A Stochastic Search Approach for Tool Requirement Planning

D. Ganeshwar Rao[1*], C. Patvardhan[2] and Ranjit Singh[3]

ABSTRACT

In an automated manufacturing unit, tool requirement planning is the primary function of tool management. The issues such as tool procurement, tool life inventory, use of alternative tools etc. can be addressed at a level below aggregate planning and above operational implementation. At this level, it is very essential to choose the optimum set of tools out of the alternative tools available for performing operations on various parts in a production period for minimum cost. In the present work, a model has been developed to consider the above mentioned factors and provide the optimum set of tools, out of the alternative tools available, for performing operations on a batch of parts. The performance of the three stochastic search techniques has been studied. It has been observed that the modified simulated annealing techniques are powerful techniques to provide the results in a very less number of evaluations.

Keywords : *Automated manufacturing, multi-tool magazine, stochastic search, tool requirement.*

1. INTRODUCTION

TOOL management is a vital aspect of automated manufacturing machines. There have been many studies on the control components of tool management for these machines but only a few studies on the planning components are there due to the fact that tools were traditionally accorded less importance than the machines and parts. Therefore, these were not considered at the planning stage but treated as last minute measures for the solution of production problems arising at the control stage. Major problems as a result of poor planning of tool availability were observed by Rovito and Hankins [1], Mason [2], Chung [3], Gray et al. [4], and Khator and Leung [5] including high levels of tool inventory, significant system idle time due to lack of tools, unnecessary tool handling, hampered production flow, increased queues and unnecessary tool duplicates. Therefore, the planning considerations of tool management have gained in prominence.

ElMaraghy [6] discussed various tool automation components and identified the need to plan and control tools for usage, availability, replenishment, and so on. A simulation model to understand tool flow problems in FMS was developed by Bell and DeSouza [7]. Other models can be seen in Zeleny [8], and Gaalmann and Nawijin [9]. Planning problems include design of tool related facilities, tooling strategies, tool availability, tool machine interface, and so on. An important element is the planning of tool purchasing and regrinding with respect to the workpiece demand. This has been referred to in the literature as the problem of Tool Requirement Planning (TRP).

Khator and Leung proposed a model to incorporate tool migration policy as well as other issues such as use of alternative tool types, tool failure, regrinding of tools etc. They presented a tool planning model formulated as a linear program. However, in their model, there are 572 decision variables and 224 constraints for a small problem of 4 part types and 5 tools types with 6 operations on each part type. A rule based method for selecting tools out of the alternative tools available, for the manufacture of turned components with optimum economic performance was described in Hiduja and Barrow [10]. Yeong-Dae Kim et al. [11] have considered a tool requirements planning problem in a flexible manufacturing system with an automatic tool transporter. In their study they have determined the

1*. D. Ganeshwar Rao is Associate Professor with the Department of Mechanical Engineering, Faculty of Engineering, Dayalbagh Educational Institute, Dayalbagh, Agra (India) (corresponding author, phone: +91-9557275517; e-mail: dgrao.dei@gmail.com).
2. C. Patvardhan is Professor with the Department of Electrical Engineering, Faculty of Engineering, Dayalbagh Educational Institute, Dayalbagh, Agra (India).
3. Ranjit Singh is Professor with the Department of Mechanical Engineering, Faculty of Engineering, Dayalbagh Educational Institute, Dayalbagh, Agra (India).

number of tool copies of each tool type with the objective of minimizing total tardiness of orders with distinct due dates for a given budget for tool purchase. Martin Noël et al. [12] have focused on the problem of selecting the cutting speeds for processing a set of part types by an unsupervised metal cutting flexible machine in such a situation. Their paper presents models for determining the optimal magazine loading and cutting speeds that will meet a required service level.

Tool requirement planning is the primary function of tool management as other functions such as capacity requirement planning; machine grouping, tool-part grouping, tool placement etc. are based on this function. The issues such as tool procurement, tool life inventory, use of alternative tools etc. can be addressed at an intermediate level, i.e. below aggregate planning and above operational implementation . At this level, it is very essential to choose the optimum set of tools out of the alternative tools available for performing operations on various parts in a production period for minimum cost. Further, on arrival of a batch of parts into the system for manufacturing, it is essential to estimate the number of tools required for its production. A realistic estimation needs to consider the tool changes based on service lives of tools and a control policy regarding the unscheduled changes following the tool failures.

