

A Simple Heuristic for Linear Sequencing of Machines in Layout Design

*M. Siva Kumar, ¹M.N. Islam, N. Lenin, D. Vignesh Kumar and L. Vadivelkannan
*Department of Mechanical Engineering, National Engineering College, K.R. Nagar,
Kovilpatti, Tamilnadu – 628 503. India.*

¹*Department of Mechanical Engineering, Curtin University of Technology,
GPO Box U 1987, Perth WA 6845*

Abstract

This paper presents a simple heuristic to determine a common linear machine sequence for multiple products with different operation sequences and a limited number of duplicate machine types available for the job. The heuristic is based on minimization of the total flow distance traveled by a product on the linear machine sequence. It is assumed that the flows of products are allowed only in the forward direction, either in-sequence or by-pass. It is also assumed that backtrack movements are not allowed. The effectiveness of the proposed heuristic is demonstrated through the solutions of two typical layout design problems taken from the literature. Subsequently, a number of additional problems are solved and their results are compared with the results applying existing methods. The results indicate that the proposed method can be an effective tool in solving layout design problems.

Keywords: layout; linear sequencing; heuristics; flow distance

1. Introduction

To be converted into a desired shape or product, materials have to pass through a sequence of operations performed on a number of machines arranged in various configurations such as a straight line, a U-shape line, a serpentine line or a loop. The machine layout design involves determination of the relative positions of the machines in a layout.

The layout for a cellular manufacturing or assembly cell can be classified as: (a) unidirectional network loop layout, (b) linear single-row layout, (c) linear double-row layout, (d) circular layout and (e) cluster layout (Heragu 1997; Heragu and Kusiak 1988). A linear machine sequence is the most popular among production systems due to its simplicity and efficient flow structure (Heragu and Kusiak 1989; Kouvelis and Chiang 1992), as well as for its ability to arrange machines in a variety of flow configurations such as straight line, U-shape line, serpentine line or loop for a conveyor or an automated guided vehicle (AGV) system (Aneke and Carrie 1986). It has the benefits of shorter flow distance, easier control of the production process and easier material handling. This type of layout configuration is the most commonly used form in cellular manufacturing system (Nicholas 2001; Zhou and Irani 2003) and flexible manufacturing system (Houshyar and MCGinnnis 1990; Kouvelis and Kim 1992; Lee et al., 2001). Hence, in this work, a linear machine sequencing method has been selected.

2. Literature review

In this section, a review of the existing methods for linear sequencing of machines in layout design is presented. Heragu and Kusiak (1988) showed that the quadratic assignment formulation cannot be used to model the machine layout

* Corresponding author Email: lawan_sisa@yahoo.com

problems in a flexible manufacturing system and developed and successfully implemented a triangle assignment algorithm. In terms of computation time, their work produced better results than other existing methods. Heragu and Kusiak (1990) presented two efficient models; namely, a linear continuous and linear mixed integer for the facility layout problems. The proposed model doesn't require prior knowledge about the location of sites. The authors have shown that the continuous models are more useful for solving the facility layout problem than other models published in the literature. Houshyar and McGinnis (1990) introduced a heuristic for assigning facilities to locations to minimize the work-in-process travel distance in a straight track. The authors have shown that the performance of their heuristic performs better than the modified and classical lower bound methods for the test problems under consideration. Heragu and Alfa (1992) presented experimental analysis of simulated annealing-based algorithms, namely, a modified penalty algorithm, the simulated annealing algorithm and a hybrid simulated annealing algorithm for single-row layout problems with facilities of unequal area and multi-row layout problems with facilities of equal area. The authors have concluded that the hybrid algorithm produces better quality solutions than the other algorithms—at the expense of slightly more computation time. Kouvelis and Chiang (1992) utilized a simulated annealing procedure to determine a flow line (or single-row layout) under the assumptions that the number of machines is fixed and backtrack movements are allowed. Their objective was to minimize the total backtrack distance. Ho et al. (1993) proposed two flow analysis methods for multi-flow line layout design to get smaller flow distance in which the first method adopted the traditional line structure for analysis and the second method implemented a network structure for the analysis. The authors also have developed a heuristic pattern matching method for single-row layout problems in flexible manufacturing systems in which a linear machine sequence is initially constructed by the product with the largest number of operations.

Braglia (1997) concluded that the linear machine sequencing problem is a non-polynomial (NP) hard combinatorial problem. The number of possible sequences even grows exponentially as duplicate machines are allowed. Moreover, the set of all feasible sequences is not simply the set of simple permutations of a fixed number of machines, as they must satisfy different operation sequences of all products. Ho and Moodie (1998) investigated the machine layout problem with a linear single-row flow path for an automated manufacturing system using a simulated annealing algorithm. The authors also have investigated the effect of flow path characteristics on machine layouts, and provided vital information on selecting appropriate flow line analysis methods and to determine the appropriate evaluation criteria for different layout problems. Azadivar and Wang (2000) presented a facility layout optimization technique with the considerations of the dynamic characteristics and operational constraints of the system. Hakim (2000) demonstrated the difficulties in applying classical crossover and mutation operators for solving facility layout problems and also studied the applicability of genetic operations such as diagonal crossover and cloning in generating feasible offspring.

Chen et al. (2001) addressed the problem of determining a common linear machine sequence for multiple products having different operation sequences and a limited number of duplicate machine types. The authors have considered the objective of minimizing the total flow distance traveled by the products on this linear flow line using modified simulated annealing algorithm. Braglia (2003) proposed the adoption of indices to identify the preferred strategy for a single-row layout. Diponegoro and Sarker (2003) presented a two-stage solution methodology to ease the computation

and to obtain a better solution for the problem of minimizing the travel distance that involves sets of identical machines. This problem has often been formulated as a tertiary assignment problem because of its combinatorial nature. Dunker et al. (2003) presented a co-evolutionary approach to the numerical optimization of large-facility layouts and used a mixed-integer model. The authors have suggested some improved mutation and cross-over operators for GA. Shayan and Chittilappilly (2004) reported on the design, development and experimentation results of a new genetic algorithm named GA.FLP.STS to solve facility layout problems.

Ficko et al. (2004) discussed a model for designing the flexible manufacturing system (FMS) in one or multiple rows with genetic algorithms. The authors have established the most favorable number of rows and the sequence of devices in the individual row by applying genetic algorithms. Chrysostomos and Vlachos (2005) used the linear programming model for minimal backward-flow, to determine the optimal linear machine sequence in a manufacturing cell and applied a modified ACS algorithm for the conditions and parameters of the linear machine layout problem. Anjosa et al. (2005) introduced a semi-definite programming approach to determine the optimal linear placement of facilities with varying dimensions on a straight line for the one-dimensional space-allocation problem (ODSAP), also known as the single-row facility layout problem. Pillai et al. (2005) addressed the determination of a linear sequence that minimizes the total distance traveled by multiple items with different operation sequences. The authors have considered that each type of machine available is limited and adopted a simulated annealing algorithm to find the best solution.

Solimanpur (2005) et al. formulated the single-row machine layout problem as a non-linear 0-1 programming model in which the distance between the machines is sequence dependent. The authors have developed an ant colony algorithm to solve this problem. Andre and Amaral (2008) proposed a mixed 0-1 linear program for the one-dimensional facility layout problem to minimize the weighted sum of the distances. Teo and Ponnambalam (2008) proposed a hybrid ACO/ PSO heuristic to solve single-row layout problems. Lin (2009) proposed a hierarchical order-based genetic algorithm to minimize the moving distance among cutting pieces during apparel manufacturing for U-shaped single-row machine layout.

