APEM journal **Advances in Production Engineering & Management** 3 (2008) 4, 193-204
ISSN 1854-6250 Scientific paper

SCATTER SEARCH ALGORITHM FOR SINGLE ROW LAYOUT PROBLEM IN FMS

Satheesh Kumar R.M.; Asokan P.; .Kumanan S.; Varma B
Department of Production Engineering
National Institute of Technology, Tiruchirappalli – 620015,INDIA
E-mail: rm_satkumar@yahoo.co.in

Abstract:
Flexible manufacturing Systems are automated manufacturing systems which have job shop flexibility and flow shop efficiency. One of the major concerns in Flexible Manufacturing System (FMS) is the minimization of the total flow of material through the system. Facility layout design deals with the arrangement of machines, in order to minimise the total flow of parts in the system. The commonly used layout configuration in FMS is the single row layout, because it provides better control of operations and has ability to support different type of material handling system such as conveyors, Automated Guided Vehicles (AGV). Due to the computational complexity involved in facility layout problem, researchers have developed several heuristics to solve the problem. In this paper, Scatter Search (SS) technique is proposed to solve single row unequal area facility layout problem. The validity of the proposed approach is tested on benchmark problems available in the literature and the results are presented.

Key Words: Facility layout, Flexible Manufacturing System, Single row layout, Scatter Search

1. INTRODUCTION

The layout arrangement is an inevitable problem in all industrial plants and the decisions regarding the layout of machines receive intensive attention in production and operations management. The facility layout problem is belongs to NP-Complete type [1] and the complexity of the problem exponentially increases with the number of facilities. Traditionally, the layout problem has been modelled as a Quadratic Assignment Problem (QAP) and it require that the locations are known in advance and therefore the distances between the locations are predefined numerical values.

In recent years, new manufacturing technologies such as FMS, Group Technology (GT) have been implemented in industries around the world to remain competitive in the global market. The arrangement of machines is an important step in designing FMS as it critically affects operating efficiency, system capacity and system flexibility. The different types of layouts used in FMS are (i) Single row layout (ii) Multi-row layout (iii) Loop layout (iv) Cluster layout and (v) Circular layout. The single row layout is widely implemented configuration in FMS, in which machines are arranged along a straight line and a material handling device moves the parts from one machine to another. In this paper, single row unequal area layout problem is considered to reflect real manufacturing facilities and also clearance between the machines is assumed different. Unequal machine area problem is hard to solve when compared with equal area problem because of additional constraints into the problem formulation.

2. BACKGROUND

This section presents an overview of the single row layout optimisation problems attempted by researchers.

Heragu and kusiak [1] have illustrated the basic types of FMS layouts and presented two algorithms for solving the problem. Houshyar and Mcginnis [2] have proposed a heuristic procedure to determine an assignment of machines to locations on a linear track. Heragu and Kusiak [3] have formulated facility layout problem as a linear mixed integer programming model and solved both the single row and multi row problems with heuristics. Kouvelis and Chiang [4] have proposed a Simulated Annealing (SA) algorithm for the solution of single row layout problem. Heragu [5] has surveyed the different models of the layout problem and compared various heuristic algorithms for solving the models. Heragu and Alfa [6] have solved single row layout problems with facilities of unequal area and multi row layout problems with facilities of equal area. The 2 way, 3 way exchange algorithm, a modified penalty algorithm and hybrid simulated annealing algorithm have been employed to solve the problem. Two flow analysis methods have been proposed by Ho et al [7] for the design of multi product flow line. The method considered workstation availability and sequence similarity in construction process. A network based approach was used to analyse the material flow in the system.

Kumar et al [8] have proposed a greedy constructive approach for the single row facility layout problem to minimize the material handling cost. A multipass heuristic was developed by Sarkar et al [9] to solve single row layout problem and a depth first insertion heuristic was employed to improve the solution obtained by multipass heuristics. Kouvelis and Chiang [10] have investigated the row layout problem and employed a dynamic layout program to solve general case of row layout problem and integer program to solve equidistance row layout problem. Braglia [11] solved the single row layout problem with a modified heuristic derived from a heuristic developed for the flow shop scheduling problem.