Most of the tool management problems fall under the class of NP-complete problems. This precludes the possibility of finding efficient polynomial time algorithms for them. Stochastic search techniques are the powerful techniques used to yield near optimum solutions at a considerably lower computational effort. It is evident from the literature survey that the relative effectiveness of these techniques has not been studied in solving the tool requirement planning problems. Therefore, an attempt has been made to fill this gap in this paper.

The tool requirement planning problem has been formulated as an optimization problem and the intelligent search techniques viz. Simulated Annealing and Genetic Algorithms have been adapted for selection of the optimum set of tools out of the alternative tools available, to perform operations and to estimate the number of tools of each type required, considering the factors such as cost, useful service lives, regrindings, and unexpected breakages of each tool at intermediate level with an objective to minimize the overall cost..

2. MODEL FOR INTERMEDIATE TOOL REQUIREMENT PLANNING

It is essential to estimate the number of tools required for the production of a set of parts in the given production period (typically a week or a month depending on the tool procurement period). Since most VMCs are capable of accommodating mid-variety and mid-volume of parts, there must be a system which estimates the number of tools required for different part varieties with varying production requirements. These variations, both in variety and requirements, influence the number of tools required for the production period. Further, during machining, the tools are subjected to replacements. These tool replacements also influence the net requirement of tools. Tools incur carrying costs, purchasing costs, regrinding costs and costs of tool failure. A proportional relationship between tool usage and workpiece demand cannot necessarily be assumed to exist for the planning of tool life requirements. This is due to the reasons such as use of alternative tool types, tool failure, reduction in life due to tool regrinding, tool life inventory of two types, i.e. purchased and reground etc..

In the present work, a TRP model has been developed considering the above mentioned points. The model has been validated by computational experience on "realistic" randomly generated data.

2.1 Generation of Experimental Data

The inputs required to randomly generate a tool-operation matrix showing the alternative tools available for various operations are:

(i) The total number of tools available to perform operations on a particular part type and

(ii) The total number of operations to be performed on each part type.

Accordingly, a tool-operation incidence matrix of size m x n is generated for each part type, where m is the number of tools and n the number of operations. The entry aij in the matrix is 1 if tool i is able to process operation j and 0 otherwise. After generating the tool-operation matrix, the algorithm generates the processing times of each operation depending on the material of tool and part and other operation parameters such as feed, depth of cut etc. Also, the tool life of each tool is generated and stored. The tool costs and tool lives have been randomly generated keeping in view the data regarding the actual ranges collected from various industries. Also the tool regrinding cost and tool life after regrinding has been taken as per the data obtained from an industry. All the above generated and collected data is stored in a database.

3. ADAPTATION OF STOCHASTIC SEARCH TECHNIQUES

The problem is to find the optimum set of tools out of the alternative tools available to perform operations on part types with an objective to minimize the overall cost (including tool costs, regrinding costs, operations costs, breakage costs etc.). As the number of alternative tools and operations increase, the computational effort increases exponentially. Hence, the stochastic search techniques have been applied to determine the optimal assignment of tools out of the available tools for various operations, considering the issues identified in the previous section. The following assumptions have been made:

1. All decision variables as well as parameters are deterministic.
2. The number of machines in the system does not affect the tool requirement, and the cost of operation per unit time is same for all machines.
3. There are two distinct categories of tool life inventory: purchased and reground. The system is capable of monitoring both.
4. Reground tools can perform machining operations as efficiently as newly purchased tools. However the possibility of failure is higher.
5. Machines never breakdown.

