Proceedings of the 2004 IEEE
International Symposium on Intelligent Control
Taipei, Taiwan, September 2-4, 2004

A Random Search Approach to the Machine Loading Problem of an FMS

Felix T. S. Chan, Rahul Swarnkar and Manoj K. Tiwari

***Abstract*—This paper discusses a modelling framework that addresses operational planning problems of flexible manufacturing systems (FMSs). A generic 0–1 mixed integer programming formulation integrating the part selection and loading problems has been proposed. The constraints considered in the problems are mainly the availability of tool slots and machining time on the machining centres. The above problem is solved using an algorithm based on Simulated Annealing (SA). The potential capability of the approach is demonstrated via a small set of test problems.**

I. Introduction

THE market nowadays is becoming more efficient and customer oriented. There is a growing need for flexibility, efficiency and high quality forcing major changes in the manufacturing industry. It has been observed that concept of the flexible manufacturing system (FMS) has provided viable answer to the problem pertaining to flexibility and efficiency in recent years. FMSs are beneficial over traditional manufacturing due to their ability to comply with the highly uncertain market. The recent trends in manufacturing include shorter product life cycle, frequent design changes and continuously changing demand. There are two types of decision problems associated with FMSs: Design Problems (selection of machines, robot layout decisions, automated guided vehicles path selection, etc.); and Operational Problems (part type selection, resource grouping, production ratio determination, allocation of resources and loading). The machine loading problem in an FMS is specified as to assign the machines operations of selected jobs and the tools necessary to perform these operations by satisfying the technological constraints (available machining time and tool slots constraints) in order to ensure the minimum system unbalance and maximum throughput, when the system is in operation. System unbalance is the summation of idle times remaining on the machines after allocation of all feasible jobs in the system whereas System Throughput refers to the unit of jobs to be produced. Stecke [1] discussed six objectives pertaining to the machine loading problem of FMSs:

1) Balancing the machine processing time
2) Minimizing the number of movements
3) Balancing the workloads per machine for a system of pooled machines of equal sizes.
4) Unbalancing the load per machine for a system of groups of pooled machines of unequal sizes.
5) Filing the tool magazines as densely as possible
6) Maximizing the sum of operations priorities

From the available literature, it is evident that majority of the performance measures of loading are quite stringent and frequently involve multiple objectives [1-4]. Numerous solution methodologies have been developed to solve the machine loading problems of FMSs. Mukhopadhyay and Tiwari [5] have solved the machine loading problem using the principle of conjoint measurement. Mukhopadhyay *et al.* [6] have prioritized loading of machine, tool and parts in random FMS through eigenvalue analysis.

Chan [7] discussed the effect of universal loading station along with operational control rules. Shanker and Tzen [3] discussed bi–criterion objective for loading problem that includes balancing loads and meeting due date of jobs. Ammons *et al.* [8] considered a bi–criterion objective viz. balancing workloads and minimizing workstation visits to resolve the loading problem. Rajagopalan [9] clubbed the loading problem with other problems inherently found in planning stage such as job selection and production ratio determination with the aim to achieve better production schedules without too much iteration. Mukhopadhyay *et al.* [10] and Tiwari *et al.* [11] attempted machine loading problem using heuristic approaches with an objective to minimize system unbalance, thereby maximizing the throughput. Several objective functions such as maximization of workload balance on machines, minimization of system unbalance, maximization of system utilization, minimization of flow time, etc. have been considered by researchers for solving the machine loading

Manuscript received January 31, 2004.

Felix T. S. Chan is with the Department of Industrial and Manufacturing Systems Engineering, University of Hong Kong, Pokfulam Road, Hong Kong. (Phone: (852) 2859 7059; fax: (852) 2858 6535; e-mail: ftschan@hkucc.hku.hk).

Rahul Swarnkar is with Department of Industrial and Manufacturing Systems Engineering, University of Hong Kong, Pokfulam Road, Hong Kong. (e-mail: rahulswarnkar@hotmail.com).

Manoj K. Tiwari is with the Department of Manufacturing Engineering, National Institute of Foundry and Forge Technology, Ranchi – 834 003, India. (e-mail: mkt09@hotmail.com)

problems in FMS [12-13]. Some researchers have address the loading problem with these objectives utilizing heuristic approach based on evolutionary and hybrid algorithms [14-18].