Ho and Moodie [12] have proposed two phase layout procedures that combine flow line analysis with simulated annealing for linear single row flow path. Chen at al [13] have solved linear layout problem for multi products with different operation sequences with limited number of duplicating machines of each type. A modified simulated annealing algorithm was employed to solve the problem. Djellab and Gourgand [14] have presented a heuristic procedure for single row machine layout problem in which an initial feasible layout was constructed in the fist stage and in the next stage it make use of the current feasible layout to construct another layout.

Ponnambalam and Ramkumar [15] have developed two heuristic search algorithms one combining flow line analysis-5 and Genetic algorithm (GA) and the other combining flow line analysis-6 with GA for the design of single row layout problem with an objective of maximizing the number of in sequence movements. Diponegoro and Sarkar [16] have solved linear track problem with introduction of more than one identical machine. The objective was to obtain a machine sequence and flow distribution for identical machines to minimize the total flow distance.

El-Baz M.A [17] has employed GA to solve different type of layout problems. Ficko et al [18] have presented a model for designing of the FMS in one or multiple rows with GA. Amaral A.R.S [19] has proposed a mixed integer linear programming model for the single row facility layout problem. Ponnambalam S. G. et al [20] have proposed three hybrid algorithms to solve single row layout problems. Anjos et al [21] have constructed a Semi Definite Programming (SDP) relaxation which provides a lower bound on the optimal value of single row facility layout problem. Solimanpur et al [22] have formulated a 0-1 non linear mathematical model for the single row layout problem and employed an ant algorithm to solve the model.

The above Literature review reveals the continuous interest shown by the researchers in solving layout problem in FMS. As the problem belongs to NP-complete type, most of the researchers developed heuristics and meta-heuristics to obtain optimal layout. In this paper, an attempt has been made to explore the prospective of SS algorithm in solving layout problems.

3. MATHEMATICAL MODEL

The objective of the single row layout problem is to minimize the total distance travelled by the material handling device. The assumptions made in the model are (i) machines are arranged along a straight line. (ii) machines are oriented in only one direction. (iii) machines are rectangular with unequal sizes. (iv) clearance between each pair of machines is a variable. (vi) distance between the machines is calculated with respect to their centroids. The following mathematical model is used to calculate the objective function [22]:

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

Such that:

$$d_{hl}=\sum_{k=1}^{M}\sum_{t=h+1}^{l-1}L_k x_{kt}+\sum_{k=1}^{M}\sum_{r=1}^{M}\sum_{t=h}^{l-1}S_{kr}x_{kt}x_{r,t+1}+\frac{1}{2}\sum_{k=1}^{M}L_k x_{kh}+\frac{1}{2}\sum_{k=1}^{M}L_k x_{kl} \quad (2)$$

$$\sum_{i=1}^{M}x_{ih}=1; \quad h=1,2,3\ldots\ldots\text{M} \quad (3)$$

$$\sum_{h=1}^{M}x_{ih}=1; \quad i=1,2,3\ldots\ldots\ldots\text{M} \quad (4)$$

$x_{ih}\in\{0,1\}, d_{hl}$

is a real number ; i,h,l = 1,2,3…….M (5)

Notations

- M total number of machines and locations
- f_{ij} the number of trips from machine i to machine j (f_{ii}= 0)
- L_i the length of machine i
- S_{ij} the clearance required between the machines i and j
- d_{hl} the distance between the centroids of locations h and l ($d_{hl} = d_{lh}$: $\forall h,l$)
- x_{ih} binary decision variable which is 1 if i is located at location h and 0 otherwise.

The objective function (1) represents the total distance travelled by material handling device. Constraints (2) account for the distance between the location h and l. Constraints (3) ensure that only one machine is assigned to each location. Similarly, constraints (4) guarantee that each machine is assigned only one location. Constraint (5) define the domain for x and d variable.

