International Journal of Industrial Engineering & Production Research

September 2015, Volume 26, Number 4
pp.243 -253

http://IJIEPR.iust.ac.ir/

pISSN: 2008-4889

A Hybrid Simulated Annealing Algorithm for Single Row Facility Layout Problem

H. Hosseini Nasab *

Hassan Hosseini Nasab, Industrial Engineering Department, Yazd University, Yazd. Iran

KEYWORDS

Fuzzy knowledge based, Facilities planning and design, Flexible manufacturing systems, Genetic & ICA algorithms, Automated Guided Vehicle

ABSTRACT

This article addresses a single row facility layout problem where the objective is to optimize the arrangement of some rectangular facilities with different dimensions on a line. Regarding the NP-Hard nature of the considered problem, a hybrid meta-heuristic algorithm based on simulated annealing has been proposed to obtain a near optimal solution. A number of test problems are randomly generated and the results obtained by the proposed hybrid meta-heuristic are compared with exact solutions. The results imply that the proposed hybrid method provides more efficient solutions for the large-sized problem instances.

1. Introduction

Single row facility layout problem (SRFLP) has widely attracted the attention of many researchers since 1950s. It might be applied to determine the place of a number of facilities on the plant with the purpose of minimizing the total cost of handlings between the facilities. In this paper, SRFLP where facilities with equal or unequal dimensions are arranged on a line is addressed. SRFLP can be performed in various ways and has significant effects on the overall efficiency of production system.For instance, in flexible manufacturing systems (FMSs), an optimized linear layout of facilities can improve the efficiency of automated guided vehicles (AGVs). The application of SRFLP can easily be extended to non-manufacturing problems, such as assigning the service rooms along a corridor, storing files on the cylinders of a disk, arranging the books on a shelf of the library, and the like[1].

Koopmans and Beckman [2] were the pioneers to formulate the facility layout problem as a quadratic assignment problem (QAP). Afterwards, other researchers tried to formulate the problem as a linear programming model. In this direction, Lawer [3] presented a linear integer programming model for the facility layout problem, Love and Wong [4] proposed a linear mixed-integer programming model which was not a linearization attempt of QAP, and Kaufman and Broeckx [5], Bazaraa and Sherali [6] and Burkard and Bonniger [7] Sadrabadi and Sadjadi [8] also utilized mixed-integer programming models for the facility layout problem.

Kumar et al. [9] introduced a constructive heuristic which provided solutions for SRFLP in order to minimize the materials handling cost. Their heuristic assigned the facilities with the largest number of moves among

* **Corresponding author:** *H. Hosseini nasab*
Email: hhn@yazd.ac.ir

them to adjacent locations on the line. Braglia [10] presented a procedure to determine the optimal set of parameters related to heuristics based on simulated annealing (SA) to solve SRFLP. He considered search for the best set of parameters as a second optimization problem solved by genetic algorithm (GA) and tested the performance of his approach in a particular case of backtracking minimization in a SRFLP for FMSs. TavakoliMoghadam and Vasei [11] used a special design of a simulated annealing (SA) and branch-and-bound method for a single machine sequencing problem in order to find the sequence of jobs minimizing the sum of the maximum earliness and tardiness with idle times. They claimed that the method can be used for other types of sequencing problems, such as job shop and flow shop. Yaghini and Akhavan Kazemzadeh [12] proposed a SA solution method with an innovative solution representation and neighborhood structure for unsplittable multi-commodity capacitated network design problems.The results show that SA can find near optimal solution in much less time than exact algorithm. Kim et al. [13] developed an algorithm based on the combination of SA technique and graph theory to solve SRFLP. Ho and Moodie [14] proposed a two-phase layout procedure which combined flow line analysis with SA. Ficko et al. [15] extended a model for a single-row/multiple-row FMS and solved it by GA algorithms. Solimanpur et al. formulated SRFLP as a non-linear 0-1 programming model and developed an ant colony algorithm to solve it [16]. In addition, they proposed a technique to enhance the efficiency of the proposed algorithm. Anjos et al. [17] constructed a semi-definite programming relaxation providing a lower bound on the optimal value of SRFLP. They were the first to present the non-trivial global lower bound for SRFLP. As well, Anjos and Vannelli [18] demonstrated the combination of a semi-definite programming relaxation with cutting planes is able to compute the globally optimal layouts for large SRFLPs with up to 30 facilities. Amaral [19] presented a partial linear description whose integral points were the incident vectors of a layout and proposed a new lower bound for the problem by optimizing a linear program over the partial description.

