Scientia Iranica B (2022) 29(5), 2290–2303

SCIENTIA IRANICA

Sharif University of Technology

Scientia Iranica

Transactions B: Mechanical Engineering

http://scientiairanica.sharif.edu

Design and optimization of layout problem using meta-heuristic algorithms

S. Suresh Balaji[a,*], S. Ragunathan[b], K. Visagavel[a], and S. Ganesh Kumar[c]

a. *Department of Mechanical Engineering, Knowledge Institute of Technology, Salem, Tamilnadu, India.*
b. *Department of Mechanical Engineering, SRG Engineering College, Namakkal, Tamilnadu, India.*
c. *Department of Mechanical Engineering, Er. Perumal Manimekalai College of Engineering, Hosur, Tamilnadu, India.*

Received 30 March 2021; received in revised form 4 December 2021; accepted 8 June 2022

KEYWORDS
Layouts;
Meta-heuristics;
Artificial immune system;
Tabu search;
Improved tabu search.

Abstract. Loop layout is a common layout in Flexible Manufacturing Systems (FMSs), in which machines are arranged around the loop and materials are transported in a unidirectional route only. The objective of the Loop Layout Problem (LLP) is to regulate machines around a loop to minimize the maximum congestion within a set of parts. Artificial Immune System (AIS), Tabu Search (TS), and Improved Tabu Search (ITS) algorithms are employed to solve these loop layout problems. The algorithms are tested and validated through large- and small-sized randomly generated hypothetical problems with a minimum required machine sequence. The efficiency of algorithms is compared with that of existing algorithms for benchmark problems. Computational results reveal that ITS algorithm outperforms AIS, TS, and existing algorithm for large-sized hypothetical problems.

1. Introduction

One of the principal concerns regarding the layout of a loop flexible manufacturing process is distributing various resources for producing higher efficiency. Machines in the present FMS are a vital resource; thus, it is crucial to ensure that a machine does not need to remain idle based on the design of a material processing system. Hence, a vital factor in the functioning of production systems is to determine a layout so that material movement between the machines can be efficient the most. Different types of machine layout include line layout, open field layout, loop layout, ladder layout, and robot centered cell.

An overview of the loop layout problems has been made by many researchers who have already attempted to solve LLPs. Common approaches are linear programming as well as heuristic and meta-heuristic methods. Afentakis [1] proposed solutions for the LLPs of a loop network and developed a loop layout with a graph where the nodes resemble the processing elements and the links of the Flexible Manufacturing System (FMS) are resembled by edges. Efficient material flow, operational simplicity, and flexibility are some features of the loop network. Cheng and Gen [2] suggested using hybrid genetic algorithm technique and neighborhood search methodology to arrive at the min_sum and min_max LLP. Bennell et al. [3] described TS methodology along with a randomized insertion technique to provide solutions for min_max LLP.

Yang et al. [4] detailed a two-step heuristic approach to FMS with one-loop directional flow patterns. The problem was in the form of integer linear programming, which might be defined optimally for similar instant smaller sizes. Many researchers have adopted a mixed model for providing a solution for a new field layout with closed-loop configuration problems. Kumar et al. [5] suggested a dual phase, one of which involves AIS technique to determine a solution for

*. *Corresponding author.*
E-mail address: balajissindia@gmail.com (S. Suresh Balaji)

doi: 10.24200/sci.2022.58011.5519

the unidirectional LLP by considering min_sum and min_max objectives while shortcuts are formed in the path of flow for the materials in the layout to improve the overall system performance.

Kumar et al. [6] proposed a solution for determining the order of the machines over a loop network to reduce the count of loop traverse of machinable parts. Nearchou [7] determined DEA to provide a solution to the LLP and a methodology for decoding the chromosomes for problems characterized by permutation property. A simulated annealing technique was proposed by Chae and Peters to determine a solution to reduce the cost of material handling [8].

AsefVaziri et al. [9] suggested ant colony method for improving loop-dependent network design with a shortcut. Hong et al. [10] performed modified ranking and achieved local improvement using genetic algorithm. Lasrado and Nazzel [11] implemented queuing theory and genetic algorithm to form a closed-loop configuration. Glover and Laguna [12] developed tabu search which was applied to lead optimization algorithms in the search for a globally optimal solution. Uneven area configuration layout cases where the motive is to determine a flexible bay configuration with desirable solutions [13]. Potts and Whitehead [14] proposed a combination of scheduling layout and machine layout for a manufacturing facility and presented solutions with the help of an integer programming model. Yang et al. [15] and Yang Peters [16] applied fabrication facilities to find solutions for forming a layout material processing facility design problem which remained dependent on a spine layout systems.

Suhardini et al. [17] developed systematic layout planning to design the industry layout in order to increase production capacity. Kang et al. [18] developed a method for determining the optimal location of production cells in a material handling system with a central loop. The loop layout problem received very little attention in the extensive literature review [19].

Alduaji and Hassin [20] devised a linear programming methodology for a circular open field layout in designing FMSs with the goal of reducing production costs to absolute minimum. Hungerlander et al. [21] described a directed circular facility layout problem in which the total weighted sum of all pairs of machine center-to-center distances measured was minimized. The best arrangement of a series of cutting tools on a tool turret is one of the most common real-world implementations of the loop layout problem.

One of the most important aspects to consider for a successful and cost-effective operation of an automated guided vehicle system is facility layout design technique [22]. Rai and Jayswal [23] developed a particle swarm optimization algorithm for solving the loop layout problem. The monarch butterfly optimization approach was presented by the layout design, and the quickest single-loop material handling path was solved by Kim and Chae [24]. Wenhan et al. [25] described a hybrid population-based incremental learning technique for solving the closed-loop layout problem by simultaneously searching for the best facility placement order and the size of the rectangle loop. A multi-stage stochastic programming approach was utilized by Mohammadia et al. [26] for a sustainable closed-loop supply chain configuration design with financial decisions.

The major purpose of the circular layout challenge is to reduce the overall transportation cost of material flow between facilities. To this end, the appropriate facility placement sequence as well as the rectangle loop's optimal size must be determined at the same time. Although many metaheuristic-based solutions for solving the circular loop layout problem have been proposed, those approaches simply use metaheuristics to find the best facility placement sequence, with the enumeration method determining the ideal size of the rectangle loop [27].

Saravanan and Ganesh Kumar [28] reviewed the loop layout problem and described the min_max technique to reduce substantially the congestion in parts of the sub family given that their attempted minimization was the same as the amount of congestion reduction for other parts. In doing so, the min_max approach has been improved; it was found that complicated problems have been adopted using recent optimization methods with optimal solutions for LLPs and the results have been evaluated based on the earlier results of the algorithms. There is no benchmarking for the loop layout problem with distance for huge machines in the literature. However, further research is still required to look into large-scale issues. Underutilized metaheuristic algorithms are required to test this type of the loop layout problem.