4. SCATTER SEARCH

4.1 Introduction

Glover (1977) introduced SS as a heuristic for solving Integer Programming Problems [23]. As like GA, SS is also belongs to evolutionary computation family from the point of view that they build, maintain and evolve a population of solutions for the purpose of generating new trial solutions.

In SS the initial population is created with good and diversified solutions. Then a reference set (RefSet) is generated from initial population of solutions. It uses Refset to combine its solutions and constructs other solutions. Size of the RefSet in SS is relatively small when compared to the population size of other evolutionary algorithms. In other evolutionary algorithms like GA, reproduction is based on probabilistic selection of parents, where as in SS, it is based on deterministic selection of reference solutions. For combining the solutions, SS operates on unifying principles based on strategic designs, where other approaches use randomisation methods like cross over and mutation.

The applications of SS method includes Vehicle routing, Quadratic Assignment, Financial Product Design, Neural Network Training, Job Shop Scheduling, Flow Shop Scheduling, Graph Drawing, Linear Ordering, Unconstrained Continuous Optimisation, Bit representation, Optimising Simulation and Complex System Optimisation [24].

4.2 Scatter Search algorithm -steps

Step 1 Diversification Generation Method

Generate a set of solutions which focuses on diversification in a systematic way.

Step 2 Improvement method

In order to obtain a set of solutions with reasonable quality an improvement method is applied.

Step 3 Reference set update method

The Refset is a collection of B solutions that are used to generate new solution by way of applying a solution combination method. The initial Refset contains B/2 best solutions and B/2 diverse solutions.

Step 4 Subset generation method

This method combines all pairs of solutions in the current Refset. Different subset type is formed by combining two or more Refset solutions.

Step 5 Combination method

This method combines the solution in each subset to form new solutions. Again the improvement method is applied on the combined new solution to get an improved solution. The Refset is updated with improved solutions and the procedure is repeated until the convergence of Refset. The flow chart of the proposed SS algorithm is shown in fig 1.