Samarghandi et al. [20] used a particle swarm optimization (PSO) algorithm to solve SRFLP and employed a new coding and decoding technique to efficiently map the discrete feasible space of SRFLP to a continuous space. Samarghandi and Eshghi [21] proved a theorem to find the optimal solution of a special case of SRFLP. They also used the theorem to build a new algorithm based on tabu search (TS) for SRFLP. Sanjeevi and Kianfar [19] presented a polyhedral study of the triplet formulation of SRFLP introduced by Amaral [22]. Additionally, they demonstrated the linear program solved over these valid inequalities gives the optimal solution for all instances studied by Amaral. Datta and Amaral [23] applied a permutation-based GA to the NP-Hard problem of SRFLP. In their proposed GA, chromosomes were obtained using some rule-bases as well as random permutations of facilities. They improved the optimum solution by means of specially designed crossover and mutation operators.

Azadeh et al. [24] developed an integrated fuzzy simulation-fuzzy data envelopment analysis algorithm (FSFDEA) to solve a special case of SRFLP. They also employed discrete-event-simulation to model the different layout formations and used an adjusted measure range as a data envelopment analysis model to rank the simulated results and find the optimal layout design. Their proposed algorithm was capable to model and optimize the small-sized SRFLPs in stochastic, uncertain and non-linear environments.

The rest of this paper is organized as follows. Section 2 discusses on FMSs. Definition of the problem is presented in Section 3. The method of obtaining the optimal solution for the given problem is brought in Section 4. Section 5 is devoted to a meta-heuristic algorithm proposed to solve SRFLP. Experimental results are analyzed in Section

6. Finally, the concluding remarks are summed up in the last section.

2. Flexible manufacturing system

FMS stands for the system which involves high variety and high quantity of products. The purpose of the layout problem in an FMS is to distribute different resources to get maximum efficiency from the services. On the other hand, FMS is designed to optimize the production flow from the first stage to the last, like raw materials to the finished products. The layout has a significant effect on production time and cost, especially in the large FMS. It is estimated that 20% to 50% of the manufacturing costs are related to materials and work in process handlings. Hence, an optimized arrangement of facilities can reduce the manufacturing costs about 10% to 30%.

The optimum arrangement of devices and facilities is one of the main requirements in designing FMSs. A good design can improve the efficiency of operation and reduce the costs. Thus, FMS has different requirements for transportation rather than other manufacturing systems and a well-designed transportation system has an important role in the operation of an FMS.

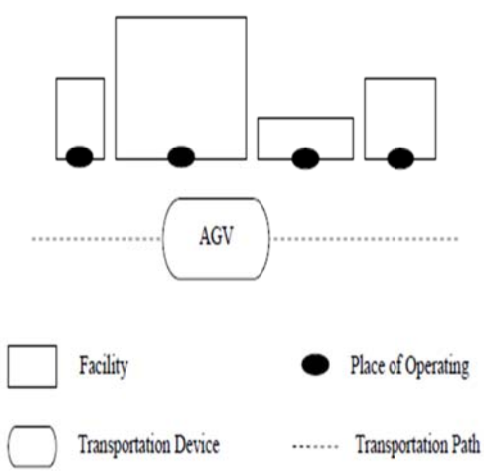

Fig. 1: The position of an AGV that serves to some facilities in a single row

Various transportation devices can be used in an FMS. One of the most important devices is AGV. The facilities fed by AGVs are usually placed in a single row or multiple rows. Figure 1 demonstrates the position of an AGV that serves to some facilities in a single row. AGVs are enjoyed the flexibility and increasingly used in modern factories, especially in FMSs. An inappropriate facilities layout uses a greater number of AGVs with lower efficiency.

Industrial robots are specialized to operate certain jobs in a work station. The arm of a robot can transport the work pieces to put in some facilities close too, as shown in Figure 2. So, an appropriate arrangement of facilities can reduce the number and cost of transportations.

