagnation = 0;
      temperature = Temperature(sol);
      bestSol = BestSol(sol);
  end
  temperature = temperature * alpha;
  localStagnation = localStagnation + 1;
  globalStagnation = globalStagnation + 1;
end
return globalBestSol;
```

Fig. 7. The pseudo-code of the algorithm

The seventh step: Restart Mechanism

If the value of $localStagnation$ is greater than $maxLocalStagnation$ and the value of $globalStagnation$ is less than $maxGlobalStagnation$, a solution among the best solutions of the global memory buffer is randomly selected to the amount of $restartMBRatio$ and it is considered to be current solution, and the temperature and the best local solution of the algorithm are updated using the temperature and the best local solution stored in the extracted solution and $localStagnation$ is set to be zero. This phase increases the diversification of the algorithm. As $restartMBRatio$ increases and converges to 1, the searching ability of the algorithm becomes shallower and the search space becomes more extended. In contrast, as this value reaches zero, the focus of the algorithm becomes performing in-depth movements increasing the intensification.

As can be seen in lines 28 to 33 of the DMSA algorithm, if the maxLocalStagnation variable is a very large number or the value of variable maxGlobalStagnation is zero, the algorithm never uses the restart mechanism and reaches the final temperature and terminates. In this case, the algorithm turns into the standard SA. If the value of variable maxLocalStagnation is a small number or the value of variable maxGlobalStagnation is a very large number, the algorithm frequently uses the restart mechanism and may not be able to search for the solution space properly or even terminate. Therefore, it can be said that the standard SA algorithm is a special case of the DMSA algorithm in which the restart mechanism and memory buffer have not been used.

The eighth step: Temperature Reduction

In this step, temperature is decreased and one unit is added to both $localStagnation$ and $globalStagnation$, and index i set reset to 1. The third to seventh steps continue until the final temperature is reached. A Pseudo-code of the algorithm is depicted in Fig. 7.

4. Numerical examples

Two test cases from MLFLP-related papers and 11 test cases from the QAP benchmark library have been selected to examine the algorithm in large-scale problems.

The algorithm was programmed in MATLAB® 2013 and executed on a Core™ i7 computer with 8GB of ram running Windows™ 7 operating system. The parameters of the algorithm for these experiments are presented in Table 4. The parameters mentioned in this table are obtained experimentally. Given that the initial temperature for all test cases is 500°C, the Boltzmann constant is chosen in such a way that the average probability of choosing bad solutions at the initial temperature of T0 is about 60 to 70% and also this probability is close to zero at the final temperature.

Table 4
Parameter setting for the numerical experiments

Parameter	problem1	problem2	tho30	tho40	wil50	wil100	sko100a-f	tho150
$maxGlobalStagnation$				600				
$bufferSize$				$localIteration$				
T_0				500				
T_f				0.1				
$alpha$				0.99				
$restartMBRatio$				0.1				
$localIteration$			1000					2000
$globalMBRatio$	0.9					0.95		
$maxLocalStagnation$	12					15		
K_b	4	20	10	10	1	1	1	30

Problem 1

This problem was introduced by Kazerooni et al. (1996). The problem was determining the layout of 24 types of machines on a 5×6 grid of the locations of the machines. The part list and production data related to 38 products are given in Table 5.

Using the data in Table 5, the material flow between departments is shown in Table 6. For example, the data flow between departments 1 to 13 in Table 5 (760), are obtained from the total material flow of these two departments in Table 6 in the production path of products No. 1 (130), No. 9 (85), No. 16 (95) , No. 17 (160), No. 31 (130) and No. 38 (160).

The material handling cost between the machines was considered to be 1 for all cases. By considering 24 machines and 6 candidate locations, the total number of possible combinations to be searched for finding the optimal solution is $\frac{30!}{6!} = 3.68 \times 10^{29}$ which makes the full enumeration impossible. In this study, this problem was modeled in the GAMS software with a time constraint of 86400 seconds. The total cost associated with the solution obtained by the CPLEX solver was

14251. Mak et al. (1998) solved this problem using GA and compared their results those of Chan and Tansri (1994) who utilized the three crossover operators of partially mapped crossover (PMX), order crossover (OX), and cyclic crossover (CX). El-Baz (2004) and Sadrzadeh (2012) also improved the problem's solutions using GA. Ou-Yang et al. (2013) solved the aforementioned problem via the Heuristic Artificial Particle Swarm Optimization (HAPSO) algorithm. Table 7 demonstrates a comparison among the results of applying these methods on the problem. In the Table, the number of times that the best solution was obtained in 30 runs is shown. The GA of Chan and Tansri (1994) with the PMX, OX and CX operators generated inferior solutions compared to the one proposed by Mak et al. (1998) which was able to obtain a solution with a total cost of $12982 which could be reached by the latter algorithm in 11 runs out of 30. El-Baz (2004) acquired a better solution with the total cost of $11862 in 23 runs out of 30. Sadrzadeh (2012) obtained a solution with a total cost of $11669 in 28 runs out of 30. Ou-Yang et al. (2013) employed a hybrid PSO algorithm called HAPSO and acquired a solution with a total cost of $11632. Although their best solution is better than the one obtained from the algorithm of Sadrzadeh (2012), their average and worst solutions were worse than those of that algorithm. The best solution belongs to the proposed SA algorithm with the total cost of $11402 which generated a better solution compared to other algorithms. Moreover, the average and worst solutions obtained by the proposed algorithm is better than those obtained by the other algorithms which imply its higher effectiveness. Fig. 8 and Fig. 9 depict the best solution obtained by the proposed algorithm and those obtained by GAMS and other algorithms, respectively. It can be seen that the best solution has been improved by 1.98%. As can be seen, the DMSA algorithm is not only better at finding the best solution to the problem than the standard SA algorithm but also the quality of the final solutions has been improved according to the average of the solutions and the worst solution.