Lee et al. (2001) proposed both heuristic and branch and bound algorithms to solve the unidirectional loop layout problem. Nearchou (2006) used a differential evolution algorithm (DEA) to solve the loop layout problem and proposed a mapping mechanism for encoding the floating-point chromosomes for combinatorial problems with permutation property. Chae and Peters (2006) used a single loop as a structural frame for positioning cells and solved an open-field type layout design problem with no predetermined flow path configuration and assumed that the machines or cells were located inside of a single loop.

Wang et al. (1998) formulated a model to minimize the total material handling distance on the shop floor in both inter-cell and intra-cell facility layouts for cellular manufacturing systems. The authors have used the improved simulating algorithm to solve this problem. Kim and Kim (2000) projected that nearly 15–70% of the manufacturing costs are due to material handling. This may be reduced by efficient distribution of different resources. Aiello et al. (2002) proposed an integrated approach to the facilities and material handling system design with the objective of the minimization of materials-handling costs. Adel El-Baz (2004) described a genetic algorithm (GA) to solve the problem of optimal facilities layout in manufacturing systems design to minimize the materials-handling costs. Hicks (2004) described that

a genetic algorithm-based optimization method produced layouts that significantly minimizes material movement for a given schedule of work in both green field and brown field situations. Balamurgan et al. (2008) used a Genetic Algorithm to solve a multiple objective machine layout problem with unequal area and fixed shape. The proposed model minimizes the cost associated with material handling during normal and breakdown periods and the cost of unusable space. Satheesh Kumar et al. (2010) employed an artificial immune system (AIS) algorithm to minimize the materials-handling cost both in single-row and loop layout problems in FMSs.

Even though numerous attempts have been made in earlier literature in the field of linear sequencing of machines to minimize flow distance, improved heuristics are still needed to minimize both flow distance and the total number of machine types. Hence, in this work an attempt has been made to develop a simple heuristic to determine a linear machine sequence with minimum flow distance.

3. Problem definition

Locations of machines in the linear machine sequence of a single-row layout design play a vital role in determining the flow distance of multiple products. The single-row layout design of duplicate machines with multiple products is a non-polynomial problem solution which by the existing methods produces inconsistent results due to its randomness.

Let n be the number of products. The number of products along with their machine sequences and flow distances are known. The number of machine types and its duplicates are also known. It is proposed to determine the linear sequence of the machine arrangement which minimizes the total flow distance.

4. Mathematical model

The following assumptions are made to determine the linear machine sequence:

- The number of products, the flow distance and the machine type sequences are known along with the availability of duplicate machine types for use.
- The products are entered at their respective starts of the machine types.
- The products' flow distances are considered up to the end machine types of the respective product.
- Backtracking is not permitted

The objective of the proposed work is to minimize the total flow distance which can be calculated by applying Eq. (1). The constraints are presented in Eq. (2) to Eq. (6):

$$td = \sum_{i=1}^{np} \sum_{j=1}^{nm_i} d_i (L_{ij+1} - L_{ij}) \quad (1)$$

where,

td – total flow distance

d_i – i^{th} product flow distance

L_{ij+1} – i^{th} product's $j+1^{\text{th}}$ machine location in the machine sequence

L_{ij} – i^{th} product's j^{th} machine location in the machine sequence

np – number of products

nm_i – number of machines in i^{th} product's machine sequence

$$L_{ij+1} > L_{ij} \quad (2)$$

$$L_{ij} > L_{i1} \quad (3)$$

$$nm_k \leq ndm_k \quad (4)$$

where,

nm_k – number of k^{th} machine available in the final linear machine sequence
 ndm_k – number of duplicate k^{th} machine type available for usage

$$tm = \sum_{k=1}^{nmt} ndm_k \quad (5)$$

where,

tm – total number of machines available for use

nmt – number of machine types

k – index to represent the machine type $k = 1, 2, 3, \dots, nmt$

$$nm_s \leq tm \quad (6)$$

where,

nm_s – total number of machines available in the final linear sequence.

Eq. (2) states that the location of a $j+1^{\text{th}}$ machine should always be greater than the location of a j^{th} machine in the linear machine sequence. Eq. (3) reveals that the location of a $j+1^{\text{th}}$ machine of the individual product machine sequence should be always greater than the location of the first machine in the linear machine sequence. Eq. (4) indicates that the number of k^{th} machine types available in the final linear machine sequence should be less than or equal to the number of duplicate k^{th} machine types available for usage. Eq. (5) demonstrates that the total number of machines is equal to the sum of the duplicates of individual machine types. Eq. (6) shows that the total number of machines in the linear sequencing must be less than or equal to the total number of machines available for usage including the duplicate machines.

5. Methodology

A simple heuristic has been developed in this work with the aim of reducing the computation time and improving the consistency of the solutions with minimum total flow distance. The detailed algorithm is presented below.

Step 1: Read the number of machine types (nm), number of duplicate machines in each type ($mtn[]$), number of products (np), number of machine types in each product ($nmp[]$), machine type sequence for each product ($pseq[][]$), distance of each product ($pd[]$).

Step 2: Arrange the product in descending order based on product demand and store in product number ($pno[]$).

Step 3: Assign the machine type for the first product's machine sequence and store in $b[]$. Update the availability of the machine type in $mtn[]$.

Step 4: For each of the remaining products in $pno[]$, perform the steps 5 to 12.

Step 5: For each machine type (mno) in the product sequence $pseq[][]$, perform step 6.

Step 6: If the machine mno is unassigned then add the machine in front of the existing machine sequence $b[]$ and update its availability. Go to step 5.

Step 7: If the machine type is assigned, check the machine type mno in the existing sequence $b[]$.

Step 8: If available, check the availability of the remaining machine type of $pseq[][]$ in $b[]$.

Step 9: If all the machine types are available, then go to step 4.

Step 10: If not, then check whether machine type mno is assigned. If the machine type mno is unassigned then insert the mno in the appropriate location (the location that doesn't affect the existing product machine sequence in $b[]$) after the location of the previous machine type in the existing machine sequence $b[]$ and update the availability. Go to step 5.

Step 11: If the machine type mno is unassigned but the insertion in $b[]$ affects the existing product sequence, then add the mno at the end of $b[]$ and update the availability. Go to step 5.

Step 12: If the machine type mno is assigned, then the existing sequence is not feasible; stop the program.

Step 13: Display the linear sequence of machine type $b[]$.

Step 14: For each product, check whether the product machine type sequences are in successive order in the final linear machine type sequence $b[]$. If not, try to shift the machine type as far as possible and close to the previous machine either without affecting the other product machines sequences or allowing the minimization of flow distance to occur.

6. Numerical Illustration

A problem discussed in Pillai et al. (2005) has been considered as an example problem to illustrate and to implement the proposed heuristic algorithm. Two procedures are established by implementing the proposed heuristic. In the first procedure, the same numbers of duplicate machine types are considered as considered by Pillai et al. (2005). In the second one, it is assumed that there is no duplicate machine type available for machine type 6.

6.1 Procedure for determining the linear machine sequence for case 1

Table 1 represents the number of machine types and the duplicates available for the job. The number of products, the machine sequence and the flow distance are listed in Table 2.