In this paper, loop layout problems have been used by considering large-sized problems with unit distance to minimize the maximum congestion in the family of parts. Large-sized problems with unit distance have been assumed and incorporated to minimize the maximum congestion among the family of parts. In order to overcome min_max congestion issues, an improved TS strategy has been developed. It has been enhanced to solve benchmarking problems compared to TS, AIS and Sheep Flock Heredity Algorithm (SFHA). Classic TS paradigm has been reconstructed with the solutions of the above process served as ITS method by solving issues.

2. Problem description

In this paper, a loop layout is considered for a spine layout given that a spine design is relative to the loop arrangement system for an FMS. The dimensions and clearance among the machines are considered equal with the unit distance. An FMS system is a loop layout in which machines must be placed in a loop structure and the materials be transferred in a cycle form, as shown in Figure 1.

In a unidirectional loop design, only a loading/unloading location exists where parts leave and enter the manufacturing system. The number of machines includes n and $M = 0, 1, 2, \cdots n$, where 0 denotes the loading/unloading station. The loop arrangement model is in the form of transformation of machines (m_1, m_2, m_3, m_n) with a prefix of loading/unloading station 0. Every part is characterized by its machining operation to be performed in a sequence. Now, machining is performed on machine j, followed by machine i. When the location of machine j is under machine i, then the part must move over the area, which is called a reload. The actual number of reloads required to finish the machining is termed as traffic congestion. Our objective is to provide the best optimal layout sequence through optimization by considering a set of machining operation constraints in hypothetical test problems given in Table 1.

A min_sum case involves an attempt to minimize the congestion of all parts, while a min_max case ensures minimizing the highest congestion existing among parts of the family [29]. Min_max and algorithm performance are ascertained as per the following formulas:

1. The function for the min_max LLP is given below:

$$Cost_{\min\_\max} = \max(reload_1, reload_2, reload_M). \quad (1)$$

   Min_max aim is to account for high variability up and down in the mix and quantity of demand. Over time, uniform loading of machines should be observed so that the number of reloads in total shafts is high for this layout arrangement;

2. Each resolution determined by the algorithm cannot be a desirable solution; however, few solutions are close enough to optimality. The efficiency of an algorithm in producing near-optimal solution is calculated through the average percentage solution effort (SE%) as expressed through Eq. (2):

$$SE(\%) = \frac{ne_{opt}}{ne_{total}} \times 100. \quad (2)$$

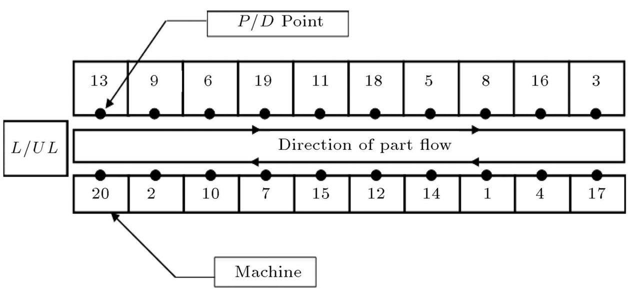

Figure 1. Arrangement of loop arrangement.

   ne_{opt} denotes the number of estimations performedthrough the proposed algorithm to obtain an optimal solution (very close to the best solution), ne_{total} denotes the total number of estimations performed through the algorithm for all solutions.

For the sake of clarity, illustration regarding the first problem is presented here. The required machine progress for each part is given in Table 1.

The layout considered is 13-9-6-19-11-18-5-8-16-3-17-4-1-14-12-15-7-10-2-20, which means the location of the machines in the loop including machine 13 in the first position, followed by machines 9, 6, etc. Only one loading/unloading is taken as the beginning point of the spine loop. Commonly, by evaluating the development of the demanded part given above, part 1 must visit 6, 3 (upper row), and 4 (lower row), and need to cross the loading/unloading (one reload) after the mentioned part. It continues to visit 18, 5, 1, 14, 7 and needs one reload; next, part 1 visits machine 11 and requires one reload. After that, part 1 visits machines 13, 9, 16, 17, and 2. Further, one reload is needed to visit machines 8, and 15 so that the process can be complete. Similarly, part 2 requires five reloads, part 3 five reloads, part 4 one reload, part 5 seven reloads, part 6 one reload, part 7 five reloads, part 8 three reloads, part 9 one reload, and part 10 five reloads. The cost min_sum value is calculated as follows:

$$\begin{aligned}\text{Cost}_{\min\_\max} = &\max(\text{part 1 reload, part 2 reload,}\\ &\text{part 3 reload, part 4 reload,}\\ &\text{part 5 reload, part 6 reload,}\\ &\text{part 7 reload, part 8 reload,}\\ &\text{part 9 reload, part 10 reload)}\\ &= \max(4, 5, 5, 1, 7, 1, 5, 3, 1, 5) = 7,\end{aligned}$$

$$\begin{aligned}\text{Cost}_{\min\_\max} = &\text{part 1 reload, part 2 reload,}\\ &\text{part 3 reload, part 4 reload,}\\ &\text{part 5 reload, part 6 reload,}\\ &\text{part 7 reload, part 8 reload,}\\ &\text{part 9 reload, part 10 reload}\\ &= 4 + 5 + 5 + 1 + 7 + 1 + 5\\ &+ 3 + 1 + 5 = 37.\end{aligned}$$

Total spine layout distance = 22-unit distance

Total travelling distance = 814

Table 1. Minimum required sequence for hypothetical problem.

P. no	Number of machines and parts	Part no.	Machine progress
1	20 & 10	1	6-3-4-18-5-1-14-7-11-13-9-16-17-2-8-15
		2	17-9-11-8-10-13-2-16-4-20-18-15-6-3-7
		3	13-2-6-3-14-12-15-17-8-1-10-7-20-19
		4	7-2-6-11-8-16-1
		5	3-17-1-2-20-8-6-19-14-11-15-12-7-16-10-18-4-13-9-5
		6	9-2-6-7
		7	15-9-19-12-3-6-5-8-14-7-1-2-13-4
		8	7-19-5-4-9-16-3-14-13-11-2
		9	3-4-1-6-11-20
		10	12-6-17-15-13-9-18-14-19-7-11-2-4
2	60 & 5	1	6-11-13-19-14-24-27-31-25-41-50-38-1-58-48-55-2-9-36-53-60
		2	2-9-16-20-29-34-39-42-44-46-50-1-25-45-38-53-52-57-60
		3	5-10-12-22-26-30-33-40-42-25-14-54-18-49-51-1-8-35
		4	8-15-17-21-25-32-38-41-45-54-11-44-12-26-56-58-59
		5	1-3-4-18-23-25-28-35-37-40-43-58-60
3	60 & 10	1	5-20-32-43-6-8-12-55-58-19-16-17-35-39-22-40
		2	59-55-5-7-15-18-25-36-43-53-28-6-35-46-51-13-58-2-3-14-41-39
		3	25-29-21-45-48-15-58-51-4-13-59-53-60-27-16-29-36-33-31-48
		4	1-7-15-23-29-36-44-5-8-14-24-33-41-48-57-51-55-60
		5	22-28-45-39-16-5-15-25-43-57-60-34-6-9-32-31-48-55-38-19-8-53
		6	5-53-44-26-58-60-34-38-25-19-28-35-41-54-10-20-48-57
		7	46-35-53-60-15-6-8-1-34-18-44-39-51-50-30-42-10-57
		8	1-8-9-15-24-38-46-53-35-29-58-59-30-21-50
		9	3-4-6-32-36-11-19-52-59-2-44-47-49-24-29-20-46
		10	46-53-57-60-10-15-24-36-49-50
4	60 & 15	1	11-13-8-7-25-24-21-51-9-14-29-56-6-32-33-34-42-48-44-58-2-3-60-17-5-22
		2	57-60-59-4-1-15-37-14-5-7-50-26-24-44-36-42-10-2
		3	27-30-29-47-15-20-53-58-23-8-7-34-51-6-55-60-24-36
		4	9-12-6-17-15-13-30-41-54-58-60-50-43-26-18-14-7-11-23-2-4-25-24
		5	33-6-3-41-18-53-56-30-5-1-14-24-38-26-7-11-36-47
		6	7-22-60-11-31-18-51-44-1
		7	15-19-29-22-33-56-48-44-7-28-23-1-39-24-27-2- 13-4-26-46-11-10-35-21-52-20-18
		8	42-51-6-8-14-9-11-3-35-22
		9	21-40-10-6-11-32
		10	11-13-31-12-1-44-8-59-29
		11	23-6-58-9-34
		12	10-8-7-59-16
		13	31-7-12-20-22-18-6-26-39-44-11-55-52-17-16-21- 10-48-43-4-27-24-25-13-30-40-9-5
		14	p5-41-50-37-13-28-19-9-11
		15	21-41-6-11-22-52-14