Table 5
Part list and production data of the products for Problem 1

Product Number	Production Volume	Production Route	Product Number	Production Volume	Production Route
1	130	22–1–13–21	20	130	10–17–12
2	150	3–20–24	21	105	4–16
3	125	14–7–23–24	22	130	2–5–11–19
4	145	15–6–18–8–12	23	150	20–12
5	65	15–6–18–8–12–5	24	185	7–14–23
6	78	9–17–10	25	145	15–6–18–8–10
7	95	9–17–10	26	65	15–6–18–8–12
8	160	4–16	27	95	9–17
9	85	22–1–13–21	28	160	6–18–8–12
10	105	2–11–19–5–21	29	85	3–20–17
11	130	3–20	30	105	14–7–23–24–16
12	140	3–20	31	130	22–1–13–21–2
13	150	2–11–19	32	150	3–20
14	185	2–11–19–5	33	125	11–19–5
15	78	3–20	34	145	20–12–21
16	95	22–1–13–21	35	65	16–11–14
17	160	1–13–22	36	78	4–16
18	85	15–6–18–8–12	37	95	4–16
19	105	4–16	38	160	1–13–19

Table 6
Flow data for problem 1

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1													760									440		
2					130						440										130			
3																				733				
4																543								
5											130	65							415		105			
6															505			665						
7														415									230	
8										145		520						665						
9																	268							
10																	303							
11														65		65			695					
12																	130			295	145			
13																			160		440	160		
14																							185	
15																								
16																								105
17																				85				
18																								
19																								
20																								150
21																								
22																								
23																								230
24																								

Table 7
A comparison of the performances of different methods on problem 1 over 30 runs

Method	Successful Hits	Best Cost	Average Cost	Worst Cost	Average Time (sec)
Proposed algorithm (DMSA)	7	11402	11584.7	11892	1182
Standard SA (This paper)	1	11552	11932.7	12362	430
CPLEX (This paper)	-	14251	-	-	86400
Ou-Yang et al. (2013)	12	11632	11764	11981	Not reported
Sadrzadeh (2012)	28	11662	11676	11951	Not reported
El-Baz (2004)	23	11862	Not reported	Not reported	Not reported
Mak et al. (1998)	11	12982	15087.7	18657	Not reported
Chan and Tansri (1994) with PMX operator	0	14947	18355.9	20654	Not reported
Chan and Tansri (1994) with OM operator	0	22406	24301.7	26926	Not reported
Chan and Tansri (1994) with CX operator	0	14717	17216.5	20654	Not reported

22	1	13	19	11	14
6	18	21	5	2	7
15	8	12	20	24	23
	10	17	3	16	
		9		4	

Fig. 8. The best solution obtained by the proposed algorithm for problem 1

	3	24	16	4	22
9	20	23	5	13	1
17	12	7	19	21	
10	8	14	11	2	
	18	6	15		

(a)

	3	24	16	4	22
9	20	23	5	13	1
17	12	7	19	21	26
10	8	14	11	2	25
	18	6	15		

(b)

	4	23	7	14	
	16	24	1	22	
	3	20	13	19	11
9	17	12	21	5	2
	10	8	18	6	15

(c)

	9	17	10		
3	20	12	8	18	6
23	24	16	4		15
7	14	2	11	19	5
		21	13	1	22

(d)

		9	4	23	7
	10	24	16		14
	3	17	2	11	22
6	15	20	5	19	
18	8	12	21	13	1

(e)

Fig. 9. The best solutions obtained by (a) Mak et al. (1998) (b) El-Baz (2004) (c) Sadrzadeh (2012) (d) Ou-Yang et al. (2013)
(e) CPLEX (this paper) for problem 1

Problem 2

The second example is taken from Francis et al. (1974) which is deciding the layout of 12 departments where the required areas of the departments are presented in Table 8. In addition, the flow data and the cost data are presented in Table 9 and Table 10, respectively. Facilities are located on the factory floor with an area of 16 × 16 with square blocks. The proposed space-filling curve layout is shown in Fig. 10, where the representation is a horizontal sweeping. In order to make the algorithm more flexible in finding the optimal solution, the remaining empty blocks were considered as four departments with zero material flow with other departments and areas of 10, 10, 10, and 6 blocks.

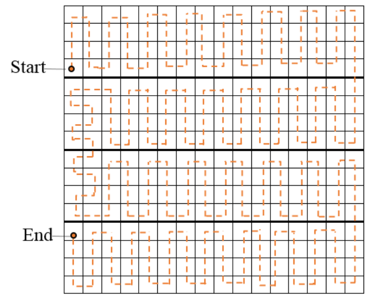

Fig. 10. space filling curve for layout of problem 2

González-Cruz and Martínez (2011) solved this problem using an entropy-based algorithm. The best solution obtained by them was \$83883 and they compared their solution with the best solution of the CRAFT algorithm of Francis et al. (1974) which had a cost of \$87693. The best solution of the CRAFT algorithm is shown in Fig. **11** and the best solution of the entropy-based algorithm of González-Cruz and Martínez (2011) is displayed in Fig. 12

Sadrzadeh (2012) solved the problem with the GA and obtained a layout with a cost of 77844\$ shown in Fig. 13. Tavakkoli-Moghaddam et al. (2015) with a four-dimensional firefly algorithm acquired the solution displayed in Fig. 14 with a total cost of 71679\$. The proposed memory-based SA algorithm with a total cost of \$69022 offered a better solution compared to other algorithms in the reviewed literature and is shown in Fig. **15**. A comparative summary of the methods of this problem is given in Table 11. For the first time for this problem, the average of the best solution, worst solution and also the algorithm run time are mentioned. As you can observe, the average of the best and worst solutions in 30 executions of the proposed algorithm is better than the best existing solution in the literature which indicates the higher effectiveness of this algorithm. It can be seen that the best solution for this example has been improved by 3.71%. Here, although the best solution from the DMSA algorithm is the same as that of the standard SA algorithm, as can be seen, the quality of the solution obtained from the memory-based SA algorithm is better, so that the average solution and the worst solution have been improved. Additionally, the proposed algorithm has been able to achieve the best solution several times more.