Table 1. Number of machine types and its duplicates

Machine types	1	2	3	4	5	6	7
No of machines available	2	1	1	1	1	1	1

Table 2. Machine sequence of individual products

S.No.	Machine type sequence ($pseq[][]$)	Distance ($pd[]$) in units
1	1-3-2-6-5	800
2	4-6-1-7	400
3	4-1-6-5	300
4	4-3-2-5	200
5	4-1-3-2	100

Step I: As per step 2 of the methodology described in Section 5, the product numbers are arranged in descending order based on the flow distance ($pd[]$). Since the problem is already in descending order, Table 2 can be used as it is for executing the next step.

Step II: Select the first product machine type sequence 1-3-2-6-5. Assign machine types 1, 3, 2, 6 and 5. Store in $b[]$. Update the availability of machine types; the same is given below in Table 3.

Table 3. Availability of duplicate machine type

Machine types	1	2	3	4	5	6	7
No of machines available	1	0	0	1	0	1	1

The linear machine type sequence ($b[]$) after the first product is 1-3-2-6-5.

Step III: The next product sequence is 4-6-1-7. As per step 6, machine types 4, 6, 1 and 7 are all unassigned, hence add the machines in front of the existing sequence $b[]$. The linear machine type sequence ($b[]$) after the second product is 4-6-1-7-1-3-2-6-5. All the machine types are assigned.

Step IV: The next product machine type sequence is 4-1-6-5. As per step 7, the machine type 4 is assigned and it is available in the existing sequence $b[]$. As per step 8, the remaining machine types 1, 6 and 5 are also available in $b[]$. Hence, the existing sequence is suitable for product number 3 without any modification. The linear sequence after the third product is 4-6-1-7-1-3-2-6-5.

Step V: In the next two product machine type sequences, the same sequences as steps 7 and 8 are applicable. Hence, the linear machine sequence after considering all the product machine type sequences is 4-6-1-7-1-3-2-6-5.

Step VI: As per step 14, it is not possible to shift the machine types to make the sequence in a successive manner without affecting the other product machine type sequence. Hence, the final linear machine sequence is 4-6-1-7-1-3-2-6-5.

Step VII: The flow distance for individual products is shown in Figure 1.

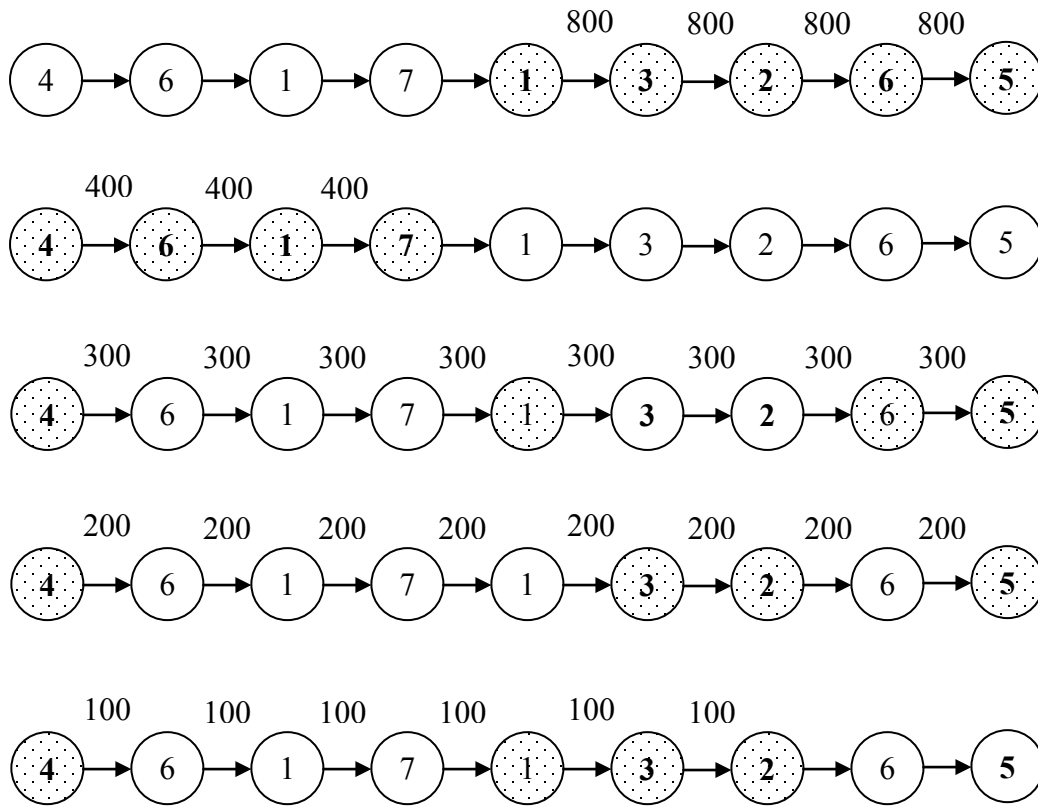


Figure 1. Flow distance of individual product for case 1

The total flow distance is computed based on Eq. (1) and it is given by
 $td = 4 \times 800 + 3 \times 400 + 8 \times 300 + 8 \times 200 + 6 \times 100$
 $td = 9000$ units

6.2 Procedure for determining the linear machine sequence for case 2

In this procedure the individual product machine type sequences are assumed to be the same as given in Table 2. The availability of duplicate machine types is listed in Table 4.

Table 4. Number of machine types and its duplicates

Machine types	1	2	3	4	5	6	7
No of machines available	2	1	1	1	1	1	1

Step I: As per step 2 of the methodology detailed in Section 5, the product numbers are arranged in descending order based on the flow distance ($pd[j]$). Since the problem is already in descending order, Table 2 can be used as it is for executing the next step.

Step II: Select the first product machine type sequence 1-3-2-6-5. Assign machine types 1, 3, 2, 6 and 5. Store in $b[j]$. Update the availability of machine types; the same are given below in Table 5.

Table 5. Unassigned machine types

Machine types	1	2	3	4	5	6	7
No of machines available	1	0	0	1	0	0	1

The linear machine type sequence ($b[]$) after the first product is 1-3-2-6-5.

Step III: The next product sequence is 4-6-1-7. Machine type number (mno) 4 is unassigned and unavailable (as per Step 6) in the existing sequence $b[]$, hence, add the machine types in front of $b[]$ and update the availability of the machine type. Machine type number 6 is assigned and available in the existing machine sequence $b[]$ and is shown by a bold letter in $b[] = 1-3-2-6-5$

Step IV: Machine types 1 and 7 are not available in the existing sequence $b[]$ after machine type 6 and duplicate machine types are available for use, i.e., unassigned. As per step 10, machine types 1 and 7 are unassigned and the insertion of machine types 1 and 7 affects the product sequence in $b[]$. Hence, as per step 11, add the machine types 1 and 7 at the end of sequence $b[]$ and update the machine types' availability. The machine type sequence is $b[] = 4-1-3-2-6-5-1-7$. The linear machine type sequence after the second product sequence is $b[] = 4-1-3-2-6-5-1-7$

Step V: The next product sequence is 4-1-6-5. Machine type 4 is assigned (shown as a bold letter in the above table) and available in the existing sequence $b[]$. Part of the remaining machine types 1, 6 and 5 are also available. This is shown in bold letters in sequence $b[] = 4-1-3-2-6-5-1-7$. The linear machine type sequence after the third product sequence is $b[] = 4-1-3-2-6-5-1-7$

Step VI: The next product sequence is 4-3-2-5. As per step 7, the machine type 4 is assigned (shown as a bold letter in the above table) and available in the existing sequence $b[]$. As per step 8, part of the remaining machine types 3, 2 and 5 are also available in $b[]$. This is shown in bold letters in sequence $b[] = 4-1-3-2-6-5-1-7$. The linear machine type sequence after the third product sequence is $b[] = 4-1-3-2-6-5-1-7$.