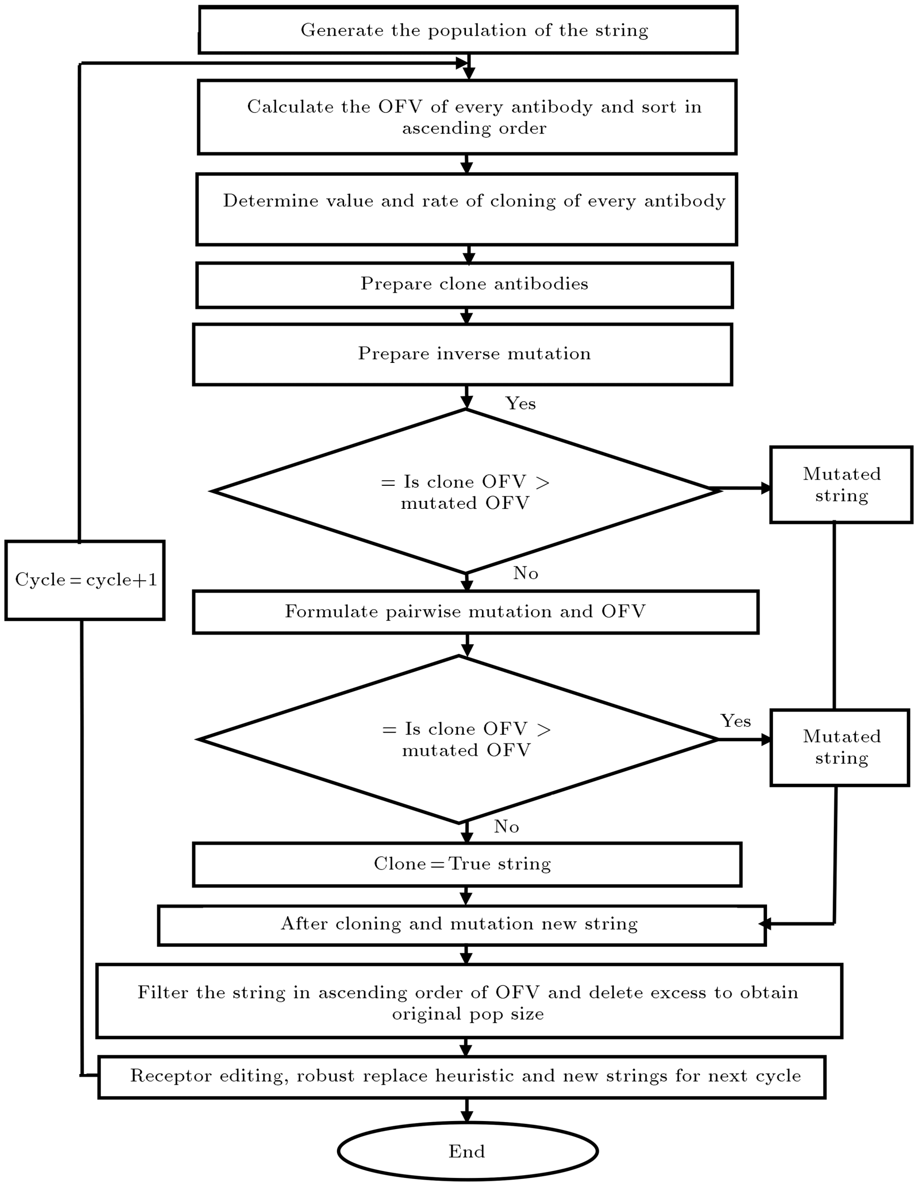

Figure 2. Working process of AIS algorithm.

3. Methodology

3.1. Artificial immune system algorithm

AIS is an efficient search methodology resembling the procedure of analytical immunology and the immune system found in organisms. In AIS, clonal choice and affinity maturation of the immune system causing immune systems to grow to prevent the infection of foreign organism in the human body bear a resemblance [30]. These antibodies produced antigens; they are multiplied by cloning based on the selection process. The cloned cells mutate and produce an affinity among antibodies, helping antigens neutralize and eliminate infectious organisms. New antibodies are developed by receptor editing and mutations, and the clones are proportional to the affinity to the antigen. As a route to transformation, a considerable number of bad cells are removed [31]. The working process of AIS is described in Figure 2. AIS algorithm consists of the following steps:

Step 1: Initialize a random population;

Step 2: Perform affinity assessment for every individual;

Step 3: Clonal choice; select the 'n' good string depending on the affinity;

Step 4: Clonal expansion; clone the 'n' good strings with the reproduction size, thus increasing with the affinity;

Step 5: Affinity maturation; mutate every string to obtain the antibodies and then, retain the strings for the next new population;

Step 6: Meta dynamics; restore the 'r' strings with new reproduced ones. Strings of lower affinity have the highest chance to be replaced;

Step 7: Refer to Steps 2 to 8 unless a certain level of criterion is achieved.

Numerical example of AIS:

Step 1: The first population (P) strings are obtained randomly up to the limited size. For instance, the sequence 8 6 4 3 1 5 7 2 10 13 9 12 11 14 18 17 16 15 20 19 is considered a layout string;

Step 2: The Objective Function Value (OFV) and the affinities are computed for randomly produced s in the population. The affinity value is determined using the formula:

$$\text{Affinity} = 1/\text{OFV}. \tag{3}$$

Step 3: Choose the cloning individuals and lay the choice of the individuals on the affinity value;

Step 4: Obtain the cloning rate and describe it below:

$$\text{Cloning rate} = \frac{\text{Solution affinity value } x \text{ population size}}{\text{Total affinity in the population}}. \tag{4}$$

This step involves creation of more individuals through cloning with minimum OFVs in comparison to maximum OFVs. This step helps determine the alternative clones that exist in the population.