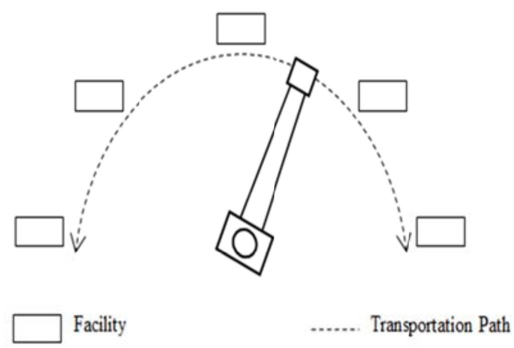

Fig. 2: The position of an industrial robot that serves to some facilities in a single row

Designing an FMS involves arranging unequally large devices. Therefore, only the methods for arranging differently large devices can be used. Unequal-area layout problems are more difficult to solve rather than equal-area ones because of having additional constraints in the problem formulation.

The problem of facilities arrangement is NP-Hard [25] and its solution time grows exponentially as the number of facilities increases.

The focus of this paper is on the single-row layout problem. Some of the characteristics of the given problem are listed as follows:

- All facilities are in the form of rectangular shapes.
- All facilities are operated in the center of the space.
- The available area for FMS is not limited along width.

The clearance between the facilities is zero.

3. Problem Definition and Notations

In this section, a mathematical programming model is presented for the considered production problem. The following notations are used in the presented model.

3.1. Parameters

M : Number of facilities and locations

f_{ij} : Number of trips between facilities *i* and *j* (f_{ii}=0)

c_{ij} : Transportation cost between facilities *i* and *j* for a distance unit (c_{ii}=0)

L_i : Length of facility *i*

3.2. Decision variable

d_{hl} : Distance between the center of locations *h* and *l* ($d_{hl} = d_{lh}$)

$$x_{ih} = \begin{cases} 1 & \text{if facility } i \text{ is located at location } h \\ 0 & \text{Otherwise} \end{cases}$$

3.3. Mathematical formulation for the considered production problem

The considered production problem is formulated as follows.

$$Min \quad \sum_{i=1}^{M}\sum_{j=1}^{M}\sum_{h=1}^{M-1}\sum_{l=h+1}^{M}(c_{ij}f_{ij}+c_{ji}f_{ji})d_{hl}x_{ih}x_{jl} \tag{1}$$

s.t.

$$d_{hl} = \sum_{k=1}^{M}\sum_{t=h+1}^{l-1} L_k x_{kt} + \frac{1}{2}\sum_{k=1}^{M} L_k x_{kh} + \frac{1}{2}\sum_{k=1}^{M} L_k x_{kl}; \quad 1 \le h < l \le M \tag{2}$$

$$\sum_{i=1}^{M} x_{ih} = 1; \quad h = 1,2,...,M \tag{3}$$

$$\sum_{h=1}^{M} x_{ih} = 1; \quad i = 1,2,...,M \tag{4}$$

$$x_{ih} \in \{0,1\}; \quad i,h = 1,2,...,M \tag{5}$$

$$d_{hl} \ge 0; \quad h,l = 1,2,...,M \tag{6}$$

According to Equation (1), the objective of the problem is to minimize the total transportation costs. Equation (2) determines the distance between locations *h* and *l*. Equation (3) ensures just one facility is assigned to each location and similarly, Equation (4) ensures each facility is just assigned to one location. Equation (5) defines the binary decision variable. Finally, Equation (6) denotes a non-negative variable.

4. Method of obtaining the optimal solution for the given problem

In order to find the best or optimum (if possible) solution for the problem described in Section 3, all of the feasible permutations and corresponding costs are created by coding through Turbo Pascal 7 software.

Consider the following example. *L*, *C* and *F* show the matrices that imply the unit transportation cost and number of trips from each facility to itself is equal to zero, and the unit transportation cost and number of trips from a facility to another one may be different from the reverse path.