Step VII: The last product machine type sequence is 4-1-3-2. As per steps 7 and 8, the machine type number 4 and the remaining machine type numbers 1, 3 and 2 are all available in the existing sequence $b[]$. This is shown in bold letters in sequence $b[] = 4-1-3-2-6-5-1-7$. The linear machine type sequence after the last product sequence is $b[] = 4-1-3-2-6-5-1-7$.

Step VIII: The successive order of the first product machine type sequence 1-3-2-6-5 in $b[]$ is shown in bold letters 4-1-3-2-6-5-1-7.

Step IX: Similarly for the next product (4-6-1-7), the machine type sequences are not in a successive manner and shifting of the machine type would affect the first product machine type sequence. Hence, leave the assignment as it is.

Step X: Similarly for the next product (4-1-6-5), the machine type sequences are not in a successive manner and shifting of the machine type would affect the first product machine type sequence. Hence, leave the assignment as it is.

Step XI: The next two product sequences 4-3-2-5 and 4-1-3-2 are similar to the third product. Hence, leave the assignment of machine types too. The final linear machine type sequence is 4-1-3-2-6-5-1-7.

Step XII: Figure 2 represents the flow distance of individual products of the example problem.

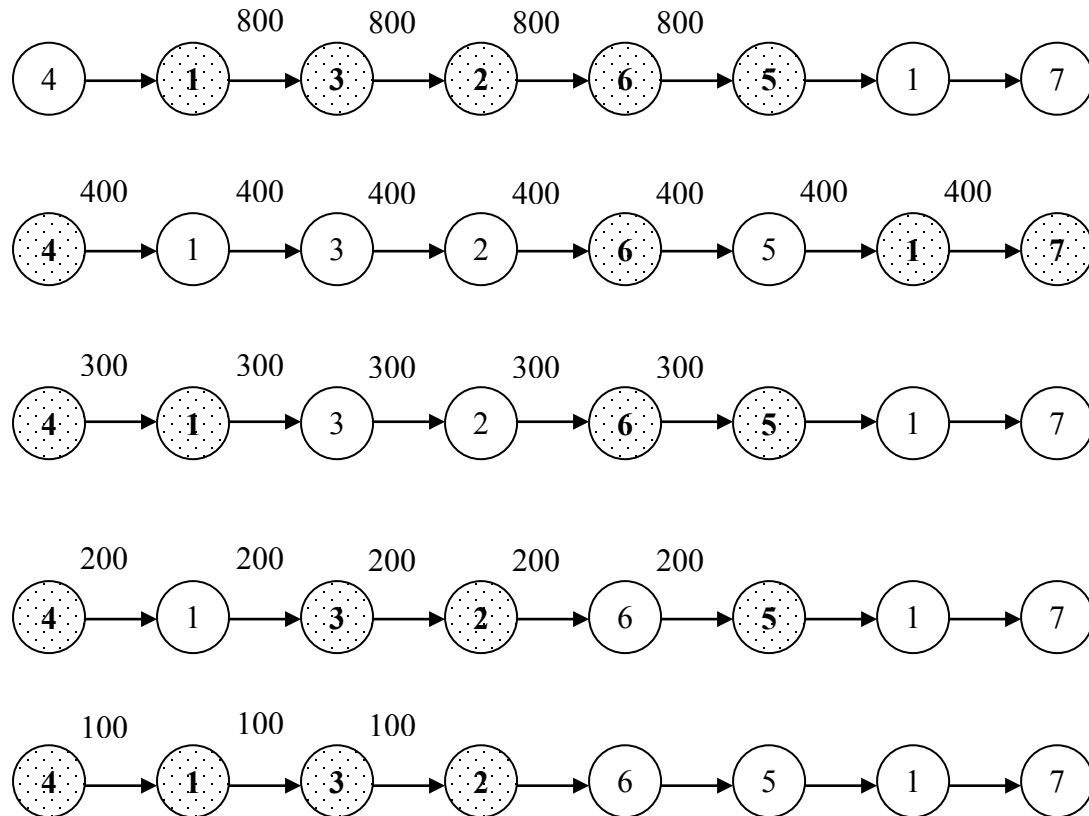


Figure 2. Flow distance of individual product for case 2

The total flow distance as per Equation (1) is equal to

$$td = 4 \times 800 + 7 \times 400 + 5 \times 300 + 5 \times 200 + 3 \times 100$$

$$td = 8800 \text{ Units}$$

The total flow distance and total number of machine types are 8800 units and 8 respectively. Table 6 represents the stepwise procedure to obtain the final sequence of linear machines for the problems discussed by Pillai et al (2005) and Chen et al (2001) using the proposed algorithm.

Table 6. Final sequence of example problems dealt by Chen et al (2001) and Pillai et al (2005)

P	A	B	C	D	E	F	G	S
1 Pillai et al + Chen et al	4	1-2-4-6-7					1-2-4-6-7	3
	1	3-2-4-5-6	3-2-4-5-6		1-2-4-6-7		3-2-4-5-6-1-2-4-6-7	6
	2	3-2-5-6-4		3-2-5-6-4	3-2-4-5-6-1-2-4-6-7		3-2-4-5-6-1-2-4-6-7	7,8,9
	3	3-2-1-4-6-8		3-2-1-4-6	3-2-4-5-6-1-2-4-6-7	8	3-2-4-5-6-1-2-4-6-7-8	7,8,11
2 Pillai et al + Chen et al	1	2-3-4-6-8-9-7					2-3-4-6-8-9-7	3
	3	2-4-6-8-9-13		2-4-6-8-9	2-3-4-6-8-9-7	13	2-3-4-6-8-9-7-13	7,8,11
	2	14-2-3-4-5-10-11-12	14	2-3-4	2-3-4-6-8-9-7-13	5-10-11-12	14-2-3-4-6-8-9-7-13-5-10-11-12	6,7,8,11
	4	1-2-3-5-11-12	1	2-3-5-11-12	14-2-3-4-6-8-9-7-13-5-10-11-12		1-14-2-3-4-6-8-9-7-13-5-10-11-12	6,7,8,9
3 Pillai et al	1	1-8-9-6-4					1-8-9-6-4	3
	2	5-3-2-7	5-3-2-7		1-8-9-6-4		5-3-2-7-1-8-9-6-4	6
	3	5-3-2-9		5-3-2-9	5-3-2-7-1-8-9-6-4		5-3-2-7-1-8-9-6-4	7,8,9
	4	3-7-6-4		3-7-6-4	5-3-2-7-1-8-9-6-4		5-3-2-7-1-8-9-6-4	7,8,9
	5	3-2-7-9-10			5-3-2-7-1-8-9-6-4	10	5-3-2-7-1-8-9-6-4-10	7,8,11
4 Pillai et al	1	1-3-2-6-5					1-3-2-6-5	3
	2	4-6-1-7	4-6-1-7		1-3-2-6-5		4-6-1-7-1-3-2-6-5	6
	3	4-1-6-5		4-1-6-5	4-6-1-7-1-3-2-6-5		4-6-1-7-1-3-2-6-5	7,8,9
	4	4-3-2-5		4-3-2-5	4-6-1-7-1-3-2-6-5		4-6-1-7-1-3-2-6-5	7,8,9
	5	4-1-3-2		4-1-3-2	4-6-1-7-1-3-2-6-5		4-6-1-7-1-3-2-6-5	7,8,9

P- Problem number; A- Product number (Pno[]); B- Product's machine type sequence (pseq[][]); C- Machine types to be added in front of existing sequence b[] when unassigned; D- Machine type available in existing sequence b[] when the machine type is assigned; E- Existing sequence b[]; F- Unassigned machine types to be added after existing sequence b[] when the previous machine type assigned and available in the existing sequence b[]; G- Linear Sequence; S- As per algorithm step number

7. Computation results and discussions

The heuristic proposed in this work has been implemented in solving 13 problems out of which the first six problems were taken from Chen et al. (2001) and Pillai et al. (2005). The input data such as the number of machine type, its duplicate number, the number of products and the machine type sequences are listed in Appendix Table A1. The final linear machine sequence, the flow distance and the total number of machine types available in the final linear sequence are presented in Table 7.