The objective of this paper is to solve the machine loading problem, which includes a bi–criterion objective function of minimizing the system unbalance and maximizing the throughput in presence of available machine and tool slots as constraints. These objectives result into higher machine utilizations, higher system output and often lead to limiting the job tardiness.

In this paper, the combined part sequencing and machine loading problem have been solved by a heuristic approach based on the simulated annealing (SA) approach. The SA based approach makes use of a perturbation technique to generate a number of solutions and carries out iterative improvements beginning from a initial feasible solution thereafter searching for better solutions. The proposed heuristic has been employed on ten sample problems adopted from [10].

II. Problem Environment

A. Problem Description

The FMS under consideration in this paper consists of a number of multifunctional CNC machines, tools with capability to perform several operations and automated material handling devices, where several types of jobs arrive with different processing requirements. Jobs are available in batches and some of them are to be selected for processing during a given planning horizon. Job selection is carried out in the beginning of planning period. A job includes one of more operations and each of them can be performed by one or more number of machines. The details related to production requirement of job, number of operations for each job and their machining time, numbers of tool slots required are known in advance. Essential and optional types of operation are associated with each job. Essential operations of a job are the operations that can be performed only on a particular machine using certain number of tool slots whereas optional operations are those which can be carried out on a number of machines with same or varying processing time and tools slots. In this problem, the flexibility lies in the selection of machines for processing optional operations of the jobs. The complexity associated with a typical machine loading problem enlisted in Table I can be best exemplified as follows:

There are 8 jobs that can be sequenced in 8! ways, all together 2592 operation–machines–allocation combinations are possible for one of the job sequences. Hence, for 8! job sequences, total number of possible allocation turns out to be 8! × 2592 = 104509440. Some of these operation allocations are not feasible because they are not able to satisfy system constraints (available machining time and tool slots). Enumerating an optimal / near optimal solution in such a huge search space is computationally prohibitive and needs some heuristic approach. Shanker and Srinivasulu [4], Mukhopadhyay *et al.* [10], Tiwari *et al.* [11] developed heuristic solutions using minimization of system unbalance and thereby maximizing the throughput as objective functions in presence of system constraints. Due to the requirement of large computational time in presence of huge search space to arrive at optimal / near optimal solution, the authors have proposed simulated annealing based algorithm to resolve the complexities of the above machine loading problem. The search procedure works with iterative improvement, beginning at an initial feasible solution and continuing to better solutions in successive stages.

TABLE I
PROBLEM DESCRIPTION FOR THE TEST PROBLEM

Part type	Batch size	Operation number	UPT	Tool slot needed	Machine number
1	8	1	18	1	3
2	9	1	25	1	1, 4
		2	24	1	4
		3	22	1	2
3	13	1	26	2	4, 1
		2	11	3	3
4	6	1	14	1	3
		2	19	1	4
5	9	1	22	2	2, 3
		2	25	1	2
6	10	1	16	1	4
		2	7	1	4, 2, 3
		3	21	1	2, 1
7	12	1	19	1	3, 2, 4
		2	13	1	2, 3, 1
		3	23	3	4
8	13	1	25	1	1, 2, 3
		2	7	1	2, 1
		3	24	3	1

UPT: Unit Processing Time

B. Model Formulation

1) Formulation

The problem described above is formulated as bi–criterion objective problem, which is a combination of two objectives as presented in Eq. (1) and Eq. (2), the notation used in the formulation is presented in Table II.

$$\text{Max } f_1 = \frac{M \times L - \sum_{m=1}^{M} \left(T_m^- - T_m^+\right)}{M \times L} \tag{1}$$

$$\text{Max } f_2 = \frac{\sum_{j=1}^{J} B_j \times y_j}{\sum_{j=1}^{J} B_j} \tag{2}$$