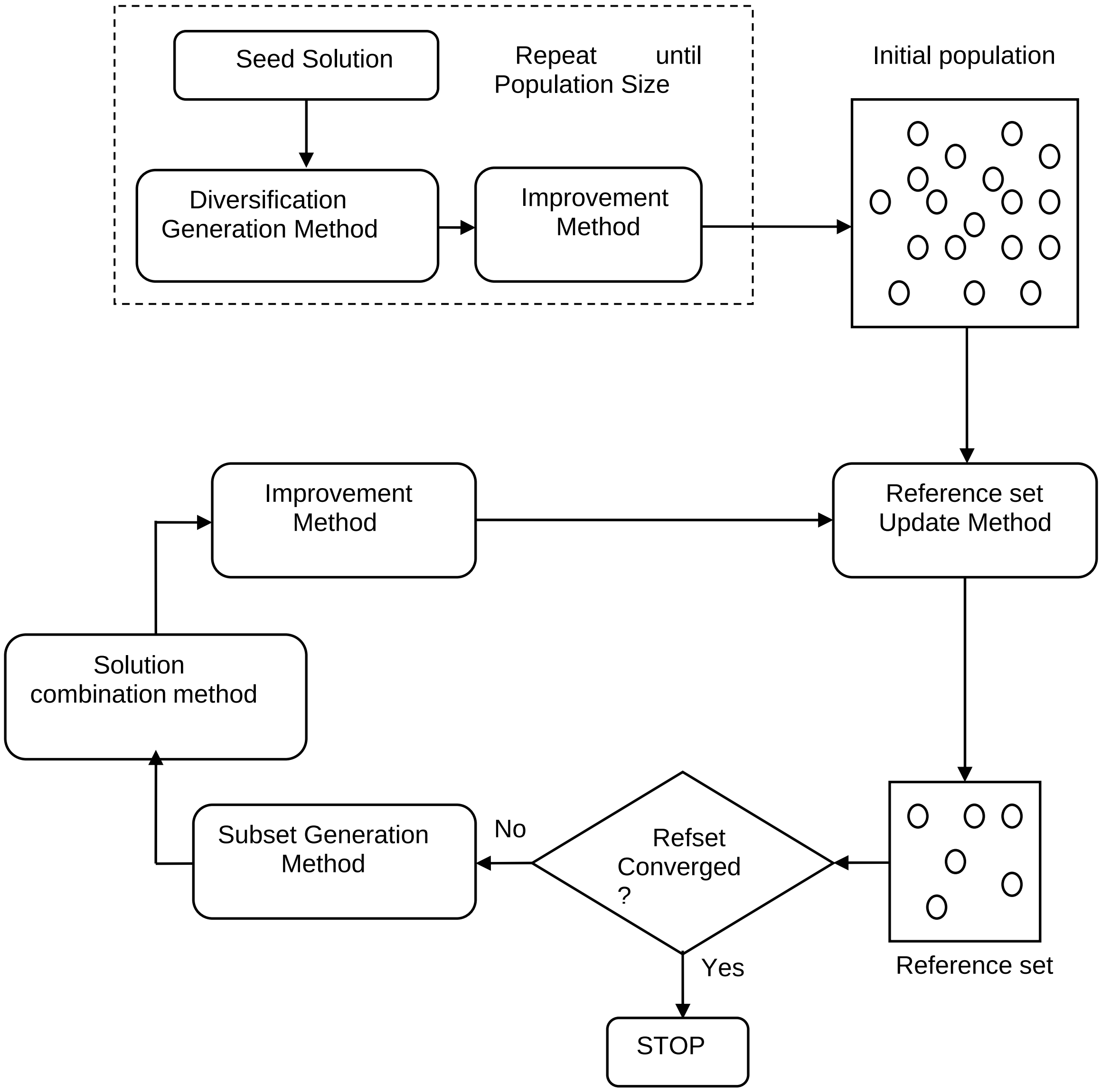

Figure 1: Flow chart of proposed Scatter Search algorithm.

4.3 Numerical illustration

Step 1: Generate a random sequences of population size P.
Step 2: Diversification generation method

On each randomly generated sequence apply diversification generation method which is used to generate diverse solutions. Glover [24] suggested a method for generating diversified solution using the following expression.

In this method, the subsequence is generated from the original sequence using the following expression:

$$P(h{:}S) = (S, S + h, S + 2h \ldots. S + rh) \quad (6)$$

Where:

- $P(h{:}S)$ – subsequence
- h – generally value ranges from 1 to $N/2$.

  preferably 'h' allowed to take the two values closest to the square root of 'N'.

This 'h' value gives maximum separation of each element from each other element in the new diversified sequence.

- S – positive integer between 1 and h.
- r – largest non negative integer such that $S + rh \leq N$.
- N – Total number of machines.

The new diversified solution is formed by combining different subsequences. The new diversified sequence is defined by the following expression:

$$P(h) = (P(h{:}\,h), P(h{:}h\text{-}1) \ldots P(h{:}1)) \quad (7)$$

This is illustrated through following example.

Number of machines - 10
Let us take the randomly generated sequence $P =$ (1 2 3 4 5 6 7 8 9 10)
$h = 3$
Then the new diversified sequence is
$P(3) = (P(3{:}\,3), P(3{:}2) \ldots P(3{:}1))$
$= ((3,6,9),(2,5,8),(1,4,7,10)$
$= (3,6,9,2,5,8,1,4,7,10)$

Step 3: Improvement method

Improvement method is used to obtain a better solution through some heuristics. In the proposed algorithm a left and right insertion heuristic is used. In this method, the machine sequence is divided into two halves. If the total numbers of machines are odd in number more than half the number is taken in the left side. The machine which is on the left side is inserted in the right side and machine which are on the right side is get inserted in the left side to obtain new sequences. Among the generated new sequences the sequence with minimum objective value is retained to form the Refset. The improvement heuristic is shown in fig 2.