Step 5: Consider the existing string for inverse and pairwise mutations as follows:

- *Inverse mutation:* It is obtained through reversing of the machines between 5 and 14 as follows:

  Original string $\longrightarrow$8 6 19 4 **3 1 57 2 10 13 9 12**

  11 14 18 17 16 15 20,

  Mutated string $\longrightarrow$8 6 19 4 **11 12 9 13 10 2 7 5 1**

  3 14 18 17 16 15 20.

  Mutated string is lower than the existing string; then, the modified string replaces the existing one. The pairwise mutation is obtained over the original string.

- *Pairwise Mutation:* It is obtained through the exchange of the machines between 7 and 17, as follows:

  Original string $\longrightarrow$8 6 19 4 3 1 **5** 7 2 10 13

  9 12 11 14 18 **17** 16 15 20

  Mutated string $\longrightarrow$8 6 19 4 3 1 **17** 7 2 10 13

  9 12 11 14 18 **5** 16 15 20

  In the post process after pairwise mutation, OFV of the mutated string is smaller than that of the previous string. Then, the original string is eliminated through mutated string; otherwise, the initial string is maintained.

Step 6: Eliminate R% of the solutions having a high value with the same R% of randomly created solutions;

Step 7: Implement Steps 2 to 7 for necessary iterations.

Parameter settings are population size (P) = 60, the number of low-affinity antibodies to be replaced (R) = 20, and termination criteria (n) = 300 iterations.

3.2. Tabu Search (TS) algorithm

Normally, the Tabu Search (TS) is framed [32–33] from the local search technique. Tabu search procedure of the chains changes from one location to another. The solution obtained by TS is the best local optimal that can be determined. Accordingly, a superior part of the solution region is discovered, while comparison of local search and Tabu search involves a wider space for achieving better solutions. The return to the local optimal resulting from TS is dependent on the technique of prohibitions in which few moves are frozen from period to period. However, it accounts for few complications including the maximum number of local optimal solutions in the region and lookout pattern, and this state is termed as deterministic chaos. The final local solution and the generations are left; however, the solution lookout is in a narrow solution region. Therefore, if this portion does not have the global domain, it can be identified by the search technique with the restricted element of the key region.

3.3. Improved Tabu Search (ITS) algorithm

ITS is recommended to overcome the difficulties of the conventional tabu search and idea of intensification and reconstruction structure which is grounded on three stages of intensification and reconstruction which accepts a solution. The ITS proceeds a new optimized solution which is regenerated further, and this process continues. The best output identified through the iterative operation is determined and stored, and this outcome would be the prospective consequential result of ITS. The same process is obtained for the number of creations to attain the optimal result [34–38]. The process of ITS is detailed in Figure 3.

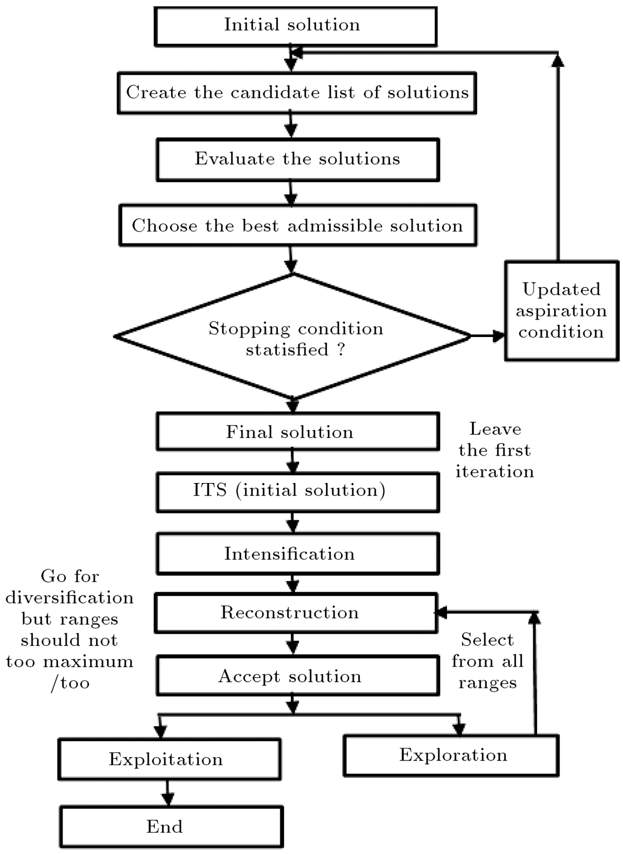

Figure 3. The process of the ITS algorithm.

Step 1: Initiate the first string S_0. Select the number of iterations $k = 1$. Select S_1 and set $S_0 = S_1$;

Step 2: Select the best S_c in the local string N (S_k):

(a) If the move $S_k \rightarrow S_c$ tabu list, set $S_{k+1} = S_k$ and go to Step 3;

(b) If $S_k \rightarrow S_c$ is not tabu list, set $S_{k+1} = S_c$

Increase the reverse move to the beginning of the tabu list and remove the entry on the end. If the objective function $G(S_c) < G(S_0)$, set $S_0 = S_c$.

Step 3: Set $k = k + 1$. Stop if end criteria are completed; otherwise, go to Step 2;

Step 4: Intensification: Reconstruct the current string;

Step 5: Reconstruction: Keep a certain level of diversification;

Step 6: Accept solution.

The exploitation and exploration of the alternative determine the selection of candidate strings for the diversification operator until a criterion is met.

3.3.1. ITS numerical example

The numerical example is illustrated in the case of a problem with 5 m/cs and 3 parts. The required machine sequences for each part are assumed as given below:

- Part 1 : 3-2-1
- Part 2 : 1-2
- Part 3 : 4-1-5

Initial sequence is 5 2 1 4 3 (initial population size 5). A numerical example of the ITS algorithm is illustrated through a step-by-step procedure:

Step 1: $S_0 = S_1$ =(5-2-1-4-3). $G(S_1) = 4$. Set $L = \{\}$

Step 2: $N(S_1)$={ (2-5-1-4-3)
(5-1-2-4-3)
(5-2-4-1-3)
(5-2-1-3-4)}5

With respective reload costs = $\{4, 4, 3, 4\}$
$S_c = S_0 = S_2$ = (5-2-4-1-3)
Set $L = \{(1, 4)\}$

Step 3: $k = 4$, stop if criteria are satisfied; otherwise, refer to Step 2;

Step 4: Intensification. This phase enhances the present solution and is applied to the freshly rebuilt solution, unlike the first iteration, which intensifies the initial solution (i.e., the "output" of construction). Experimentation indicates that the expensive runs of tabu search-based improvement technique are unnecessary. The simple tabu search iterations save a significant amount of CPU time. When used in conjunction with the diversification operators, classic tabu search might find more near-optimal solutions than extensive runs of basic tabu search.