Hence, the overall objective function becomes

$$\text{Max } f = \alpha_1 f_1 + \alpha_2 f_2 \tag{3}$$

where α_1 and α_2 are the weights assigned to the objective functions, such that

$\alpha_1 + \alpha_2 = 1$

subject to

$$\sum_{m=1}^{M}\left(T_m^- - T_m^+\right) \geq 0 \quad (4)$$

$$\sum_{o=1}^{O_j} T_{jmo} y_{jmo} \leq T_{jm}^a; \quad j = 1,2\ldots J \quad (5)$$

$$\sum_{o=1}^{O_j} \tau_{jmo} y_{jmo} \leq \tau_{jm}^a \quad j = 1,2\ldots J \quad (6)$$

$$\sum_{m \in S_{jo}} y_{jmo} \leq 1; \quad j = 1,2\ldots J \quad o = 1,2\ldots O_j \quad (7)$$

$$\sum_{m=1}^{M}\sum_{o=1}^{O_j} y_{jmo} = y_j \times O_j; \quad j = 1,2,\ldots J \quad (8)$$

$$T_{jm}^r y_{jm} \geq 0; \quad j = 1,2\ldots J \quad m = 1,2\ldots M \quad (9)$$

$$\tau_{jm}^r y_{jm} \geq 0; \quad j = 1,2\ldots J \quad m = 1,2\ldots M \quad (10)$$

$$y_j = \begin{cases} 1; \text{if job } j \text{ is selected} \\ 0; \text{otherwise} \end{cases} \quad (11)$$

$$y_{jm} = \begin{cases} 1; \text{if job } j \text{ is assigned to machine } m \\ 0; \text{otherwise} \end{cases} \quad (12)$$

TABLE II
NOTATIONS USED IN FORMULATION

j	Index of job; $1 \leq j \leq J$
m	Index of machines; $1 \leq m \leq M$
o	Index of operation; $1 \leq o \leq O_j$
J	Total number of jobs
M	Total number of machines
O_j	Total number of operations for job j
B_j	Batch size of job j
L	Length of scheduling period
T_m^-	Under utilized time on machine m
T_m^+	Over utilized time on machine m
T_m	Tool slot capacity on machine m
S_{jo}	Set of machines on which operation o of job j can be performed
τ_{jm}^r	Time remaining on machine m after allocation of operation o of job j
τ_{jm}^a	Time available on machine m after allocation of operation o of job j
τ_{jmo}	Time required by machine m for operation o of job j
T_{jm}^r	Tool slot remaining on machine m after allocation of operation o of job j
T_{jm}^a	Tool slot available on machine m after allocation of operation o of job j
T_{jmo}	Tool slot required by machine m for operation o of job j

y_j, y_{jk}, y_{jmo} are the decision variables

$$y_{jmo} = \begin{cases} 1; \text{if operation } o \text{ of job } j \text{ is} \\ \quad \text{assigned to machine } m \\ 0; \text{otherwise} \end{cases} \quad (13)$$

The first objective is to minimize the system unbalance, which is equivalent to maximize the system utilization and is given by (1), whereas (2) shows the maximization of throughput or equivalently, maximizing the system efficiency. (4) ensures that the system unbalance of the system never becomes negative. Equations (5) and (6) ensure that the number of slots and time needed for the operation of the jobs to be performed on a machine must always be less than or equal to the tool slots and time available in that machine. Despite the flexibility exists in the selection of a machine for optimal operations, once a machine is selected; the operation has to be completed on the same machine, this constraint is expressed in equation (7). Constraint (8) implies that once a job is considered for processing, all the operations are to be completed before undertaking a new job. Constraints (9) and (10) ensure that the number of tool slots and remaining time on any machine after any assignment of job should always be positive or zero. Constraints (11) through (13) make the decision variables possessing the value of 0 and 1 integers.

III. SIMULATED ANNEALING

A. Basic Concepts

Kirkpatrick *et al.* [19] first introduced simulated annealing (SA) to solve difficult combinatorial optimization problems. It is a controlled random search technique and draws its analogy from physical annealing of metals. In the annealing process a metal is heated to its crystallization temperature and then cooled slowly so that uniform hardness is achieved throughout the metal. SA simulates the basic concepts of annealing of metals to arrive at the near global optimum solution for complex engineering problem. Figure 1 shows SA algorithm for minimization problem. This algorithm is generic in nature and has to be modified as per the requirements of the problems. The control parameters involved at each step in the SA algorithm is temperature (T).

1. Get an initial solution S
2. Get an initial temperature $T > 0$
3. While not frozen, Do:
    - 3.1. Perform the following loop n times
        - 3.1.1. Pick a random neighbor S' of S
        - 3.1.2. Let $\Delta = f(S') - f(S)$
        - 3.1.3. If $\Delta < 0$ (downhill move) then $S = S'$
        - 3.1.4. If $\Delta \geq 0$ (uphill move) then $S = S'$ with probability $P(\Delta, T)$
        - 3.1.5. If $f(S) \geq f(S_{best})$ then $S_{Best} = S$
    - 3.2. T = New value of temperature
4. S_{Best} is the best solution

Fig. 1. A generic SA algorithm

Based on iterative improvement, SA is a heuristic method with the basic idea of generating random displacement from any feasible solution. This process accepts not only the generated solutions, which improve the objective function but also those, which do not improve it with the probability exp(-$\Delta f/T$), a parameter depending on the objective function and decreasing temperature.