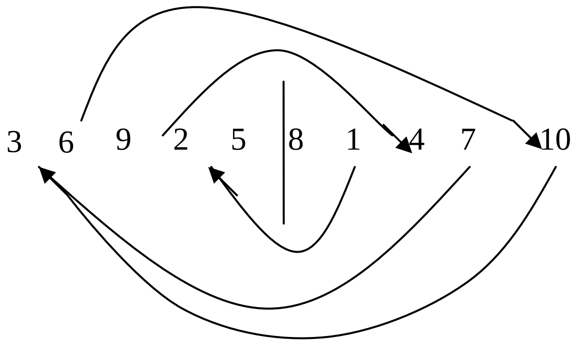

Figure 2: Illustration of Improvement Heuristic.

Step 4: Reference set update method

This method builds and updates Refset consisting of B solutions. A set of good representative solutions of the population is chosen to generate the Ref set. The good solutions are not limited to those with the best objective function values (OFVs) and some inferior solution is also to be included in the Refset to introduce diversity. The considered Refset B consists of B1 + B2 solutions. B1 consists of best OFVs and B2 consists of diverse solutions.

In the proposed method the set of improved solutions obtained from the improvement method is ordered based on OFV. The superior solutions will be in the top of the list and inferior solutions are in the bottom of the list. From the list, 50% of the superior solutions and 50% of the inferior solutions are extracted to form the Refset B. Suppose if the Refset size is 4, it consists of 2 best solutions and 2 diverse solutions. An example Refset of size 4 is given below.

Solution 1 :	5	4	7	9	2	1	10	8	6	3	Best solutions
Solution 2 :	7	1	10	5	4	2	8	3	6	9	
Solution 3 :	6	8	3	5	10	7	4	2	9	1	Diverse solutions
Solution 4:	10	6	2	9	4	1	5	3	8	7	

Step 5: Subset Generation method

Subset generation method produces the subset of solutions of the Refset. This method facilitates the construction of combined solutions. Different collections of subset are formed by combining two or more solutions from the Refset [24]. The example shows the subsets produced for 5 machine problem.

SubsetType 1: all 2-element subsets.
(1,2), (1,3),(1,4),(1,5),(2,3) ….(4,5) Thus Type-1 subset consists of $(b^2 - b) / 2$ pair wise combinations of the solutions.

SubsetType 2: 3-element subsets derived from the 2-element subsets by augmenting each 2-element subset to include the best solution not in this subset.
(1,2,3),(1,2,4),(1,2,5),(1,3,5)……..(3,4,5)

SubsetType 3: 4-element subsets derived from the 3-element subsets by augmenting each 3-element subset to include the best solutions not in this subset.
(1,2,3,4),(1,2,3,5)…………(2,3,4,5)

SubsetType 4: the subsets consisting of the best i elements, for i = 5 to b.
(1,2,3,4,5)

Step 6: Combination method

Combination method in SS does the work of mutation operators in GA. Different combination methods were discussed elaborately by Marti et al [23].

The combination method scans the Refset solutions from left to right and uses the rule that each reference solution votes for its first element that is not included in the trial solution (referred to as the incipient element). The voting determines the next element to enter the first still unassigned position of the trial solution. This is a min-max rule in the sense that if any element of the reference solution is chosen other than the incipient element, then it would increase the deviation between the reference and trial solutions. Similarly, if the incipient element were placed later in the trial solution than its next available position, this deviation would also increase. So the rule attempts to minimize the maximum deviation of the trial solution from the reference solution under consideration, subject to the fact that other reference solution is also competing to contribute [23].

The solutions in the produced subsets are combined to produce a new solution, and it is carried out as follows. Let us take for example subset (1, 2) in which solution 1 and solution 2 in the Refset are considered for combination.

Solution 1: 5 4 7 9 2 1 10 8 6 3

Solution 2: 7 1 10 5 4 2 8 3 6 9

In the above illustration arrow mark indicates the direction of scanning the Solution1 and Solution 2 to form combined solution. Once all the elements are included in the combined solution the scanning stops. The combined sequence obtained is given below.