Step 5: Reconstruction. A method of solution reconstruction allows you to move away from the current local optimum to other sections of the solution space. As a result, maintaining a degree of variety at this step is critical. If it is too high, the resulting algorithm may resemble a clean random multi-start, and if it is too minimum, the procedure may return to the same solutions on which the reconstruction was performed. It operates in the same manner as GA's mutation procedure does. During the reconstruction method, at the layout solution of K percentage for total nC2 (total number of pairings that may be exchanged) the pair of facilities are interchanged with locations.

Step 6: Solution acceptance. In exploration, only the currently best local optimum is chosen. Each locally optimal solution might be a possible candidate for study diversification. Under extreme conditions, it is possible to create a new solution from the ground up. It is feasible to achieve the so-called where-you-are approach. In this respect, any new local optimum is accepted for the reconstruction process regardless of solution quality. The exploitation approach is employed for candidate selection under the diversification operator in the described ITS-based heuristic strategy.

The ITS's usual flow is as follows. It all starts with the development of a one-of-a-kind solution using the traditional TS technique. As a result, the best option was identified first. A given solution is also reconstructed, resulting in a new one. The purpose of rebuilding is not to completely eliminate the current solution. However, since parts of this optimum may mirror those of the globally optimal solution, it is ideal that the ultimate solution would inherit certain traits from prior local optimums. As soon as the perturbation procedure is completed, the result is passed to the TS method, which begins immediately. The ITS then returns to form a new optimized solution, which is rebuilt (or be replaced with another local optimum), and so on. Throughout the iterative process, a better solution is discovered and remembered. This method is used for the needed iterations to get the best result.

Parameter setting includes initial population sizes = 19, 59. When the number of disimproving moves reaches maximum, no neighbor is generated or an infeasible solution occurs for TS, $k = 40\%$, number of iterations = 300.

4. Result and discussions

To describe the efficacy of the AIS, TS, and ITS, the hypothetical test problems mentioned in Table 1 and benchmark problem taken from Saravanan and Ganesh Kumar are assessed [39]. First, 1 to 4 problems are randomly created and hypothetical test instances are examined. Problem 1 consists of 20 machines with 10 parts with small machine and small part in size. Problems 2 to 4 (60 machines with 5 parts, 60 machines with 10 parts, and 60 machines with 15 parts) with machines and the part size increase. The large number of machines is assumed to be the same as that in the previous papers and the technique efficacy is tested. The results of AIS and ITS are given in Table 2. In this latter, the 5th and 6th test problems were adopted from the literature for layout problems with 50 machines and 10 and 20 parts. The comparative results of AIS, TS, ITS and existing SFHA are presented in Tables 3 and 4. The projected system is coded in the MATLAB and language overall tests are computed using a Pentium-IV Microsoft windows system.

The performance of the AIS and ITS is compared in the case of the hypothetical problems based on the min_max approach, unit distance, and solution effort. Further, the benchmark problems based on the min_sum and min_max approaches were resolved without distance and solution effort. From Table 2 and Figures 4, 5, and 6 for the hypothetical test problem 1, AIS achieves an optimum result compared to ITS. ITS for test problems 2 to 4 outperforms AIS with a minimum distance. For the test problems, ITS outperforms AIS in terms of percentage Solution

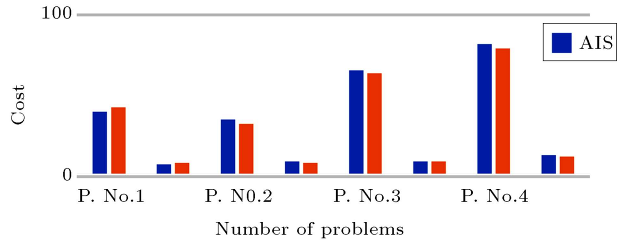

Figure 4. Comparison of ITS and AIS for large-sized loop layout problem with min_max.

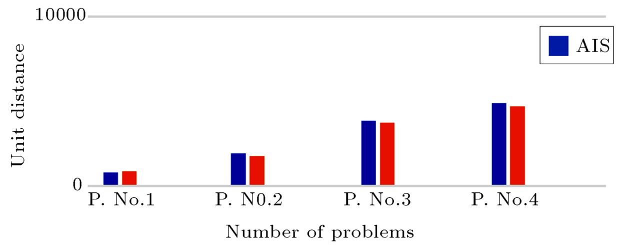

Figure 5. Comparison of ITS and AIS for large-sized loop layout problem with unit distance.

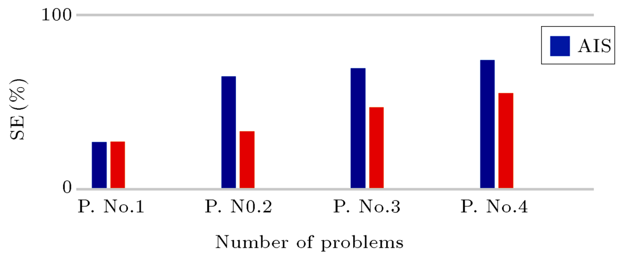

Figure 6. Comparison of ITS and AIS for large-sized loop layout problem with SE (%).

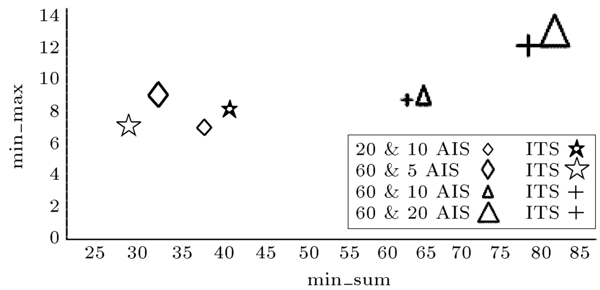

Figure 7. Comparison of min_max vs min_sum for problems with 20 M/Cs 10 parts, 60 M/Cs 5 parts, 60 M/Cs 10 parts, and 60 M/Cs 15 parts.

Effort (SE)%. The lower value SE% indicates the quick convergence which leads to the optimum result for large-sized problems. Machine size and part size increased, which can change the simple problem into a complex problem.

The proximity of solutions in both approaches specifies the consistency of the performance of a specific tool. As can be seen in Figure 7, the other method presents scattered solutions for min_max and min_sum. However, ITS presents minimum point for min_max

Table 2. Outcomes of AIS and ITS for the hypothetical problem.