$$L = \begin{bmatrix} 5 & 2 & 7 & 1 & 9 & 3 & 2 & 7 & 5 & 3 \end{bmatrix}$$

$$C = \begin{bmatrix} 0 & 5 & 1 & 7 & 3 & 9 & 4 & 2 & 9 & 2 \\ 1 & 0 & 8 & 3 & 4 & 2 & 4 & 1 & 7 & 3 \\ 6 & 1 & 0 & 8 & 3 & 5 & 8 & 9 & 2 & 3 \\ 4 & 2 & 7 & 0 & 9 & 6 & 7 & 1 & 3 & 4 \\ 8 & 6 & 9 & 3 & 0 & 1 & 2 & 6 & 7 & 4 \\ 7 & 9 & 4 & 7 & 1 & 0 & 3 & 4 & 3 & 7 \\ 1 & 4 & 6 & 2 & 8 & 5 & 0 & 8 & 3 & 4 \\ 5 & 3 & 7 & 8 & 5 & 2 & 4 & 0 & 9 & 2 \\ 5 & 7 & 9 & 4 & 5 & 2 & 7 & 4 & 0 & 3 \\ 8 & 6 & 7 & 9 & 3 & 4 & 1 & 6 & 4 & 0 \end{bmatrix}$$

$$F=\begin{bmatrix} 0 & 4 & 1 & 6 & 4 & 7 & 4 & 2 & 3 & 1 \\ 5 & 0 & 3 & 7 & 8 & 9 & 6 & 5 & 8 & 4 \\ 2 & 3 & 0 & 8 & 5 & 7 & 9 & 1 & 2 & 7 \\ 5 & 7 & 5 & 0 & 9 & 3 & 2 & 7 & 4 & 5 \\ 7 & 3 & 4 & 8 & 0 & 6 & 2 & 8 & 5 & 8 \\ 5 & 6 & 3 & 8 & 5 & 0 & 8 & 9 & 1 & 2 \\ 4 & 2 & 3 & 6 & 7 & 4 & 0 & 2 & 9 & 4 \\ 5 & 4 & 7 & 6 & 3 & 4 & 8 & 0 & 9 & 4 \\ 4 & 6 & 3 & 7 & 6 & 9 & 8 & 2 & 0 & 7 \\ 9 & 4 & 5 & 7 & 3 & 1 & 7 & 3 & 8 & 0 \end{bmatrix}$$

If number of facilities is equal to M, $M!$ permutations will be created to arrange the facilities in a single line. Since each permutation and its reverse permutation are the same solution, total number of solutions is equal to $M!/2$. The optimal cost and the corresponding computational time (using a personal computer, including: Dual-Core CPU 2×0.9GHz and 2GB RAM) for different values of M are brought in Table 1. For the values of M smaller than 10, only the first M values of the model's parameters (L, C and F) have been supposed.

Tab. 1. Optimal results for the model including the small number of facilities

M	Optimal Permutation	Optimal Cost	Computational Time (S)
5	1-5-4-2-3	3970.5	0.0025
6	3-2-6-4-1-5	6198.5	0.0085
7	3-7-2-6-4-1-5	8688.5	0.1040
8	5-1-2-4-6-7-3-8	13836.0	1.1755
9	5-1-6-4-2-9-7-3-8	21282.5	14.3762
10	5-1-10-4-6-2-9-7-3-8	27242.0	185.0150

Figure 3 shows the computational time in terms of the number of facilities. As seen, the computational time grows exponentially when the number of facilities increases. Since the considered problem is strongly NP-Hard, it is not able to obtain optimal solutions for the large-sized problems in a reasonable computational time. This issue necessitates the use of meta-heuristic algorithms to achieve near-optimal solutions in the real-world applications.

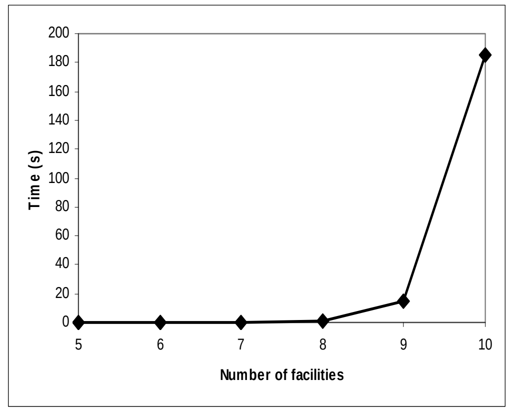

Fig.3. Computational time in terms of the number of facilities

5. The Proposed Meta-Heuristic Algorithm

The proposed algorithms to solve the layout problem are categorized into three general classes in the literature:

- Constructive
- Improving
- Hybrid

Constructive methods create a feasible layout with respect to constraints and limitations. These methods consider the possible minimum total cost while creating a layout. Improving methods generate a primary feasible layout and try to improve it in order to achieve the minimum total cost. Hybrid methods build a feasible layout at first and then improve it which has been addressed in this paper. In this paper, an algorithm which applies four heuristic methods to generate an initial solution (layout) is proposed. Then, the generated initial solution is improved using SA algorithm. As a matter of fact, a hybrid SA algorithm is used to solve the given problem. The steps of the proposed algorithm are described as the following.