B. Proposed Heuristic

The initial solution procedure on which the proposed SA based heuristic is applied was adopted from Tiwari *et al.* [11]. They have suggested a solution methodology to the loading problem in FMS by considering the "utilization of maximum remaining available time" with the objective of minimizing system unbalance and thereby maximizing throughput. The initial job sequence selected for the heuristic is by SPT.

C. Search parameters

The proposed SA based heuristic is to be applied to the maximization problem considered in this article. The algorithm runs with any random sequence of jobs as initial solution, but it is experimentally observed that the search, if started with some specific sequencing rule (SPT, LPT etc.), performs better. This sequencing rule may be different for different types of FMS and is determined experimentally. As quoted by previous researchers, SPT makes the algorithms work better when used as initial solution. The performance of the algorithm is governed by various parameters, which are as follows:

1) Neighborhood generation (Perturbation)

The method used in this article to generate new solution performs better than some method available in literature. For example Sridhar and Rajendran [20] used adjacent interchang method to generate new sequences for solving the scheduling problem in cellular manufacturing. In adjacent interchanged method proposed by them, a random number is generated between one and the total number of jobs and the job corresponding to this position is interchanged with one of its adjacent neighborhood at random. The drawback is that it does not destroy seed sequence entirely. Another perturbation method called Modified Insertion Scheme (MIS) was devised by Mukhopadhyay *et al.* [21], in which two random numbers are generated between one and the total number of jobs. The jobs occupying these positions are interchanged and hence an entirely new solution is generated. MIS was successful in overcoming the above drawback, however it may interchange two previously assigned jobs, which leads to little improvement in the solution quality. In this paper attempt has been made to overcome these drawbacks by another perturbation method, in which the position a randomly selected assigned job is interchanged with that of an unassigned one in the candidate solution. This adds to the stochastic nature of the algorithm in searching newer solutions.

2) Transition probability

When a random neighboring solution is generated, its function value is evaluated. If the function value is improving, it is accepted; otherwise it is accepted only if the transition probability is higher than a uniform random number. For every perturbed solution that is inferior to the candidate solution, transition probability '*P*' is calculated, which is given by

$$P = \exp(-\Delta S/T)$$

Where ΔS is the difference of function values of current solution and the neighboring solution and T is the temperature. Depending on this probability, the solution is rejected or accepted. The probability function has higher value at small ΔS and larger T; and lower value at large ΔS and small T, owing to which the inferior solutions are easily accepted at the initial stages of search, but not at last as the algorithm assumes that the best solution obtained so far is near optimal.

The search initially starts with a high temperature with a better chance of escaping local optima (i.e. transition from a low cost solution to a high cost solution). As the temperature reduces the transition probability approaches zero. When the temperature approaches zero, the search is no longer able to escape local optima and neighboring solution is accepted only if it shows improvement in function value. If the transition from the current solution to the neighboring solution is rejected, another solution in the neighborhood is selected and evaluated.

3) Annealing schedule

The temperature decline is performed using a function known as annealing schedule. In this article, the following annealing schedule is used

$$T = T_0 / (1 + \ln i)$$

Where T is the current temperature, T_o the initial temperature and i the number of iterations. In the annealing schedule adopted in this article, the cooling rate is high in the beginning and low in the end (Figure 2). This schedule is adopted keeping in mind the fact that the chances of getting better results are higher in the beginning and low therefore there is not much use of higher transition

Fig. 2: Cooling curve

probability to escape the local optima. After a few iterations, attempt has been made to prevent the temperature from decreasing abruptly, which may result in frequent rejection of perturbed solution.

4) Rejecting the inferior candidate solution

SA keeps track of the number of rejection of perturbed solutions. The counter gets increased by 1 whenever a solution is not selected through the probability consideration. If it reaches a predetermined fixed value, it signifies that there are no superior solutions in the neighborhood i.e. search has reached a near-optimal solution. At this moment the search is stopped or started again depending upon the other criteria.