P. no	NoM & NoP	Algorithm	Cost min_sum	Cost min_max	Unit distance	SE%	Congestion for each part	Desirable order of machines
1	20 & 10	**AIS**	**37**	**7**	814	27.2	4-5-5-1-7-1-5-3-1-5	**13-9-6-19-11-18-5-8-16-3-17-4-1-14-12-15-7-10-2-20**
		ITS	40	8	880	27.5	6-5-4-1-8-1-4-3-1-7	11-12-8-16-3-10-18-14-5-13-7-15-20-17-4-1-9-2-6-19
2	60 & 5	AIS	32	9	1984	64.9	9-7-6-5-5	2- 41- 46- 36- 14- 19-58-10- 49- 53- 32- 22-27-5- 31- 29- 44- 56-35-39- 17- 30- 52- 33-11-16- 55- 6- 50- 28-24- 9-3- 42- 47- 25-20- 60-45- 23- 18-57- 1- 40-13-8- 48-7- 12- 51- 37- 15-59- 54- 38- 43- 21-4- 26- 34
		ITS	**29**	**8**	**1798**	**33.3**	**6-8-6-6-3**	**1-2- 19- 6- 9- 58-41-23- 36- 31- 53-5- 44-48- 26- 8- 52-30- 42-20- 25- 3- 16-55- 59- 28- 33- 11-37- 15- 50- 40- 45-14- 39- 17- 22- 27-12- 51- 56- 34- 47-54- 4- 29- 7- 46- 32-10- 18- 57- 43- 21-60- 38- 24- 13- 35- 49**
3	60 & 10	AIS	64	9	3968	69.4	7-9-8-8-7-6-8-6-3-2	54-10-20-49-15-46-12-59-37-24-7-29-3-25-34-51-39-17-4-26-21-43-1-23-18-40-28-50-45-6-42-32-31-53-48-9-57-14-36-11-33-58-19-2-16-38-55-13-35-30-52-5-27-22-44-41-60-8-47-56
		ITS	**62**	**9**	**3844**	**47.1**	**5-7-5-5-7-9-8-7-6-3**	**55-24-58-19-28-36-6-45-2-33-41-31-9-44-48-27-8-47-52-30-42-20-25-3-16-11-49-32-10-22-1-5-57-35-40-18-12-7-46-51-29-15-54-4-43-21-60-50-14-13-38-39-59-37-53-17-56-34-23-26**

Table 2. Outcomes of AIS and ITS for the hypothetical problem (continued).

P. no	**NoM & NoP**	**Algorithm**	**Cost min_sum**	**Cost min_max**	**Unit distance**	**SE%**	**Congestion for each part**	**Desirable order of machines**
4	60 & 15	AIS	81	13	5022	74.2	9-7-6-10-8-4-10-3-0-3-1-1-13-4-2	60-12-34- 21-17-51-56-43-4-23-45-40-1-49-10-6-27-15-37-59-20-54-5-44-30-8-52-47-41-19-58-7-11-14-26-48-39-25-50-32-28-57-35-18-31-9-13-53-46-29-42-24-3-22-55-16- 33-38-36-2
		ITS	**78**	**12**	**4836**	**55.3**	**8-6-6-10-7-4-11-3-1-2-2-1-12-4-1**	**7-25-59-34-46-29-37-51-24-3-56-12-21-48-43-4-23-45-40-1-50-10-6-28-54-39-17-9-35-32-5-44-49-27-55-33-38-16-11-2-41-36-15- 19-58-18-22-13-52-57-26-30-8-60-20-42-14 31-53-47**

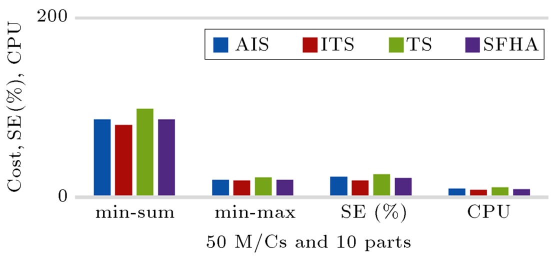

Figure 8. Comparison of the ITS with others for min_max approach, SE (%), CPU to the 50-machine, 10-part benchmark problem [16].

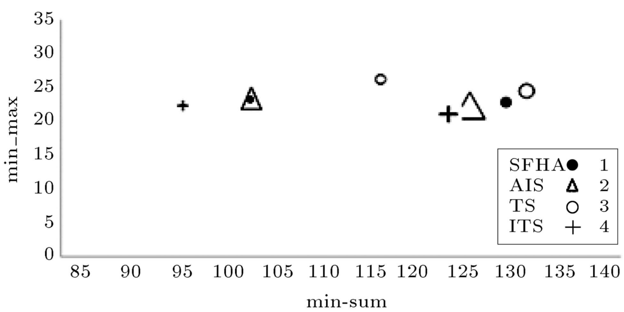

Figure 9. Comparison of the min_max and min_sum approaches to the benchmark problems with 50 machines in 10 and 20 parts [16].

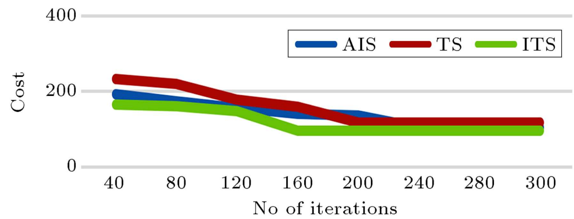

Figure 10. Comparison of the algorithm convergence for min_sum approach to the problems with 50 machines and 10 parts.

and min_sum and displays closer solutions for both, thus specifying the consistency of the tool performance.

From Tables 3–4 and Figure 8–9 for the benchmark test problem 5, ITS achieves the optimum result compared to IT, AIS, and SFHA with less solution effort. Moreover, it outperforms AIS and TS [40,41]. ITS outperforms TS, AIS, and SFHA in the case of test Problem 6. When the performance of the ITS method was evaluated with the min_max, the findings indicate that it performed well for all of the large-sized test issues. The percentage of solution effort represents the algorithm pace of convergence to the best answer. In a number of test issues, ITS outperformed TS, AIS [42], and SFHA at a faster convergence rate.

Figure 10 shows that the answer reaches convergence within 300 cycles for configurations. Best congestion is 95 produced and the ITS gives almost faster convergence for all of the layout problems than TS and AIS. TS produced the same results at the end of the iterations.

The results from the experiments indicate that the AIS exhibits proper performance in finding a near-optimum solution to the 20-machine size problem. When the problem size increased from 20 to 50 or 60 machines, the performance of ITS in determining the

Table 3. Comparison of the improved tabu search algorithm with other algorithms for 50-machine, 10-part problem [16].

Algorithm	Cost min_sum	Cost min_max	SE (%)	CPU time	Congestion for each part	Optimal order of machines
min_max approach						
SFHA	103	23	25.4	10.3	23-10-4-10-11-20-9-6-8-2	50-41-14-23-34-6-18-32-17-15-39-44-19-47-48-11-4-31-30-25-13-40-24-22-12-46-2-28-33-16-49-43-36-10-7-37-8-20-42-45-3-38-26-1-5-27-9-21-29-35
AIS	103	23	27.1	10.9	23-6-6-8-12-22-11-6-6-3	49-23-43-32-22-14-44-19-36-10-33-37-31-26-24-15-18-34-47-40-3-27-1-20-41-35-50-46-28-13-2-11-38-30-39-42-25-6-4-29-5-9-21-48-7-12-17-8-45-16
TS	117	26	30.2	12.7	26-9-7-10-13-25-14-6-6-1	41-44-36-29-15-17-31-35-27-32-30-19-40-1-24-7-3-43-16-20-14-37-18-12-46-22-21-38-10-23-8-48-9-4-50-26-45-28-42-2-5-6-49-25-13-11-39-33-34-47
ITS	**95**	**22**	21.9	9.8	22-8-5-10-8-19-11-4-6-2	**8-40-9-18-22-47-5-34-7-35-16-36-33-44-27-10-19-50-3-38-43-15-26-21-25-37-17-39-20-42-45-13-30-1-6-12-32-4-46-48-31-41-11-2-28-24-14-23-29-49**

best solution was enhanced. A large number of local optima across the solution space, repeating sequences or presence of cycle search configuration, and chaotic attractors are some of the downsides of traditional TS. Although the TS chaotic attractors are distinguished by "getting stuck" because of the absence of cycles and local optima, the search is still limited to a "small region" of the solution space [43]. As a result, the search method will only look at a small portion of the solution space.