5-1. Construction

In this section, four heuristic methods used to generate an initial solution are explained. The first heuristic method just generates a feasible solution, while the second, third and fourth heuristic methods attempt to make a feasible solution with the possible minimum total cost.

5-1-1. Random Initialization (RND)

In this method, a valid random permutation is assigned to the solution from a temporary list. Initially, the temporary list includes all facilities. One of the facilities is selected randomly from the temporary list and is assigned to the first place of the solution. Then, the facility is removed from the temporary list. This process is repeated for all M places of the solution.

5-1-2. Cost-Based Permutation (CBP)

The idea behind this method is that a good permutation will be obtained if the facilities with the lowest transportation cost are placed at the beginning or at the end of the permutation and the facilities with the highest transportation cost are placed in the middle of the permutation. Thus, the index value of each facility is calculated as Equation (7) and facilities are sorted in an ascending order of the index value. If the ordering is (1), (2), …, (*M*), the permutation will be (1)(3)…(*M*-2)(*M*)(*M*-1)…(4)(2) for odd values of *M* and (1)(3)…(*M*-1)(*M*) …(4)(2) for even values of *M*.

$$I = \sum_{j=1}^{M} c_{ij}, \qquad i \in \{1,...,M\} \tag{7}$$

5-1-3. Length-Based Permutation (LBP)

In the method proposed by Samarghandi and Eshghi [21], if the unit cost of transportation between all facilities is equal to a constant value and facilities are sorted in an ascending order in terms of the facilities' length, the optimal permutation will be (*M*)(*M*-2)…(3)(1)(2)…(*M*-3)(*M*-1) for odd values of *M* and (*M*)(*M*-2)…(2)(1)(3)…(*M*-3)(*M*-1) for even values of *M*. It should be mentioned, although the unit costs of transportation between the facilities are not equal, LBP method is used to obtain the initial solution.

5-1-4. Length & Cost-Based Permutation (LCBP)

In this method, a combination of CBP and LBP methods is used to select the permutation. The index of each facility is calculated as Equation (8). After sorting the facilities in an ascending order in terms of the index value, the permutation will be (1)(3)…(*M*-2)(*M*)(*M*-1)…(4)(2) for odd values of *M* and (1)(3)…(*M*-1)(*M*) …(4)(2) for even values of *M*.

$$I = \sum_{j=1}^{M} c_{ij} / L_i, \qquad i \in \{1,...,M\} \tag{8}$$

5-2. Improvement

In this section, the primary solutions obtained by the four aforementioned constructive methods are improved by SA algorithm.

The SA technique developed by Metropolis et al. [26] is a meta-heuristic optimization algorithm based on iterative improvement of the solution. Various applications of this technique to solve the classical combinatorial optimization problems in the field of manufacturing systems have been observed in the recent years. The basic idea of SA is very simple. In the search for a neighborhood, a better solution is always accepted. Additionally, a worse solution is also accepted with a certain probability *P* that is generally given by *EXP*(-*DELTA*/*T*), where *DELTA* is the increasing of the objective function value (cost) and *T* is a control temperature. The value of *T* begins from an initial temperature and is slowly reduced to a final value according to a cooling function. The probability of accepting the poor movements decreases with successive stages. A given SA may iterate for a number of times at each temperature. The steps of the proposed algorithm are as follows.

Step 1. Generate an initial solution (*S*).

Step 2. While G ≤ a specific time do

- Step 2.1. While N ≤ G^2 do
  - Step 2.1.1. Find a neighborhood for the current solution (*S'*).
  - Step 2.1.2. If the new solution has lower cost than the current solution

($F(S')<F(S)$), accept the new solution. Else, if $EXP(-(F(S')-F(S))/T)$ is greater than a uniform random value between 0 and 1, accept the new solution.

Step 2.1.3. Memorize the best solution which has been found so far.

End while

Step 2.2. Reduce the value of T using the cooling function $\alpha(T)$.