Table 8
Departments and required areas in Problem 2

Number	Department	Area (Ft^2)	Number	Department	Area (Ft^2)
1	Receiving	600	7	Sanding	125
2	Machining	425	8	Facing	275
3	Grinding	200	9	Painting	285
4	Welding	250	10	Cleaning	150
5	X-ray	210	11	Labeling	75
6	Inspection	175	12	Storage and shipping	715

Table 9
Flow data for Problem 2

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1		1000	300									
2			1000	200								
3		200		400		600						
4					600							
5						200	400					
6							600	200		400		
7						400				200		600
8							200					
9											200	400
10									600			
11												200
12												

Table 10
Cost data for Problem 2

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**1**	**11**	**12**
1		3	1.5									
2			1.23	0.98								
3		0.98		1.23		1.1						
4					2.1							
5						1.8	1.5					
6							1	1		1.2		
7						1				1		2.4
8							1					
9											1.3	2.1
1									1.7			
11												2
12												

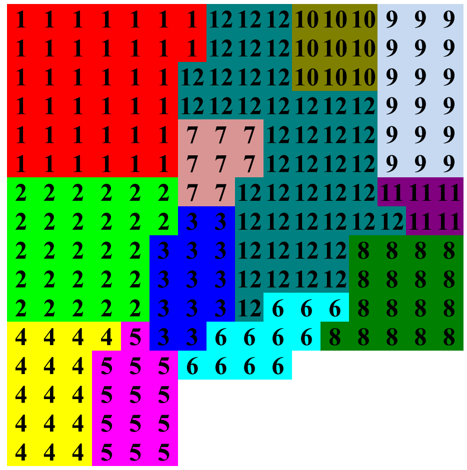

Fig. 11. The best solution obtained by CRAFT algorithm of Francis et al. (1974) for Problem 2

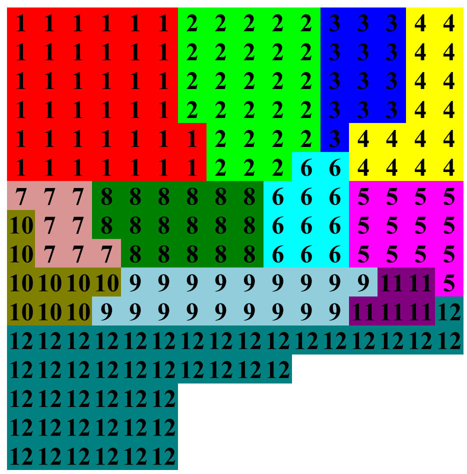

Fig. 12. The best solution obtained by entropy-based algorithm of González-Cruz and Martínez (2011) for Problem 2

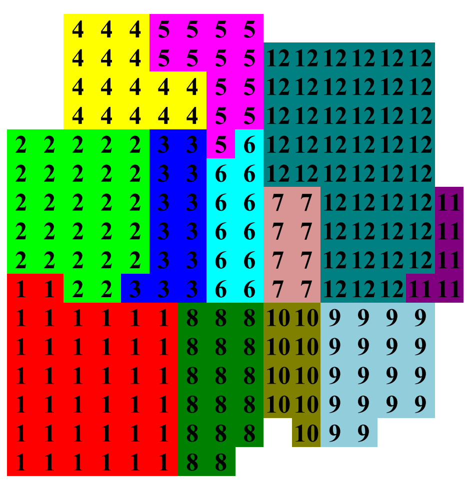

Fig. 13. The best solution obtain by genetic algorithm of Sadrzadeh (2012) for Problem 2

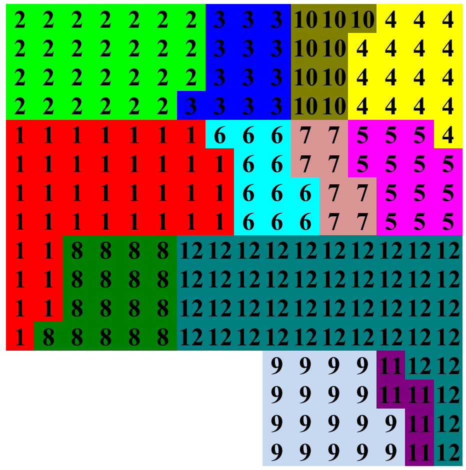

Fig. 14. The best solution obtain by QFA algorithm of Tavakkoli-Moghaddam et al. (2015) for Problem 2

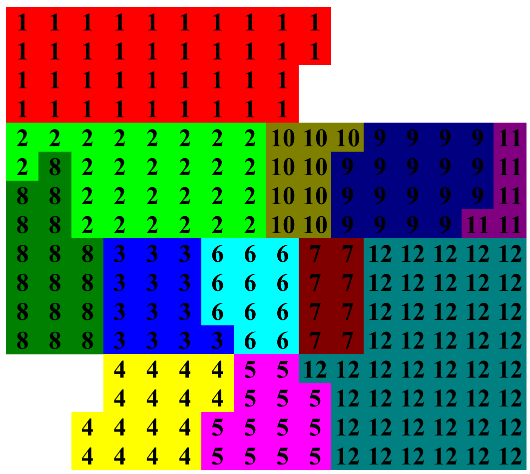

Fig. 15. The best solution obtained by proposed approach for Problem 2

Table 11
A comparison of the performances of different methods on Problem 2 over 30 runs

Method	Successful Hits	Best Cost	Average Cost	Worst Cost	Average Time (sec)
Proposed algorithm (DMSA)	26	$69022	69132	70207	1270
Standard SA (This paper)	10	$69022	69896	72633	469
Tavakkoli-Moghaddam et al. (2015)	Not Reported	$71679	Not Reported	Not Reported	Not Reported
Sadrzadeh (2012)	Not Reported	$77844	Not Reported	Not Reported	Not Reported
González-Cruz and Martínez (2011)	Not Reported	$83883	Not Reported	Not Reported	Not Reported
Francis et al. (1974)	Not Reported	$87963	Not Reported	Not Reported	Not Reported

QAP Problems

In this section, the performance of the DMSA algorithm in solving large-scale problems is examined. To this end, we have compared the best solution obtained from the implementation of the proposed SA algorithm with the standard SA algorithm and the related research in solving tho, sko, wil test cases from the QAP benchmark library.