ITS, unlike TS, widens its search space with each generation to find a better solution and avoid local minima. Reconstruction and intensification are the two primary tactics. The first strategy requires reconstructing the existing solution that shifts away from the current location to a new one in the solution space. The second technique applies local refinement to a "ruined" solution [44] based on the classic TS methodology; theoretically, the enhanced solution is better than the responses of previous iterations. The principal purpose of ITS is to find near-optimal feasible solutions by repeating these stages multiple times.

5. Conclusion

This paper proposed Improved Tabu Search (ITS) and Artificial Immune System (AIS) to solve the loop design problems using min_max approach. Computational results show that ITS algorithm outperformed the AIS algorithm for the small-size problem. Further results demonstrated that ITS outperformed AIS algorithm to solve large-sized problems with less solution effort through exploitation and exploration. ITS is a principle-based optimization policy that "reconstructs and improves" to provide better outcomes than other algorithms. In addition, for large-scale benchmark

Table 4. Comparison of the improved tabu search algorithm with others for 50-machine, 20-part problem [16].

Algorithm	Cost min_sum	Cost min_max	SE (%)	CPU time	Congestion for each part	Optimal order of machines
			min_max approach			
SFHA	129	23	55.9	14.1	6-3-5-6-8-12-23-6-7-5-4-3-3-7-5-3-2-10-7-4	50-2-29-20-40-36-41-27-26-11-34-31-19-30-22-45-39-9-46-8-23-15-18-25-4-13-17-44-5-49-38-10-43-35-37-33-28-14-48-3-16-6-7-42-21-32-47-12-1-24
AIS	125	21	56.0	15.5	7-3-4-7-8-12-21-6-7-6-4-3-1-7-4-3-0-11-7-4	37-11-50-27-30-31-1-21-24-26-23-28-9-13-4-34-19-49-22-32-45-48-6-39-25-41-14-46-33-44-3-5-42-29-16-36-10-38-15-18-43-47-12-2-40-7-8-35-17-20
TS	132	24	57.5	19.7	8-5-3-5-7-9-24-8-7-6-5-3-2-6-5-5-2-11-6-5	2-8-39-4-44-16-30-27-33-38-9-35-37-36-43-14-32-47-46-3-31-28-26-24-11-5-15-7-25-21-49-6-45-1-41-34-19-23-29-20-12-18-42-40-10-22-48-17-50
ITS	124	21	51.1	12.9	6-5-3-5-9-10-21-4-4-4-5-4-3-10-6-3-2-10-6-4	44-37-32-17-41-15-45-25-8-20-13-12-47-49-28-11-43-50-30-23-10-18-29-14-6-26-1-36-3-2-9-7-22-19-38-16-39-40-42-48-31-33-4-34-5-46-21-24-27-35

problems, ITS outperformed TS, AIS, and Sheep Flock Heredity Algorithms (SFHA). Min-max and min_sum provided a more accurate solution for both, pointing to the tool consistency in performance. The novel approach and the mathematical model can be adopted to solve bidirectional LLPs in the future.

References

1. Afentakis, P. "A loop layout design problem for flexible manufacturing systems", Int *J of Flex Manuf Syst*, **1**(2), pp. 143–175 (1989).

2. Cheng, R. and Gen, M. "Loop layout design problem in flexible manufacturing systems using genetic algorithms", *Comput Ind Eng.*, **44**(1) , pp. 53–61 (1998).

3. Bennell, J.A., Potts, C.N., and Whitehead, J.D. "Local search algorithms for the min_max loop layout problem", *J Oper Res Soc*, **53**, pp. 1109–1117 (2002).

4. Yang, T., Peters, B.A., and Tu, M. "Layout design for flexible manufacturing systems considering single loop directional flow patterns", *Eur J Oper Res.*, **164**(2), pp. 440–455 (2005).

5. Kumar, R.M.S., Asokan, P., and Kumanan, S. "Artificial immune system-based algorithm for the unidirectional loop layout problem in a flexible manufacturing system", *Int J Adv Manuf*, **40**, pp. 553–565 (2009).

6. Kumar, R.M.S., Asokan, P., and Kumanan, S. "Design of loop layout in flexible manufacturing system using non-traditional optimization technique", *Int J Adv Manuf Technol*, **38**, pp. 594–599 (2008).

7. Nearchou, A.C. "Meta-heuristics from nature for the loop layout design problem", *Int J ProdEcon*, **101**, pp. 312–323 (2006).

8. Chae, J. and Peters, B.A. "A simulated annealing algorithm based on a closed-loop layout for facility layout design in flexible manufacturing systems", **44**, pp. 2561–2572 (2006).

9. AsefVaziri, A.A., Kazemi, M., and Eshghi, K. "An ant colony system for enhanced loop-based aisle-network design", *Eur J Oper Res.*, **207**(1), pp. 110–120 (2010).