Several implementation details need to be carefully selected in any adaptation of search techniques for the solution of any combinatorial optimization problem as these implementation details affect the search power of the technique. These are described under the following sections:

(i) Representation of search space
(ii) Method of generating initial solution(s)
(iii) Definition of neighbourhood or crossover
(iv) Calculation of fitness of a string, i.e. fitness function
(v) Cooling schedule
(vi) Stopping criterion

3.1 Representation of Search Space

The search space is represented by a string Tool[i] having j spaces, where j is the total number of operations to be performed on a part type. Tool[i] denotes the tool number selected in the current string for operation i out of the possible alternatives. For example, the string [214341] indicates that for a particular part type, the tool number 2 will process the 1st operation, tool number 1 will perform 2nd and 6th operations, the tool number 4 will perform 3rd and 5th operations and tool number 3 will perform 4th operation. The string is considered to be valid if for each operation, a valid tool is chosen out of possible alternatives.

3.2 Generation of Initial Solutions

Since SAF and SAB are single point search techniques, an initial solution is generated at random ensuring that the string is a valid string. However, in case of GA an initial population of size N is generated randomly. Larger value of N yields a more exhaustive search of the search space with correspondingly greater computational effort. In the present implementation, N=20 for GA has been taken as a reasonable compromise.

3.3 Neighbourhood and Crossover

In case of SAF and SAB, the next solution is generated from the neighbourhood of the current solution. A range of operations is randomly selected and for each of these operations, the tool is randomly changed out of the possible alternatives. In case of Genetic Algorithm adaptation, two common genetic operators - crossover and mutation are used. The crossover operator involves interchanging the elements between two randomly selected points from two parents strings selected randomly. This operator is applied with a probability of 0.5. This operator mainly provides the search capability to GA. Mutation works with a single string leaving the parent intact with the population. It randomly picks and changes the tool allocation for a randomly selected operation in a randomly selected parent. This operator is used with a very low probability of 0.01 [13]. The main effect of this operator is to shift the search to a new neighbourhood thus ensuring that the whole search space is covered over the complete run of the GA.

3.4 Computation of the Fitness Function

Each candidate string in the final population represents the tools allocated for each operation of a part type, out of the alternative tool available. The total cost involved on accepting the string is calculated according to the objective function given below.

The objective is to minimize the sum of purchased inventory carrying cost, reground inventory carrying cost, tool purchase cost, regrinding cost and cost of operation. This sum can be expressed as:

$$Sum = \sum PT\_BRC_k + RT\_BRC_k + PT\_TC_k + RT\_TC_k + PT\_OPC_k + RT\_OPC_k + PT\_TCC_k + RT\_TCC_k$$

3.5 Cooling Schedule

In SAF and SAB, c_i is the control parameter called the cooling parameter in analogy with the physical annealing process in metals. The change in this parameter as iterations proceed is called the cooling schedule. In this study, the cooling schedule originally proposed by Lundy and Mees [14] has been employed:

$$c_{i+1} = c_i / (1 + \beta c_i),\ i = 1, 2, ...N-1.$$

where β is a constant whose value is specified as $\beta = (c_1 - c_N)/(c_1 c_N (N-1))$. c_1 and c_N are the initial and final temperatures.

The selection of the temperature is such that initially the probability of acceptance of a bad move i.e. when the best child is worse than the parent is high but as the temperature is successively lowered, this probability is decreased till at the end when the probability of acceptance of a bad move is almost negligible. It has been shown that the strategy enables the algorithm to seek the global optimum without getting stuck in some local optima [Laarhoven and Aarts, 1987].

3.6 Stopping Criterion

The number of evaluations has been used as the termination criterion in the present heuristics. According to the trial examples, it was observed that the solutions become stable within 1000 evaluations in most of the cases. Therefore 1000 evaluations (i.e. 5 generations in case of GA, as in each generation the number of children evaluated is 200) has been used as the termination criterion. Since the number of evaluations is same for all the three algorithms, the basis of comparison is only the solution quality, i.e. the total cost.

The search heuristics described above can be easily understood from the pseudo codes given in Tables 1 to 3.