Table 12 summarizes the results of the DMSA algorithm and standard SA algorithm for the considered test cases. Each test case was independently performed ten times, and the best, average, and worst solutions were listed in the table. The table also shows the best solution in bold font for each test case among the compared algorithms. The percentage of deviations of the best solution and the average solution is calculated by Eq. (9), in which BKS represents the best-known solution.

$$Dev = \frac{solution - BKS}{BKS} \times 100 \tag{9}$$

Table 12
The summarized results obtained by Standard SA and DMSA algorithm over 10 runs

Problem	BKS	Standard SA						DMSA					
		Best	Average	Worst	Time sec	Dev Best	Dev Avg	Best	Average	Worst	Time sec	Dev Best	Dev Avg
tho30	149936	**149936**	150572.6	151318	221	0	0.42	**149936**	150298.2	150542	718	0	0.24
tho40	240516	241028	241770.4	243600	265	0.21	0.52	**240542**	241173.8	242474	701	0.01	0.27
tho150	8133398	8151160	8159008.2	8166898	1598	0.22	0.31	**8148538**	8156319	8164216	4423	0.19	0.28
wil50	48816	48850	48902	48994	307	0.07	0.18	**48824**	48865.8	48906	1063	0.02	0.1
wil100	273038	273402	273644	273946	526	0.13	0.22	**273336**	273568.6	273798	1519	0.11	0.19
sko100a	152002	**152224**	152677.2	152868	586	0.15	0.44	152280	152426.2	152692	1448	0.18	0.28
sko100b	153890	154258	154579.2	154804	529	0.24	0.45	**154076**	154293	154582	2077	0.12	0.26
sko100c	147862	148100	148445.4	149554	526	0.16	0.39	**148090**	148360.6	148802	1706	0.15	0.34
sko100d	149576	149950	150338	150594	530	0.25	0.51	**149940**	150183.8	150512	1490	0.24	0.41
sko100e	149150	149390	149722.2	150216	525	0.16	0.38	**149292**	149814	150230	2141	0.1	0.45
sko100f	149036	149494	149773	150112	526	0.31	0.49	**149254**	149678	150058	1340	0.15	0.43

As can be seen in this table, the DMSA algorithm has performed better in all cases than the standard SA algorithm except one test case. In the tho30 test case, both algorithms have found the optimal solution of the problem. Abdel-Basset et al. (2018) compare the implementation results of the combined whale algorithm and tabu search with SA, Particle Swarm Optimization (PSO_{pure}), Particle Swarm Optimization with Variable Neighborhood Search and Genetic Algorithm (OB-GA) of Mamaghani and Meybodi (2012). They show that their combined algorithm performed better in sko100a-f test cases, but this algorithm was not successful compared to the hybrid PSO algorithm (PSO_{HC}) in sko100a, sko100d, and sko100f test cases. Table 13 summarizes the results of implementing the DMSA algorithm with the mentioned algorithms.

Table 13
A comparison of the performances of different methods on sko100a-f

Problem	BKS	(Mamaghani & Meybodi, 2012)					(Abdel-Basset et al., 2018)	This paper
		PSO_{pure}	PSO_{vns}	SA	OB-GA	PSO_{HC}	WAITS	DMSA
sko100a	152002	155462	153949	154210	153090	152922	153090	**152280**
sko100b	153890	156007	155971	154262	155030	154123	155030	**154076**
sko100c	147862	150978	149366	149542	149948	149742	149336	**148090**
sko100d	149576	152346	151746	151746	150828	150431	150828	**149940**
sko100e	149150	152349	150999	150426	150598	151426	150294	**149292**
sko100f	149036	152038	150871	150738	150402	150111	150402	**149254**

As can be observed in this table, the proposed SA algorithm has performed better in all test cases than the mentioned algorithms. Table 14 compares the best solutions for wil100 and tho150 test cases with the WAITS algorithm of Abdel-Basset et al. (2018), the genetic algorithm of Şahinkoç and Bilge (2018) and the backtracking search algorithm of Zhou et al. (2017).