End while

The flowchart of the proposed hybrid SA

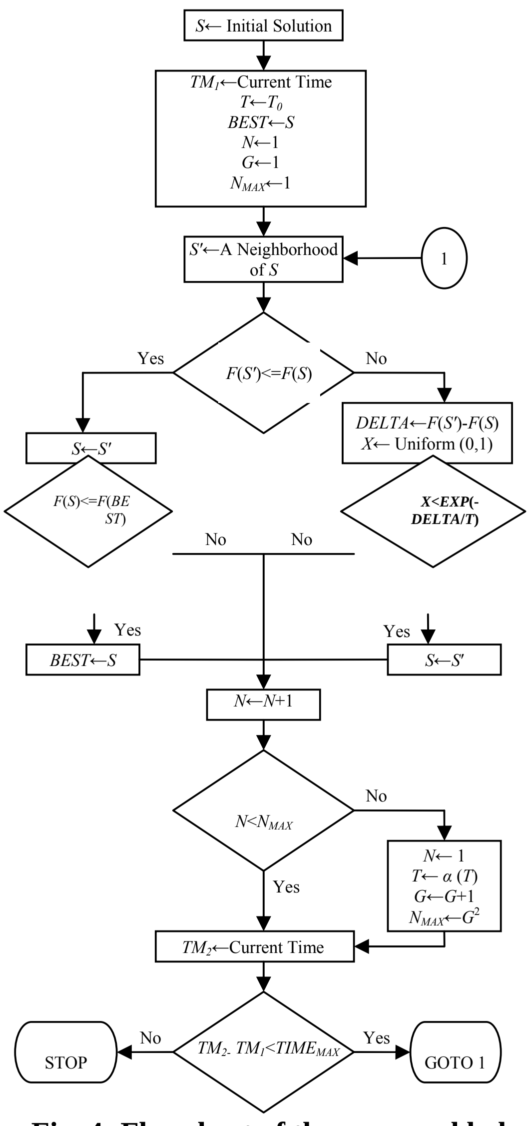

Fig. 4: Flowchart of the proposed hybrid SA algorithm

6. Experimental Results and Discussion

As expressed in Section 4, the optimal solution of the given problem with number of facilities (M) between 5 and 10 can be calculated within an acceptable time. But, when number of facilities enlarges, the computational time significantly increases and obtaining the optimal solution is impossible. To examine the performance of the proposed hybrid SA algorithm, the parameters of the model are set to the values of L, C and F matrices propounded in Section 4 as well as Equations (7) and (9). The presented model is solved for small values of M (between 5 and 10) by different initialization methods for 20 times. The average of obtained results is brought in Tables 2 to 5. The results show that the solutions obtained by the proposed method are the same as the optimal solutions obtained in Section 4.

$$T_0 = 100000, \qquad \alpha(T) = \frac{T}{1+0.001T}, \qquad TIME_{MAX} = 30S \tag{9}$$

Tab. 2: The results of the given model for small values of *M* by RND initialization

M	Initial Cost	Final Cost	Computational Time (S)	Number of Iterations
5	5023.7	3970.5	0.0000	4.60
6	8696.6	6198.5	0.0055	5.60
7	12798.0	8688.5	0.0140	6.75
8	19536.7	13836.0	0.0300	7.25
9	28718.1	21282.5	0.0630	9.15
10	38266.2	27242.0	0.1455	10.40

Tab. 3: The results of the given model for small values of *M* by CBP initialization

M	Initial Cost	Final Cost	Computational Time (S)	Number of Iterations
5	4644.5	3970.5	0.0000	3.95
6	7201.5	6198.5	0.0055	5.75
7	11691.5	8688.5	0.0165	7.30
8	20016.0	13836.0	0.0275	8.05
9	33684.5	21282.5	0.0660	8.95

10	38344.0	27242.0	0.1565	10.70

Tab. 4: The results of the given model for small values of *M* by LBP initialization

M	Initial Cost	Final Cost	Computational Time (S)	Number of Iterations
5	4100.5	3970.5	0.0000	3.60
6	6233.5	6198.5	0.0025	5.55
7	9073.5	8688.5	0.0135	6.80
8	15534.0	13836.0	0.0220	6.95
9	23257.5	21282.5	0.0690	8.95
10	30335.0	27242.0	0.1130	9.65

Tab. 5: The results of the given model for small values of *M* by LCBP initialization