5) Stopping criteria

To stop the search procedure from roaming in the solution space, two criteria are incorporated in this paper:

1) The "rejection counter" acquires its value equal to a predetermined fixed value, because it indicates that no optimal/near-optimal solution has been achieved during the last few steps, thereafter the probabilities of getting any better solution is small.
2) Number of iterations reaches the maximum number of iterations or equivalently temperature falls to minimum temperature set for the algorithm. Any further reduction in temperature would not be useful because at this low temperature the possibility of accepting inferior solutions is very small.

IV. Results and Discussion

The proposed SA approach is stochastic in nature and its search scheme prevents the entrapment of solution at local optimum points. Table III shows the values of system unbalance and throughput obtained through the proposed simulated annealing based heuristic. A study has been carried out to evaluate the computational performance of the simulated annealing based algorithm in solving the machine loading problem. Figure 3 compares the performance of the proposed algorithm with the existing approaches. In this work, attempts have been made to develop a solution methodology, which is able to encompass the following features:

1) Several combinations of job sequence are evaluated by perturbing the sequence obtained from the fixed job sequence.
2) Corresponding to each job sequence, the operation–machine–allocations are carried out to achieve the combined objectives of minimum system unbalance and maximum throughput by satisfying the system's technological constraints.
3) New sequences are generated and corresponding operation allocations are made till the optimal/near-optimal results are achieved.

TABLE III
VALUES OF SYSTEM UNBALANCE AND THROUGHPUT OBTAINED BY THE PROPOSED HEURISTIC

Test	System unbalance		Throughput	
Problem	Mean	Std. Dev.	Mean	Std. Dev.
1	15.2	1.93	50.3	2.6
2	234.4	14.2	62.8	4.00
3	128.3	3.24	73.2	4.92
4	819.0	0.00	51.0	0.00
5	364.7	9.41	78.0	6.49
6	69.5	4.21	64.4	2.37
7	177.4	9.47	54.6	3.54
8	63.8	6.34	48.2	6.27
9	309.7	16.8	88.7	6.63
10	122.1	9.24	56.2	7.35

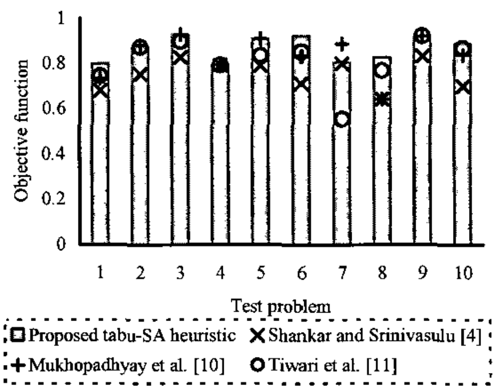

Fig. 3: Performance of the proposed heuristic over the existing approaches

V. Conclusion

The main contribution of this paper is to develop an efficient algorithm for solving the machine loading problem for random FMS. In most of the previous research, job sequencing and operation allocation problems are treated separately. In this paper, attempts have been made to vary the job sequences obtained at every iteration of the algorithm. The objectives of the loading problem considered in this research are minimization of system unbalance and maximization of throughput where the system constraints are maximum available time and tool slots on each machine. Because of the computationally complex nature of underlying loading problem having huge search space to achieve an optimal/near-optimal solution with respect to a set of objective functions and constraints, it becomes essential to use a random search technique.

The proposed algorithm performs well on the test problems. However, application of the proposed solution methodology is restricted to certain cases where there are sufficient number of pallets, fixtures and AGVs available in the shop floor. This research can be further extended by considering few more objective functions namely, minimization of path movements, tool changeovers, setup changeovers, along with measures of flexibilities associated with machines material handling, etc.

REFERENCES

[1] K. E. Stecke, "Formulation and solution of nonlinear integer production planning problem for flexible manufacturing system", *Management Science*, vol. 29, no. 3, pp. 273–288, 1983.

[2] A. Kusiak, "Loading models in flexible manufacturing systems", in *manufacturing Research and Technology, A. Raouf and S. H. Ahmed (eds)*", Amsterdam: Elsevier, 1985.

[3] K. Shanker and Y. J. Tzen, 1985, "A loading and dispatching problem in a random FMS" *International Journal of Prod. Res.*, vol. 23, no. 3, pp. 579–595, 1985.