Table 14
A comparison of the performances of different methods on tho150 and will100

Problem	BKS	(Zhou et al., 2017)	(Şahinkoç & Bilge, 2018)	(Abdel-Basset et al., 2018)	This paper
		BSAL	GA	WAITS	DMSA
tho150	8133398	8207482	8216038	Not reported	**8148538**
wil100	273038	273942	273786	273938	**273336**

According to this table, the DMSA algorithm provides a better solution than the other studies mentioned in the table.

Sensitivity Analysis

To analyze the sensitivity of the parameters, we selected problem 1. Also, the value of the three important parameters of the proposed algorithm was set according to Table 15, and the value of other parameters was considered according to Table 4. Based on the Taguchi L16 design, the algorithm was performed 30 times for each design. Table 16 summarizes the results.

Table 15
The value of parameters of the Taguchi design for problem 1

Parameters	values
restartMBRatio	0.01, 0.05, 0.1, 0.2
globalMBRatio	0.8, 0.9, 0.95, 0.99
maxLocalStagnation	6, 9, 12, 15

Table 16
Summarized results of the Taguchi design for problem 1

design	restartMBRatio	globalMBRatio	MaxLocalStagnation	Average Cost	Average time(sec)	Successful hits
1	0.01	0.8	6	11638	2111	3
2	0.01	0.9	9	11699	2268	2
3	0.01	0.95	12	11641	2074	3
4	0.01	0.99	15	11666	1651	6
5	0.05	0.8	9	11641	2635	4
6	0.05	0.9	6	11672	2704	5
7	0.05	0.95	15	11583	1165	6
8	0.05	0.99	12	11597	1317	7
9	0.1	0.8	12	11604	1455	5
10	0.1	0.9	15	11582	1033	2
11	0.1	0.95	6	11600	2504	11
12	0.1	0.99	9	11578	1662	8
13	0.2	0.8	15	11599	1062	4
14	0.2	0.9	12	11560	1300	10
15	0.2	0.95	9	11524	1716	8
16	0.2	0.99	6	11618	2722	2

According to the Table above, the algorithm successfully found the best solution in all cases, yielding an average of acceptable solutions. This demonstrated the ability of the proposed algorithm to find adequate solutions. Figures 16 and 17 illustrate the graphs of the results obtained through MINITAB.

According to Fig. 16, increasing the value of the parameter $restartMBRatio$ enhances the capability of the algorithm to escape from the local optimum and decreases the average solution. According to Fig. 17 and discussions in section 3 on the algorithm steps, changing the parameters can lead to changes in the exploration and exploitation capability of the proposed algorithm. For this problem, the proposed values for parameters restartMBRatio, globalMBRatio, and maxLocalStagnation are 0.1, 0.95, and 12, respectively.

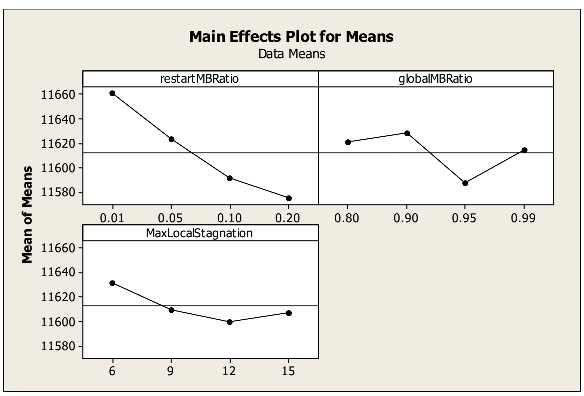

Fig. 16. The output of MINITAB for the average best solution through the DMSA algorithm

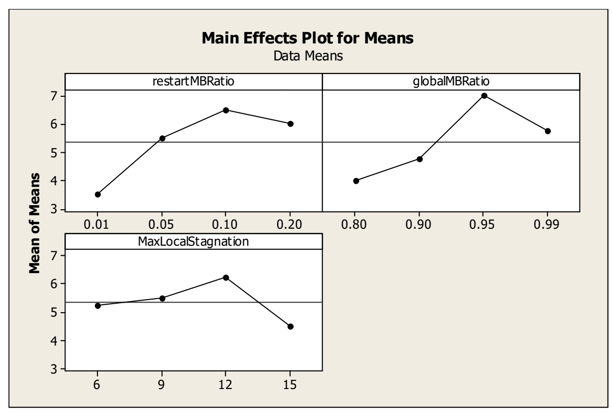

Fig. 17. The output of MINITAB for the successful hits of the DMSA algorithm

5. Conclusions

In this paper, a memory-based simulated algorithm, named DMSA, for multi-line facility layout problems was presented. The difference between this algorithm and the standard SA algorithm was the use of two memory buffers, two variables for storing the best algorithm solutions and a restart mechanism. The effectiveness of the algorithm was shown on two benchmark problems, and the results obtained were better than those obtained from the other algorithms employed for solving these problems. Since the DMSA algorithm has a general structure like the standard SA algorithm, it can be used in the approximate solution of other problems and is recommended as a problem-solving method for NP-Hard problems.

The efficiency of the proposed algorithm was demonstrated by 13 numerical examples. The first example has been presented by Kazerooni et al. (1996) and solved by Chan and Tansri (1994), Mak et al. (1998), El-Baz (2004), and Sadrzadeh (2012) using the genetic algorithm and by Ou-Yang et al. (2013) using the particle swarm optimization. The proposed algorithm could improve the best solution by 1.98%. The second example was taken from Francis et al. (1974). The best solution of the proposed algorithm was compared with that of the CRAFT algorithm (Francis et al., 1974), entropy-based algorithm (González-Cruz & Martínez, 2011), genetic algorithm (Sadrzadeh, 2012), and the quaternion firefly algorithm (Tavakkoli-Moghaddam et al., 2015). The proposed algorithm improved the best solution by 3.71%.