M	Initial Cost	Final Cost	Computational Time (S)	Number of Iterations
5	4109.5	3970.5	0.0000	3.75
6	6307.5	6198.5	0.0030	4.55
7	8711.5	8688.5	0.0190	6.95
8	15226.0	13836.0	0.0245	7.45
9	22007.5	21282.5	0.0575	8.60
10	28311.0	27242.0	0.0905	9.20

By generating *L*, *C* and *F* parameters randomly and considering other parameters as Equation (9), the given problem is also solved by different initialization methods for large values of *M* (20, 30, 40, and 50) for 20 times and the average of obtained results are presented in Table 6. The results existing in Table 6 imply that LBP and LCBP methods provide more efficient solutions than RND and CBP methods. Figure 5 demonstrates the average of initial and final solutions with 50 facilities for 20 times execution. Also, Figure 6 shows the variation trend of the solutions using the four initialization methods and 50 facilities in terms of the number of iterations for a single run.

Tab. 6: The average of obtained results for large values of *M*

M	RND initialization method		CBP initialization method		LBP initialization method		LCBP initialization method	
	Initial Cost	Final Cost	Initial Cost	Final Cost	Initial Cost	Final Cost	Initial Cost	Final Cost
20	320503.1	217606.9	318680.5	217551.7	232700.5	217116.0	230704.5	216193.3
30	1487723.3	1119504.5	1397591.5	1121839.3	1161532.5	1124991.3	1189814.5	1119327.7
40	2841440.9	1929689.9	2796445.5	1934300.8	2058709.5	1929826.1	2059714.5	1930361.2
50	7028388.6	5300225.5	6784223.0	5302284.5	5511382.0	5287334.5	5453452.0	5291730.9

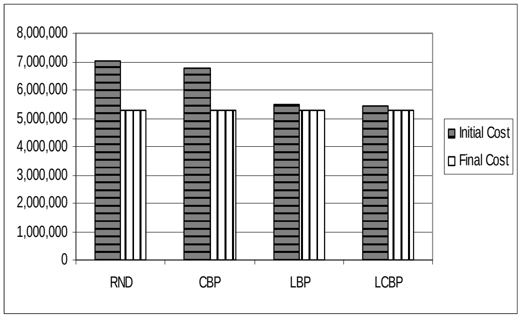

Fig. 5: The average of initial and final solutions with 50 facilities for 20 times execution.

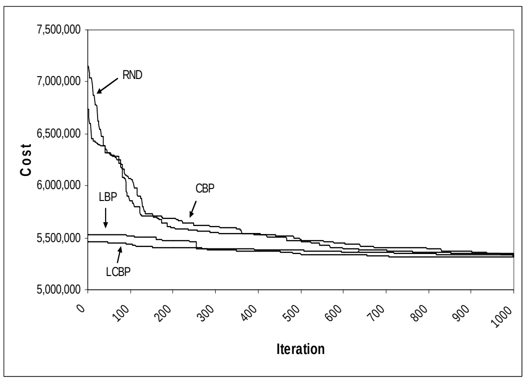

Fig. 6: Variation trend of the solutions using the four initialization methods and 50 facilities for a single run

7. Concluding Remarks

In this paper, a single row facility layout problem with the objective of optimizing the arrangement of some rectangular facilities with different dimensions on a line has been presented. As expressed in the text, the optimal solution of the given problem with number of facilities between 5 and 8 can be calculated within an acceptable time of 0.0025 to 1.1755. But, when the number of facilities enlarges from 8 to 10, the computational time significantly increases from 1.1755 to 185.0150. it emphasizes that the computational time grows exponentially, so for a large number of facilities, obtaining the optimal solution is almost impossible. To overcome this problem, a hybrid meta-heuristic algorithm based on the RND, CBP, LBP and LCBP methods and simulated annealing algorithm has been proposed. Computational results have shown that the proposed meta-heuristic algorithm obtains the optimal solution between 0.0000 to 0.1130 seconds, indicates that the reduction of 184.902 seconds of CPU time which is an acceptable performance to generate optimal solutions for the small-sized problems and near-optimal solutions for the large-sized problems. As well, the LBP and LCBP methods used in this paper are capable to supply more efficient solutions than the RND and CBP methods.

References

[1] Picard J., Queyranne M., (1981), On the one-dimensional space allocation problem, Operations Research, 29(2), 371-391.