Combined sequence: 5 7 4 1 10 9 2 8 3 6

Similarly, different combined solutions are obtained by combining the solutions in the subsets. The combined solutions are improved again by left and right insertion heuristic and the Refset is updated with new improved solutions. The procedure is repeated for the new updated Refset until the convergence of solutions occurs.

5 COMPUTATIONAL RESULTS

The proposed SS algorithm is coded in C++ and run on a Pentium IV 256 MHz computer. To validate the performance of the algorithm, it is tested on a set of problems available in the literature [22] and the results are compared with other algorithms.

For the first set of problems, flow matrix is obtained from Nugent et al [25] and the dimensions of the machines are taken from Heragu and kusiak [1]. The clearance between the machines are assumed as 0.01 unit distance

For the second set of problems, clearance between the machines is assumed different and the data for the clearance is obtained from Nugent et al [25]. The distances between the departments given in Nugent et al are assumed as clearance between the machines.

Table I shows the comparative results of SS algorithm with the other algorithms available in the literature for the first set of problems. The optimal OFVs are highlighted in boldface in the table.

For the test problems one to seven, the proposed algorithm out performed all other heuristics available and give comparable results with Ant Colony Optimisation (ACO) algorithm. SS has produced improved results for the 30 machine problem over the existing algorithms. The CPU time given is in second and indicated within square brackets in Table I.

Table I: Comparative result of the SS with existing algorithms for the first set of problems.

P. No.	No of M/cs	Hall [26] OFV [Time]	Neghabat [27] OFV [Time]	Heragu & Kusiak [1] OFV [Time]	Kumar et al [8] OFV [Time]	ACO [22] OFV [Time]	Proposed method OFV [Time]
1	5	1.1409 [0.018]	**1.100** [0.009]	1.165 [0.007]	1.140 [0.008]	**1.100** [0.00]	**1.100** [0.00]
2	6	**1.990** [0.034]	2.030 [0.015]	2.085 [0.008]	1.990 [0.010]	**1.990** [0.00]	**1.990** [0.00]
3	7	5.530 [0.052]	**4.730** [0.500]	5.420 [0.017]	5.150 [0.033]	**4.730** [0.00]	**4.730** [0.00]
4	8	7.035 [0.069]	**6.295** [5.500]	7.995 [0.035]	6.775 [0.067]	**6.295** [0.00]	**6.295** [0.00]
5	12	27.805 [0.120]	24.675 [1800.0]	31.525 [0.067]	24.715 [0.100]	**23.365** [0.06]	**23.365** [0.10]
6	15	48.375 [0.153]	51.500 [1800.0]	62.624 [0.100]	54.830 [0.600]	**44.599** [0.18]	**44.599** [0.30]
7	20	153.260 [0.255]	169.920 [1800.0]	178.149 [0.600]	141.040 [4.200]	**119.71** [1.80]	**119.71** [2.80]
8	30	395.770 [0.624]	466.800 [1800.0]	414.400 [4.800]	405.520 [14.800]	334.87 [37.3]	**331.39** [42.5]

The comparative results of SS algorithm with the other algorithms for the second set of problems are shown in Table II. The proposed SS algorithm is compared with 2-opt algorithm and ACO. Except test problem one & two the SS algorithm has obtained improved results for all the test problems. In particular SS has produced improved results for the large size instances of 20 & 30 machine problems with significant reduction in the OFV. To compare the results of SS algorithm with other algorithms, the OFV is averaged over the five runs of SS algorithm on each problem and the values are indicated in Table II. The minimum OFV obtained among the 5 runs by SS algorithm and the corresponding layout for the second set of problem is shown in Table IV.

Table II: Comparative results of the SS with other algorithms for the second set of problems.