The DMSA algorithm was examined in 11 test cases of the QAP library benchmark, including tho, wil, and sko test cases, which performed better than the standard SA algorithm in 10 of 11 test cases. The results of this algorithm in solving the sko100a-f test case were compared with the combined whale and tabu search algorithm Abdel-Basset et al. (2018) and the combined PSO algorithm Mamaghani and Meybodi (2012) and it reached a better solution in all six tests. Moreover, a comparison was made with the GA of Şahinkoç and Bilge (2018), the backtracking search algorithm of Zhou et al. (2017), and the algorithm of Abdel-Basset et al. (2018) in solving the wil100 test case and tho150, where better solutions were obtained, and this shows the suitable efficiency of the algorithm in solving complex large-scale optimization problems.

Current research can be expanded in several ways. More specifically, the proposed algorithm can be utilized for solving other problems; and its performance can be compared with that of other methods. Another significant research may be creating a bed for a more thorough comparison for example based on criteria such as computational time.

References

Abdel-Basset, M., Manogaran, G., El-Shahat, D., & Mirjalili, S. (2018). Integrating the whale algorithm with tabu search for quadratic assignment problem: a new approach for locating hospital departments. *Applied Soft Computing, 73*, 530-546 .

Ahmadi, A & ,.Jokar, M. R. A. (2016). An efficient multiple-stage mathematical programming method for advanced single and multi-floor facility layout problems. *Applied Mathematical Modelling, 40*(9-10), 5605-5620 .

Aiello, G., La Scalia, G., & Enea, M. (2012). A multi objective genetic algorithm for the facility layout problem based upon slicing structure encoding. *Expert Systems with Applications, 39*(12), 10352-10358 .

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

Amar, S. H., Abouabdellah, A., & El Ouazzani, Y. (2018). Interactive facility layout problem: a bi-objective design. *International Journal on Interactive Design and Manufacturing (IJIDeM), 12*(1), 151-159 .

Anjos, M. F., Fischer, A., & Hungerländer, P. (2018). Improved exact approaches for row layout problems with departments of equal length. *European Journal of Operational Research, 270*(2), 514-529 .

Armour, G. C & ,.Buffa, E. S. (1963). A heuristic algorithm and simulation approach to relative location of facilities. *Management Science, 9*(2), 294-309 .

Asef-Vaziri, A., Jahandideh, H., & Modarres, M. (2017). Loop-based facility layout design under flexible bay structures. *International Journal of Production Economics, 193*, 713-725 .

Asl, A. D., & Wong, K. Y. (2017). Solving unequal-area static and dynamic facility layout problems using modified particle swarm optimization. *Journal of Intelligent Manufacturing, 28*(6 .۱۳۳٦-۱۳۱۷ ,(

Atta, S., & Mahapatra, P. R. S. (2019). Population-based improvement heuristic with local search for single-row facility layout problem. *Sādhanā, 44*(11), 1-19 .

Balakrishnan, J., Cheng, C. H., Conway, D. G., & Lau, C. M. (2003). A hybrid genetic algorithm for the dynamic plant layout problem. *International Journal of Production Economics, 86*(2), 107-120 .

Barzinpour, F., Mohammadpour Omran, M., Hoseini, S. F., Fahimi, K., & Samaei, F. (2019). A New Optimization via Invasive Weeds Algorithm for Dynamic Facility Layout Problem. *Journal of Quality Engineering and Production Optimization, 1*(1), 11-20 .

Bashiri, M., & Dehghan, E. (2010). Optimizing a multiple criteria dynamic layout problem using a simultaneous data envelopment analysis modeling *.Int J Comput Sci Eng, 2*(1), 28-35 .

Baykasoglu, A., Dereli, T., & Sabuncu, I. (2006). An ant colony algorithm for solving budget constrained and unconstrained dynamic facility layout problems. *Omega, 34*(4), 385-396 .

Besbes, M., Zolghadri, M., Affonso, R .C., Masmoudi, F., & Haddar, M. (2019). A methodology for solving facility layout problem considering barriers: genetic algorithm coupled with A* search. *Journal of Intelligent Manufacturing*, 1-26 .

Burkard, R. E., & Rendl, F. (1984). A thermodynamically motivated simulation procedure for combinatorial optimization problems. *European Journal of Operational Research, 17*(2), 169-174 .

Chan, K., & Tansri, H. (1994). A study of genetic crossover operations on the facilities layout problem. *Computers & Industrial Engineering, 26*(3), 537-550 .

Chen, C., & Tiong, L. K. (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 .

Chen, G., & Rogers, J. (2009). *Managing dynamic facility layout with multiple objectives.* Paper presented at the PICMET'09-2009 Portland International Conference on Management of Engineering & Technology.

Cravo, G. L., & Amaral, A. R. (2019). A GRASP algorithm for solving large-scale single row facility layout problems. *Computers & Operations Research, 106*, 49-61 .

de Sousa Junior, W. T., Montevechi, J. A. B., de Carvalho Miranda, R., de Oliveira, M. L. M., & Campos, A. T. (2020). Shop floor simulation optimization using machine learning to improve parallel metaheuristics. *Expert Systems with Applications, 150*, 113272 .

Domschke, W., & Krispin, G. (1997). Location and layout planning. *Operations-Research-Spektrum, 19*(3), 181-194 .

Drezner, Z. (1987). A heuristic procedure for the layout of a large number of facilities. *Management Science, 33*(7), 907-915 .