10. Hong, S., Johnson, A.L., Carlo, H.J., et al. "Optimizing the location of crossovers in conveyor-based automated material handling systems in semiconductor wafer fabs", *International Journal of Production Research*, **49**, pp. 6199–6226 (2011).
11. Lasrado, V. and Nazzel, D. "Design of a manufacturing facility layout with a closed-loop conveyor with shortcuts using queueing theory and GA", *Proceedings of the Winter Simulation Conference*, Jain S, Creasey R.R., Himmelspach, J., White K.P., Fu M. (Eds.) (2009).
12. Glover, E. and Laguna, M., Tabu Search, Kluwer, Dordrecht (1997).
13. Javid, A.A, and Jaafari, A.A. "The unequal area facility layout problem with shortest single-loop AGV path: how material handling method matters", *International Journal of Production Research*, **59**(8), pp. 74–82 (2020)
14. Potts. C. and Whitehead, J.D. "Workload balancing and loop layout in the design of a flexible manufacturing system", *Eur J Oper Res.*, **129**(2), pp. 326–336 (2001).
15. Yang, T., Su, C.T., and Hsu., Y.R. "Systematic layout planning: A study on semiconductor wafer fabrication facilities", *International Journal of Operations and Production Management*, **20**, pp. 1359–1371 (2000).
16. Yang, T. and Peters, B.A. "A spine layout design method for semiconductor fabrication facilities containing automated material handling systems", *Int J Oper Prod Manag*, **17**(5), pp. 490–501 (1997).
17. Suhardini, D., Septiani, W., and Fauziah S. "Design and simulation plant layout using systematic layout planning", *IOP Conf. Series: Materials Science and Engineering*, **277**, pp. 12–52 (2017).
18. Kang, S., Kim, M., and Chae, J. "A closed-loop based facility layout design using a cuckoo search algorithm", *Expert Systems with Publications, an International Journal*, **93**(3), pp. 322–335 (2018).
19. Perez-Gosende, P., Mula, J., and Diaz-Madronero, M. "Facility layout planning. An extended literature review", *International Journal of Production Rese arch*, **59**(12), pp. 3777–3816 (2021).
20. Alduaji, A. and Hassin, N.M. "Adopting a circular open-field layout in designing flexible manufacturing systems", *International Journal of Computer Integrated Manufacturing*, **33**(6), pp. 572–589 (2020).
21. Hungerlander, P., Maier, K., Pachatz, V., et al. "Exact and heuristic approaches for a new circular layout problem", *SN Appl. Sci.*, **2**(6), 1035 (2020).
22. Guttaa, P.R., Chinthalaa, V.S., Manchojua, R.V., et al. "A review on facility layout design of an automated guided vehicle in flexible manufacturing system", *Materials Today: Proceedings*, **5**(2), pp. 3981–3986 (2018).
23. Rai, R. and Jayswal, SC. "Design and optimization of loop layout in flexible manufacturing system using particle swarm optimization", *Int J Adv Technol*, **9**(2), pp. 1–5 (2018).
24. Kim, M. and Chae, J. "Monarch butterfly optimization for facility layout design based on a single loop material handling path", *Mathematics*, **7**(2), pp. 1–21 (2019).
25. Wenhan, D., Ming, Z., Lijin, W., et al. "Hybrid population-based incremental learning algorithm for solving closed-loop layout problem", *Journal of Computer Applications*, **41**(1), pp. 95–102 (2021).
26. Mohammadia, A.S., Alemtabriza, A., Pishvaeeb, M.S., et al. "A multi-stage stochastic programming model for sustainable closed-loop supply chain network design with financial decisions: A case study of plastic production of recycling supply chain", *Scientia Iranica, E.*, **27**(1), pp. 377–395 (2020).
27. Deng, W., Zhang, M., He, K., et al. "A mixed-coding harmony search algorithm for the closed loop layout problem", *ICMLC '21*, pp. 93–100 (2021).
28. Saravanan, M. and Ganesh Kumar, S. "Different approaches for loop layout problem: a review", *Int J Adv Manuf Technol.*, **69**, pp. 2513–2529 (2013).
29. Hosseini, S.S., Azimi, P., Shaarifi, M., et al. "A new soft computing algorithm based on cloud theory for dynamic facility layout problem", *RAIRO Operations Research*, **55**, pp. 2433–2453 (2021).
30. De Castro, L.N. and Timinis, J., *Artificial Immune System: A New Computational Approach*, Springverlag, London (2002).
31. Hungerlander, P., Maler, K., Pachatz, V., et al. "Exact and heuristic approaches for a new circular layout problem", *SN Appl. Sci.* **2.**, pp. 1035–1045 (2020).
32. Glover. F. "Tabu search: Part I", *ORSA J Comput*, **1**, pp. 190–206 (1989).
33. Glover, F. "Tabu search: Part II", *ORSA J Comput*, **2**, pp. 4–32 (1990).
34. Battiti, R. and Techchioli, G. "The reactive tabu search", *ORSA J. Comput.*, **6**, pp. 126–140 (1994).
35. Johnson, D.S. "Local optimization and the travelling salesman problem", *Lec. Notes Comp. Sci.*, **443**, Springer, Berlin, pp. 446–461 (1990).
36. Martin, O. and Otto, S.W. "Combining simulated annealing with local search heuristics", *Annals Oper. Res.*, **63**, pp. 57–75 (1996).
37. Schrimpf, G., Schneider, J., Stamm-Wilbrandt, H., et al. "Record-breaking optimization results using the ruin and recreate principle", *J. Comput. Phy*, **159**, pp. 139–171 (2000).
38. Lourenco, H.R., Martin, O.R., and Stutzle, T. "Iterated local search", In F. Glover, and G. Kochenberger, (Ed.), *Handbook of Metaheuristics*, Kluwer, Norwell, pp. 321–353 (2002).
39. Saravanan, M. and Ganesh Kumar, S. "Design and optimisation of loop layout problems flexible manufacturing system using sheep flock heredity algorithm", *Int J Adv Manuf*, **77**, pp. 1851–1866 (2015).
40. Kannan, D., Govindan, K., and Soleimani, H. "Artificial immune system and sheep flock algorithms for two-stage fixed charge transportation problem", *optimization*, **63**, pp. 1465–1479 (2014).

41. Venkumar, P. and Chandrasekar, K. "Design of cellular manufacturing system using non-traditional optimization algorithms", In *Operations Management Research and Cellular Manufacturing, Innovative Methods and Approaches*, Publisher: IGI Global, pp. 99–139 (2011).
42. Ojha, U. and Chow, M.Y. "An analysis of artificial immune system and genetic algorithm in urban path planning", *Proceedings of 36th Annual Conference of the IEEE Industrial Electronics Society*, Glendale, USA (2010).
43. Glover, F., Laguna, M., and Marti, R. "Principles and strategies of tabu search", *Handbook of Approximation Algorithms and Metaheuristics*, 2nd Ed., New York, Chapter 23, pp. 1–17 (2018).
44. Lourenco, H.R., Martin, O.C., and Stutzle, T. "Iterated local search", *International Series & Management Science, Handbook of Metaheuristics*, **57**, pp. 320–353 (2003).

Biographies

S. Suresh Balaji completed his undergraduate in Mechanical Engineearing from GCE, Salem, He completed his ME in Engineering Design specialization from Paavai Engineering College, Salem. He is currently working as Assistant Professor at KIOT, Salem. He has 15 years of teaching experience. He has published 19 papers in various national and international journals and conferences.

S. Ragunathan received BE degree in GCE, Salem in 1997 and ME degree from GCT, Coimbatore. He received his PhD in Robotics from Anna University, Chennai in 2008. He has been a Research Associate at the Anna University, Chennai from 2002 to 2006. Presently, he is the Principal in SRG Engineering College, Namakkal, Tamilnadu. He has supervised 7 PhD scholars and at present, he is guiding 2 research scholars. He has published 48 papers in various journals and conferences. He is having 15 years of research experience and 25 years of teaching experience.

K. Visagavel received his BE degree in MechEngg and ME degrees in Thermal Power Engineering from Annamalai University, India and the PhD degree from the Anna University, Chennai in 2010. Now, he is working as a Professor in and the Head of the Department of MechEngg in KIOT. He has supervised 5 PhD and published 91 papers in various journals and conferences. He has 11 years of research and 23 years of teaching experience.

S. Ganesh Kumar is an Professor in Mechanical Engineering at Er. Perumal Manimekalai College of Engineering. He Completed his PhD from Anna University, Chennai. He has rich experience in various institutions. He has published 14 technical papers in referred journals.