P.	No of	2-opt algorithm		ACO		Proposed method	
No	M/cs	OFV	Time	OFV	Time	OFV	Time
1	5	**33.95**	0.00	**33.95**	0.00	**33.95**	0.00
2	6	55.40	0.00	**52.48**	0.00	**52.48**	0.00
3	7	116.94	0.00	114.06	0.00	**111.72**	0.00
4	8	174.21	0.00	169.97	0.00	**167.85**	0.00
5	12	702.21	0.01	660.02	0.05	**652.50**	0.10
6	15	1770.53	0.05	1460.25	0.22	**1445.56**	0.35
7	20	4279.64	0.01	3853.00	2.00	**3773.13**	3.20
8	30	14569.23	0.15	11771.89	32.84	**11306.82**	46.20

In both the first and second set of problems, the computational time spent by the proposed SS algorithm is slightly higher when compared with ACO, yet the proposed SS algorithm has produced better solutions when compared with other algorithms. The optimal layout sequence obtained for the first and second set of problems is shown in table III and IV respectively.

Table III: Optimal layout sequence for the first set of problems.

P. No.	No of M/cs	OFV	Optimal layout sequence
1	5	1.100	3-2-1-5-4
2	6	1.990	6-5-4-1-2-3
3	7	4.730	7-3-6-5-4-2-1
4	8	6.295	7-6-5-4-8-1-2-3
5	12	23.365	6-5-10-2-7-1-8- 4-11-12-9-3
6	15	44.599	7-11-12-9-8-5-13-2-14-1-4-3-15-6-10
7	20	119.71	9-3-6-13-10-18-17-5-14-19-2-15-8-12-16-4-11-1-7-20
8	30	331.39	20-26-15-27-17-10-24-1-28-12-6-22-18-14-23-8-11-4-16-30-25-13-19-29-2-5-7-9-3-21

Table IV: Optimal layout sequence for the second set of problems.

P. No.	No of M/cs	OFV	Optimal layout sequence
1	5	33.95	5-4-2-1-3
2	6	52.48	6-5-4-1-2-3
3	7	111.72	5-6-7-4-3-2-1
4	8	167.85	5-6-7-8-4-3-2-1
5	12	647.86	9-12-11-7-8-4-3-2-1-5-6-10
6	15	1434.87	1-2-7-6-11-12-13-8-9-14-15-10-5-4-3
7	20	3736.20	16-11-1-6-7-12-13-8-9-14-19-20-15-10-5-4-3-2-17-18
8	30	10940.15	21-25-19-13-7-1-2-3-4-5-6-12-18-24-30-29-23-17-11-10-9-8-14-20-26-27-28-22-16-15

6. CONCLUSIONS

In this paper, SS algorithm is employed to solve single row unequal area facility layout problem. The efficiency of the algorithm is tested on benchmark problems available in the literature. From the experimental results, it is observed that the proposed SS algorithm performed well on number of test problems. It is noteworthy that the proposed SS algorithm yields better solutions for the large size instances. The computational time required to obtain the optimal layout is slightly higher for large size instances. As a future work the proposed SS algorithm can be employed to solve other type of layout problems such as multi-row and loop layout problem in FMS.

REFERENCES

[1] Heragu, S.S.; Kusiak, A. (1988). Machine layout problem in flexible manufacturing systems, *Operations research*, Vol.36, No.2, 258-268.

[2] Houshyar, A.; Mcginnis, L. F. (1990). A heuristic for assigning facilities to locations to minimize WIP travel distance in a linear facility, *International Journal of Production Research*, Vol.28, No.8,1485-1498.

[3] Heragu, S.S.; Kusiak, A. (1991). Efficient models for the facility layout problem, *European Journal of Operation Research*, 53, 1-13.

[4] Kouvelis, P.; Chiang,WC. (1992). A simulated annealing procedure for single row layout problems in flexible manufacturing systems, *International journal of production research*, Vol.30, No.4, 717-732.

[5] Heragu,S.S. (1992). Invited Review. Recent models and techniques for solving the layout Problem, *European Journal of Operation Research*, 57, 136-144.

[6] Heragu, S.; Alfa, A.S. (1992). Experimental analysis of simulated annealing based annealing algorithms for the layout problem, *European Journal of Operation Research*, 57,190-202.