El-Baz, M. A. (2004). A genetic algorithm for facility layout problems of different manufacturing environments. *Computers & Industrial Engineering*, *47*(2-3), 233-246.

Feng, J., & Che, A. (2018). Novel integer linear programming models for the facility layout problem with fixed-size rectangular departments. *Computers & Operations Research, 95*, 163-171 .

Francis, R. L., White, J. A., & McGinnis, L. F. (197 .(٤*Facility layout and location: an analytical approach* (Vol. 31): Prentice-Hall Englewood Cliffs, NJ.

Gai, L., & Ji, J. (2019). An integrated method to solve the healthcare facility layout problem under area constraints. *Journal of Combinatorial Optimization, 37*(1), 95-113 .

Glover, F. (1989). Tabu search—part I. *ORSA Journal on computing, 1*(3), 190-206 .

González-Cruz, M. C., & Martínez, E. G.-S. (2011). An entropy-based algorithm to solve the facility layout design problem. *Robotics and Computer-Integrated Manufacturing, 27*(1), 88-100 .

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

Guan, C., Zhang, Z., Liu, S., & Gong ,J. (2019). Multi-objective particle swarm optimization for multi-workshop facility layout problem. *Journal of Manufacturing Systems, 53*, 32-48 .

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

Hammad, A., Akbarnezhad, A., & Rey, D. (2016). A multi-objective mixed integer nonlinear programming model for construction site layout planning to minimise noise pollution and transport costs. *Automation in construction, 61*, 73-85 .

Hasani, A., & Soltani, R. (2015). A hybrid meta-heuristic for the dynamic layout problem with transportation system design. *International Journal of Engineering, 28*(8), 1175 .١١٨٥-

Hasda, R. K. (2017). *Contribution to the optimization of Unequal Area Rectangular Facility Layout Problem.* Ecole centrale de Nantes .

Herrán, A., Colmenar, J. M., & Duarte, A. (2021). An efficient Variable Neighborhood Search for the Space-Free Multi-Row Facility Layout problem. *European Journal of Operational Research* .

Holland, J. (1975). An efficient genetic algorithm for the traveling salesman problem. *European Journal of Operational Research, 145*, 606-617 .

Hu, B., & Yang, B. (2019). A particle swarm optimization algorithm for multi-row facility layout problem in semiconductor fabrication. *Journal of Ambient Intelligence and Humanized Computing, 10*(8), 3201-3210 .

Kalita, Z., & Datta, D. (2018). A constrained single-row facility layout problem. *The International Journal of Advanced Manufacturing Technology, 98*(5-8), 2173-2184 .

Kang, S., Kim, M., & Chae, J. (2018). A closed loop based facility layout design using a cuckoo search algorithm. *Expert Systems with Applications, 93*, 322-335 .

Kazerooni, M., Luong, L., Abhary, K., Chan, F., & Pun, F. (1996). *An integrated method for cell layout problem using genetic algorithms.* Paper presented at the Proceedings of Twelfth International Conference on CAD/CAM Robotics and Factories of the Future, 14 ١٦-August 1996, Middlesex University London.

Kheirkhah, A., & Bidgoli, M. M. (2016). Dynamic facility layout problem under competitive environment: a new formulation and some meta-heuristic solution methods. *Production Engineering, 10*(6), 615-632 .

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *science, 220*(4598), 671-680 .

Kothari, R., & Ghosh, D. (2014). A scatter search algorithm for the single row facility layout problem. *Journal of Heuristics, 20*(2), 125-142.

Kulturel-Konak, S. (2017). A Matheuristic Approach for Solving the Dynamic Facility Layout Problem. *Procedia Computer Science, 108*, 1374-1383 .

Kulturel-Konak, S. (2019). The zone-based dynamic facility layout problem. *INFOR: Information Systems and Operational Research, 57*(1), 1-31 .

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

Kumar, K. R., Hadjinicola, G. C., & Lin, T.-l. (1995). A heuristic procedure for the single-row facility layout problem. *European Journal of Operational Research, 87*(1), 65-73 .

Lakehal, S., Aitzai, A., & Ghedjati, F. (2021). Parallel hybrid BBO-TS algorithm for QAP-formulation of FLP. *The International Journal of Advanced Manufacturing Technology, 117*(11), 3189-3209 .

Lee, Y. H., & Lee, M. H. (2002). A shape-based block layout approach to facility layout problems using hybrid genetic algorithm. *Computers & Industrial Engineering, 42*(2-4), 237-248 .

Li, T., & Mashford, J. (1990 .(A parallel genetic algorithm for quadratic assignment .

Liu, J., Zhang, H., He, K., & Jiang, S. (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 .

Liu, S., Zhang, Z., Guan, C., Liu, J., Gong, J., & Dewil, R. (2022). Mathematical formulation and two-phase optimisation methodology for the constrained double-row layout problem. *Neural Computing and Applications*, 1-20 .

Liu, S., Zhang, Z., Guan, C., Zhu, L., Zhang, M., & Guo, P. (2021). An improved fireworks algorithm for the constrained single-row facility layout problem. *International Journal of Production Research, 59*(8), 2309-2327 .

Mak, K., Wong, Y., & Chan, F. (199 .(^A genetic algorithm for facility layout problems. *Computer Integrated Manufacturing Systems, 11*(1-2), 113-127 .

Mamaghani, A. S., & Meybodi, M. R. (2012). *Solving the Quadratic Assignment Problem with the modified hybrid PSO algorithm.* Paper presented at the 2012 6th International Conference on Application of Information and Communication Technologies (AICT.(

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

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

McKendall Jr, A. R., Shang, J., & Kuppusamy, S. (2006). Simulated annealing heuristics for the dynamic facility layout problem. *Computers & Operations Research, 33*(8), 2431-2444 .