[4] K. Shanker and A. Srinivasulu, "Some heuristic methodologies for a loading problem in a flexible manufacturing system", *International Journal of Production Research, vol. 27, no. 6, pp. 1019–1034, 1989.*

[5] S. K. Mukhopadhyay, M. K. Tiwari, "Solving machine loading problem of FMS using conjoint measurement", in *Proc. 13th International Conf. on Production research,* Jerusalem, 1995, pp. 74–76.

[6] S. K. Mukhopadhyay, B. Maity and S. Garg, "Heuristic solution to the scheduling problems in flexible manufacturing systems", *International Journal of Production Research*, vol. 29, no. 10, pp. 2003–2024, 1991.

[7] F. T. S. Chan, "Evaluations of operational control rules in scheduling a fexible manufacturing system", *Robotics and Comp. Integrated Manuf.*, vol. 15, pp. 121–132, 1999.

[8] J. C. Ammons, C. B. Lofgren and L. F. McGinnis, "A large scale machine loading problem in flexible assembly", *Annals of Operation Research*, vol. 3, pp. 319–322, 1985.

[9] S. Rajagopalan, "Formulation and heuristic solutions for part grouping and tool loading in flexible manufacturing systems", in *Proc. 2nd ORSA/TIMS Conf. on Flex. Manuf. Syss,* Ann Arbor, MI, 1986, pp. 312–314.

[10] S. K. Mukhopadhyay, S. Midha and V. A. Krishna, "A heuristic procedure for loading problem in flexible manufacturing system", *International Journal of Production Research*, vol. 30, no. 9, pp. 2213–2228, 1992.

[11] M. K. Tiwari, B. Hazarika, N. K. Vidyarthi, P. Jaggi and S. K. Mukhopadhyay, "A heuristic solution approach to the machine loading problem of FMS and its Petri net model", *International Journal of Prod. Res.*, vol. 35, no. 8, pp. 2269–2284, 1997.

[12] A. Gunasekaran, T. Martikainen and P. Yli–Olli, "Flexible manufacturing systems: an investigation for research and applications", *European Journal of Operation* Research, vol. 66, pp. 1–26, 1993

[13] F. Guerreore, S. Lozano, T. Koltai and J. Larraneta, "Machine loading and part type selection in flexible manufacturing systems", *International Journal of Prod. Res.*, vol. 37, no. 6, pp. 1303–1317, 1999.

[14] R. Swarnkar and M. K. Tiwari, "Modeling machine loading problem of FMSs and its solution methodology using a hybrid tabu search and simulated annealing-based heuristic approach", *Robotics and Computer-Integrated Manufacturing*, vol. 20, pp. 199–209, 2004.

[15] Srinivas, M. K. Tiwari, V. Allada, "Solving the machine-loading problem in a flexible manufacturing system using a combinatorial auction-based approach", *International Journal of Production Research*, vol. 42, no. 9, pp. 1879-1893, 2004.

[16] N. K. Vidyarthi and M. K. Tiwari, "Machine loading problem of FMS: a fuzzy-based heuristic approach", *International Journal of Production Research*, vol. 39, no. 5, pp. 953-979, 2001.

[17] N. K. Vidyarthi and M. K. Tiwari, "Solving machine loading problems in a flexible manufacturing system using a genetic algorithm based heuristic approach", *International Journal of Production Research*, vol. 38, no. 14, pp. 3357-3384, 2000.

[18] U. M. B. S. Sarma, S. Kant, R. Rai and M. K. Tiwari, "Modelling the machine loading problem of FMSs and its solution using a tabu-search-based heuristic", *International Journal of Computer Integrated Manufacturing*, vol. 15, no. 4, pp. 285-295, 2002.

[19] F. Kirkpatrick, C. D. Gelatte, M. P. Vecchi, "Optimization by simulated annealing", *Science*, vol. 220, pp. 671-780, 1983.

[20] J. Sridhar and C. Rajendran, "Scheduling in cellular manufacturing system: a simulated annealing approach", *International Journal of Prod. Res.*, vol. 31, no. 12, pp. 2927–2945, 1993.

[21] S. K. Mukhopadhyay, M. K. Singhm R. Srivastava, "FMS loading: a simulated annealing approach", *International Journal of Prod. Res.*, vol. 36, no. 6, pp. 1529–1547, 1998.