Table 1 : Pseudo Code of GA for TRP

Step 1:	*Randomly and independently generate N strings of tools to form the initial population.*
Step 2:	*Determine the overall cost involved in case of each string. These strings may be called parents.*
Step 3:	*Calculate the selection probability for each string of population, where the selection probability is defined as P[i] = COST[i] / TCOST;* *where, P[i] is the probability of sequence i, COST[i] is the overall cost in case of string i, and TCOST is the sum of the costs in case of all the strings.*
Step 4:	*Calculate cumulative probability as per the formula : CP[i] = CP[i-1] + P[i]; where, CP[i] is the cumulative probability of string i and P[i] is the probability of string i.*
Step 5:	*Select those strings for reproduction for which the cumulative probability is greater than ρ (i.e. CP > ρ);* *where, ρ is the random number uniformly distributed between 0 and 1.*
Step 6:	*Select the strings for crossover for which the crossover probability is greater than ρ (i.e. $P_c > \rho$); apply crossover between the pairs of selected strings of population. Replace those parents with the resulting offsprings to form a new population.*
Step 7:	*Select the strings for mutation for which mutation Probability is greater than p (i.e. $P_m > \rho$) and apply mutation operator.*
Step 8:	*Repeat steps 3 to 7 until the stopping criterion is reached.*

Table 2: Pseudo Code of SAF for TRP

Step 1:	*Let i=0. Set temperature at the maximum.* *Randomly generate a string of tools, Str(x). Determine overall cost, x, involved on considering this string.*
Step 2:	*Randomly and independently generate K strings from the neighbourhood of the current sequence Str(x). Determine cost involved in case of K strings in random order. Accept the solution if it provides lower cost than that provided by the current string. If there is no string that improves str(x), then find the best string, str(y) out of the K strings and determine the cost involved y in case of this string.*
Step 3:	*Replace str(x) by str(y) and x by y,* *either if (x > y)* *or if (exp(x-y)t > ρ),* *where, x is the cost involved on considering the current string and y the cost involved on considering the next generated string, t is the temperature coefficient and ρ,a random number uniformly distributed between 0 and 1.*
Step 4:	*Decrease the temperature according to the cooling schedule.*
Step 5:	*If i=N then stop else go to step 2,* *where, N is number of evaluations, used as stopping criterion.*

Table 3: Pseudo Code of SAB for TRP

Step 1:	*Let i=0. Set temperature at the maximum.* *Randomly generate a string of tools, Str(x). Determine overall cost, x, involved on considering this string.*
Step 2:	*Randomly and independently generate K strings from the neighbourhood of Str(x), the current sequence. Determine the overall cost involved in case of each string.* *Let the best string among the generated K strings be Str(y) and the overall cost considering this string be y. Let i=i+K.*
Step 3:	*Replace str(x) by str(y) and x by y,* *either if (x > y)* *or if (exp(x-y)/t > ρ).*
Step 4	*Decrease the temperature according to the cooling schedule.*
Step 5:	*If i=N then stop else go to step 2.*

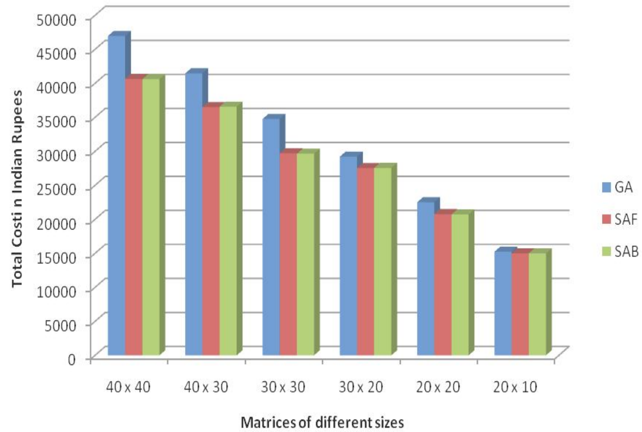

Fig. 1 : Comparison of total cost on different matrices ithree stochastic search techniques

4. COMPUTATIONAL EXPERIENCE

The method of generating "realistic" data has been given in section 2.1. The model and the adaptation of the search techniques given in section 3 have been used to compute the minimum cost for the best assignment of tool to operation out of the alternative tools available for each operation. A set of 6 tool-operation matrices of various sizes ranging from 20 tools and 10 operations to 40 tools and 40 operations has been selected and the results reported in Table 4. For each technique, the result reported is the best obtained in 10 runs to eliminate chance factor. The comparison of the total cost or the solution obtained in each of these matrices is presented in the form of a bar chart in Figure 1. The chart clearly indicates the superior performance of SAF and SAB over GA in terms of the quality of solution obtained for the same number of evaluations.