Table 7 Final linear sequence and its total flow distance

Problem No.	No of machine types	No of product	Method	Total distance in Units	Total No of machines	Final sequence
1	8	4	Proposed	2600	11	3-2-4-5-6-1-2-4-6-7-8
			Pillai et al + Chen et al	2600	11	3-2-4-5-6-1-2-4-6-7-8
2	14	4	Proposed	475	14	1-14-2-3-4-6-8-9-7-13-5-10-11-12
			Pillai et al + Chen et al	475	14	14-1-2-3-4-6-8-9-7-13-5-10-11-12
3	10	5	Proposed	12800	10	5-3-2-7-1-8-9-6-4-10
			Pillai et al	12800 #(13200)	10	5-3-2-7-1-8-9-6-4-10
4	7	5	Proposed	8800	8	4-1-3-2-6-5-1-7
			Pillai et al	9000 #(7800)	9	4-6-1-7-1-3-2-6-5
5	8	4	Proposed	1243	12	8-2-7-4-6-1-5-3-2-4-7-5
			Chen et al	1890	12	5-8-2-7-4-6-3-7-2-4-5-1
6	15	4	Proposed	890	12	2-10-12-14-13-7-11-15-5-3-1-4
			Chen et al	989	12	2-10-12-14-13-7-11-15-5-1-4-3
7	9	5	Proposed	624	14	4-9-7-6-8-5-1-6-2-7-3-1-5-8
			Chen et al	903	15	9-7-6-8-7-5-4-9-1-6-2-3-1-5-8
8	14	6	Proposed	1765	17	14-4-3-12-2-8-5-6-11-3-7-1-9-12-13-2-10
			Chen et al	3312	17	14-4-3-14-12-2-8-5-11-7-1-9-12-13-6-2-10

9	16	8	proposed	5367	22	2-5-14-12-15-5-4-1-9-6-13-9-11-15-8-1-3-12-10-2-14-10
			Chen et al	7459	21	2-4-14-12-15-5-6-13-9-1-9-6-11-8-15-12-10-2-8-1-3
10	12	6	proposed	2142	20	11-3-5-6-12-8-9-2-4-9-7-1-11-7-12-1-4-3-6-10
			Chen et al	3323	18	12-8-9-2-4-9-11-3-5-11-6-7-12-1-4-10-3-6
11	14	6	proposed	2388	14	2-4-8-5-3-11-13-14-7-12-9-1-10-6
			Chen et al	2939	14	4-2-8-5-3-11-13-14-1-10-7-12-9-6
12	15	5	proposed	1232	15	3-6-15-12-9-7-5-3-14-15-1-8-4-7-8
			Chen et al	1503	17	3-14-3-6-15-1-8-9-7-5-4-7-6-15-12-14-8
13	11	5	Proposed	2192	13	1-11-7-8-3-10-6-4-7-9-1-5-2
			Chen et al	2377	13	3-1-11-7-8-10-9-6-4-7-1-5-9

In problem numbers 4, 9 and 12, as compared with the existing method, the proposed method yields the best linear sequencing of machines in which both flow distance and total number of machines in the sequence are reduced to a great extent. In problem numbers 9 and 10, it is noted that the proposed method can produce a linear sequence of machines with less flow distance and greater number of total machines as compared with the existing method. All other problems are better in flow distance with the same number of total machine types in the final linear machine sequencing. This can be easily understood from Figure 3.

The proposed method yields a minimum flow distance due to the following reasons:

- The machine types are assigned as per the maximum flow distances of the product machine type sequences.
- The unassigned machine types are added in front of the existing machine sequence.
- If one of the machine types is assigned and it is available in the existing sequence, even though the remaining machine types are unassigned, its availability is checked in the existing sequence. If any of the remaining machine types are not available in the existing sequence and are unassigned, then the machine type is inserted in the existing sequence without affecting the previous product machine types' sequences.
- The successiveness of the sequence of individual product machine types is checked in the final linear sequence. If there is a chance to shift the machine

type close to its previous machine type without affecting the successiveness of the other product machine type sequences, the shifting is carried out.

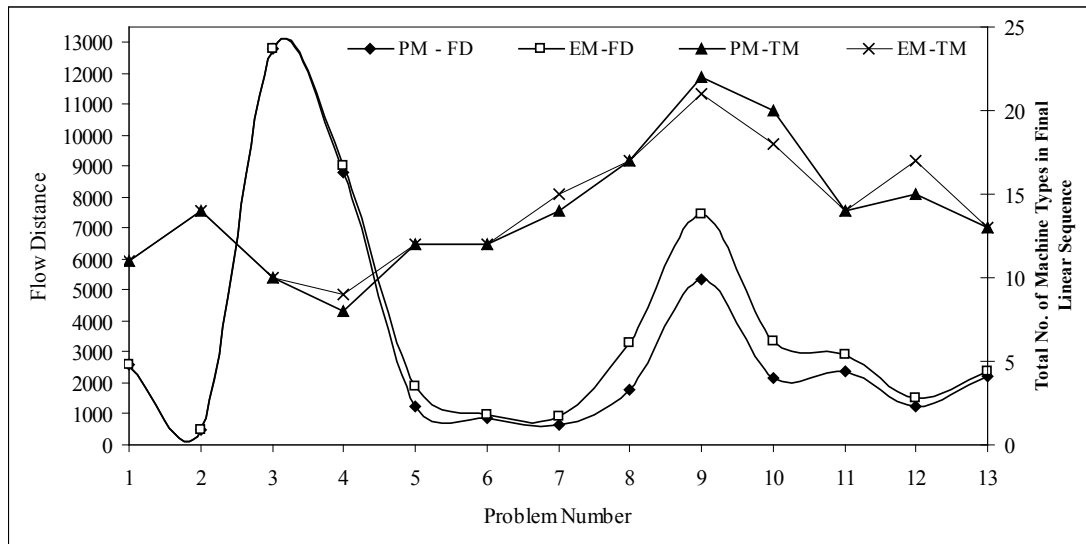


Figure 3. Comparison of flow distance between the proposed and the existing method

PM-FD Total flow distance by the proposed method; EM-FD Total flow distance by the existing method; PM-TM Total machine types in the final linear sequence by the proposed method; EM-TM Total machine types in the final linear sequence by the existing method;

Table 8 represents the computation time to determine both linear machine sequence and flow distance for the proposed and the existing method. It demonstrates that the proposed method consumes a great deal less computation time even after increasing the number of products and machine types. The existing method, however, requires nearly 50 times more than the time required for the proposed method. It is clearly shown in Figure 4 that the proposed heuristic requires very much less time as compared with the existing method.