[7] Ho, Y-C ; Lee C-En-C, Moodie C.L. (1993). Two sequence pattern, matching based, flow analysis methods for multi flow lines layout design, *International Journal of Production Research*, Vol. 31. No.7, 1557-1578.

[8] Kumar, K R.; Hadjinicola, G.C.; Lin, TL. (1995). A heuristic procedure for the single row facility layout Problem, *European Journal of Operation Research*, 87, 65-73.

[9] Sarkar, B.R.; Wilhelm, W.E.; Hogg, G L; Han M-H.(1995). Backtracking of jobs in-one dimensional machine location problems, *European Journal of Operational Research,* Vol.85, 593-609.

[10]Kouvelis, P; Chiang, W-C. (1996). Optimal and heuristic procedures for row layout problems in automated manufacturing systems, *The Journal of the Operational Research Society*, Vol.47, No.6, 803-816.

[11]Braglia, M (1997). Heuristics for single- Row layout problems in flexible manufacturing Systems, *Production Planning and Control*, Vol.8, No.6, 558-567.

[12]Ho, Y.C. Moodie, C.L. (1998). Machine layout with a linear single row flow path in automated manufacturing system, *Journal of manufacturing systems*, Vol.17/No.1,1- 21

[13]Chen, D.S.; Wang, Q; Chen, H-C. (2001). Linear sequencing machine layouts by modified simulated annealing, *International Journal of production Research*. Vol.39. No.8, 1721-1732.

[14]Djellab, H.; Gourgand, M.; (2001). A new heuristic procedure for the single row facility layout problem, *International Journal of computer integrated manufacturing*, Vol.14, No.3, 270-280.

[15]Ponnambalam, S. G.; Ramkumar, V. (2001). A genetic algorithm for the design of single row layout in automated manufacturing systems, *The International .Journal of advanced Manufacturing Technology*, 18, 512-519.

[16]Diponegoro A.; Sarkar, B. R. (2003). Flow distance reduction for a multi-product flow line with sets of identical machines, *European Journal of Operational Research,* 147, 591- 612.

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

[18]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*,Vol.157-158, 150-158.

[19]Amaral, A.R.S (2005). On the exact solution of a facility Layout problem, *European Journal of operational research*, Vol.173, 2, 508 -518.

[20]Ponnambalam, S.G.; Venkataraman, R; Sudhan, H H; Chatterjee, P V. (2005). Hybrid search algorithms for single row layout in automated manufacturing systems, *International Journal of Industrial Engineering*, 12, 2,115-124
[21]Anjos, M.F.; Kennings, A.; Vannelli, A. (2005). A semi definite optimisation approach for the single row layout problem with unequal dimensions, *Discrete optimisation,* 2, 113-122.
[22]Solimanpur, M.; Vrat, P.; Shankar, R. (2005). An ant algorithm for the single row layout in flexible manufacturing system, *Computers and Operations Research*, Vol.32,3,583-598.
[23]Marti, R.; Laguna, M.; Campos, V. (2005). Scatter search vs Genetic algorithms: an experimental evaluation with permutation problems. In: Rego, C; Alidaee, B. (eds) Metaheuristic optimisation via adaptive memory and evolution; tabu search and scatter Search, kluwer Academic publishers, Norwell, Massachusetts, 263-282.
[24]Glover, F. (1998). A template for scatter search and path relinking, In: Hao, JK; Lutton, E; Ronald, E; Schoenauer, M.; Snyers D (eds) Lecture notes in computer science, Vol.1363, 13-54.
[25]Nugent, E.; Vollman, T. E.; Ruml, J. (1968). An experimental comparison of techniques for the assignment of facilities to locations, *Operational Research*. Vol.16, 150-173.
[26]Hall, K.M. (1970). An r-dimensional quadratic placement algorithm, *Management Science,* 17(3), 219-229.
[27]Neghabat F.(1974). An efficient Equipment layout algorithm, *Operations Research*, 22, 622-628.