[2] Koopmans T.C., Beckmann M., (1957), Assignment problems and location of economic activities, Econometrica, 25(1), 53-76.

[3] Lawler E.L., (1963), The quadratic assignment problem, Management Science, 9(4), 586-599.

[4] Love R.F., Wong J.Y., (1976), Solving quadratic assignment problems with rectangular distances and integer programming, Naval Research Logistics Quarterly, 23(4), 623-627.

[5] Kaufman L., Broeckx F., (1978), An algorithm for the quadratic assignment problem using Benders' decomposition, European Journal of Operational Research, 2(3), 207-211.

[6] Bazaraa M.S., Sherali H.D., (1980), Benders' partitioning scheme applied to a new formulation of the quadratic assignment problem, Naval Research Logistics Quarterly, 27(1), 29-41.

[7] Burkard R.E., Bonniger T., (1983), A heuristic for quadratic Boolean programs with applications to quadratic

assignment problems, European Journal of Operational Research, 13(4), 374-386.

[8] Sadrabadi M. R., and Sadjadi S. J., (2009), A New Approach to Solve Multiple Objective Programming Problems, International Journal off Industrial Engineering & Production Research, 20(1), 41-51.

[9] 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.

[10] Braglia M., (1996), Optimisation of a simulated-annealing-based heuristic for single row machine layout problem by genetic algorithm, International Transactions in Operational Research, 3(1), 37-49.

[11] TavakoliMoghadam R. and Vasei M. A Single Machine Sequencing Problem with Idle.

[12] Yaghini M. and. Akhavan Kazemzadeh M.R (2102), A Simulated Annealing Algorithm for Unsplittable Capacitated Network Design, International Journal off Industrial Engineering & Production Research, 23(2), 91-100.

[13] Kim C.B., Kim S.S., Foote B.L., (1996),. A assignment problems in single-row and double-row machine layouts during slow and peak periods, Computers & Industrial Engineering, 30(3), 411-422.

[14] Ho Y.C., Moodie C.L., (1998), Machine layout with a linear single-row flow path in an automated manufacturing system, Journal of Manufacturing Systems, 17(1), 1-22.

[15] Ficko M., Brezocnik M., Balic J., (2004), Designing the layout of single- and multiple-rows flexible manufacturing system by genetic algorithms, Journal of Materials Processing Technology, 157–158, 150–158.

[16] Solimanpur M., Vrat P., Shankar R., (2005), An ant algorithm for the single row layout problem in flexible manufacturing systems, Computers & Operations Research, 32(3), 583–598.

[17] Anjos M.F., Kennings A., Vannelli A., (2005), A semi definite optimization approach for the single-row layout problem with unequal dimensions, Discrete Optimization, 2(2), 113–122.

[18] Anjos M.F., Vannelli A., (2008), Computing globally optimal solutions for single-row layout problems using semi definite programming and cutting planes, INFORMS Journal on Computing, 20(4), 611–617.

[19] Amaral A.R.S., (2009), A new lower bound for the single row facility layout problem, Discrete Applied Mathematics, 157(1), 183-190.

[20] Samarghandi H., Taabayan P., Jahantigh F.F., (2010), A particle swarm optimization for the single row facility layout problem, Computers & Industrial Engineering, 58(4), 529–534.

[21] 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.

[22] Sanjeevi S., Kianfar K., (2010), A polyhedral study of triplet formulation for single row facility layout problem, Discrete Applied Mathematics, 158(16), 1861-1867.

[23] Datta D., Amaral A.R.S., Figueira J.R., (2011), Single row facility layout problem using a permutation-based genetic algorithm, European Journal of Operational Research, 213(2), 388–394.

[24] Azadeh A., Moghaddam M., Asadzadeh S.M., Negahban (2011), A., An integrated fuzzy simulation-fuzzy data envelopment analysis algorithm for job-shop layout optimization: The case of injection process with ambiguous data, European Journal of Operational Research, 214(3), 768–779.

[25] Suresh G., Sahu S., (1993), Multiobjective facility layout using simulated annealing, International Journal of Production Economics, 32(2), 239–254.

[26] Metropolis N., Rosenbluth A., Rosenbluth M., Teller A., Teller E., (1953), Equation of state calculations by fast computing machines, Journal of Chemical Physics, 21(6), 1087-1092.