Table 8. Comparison of computation time between proposed and existing methods

P.No.	NoP	NoMT	TMT	CTPM	CTEM
1	4	8	14	0.13	3.42
2	4	12	14	0.16	4.35
3	5	10	10	0.15	4.35
4	5	7	9	0.14	3.92
5	4	8	13	0.13	3.42
6	4	14	22	0.16	4.42
7	5	9	16	0.15	4.25
8	6	14	22	0.17	6.21
9	8	15	26	0.18	8.22
10	6	12	21	0.17	5.52
11	6	14	14	0.17	6.18
12	5	15	23	0.17	5.69
13	5	11	19	0.17	4.42

P.No.- Problem number; NoP- Number of products; NoMT- Number of machine types; TMT – Total number of machine types including duplicate machine types; CTPM – Computation time for the proposed method in secs; CTEM-Computation time for the existing method in secs.

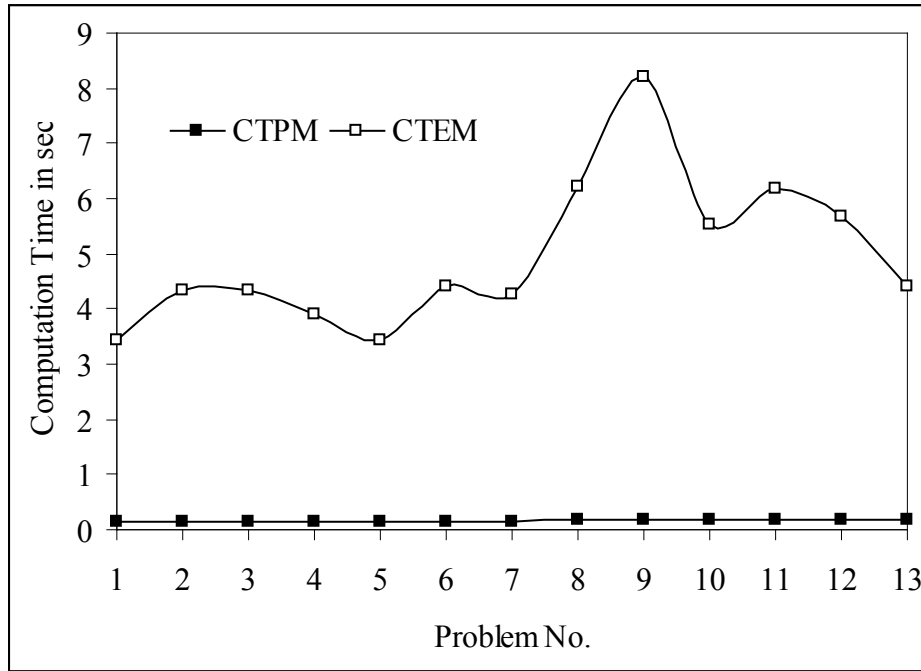


Figure 4. Comparison of computation time between proposed and existing method

8. Conclusion

In this work a simple heuristic algorithm to construct a linear sequence of machines is proposed which requires very little time compared to the existing methods. The results with regards to computation time, flow distance and total number of machine types in the final linear machine sequence demonstrate the efficiency of the proposed method. In addition to the problems discussed in the literature, a number of additional problems were generated and experimented with the proposed heuristic algorithms which are not included in this paper due to space constraint. As an extension of the present work, it is possible to consider the cost of the machine type and the total number of machines available in the final linear sequence as objective functions, which may be treated as a multi-objective problem. Optimization tools like Particle Swarm Optimization and Tabu Search can be used to solve these problems.

Appendix

Table A1. Operation sequences and flow distance of example problems

Problem no	Products	Operation sequence	Distance
1 Pillai et al + Chen et al	1	3-2-4-5-6	150
	2	3-2-5-6-4	100
	3	3-2-1-4-6-8	50
	4	1-2- 4-6-7	200
2 Pillai et al + Chen et al	1	2-3-4-6-8-9-7	20
	2	14-2-3-4-5-10-11-12	10
	3	2-4-6-8-9-13	15
	4	1-2-3-5-11-12	10
3 Pillai et al	1	1-8-9-6-4	700
	2	5-3-2-7	600
	3	5-3-2-9	500
	4	3-7-6-4	400
	5	3-2-7-9-10	300
4 Pillai et al	1	1-3-2-6-5	800
	2	4-6-1-7	400
	3	4-1-6-5	300
	4	4-3-2-5	200
	5	4-1-3-2	100
5	1	8-2-6-3	4
	2	4-6-3-7-5	80
	3	7-4-6-1	77
	4	5-3-2-4	86
6	1	14-13-7-15	34
	2	2-10-12-13	29
	3	11-15-5-3	94
	4	15-5-1-4	89
7	1	7-3-1-5-8	45
	2	6-5-1-3	8
	3	4-9-1-2	12
	4	5-1-6-2-3	41
	5	9-7-6-8	29
8	1	14-4-3-5	22
	2	14-5-6-2-10	2
	3	3-7-1-9	145
	4	3-12-5-7	11
	5	11-7-12-13-2	90
	6	12-2-8-5-6	112
9	1	14-12-15-5	127
	2	14-12-6-11	19
	3	5-15-12-10-2	111
	4	4-8-1-3	125
	5	13-9-11-15	142
	6	2-4-14-10	27
	7	4-9-11-10	18

	8	4-1-9-6-11	135
10	1	11-3-5-6-1-10	14
	2	12-8-9-4-7	84
	3	3-5-11-7-4	4
	4	2-4-9-7-1	121
	5	9-4-12-3-6	19
	6	11-7-12-1-4	150
11	1	4-5-3-9	69
	2	5-3-7-6	13
	3	13-7-12-9	113
	4	8-5-3-14	72
	5	11-13-14-7	131
	6	2-5-1-10	36
12	1	5-4-7-8	19
	2	9-7-5-14	47
	3	3-6-15-8	2
	4	6-15-12-14-8	29
	5	3-14-15-1-8	145
13	1	1 11-6-7-9	8
	2	11-8-9-5	4
	3	3-10-6-4-7	128
	4	7-8-10-9	94
	5	11-7-10-1-5	91

Table A2. Machine types and its duplicates for the example problems

Problem No.	Machines Types															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	2	2	1	2	3	2	1	1								
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
3	1	1	1	1	1	1	1	1	1	1						
4	2	1	1	1	1	2	1									
5	2	2	1	2	2	1	2	1								
6	1	1	1	2	1	2	2	2	2	2	1	2	1	1	1	
7	2	2	1	1	2	2	2	2	2							
8	1	2	2	1	1	2	1	2	2	2	1	2	1	2		
9	2	2	1	1	2	2	2	1	2	2	1	2	1	2	2	1
10	2	1	2	2	2	2	2	1	2	1	2	2				
11	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
12	1	1	2	1	1	2	2	2	2	1	2	1	2	1	2	
13	2	2	2	1	2	1	2	2	2	1	2					