Meller, R. D., & Bozer, Y. A. (1996). A new simulated annealing algorithm for the facility layout problem. *International Journal of Production Research, 34*(6), 1675-1692 .

Neghabi, H., Eshghi, K., & Salmani, M. H. (2014). A new model for robust facility layout problem. *Information Sciences, 278*, 498-509 .

Neghabi, H., & Ghassemi Tari, F. (2015). An optimal approach for maximizing the number of adjacencies in multi floor layout problem. *International Journal of Production Research, 53*(11), 3462-3474 .

Ou-Yang, C., Santosa, B., & Mustakim, A. (2013, 2013//). *Artificial Particle Swarm Optimization with Heuristic Procedure to Solve Multi-Line Facility Layout Problem.* Paper presented at the Proceedings of the Institute of Industrial Engineers Asian Conference 2013, Singapore.

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 .

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

Pan, X., Xue, L., Lu, Y., & Sun, N. (2019). Hybrid particle swarm optimization with simulated annealing. *Multimedia Tools and Applications, 78*(21), 29921-29936.

Rabbouch, B., Saâdaoui, F., & Mraihi, R. (2019). Empirical-type simulated annealing for solving the capacitated vehicle routing problem .*Journal of Experimental & Theoretical Artificial Intelligence*, 1-16 .

Rezazadeh, H., Ghazanfari, M., Saidi-Mehrabad, M., & Sadjadi, S. J. (2009). An extended discrete particle swarm optimization algorithm for the dynamic facility layout problem. *Journal of Zhejiang University-Science A, 10*(4), 520-529 .

Rosenblatt, M. J. (1979). The facilities layout problem: a multi-goal approach. *International Journal of Production Research, 17*(4), 323-332 .

Rubio-Sánchez, M., Gallego, M., Gortázar, F., & Duarte, A. (2016). GRASP with path relinking for the single row facility layout problem. *Knowledge-Based Systems, 106*, 1-13 .

Sadrzadeh, A. (2012). A genetic algorithm with the heuristic procedure to solve the multi-line layout problem. *Computers & Industrial Engineering, 62*(4), 1055-1064 .

Safarzadeh, S., & Koosha, H. (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., & Türkbey, O. (2009). A simulated annealing algorithm to find approximate Pareto optimal solutions for the multi-objective facility layout problem. *The International Journal of Advanced Manufacturing Technology, 41*(9-10), 1003 .

Şahinkoç, M., & Bilge, Ü. (2018). Facility layout problem with QAP formulation under scenario-based uncertainty. *INFOR: Information Systems and Operational Research, 56*(4), 406-427 .

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

Scholz, D., Petrick, A., & Domschke, W. (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 .

Skorin-Kapov, J. (1990). Tabu search applied to the quadratic assignment problem. *ORSA Journal on computing, 2*(1), 33-45.

Solimanpur, M., & Jafari, A. (2008). Optimal solution for the two-dimensional facility layout problem using a branch-and-bound algorithm. *Computers & Industrial Engineering, 55*(3), 606-619 .

Tam, K. Y. (1992). A simulated annealing algorithm for allocating space to manufacturing cells. *The International Journal of Production Research, 30*(1), 63-87 .

Tavakkoli-Moghaddam, R., Torabi, N., & Ghaseminejad, A. (2015). A quaternion firefly algorithm to solve a multi-row facility layout problem. *International Journal of Engineering, Transactions B: Applications, 28*(11), 1605-1613 .

Turanoğlu, B., & Akkaya, G. (2018). A new hybrid heuristic algorithm based on bacterial foraging optimization for the dynamic facility layout problem. *Expert Systems with Applications, 98*, 93-104 .

Ulutas, B., & Islier, A. A. (2015). Dynamic facility layout problem in footwear industry. *Journal of Manufacturing Systems, 36*, 55-61 .
Uribe, N. R., Herrán, A., Colmenar, J. M., & Duarte, A. (2021). An improved GRASP method for the multiple row equal facility layout problem. *Expert Systems with Applications, 182*, 115184 .
Vázquez-Román, R., Díaz-Ovalle, C. O., Jung, S., & Castillo-Borja, F. (2019). A reformulated nonlinear model to solve the facility layout problem. *Chemical Engineering Communications*, *206*(4), 476-487.
Vincent, F. Y., Redi, A. P., Hidayat, Y. A., & Wibowo, O. J. (2017). A simulated annealing heuristic for the hybrid vehicle routing problem. *Applied Soft Computing, 53*, 119-132 .
Wang, M.-J., Hu, M. H., & Ku, M.-Y. (2005). A solution to the unequal area facilities layout problem by genetic algorithm. *Computers in Industry, 56*(2), 207-220 .
Xiao, X., Hu, Y., Wang, W., & Ren, W. (2019). A robust optimization approach for unequal-area dynamic facility layout with demand uncertainty. *Procedia CIRP, 81*, 594-599 .
Zhou, J., Love, P. E., Teo, K. L., & Luo, H. (2017). An exact penalty function method for optimising QAP formulation in facility layout problem. *International Journal of Production Research, 55*(10), 2913-2929 .
Zou, D., Wang, G.-G ,.Sangaiah, A. K., & Kong, X. (2017). A memory-based simulated annealing algorithm and a new auxiliary function for the fixed-outline floorplanning with soft blocks. *Journal of Ambient Intelligence and Humanized Computing*, 1-12 .