Table A3. Final linear sequences for the example problems

P	A	B	C	D	E	F	G	S
5	4	5-3-2-4					5-3-2-4	3
	2	4-6-3-7-5	4-6	3	5-3-2-4	7-5	4-6-5-3-2-4-7-5	6,7,8,11
	3	7-4-6-1	7	4-6	4-6-5-3-2-4-7-5	1	7-4-6-5-3-2-4-7-5-1	6,7,8,11
	1	8-2-6-3	8-2	6-3	7-4-6-5-3-2-4-7-5-1		8-2-7-4-6-5-3-2-4-7-5-1	6,7,8
As per the implementation of step 14 final sequence=8-2-7-4-6-1-5-3-2-4-7-5								
6	3	11-15-5-3					11-15-5-3	3
	4	15-5-1-4		15-5	11-15-5-3	1-4	11-15-5-3-1-4	7,8,11
	1	14-13-7-15	14-13-7	15	11-15-5-3-1-4		14-13-7-11-15-5-3-1-4	6,7,8,9
	2	2-10-12-13	2-10-12	13	14-13-7-11-15-5-3-1-4		2-10-12-14-13-7-11-15-5-3-1-4	6,7,8,9
7	1	7-3-1-5-8					7-3-1-5-8	3
	4	5-1-6-2-3	5-1-6-2	3	7-3-1-5-8		5-1-6-2-7-3-1-5-8	6,7,8,9
	5	9-7-6-8	9-7-6-8		5-1-6-2-7-3-1-5-8		9-7-6-8-5-1-6-2-7-3-1-5-8	6
	3	4-9-1-2	4	9-1-2	9-7-6-8-5-1-6-2-7-3-1-5-8		4-9-7-6-8-5-1-6-2-7-3-1-5-8	6,7,8,9
	2	6-5-1-3		6-5-1-3	4-9-7-6-8-5-1-6-2-7-3-1-5-8		4-9-7-6-8-5-1-6-2-7-3-1-5-8	7,8,9
8	3	3-7-1-9					3-7-1-9	3
	6	12-2-8-5-6	12-2-8-5-6		3-7-1-9		12-2-8-5-6-3-7-1-9	6
	5	11-7-12-13-2	11	7	12-2-8-5-6-3-7-1-9	12-13-2	11-12-2-8-5-6-3-7-1-9-12-13-2	6,7,8,11
	1	14-4-3-5	14-4-3	5	11-12-2-8-5-6-3-7-1-9-12-13-2		14-4-3-11-12-2-8-5-6-3-7-1-9-12-13-2	6,7,8,9
	4	3-12-5-7		3-12-5-7	14-4-3-11-12-2-8-5-6-3-7-1-9-12-13-2		14-4-3-11-12-2-8-5-6-3-7-1-9-12-13-2	7,8,9
	2	14-5-6-2-10		14-5-6-2	14-4-3-11-12-2-8-5-6-3-7-1-9-12-13-2	10	14-4-3-11-12-2-8-5-6-3-7-1-9-12-13-2-10	7,8,11
As per the implementation of step 14 final sequence=14-4-3-12-2-8-5-6-11-3-7-1-9-12-13-2-10								

P	A	B	C	D	E	F	G	S
9	5	13-9-11-15					13-9-11-15	3
	8	4-1-9-6-11	4-1-9-6	11	13-9-11-15		4-1-9-6-13-9-11-15	6,7,8,10
	1	14-12-15-5	14-12-15-5		4-1-9-6-13-9-11-15		14-12-15-5-4-1-9-6-13-9-11-15	6
	4	4-8-1-3		4	14-12-15-5-4-1-9-6-13-9-11-15	8-1-3	14-12-15-5-4-1-9-6-13-9-11-15-8-1-3	7,8,11
	3	5-15-12-10-2	5	15	14-12-15-5-4-1-9-6-13-9-11-15-8-1-3	12-10-2	5-14-12-15-5-4-1-9-6-13-9-11-15-8-1-3-12-10-2	6,7,8,11
	6	2-4-14-10	2	4	5-14-12-15-5-4-1-9-6-13-9-11-15-8-1-3-12-10-2	14-10	2-5-14-12-15-5-4-1-9-6-1-3-9-11-15-8-1-3-12-10-2-14-10	6,7,8,11
	2	14-12-6-11		14-12-6-11	2-5-14-12-15-5-4-1-9-6-13-9-11-15-8-1-3-12-10-2-14-10		2-5-14-12-15-5-4-1-9-6-13-9-11-15-8-1-3-12-10-2-14-10	7,8,9
	7	4-9-11-10		4-9-11-10	2-5-14-12-15-5-4-1-9-6-13-9-11-15-8-1-3-12-10-2-14-10		2-5-14-12-15-5-4-1-9-6-13-9-11-15-8-1-3-12-10-2-14-10	7,8,9
10	6	11-7-12-1-4					11-7-12-1-4	3
	4	2-4-9-7-1	2-4-9-7-1		11 7 12 1 4		2-4-9-7-1-11-7-12-1-4	6
	2	12-8-9-4-7	12-8-9	4-7	2-4-9-7-1-11-7-12-1-4		12-8-9-2-4-9-7-1-11-7-12-1-4	6,7,8,10
	5	9-4-12-3-6		9-4-12	12-8-9-2-4-9-7-1-11-7-12-1-4	3-6	12-8-9-2-4-9-7-1-11-7-12-1-4-3-6	7,8,11
	1	11-3-5-6-1-10	11-3-5-6	1	12-8-9-2-4-9-7-1-11-7-12-1-4-3-6	10	11-3-5-6-12-8-9-2-4-9-7-1-11-7-12-1-4-3-6-10	6,7,8,11
	3	3-5-11-7-4		3-5-11-7-4	11-3-5-6-12-8-9-2-4-9-7-1-11-7-12-1-4-3-6-10		11 3 5 6 12 8 9 2 4 9 7 1 11 7 12 1 4 3 6 10	7,8,9

P	A	B	C	D	E	F	G	S
11	5	11-13-14-7					11 13 14 7	3
	3	13-7-12-9		13-7	11-13-14 7	12-9	11-13-14-7-12-9	7,8,9
	4	8-5-3-14	8-5-3	14	11-13- 14-7-12- 9		8-5-3-11-13-14- 7-12-9	6,7,8,9
	1	4-5-3-9	4	5-3-9	8-5-3-11- 13-14-7- 12-9		4-8-5-3-11-13- 14-7-12-9	6,7,8,9
	6	2-5-1-10	2	5	4-8-5-3- 11-13- 14-7-12- 9	1-10	2 4 8 5 3 11 13 14 7 12 9 1 10	6,7,8,11
	2	5-3-7-6		5-3-7	2-4-8-5-3 11-13-14 7-12-9-1 10	6	2-4-8-5-3-11-13- 14-7-12-9-1-10-6	7,8,11
12	5	3- 14- 15- 1- 8					3- 14- 15- 1- 8	3
	2	9 7 5 14	9-7-5	14	3- 14- 15- 1- 8		9-7-5-3-14-15-1- 8	6,7,8,9
	4	6 15 12 14 8	6-15- 12	14-8	9-7-5-3- 14-15-1- 8		6-15-12-9-7-5-3- 14-15-1-8	6,7,8,9
	1	5 4 7 8		5	6-15-12- 9-7-5-3- 14-15-1- 8	4-7- 8	6-15-12-9-7-5-3- 15-1-8-4-7-8	7,8,11
	3	3 6 15 8	3		6-15-12- 9-7-5-3- 15-1-8-4- 7-8		3-6-15-12-9-7-5- 3-14-15-1-8-4-7- 8	6
13	3	3-10-6-4-7					3-10-6-4-7	3
	4	7-8-10-9	7-8	10	3-10-6-4- 7	9	7-8-3-10-6-4-7-9	6,7,8,11
	5	11-7-10-1-5	11	7-10	7-8-3-10- 6-4-7-9	1-5	11-7-8-3-10-6-4- 7-9-1-5	6,7,8,11
	1	1-11-6-7-9	1	11-6- 7-9	11-7-8-3- 10-6-4-7- 9-1-5		1-11-7-8-3-10-6- 4-7-9-1-5	6,7,8,9

P- Problem number; A- Product number (Pno[]); B- Product's machine type sequence (pseq[][]); C- Machine types to be added in front of existing sequence b[] when unassigned; D- Machine type available in existing sequence b[] when the machine type is assigned; E- Existing sequence b []; F- Unassigned machine types to be added after existing sequence b[] when the previous machine type assigned and available in the existing sequence b[]; G- Linear Sequence; S- As per algorithm step number

REFERENCES

1. Adel El-Baz, M., 2004. A genetic algorithm for facility layout problems of different manufacturing environments. *Computers and Industrial Engineering*, 47(2-3), 233 – 246.
2. Aiello, G., Enea, M. and Galante, G., 2002. An integrated approach to the facilities and material handling system design. *International Journal of Production Research*, 40 (15), 4007-4017.
3. Al-Hakim, L., 2000. On solving facility layout problems using genetic algorithms. *International Journal of Production Research*, 38 (11), 2573-2582.
4. André, R.S. and Amaral., 2008. An exact approach to the one-dimensional facility layout problem. *Operations Research*, 56, 1026-1033.
5. Aneke, N.A. and Carrie, A.S., 1986. A design technique for layout of multi-product flowlines. *International Journal of Production Research*, 24, 471- 481.
6. Anjosa, M.F., Kenningsb, A. and Vannellib, A., 2005. A semidefinite optimization approach for the single-row layout problem with unequal dimensions. *Discrete Optimization*, 2, 113-122.
7. Azadivar, F. and Wang, J., 2000. Facility layout optimization using simulation and genetic algorithms. *International Journal of Production Research*, 38(17), 4369-4383.
8. Balamurugan, K., Selladurai, V. and Ilamathi, B., 2008. Manufacturing facilities layout design using Genetic Algorithm, *International Journal of Manufacturing Technology and Management*, 14(3-4), 461-474.
9. Braglia, M., 1997. Heuristics for single-row layout problems in flexible manufacturing problems. *Production Planning and Control*, 8, 558–567.
10. Braglia, M., Zanoni, S. and Zavanella, L., 2003. Layout design in dynamic environments: strategies and quantitative indices. *International Journal of Production Research*, 41(5), 995-1016.
11. Chae, J. and Peters, B.A., 2006. A simulated annealing algorithm based on a closed loop layout for facility layout design in flexible manufacturing systems. *International Journal of Production Research*, 44(13), 2561-2572.
12. Chen, D.S., Wang, Q. and Chen, H.C., 2001. Linear sequencing for machine layouts by a modified simulated annealing. *International Journal of Production Research*, 39(8), 1721 - 1732.
13. Chrysostomos, F. and Vlachos, A., 2005. Optimal solution of linear machine layout problem using ant colony system. *Proceedings of the 9th WSEAS*

International conference on Computers, Athens, Greece: World Scientific and Engineering Academy and Society (WSEAS), 97.

14. Diponegoro, A. and Sarker, B.R., 2003. Flow distance reduction for a multi-product flowline with sets of identical machines. *European Journal of Operational Research*, 147, 591-612.
15. Dunker, T., Radonsb, G. and Westkämpera, E., 2003. A coevolutionary algorithm for a facility layout problem. *International Journal of Production Research*, 41 (15), 3479-3500.
16. Ficko, M., Brezocnik, M. and 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.
17. Heragu, S., 1997. *Facilities Design*. USA: PWS Publishing Co.
18. Heragu, S.S. and Alfa, A.S., 1992. Experimental analysis of simulated annealing based algorithms for the layout problem. *European Journal of Operational Research*, 57, 190-202.
19. Heragu, S.S. and Kusiak, A., 1988. Machine layout problem in flexible manufacturing systems. *Operations Research*, 36, 258-268.
20. Heragu, S.S. and Kusiak, A., 1989. Machine layout: An optimization and knowledge-based approach. *International Journal of Production Research*, 28, 615- 635.
21. Hicks, C., 2004. A genetic algorithm tool for designing manufacturing facilities in the capital goods industry. *International Journal of Production Economics*, 90(2), 199-211.
22. Ho, Y.C. and 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.
23. Ho, Y.C., Lee, C.E.C. and Moodie, C.L., 1993. Two sequence-pattern, matching-based, flow analysis methods for multi-flowlines layout design. *International Journal of Production Research*, 31, 1557–1578.
24. Houshyar, A. and 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*, 28, 1485 – 1498.
25. Kim, J.G. and Kim, Y.D., 2000. Layout planning for facilities with fixed shapes and input and out points. *International Journal of Production Research*, 38(18), 4635–4653.

26. Kouvelis, P. and Chiang, W.C., 1992. A simulated annealing procedure for single row layout problems in flexible manufacturing systems. *International Journal of Production Research*, 30, 717–732.
27. Kouvelis, P. and Kim, M.W., 1992. Unidirectional loop network layout problem in automated manufacturing systems. *Operation Research*, 40, 533–550.
28. Lee, S.D., Huang, K.H. and Chiang, C.P., 2001. Configuring layout in unidirectional loop manufacturing systems. *International Journal of Production Research*, 39(6), 1183–1201.
29. Lin, M.T., 2009. The single-row machine layout problem in apparel manufacturing by hierarchical order-based genetic algorithm. *International Journal of Clothing Science and Technology*, 21(1), 31-43.
30. Nearchou, A.C., 2006. Meta-heuristics from nature for the loop layout design problem. *International Journal of Production Economics*, 101, 312–328.
31. Nicholas, J., 2001. *Competitive Manufacturing Management: Continuous Improvement, Lean Production and Customer Focused Quality*. Tata McGraw Hill, New Delhi
32. Pillai, V.M. and Gudivada, B.S., 2005. A simulated annealing algorithm for linear sequencing of machines for layout design. *Sixth Int. Conference on Operations and Quantitative Management*, August 9-11, 2005, Indore, India.
33. Satheesh Kumar, M., Asokan, P. and Kumanan, S., 2010. An artificial immune system-based algorithm to solve linear and loop layout problems in flexible manufacturing systems. *International Journal of Product Development*, 10(1-3), 165-179.
34. Shayan, E. and Chittilappilly, A., 2004. Genetic algorithm for facilities layout problems based on slicing tree structure. *International Journal of Production Research*, 42(19), 4055-4067.
35. Solimanpur, M., Prem Vrat, Ravi Shankar, 2005. An ant algorithm for the single row layout problem in flexible manufacturing systems. *Computers & Operations Research*, 32(3), 583–598.
36. Teo, Y.T. and Ponnambalam, S.G., 2008. A hybrid ACO/PSO heuristic to solve single row layout problem. *CASE 2008. IEEE International Conference on Automation Science and Engineering 23-26 Aug. 2008, Washington, DC, USA*. 597 – 602.
37. Wang, T.Y., Lin, H.C. and Wu, K.B., 1998. An improved simulated annealing for facility layout problems in cellular manufacturing systems. *Computers & Industrial Engineering*, 34(2), 309–319.

38. Zhou, J. and Irani, S.A., 2003. A new flow diagramming scheme for mapping and analysis of multi product flows in a facility. *Journal of Integrated Design and Process Science*, 7, 25-58.