Table 4: Total cost obtained from the four stochastic search heuristics for tool-operation matrices of different sizes

MATRIX SIZE	TOTAL COST OBTAINED USING FOUR SEARCH TECHNIQUES		
	GA	SAF	SAB
40 x 40	46967	40651	40619
40 x 30	41471	36506	36570
30 x 30	34760	29708	29654
30 x 20	29222	27535	27557
20 x 20	22515	20769	20724
20 x 10	15242	15000	15000

5. CONCLUSIONS

In the present paper, the stochastic search techniques have been adapted for optimum allocation of tools for operations on batches of different part types out of the alternative tools available. The results obtained for tool-operation matrices of various sizes reveal that with large sized matrices, the performance of SAF and SAB heuristics is the best, although in case of all matrices they outperform GA (Figure 1 and 2). Performance of GA may be boosted by implementing some local heuristics. The results indicate that the techniques SAF and SAB provide an effective approach for the optimum allocation of tools for minimizing the overall cost. The convergence graph in Figure 2 shows that the SAF and SAB converge very fast and the lowest value of cost is obtained within 3000 evaluations on a tool-operation matrix of size 40 x 20. GA converges at a higher value. The performance statistics of the three heuristics obtained from 100 runs of a 40 x 20 matrix, presented in the Table 4 also highlights the superiority of SAF and SAB over GA.

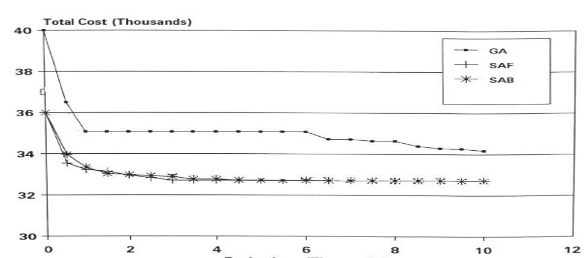

Fig. 2: Convergence graph of the three heuristics for total cost on a tool-operation matrix of size 40x20

NOTATIONS:

The various variables and parameters used in the objective function are as follows:

D_i Demand of part type i.

PT_N_k Number of purchased tool k.

RG_N_k Number of regrindings of tool k.

PT_C_k Cost of purchased tool k (Rs).

RT_C_k Cost of regrinding of tool k (Rs).

CO Unit cost of operation (common for all tools and part types, includes machining cost and inspection cost) (Rs).

PT_CC_k Unit carrying cost for purchased tool k (Rs./week).

RT_CC_k Unit carrying cost for reground tool k (Rs./pd).

PT_OT_k Total operation time taken by purchased tool k.

RT_OT_k Total operation time taken by reground tool k.

The following have been computed on the basis of the above parameters:

PT_BRN_k Number of purchased tool k lost due to catastrophic failure, given by: $PT\_BRN_k = PT\_N_k * 0.05$.

RT_BRN_k Number of reground tool k lost due to catastrophic failure, given by: $RT\_BRN_k = RT\_N_k * 0.15$.

PT_BRC_k Total cost in Rs. for purchasing tool k due to catastrophic failure of purchased tool, given by: $PT\_BRC_k = PT\_BRN_k * PT\_C_k$

RT_BRC_k Total cost in Rs. for purchasing tool k due to catastrophic failure of reground tool, given by: $RT\_BRC_k = RT\_BRN_k * PT\_C_k$

PT_TC_k Total cost in Rs. for purchasing tool k due to expiry of tool life, given by: $PT\_TC_k = PT\_N_k * PT\_C_k$

RT_TC_k Total cost in Rs. for regrinding tools k due to expiry of tool life, given by: $RT\_TC_k = RT\_N_k * RT\_C_k$

PT_OPC_k Total cost in Rs. for operation using purchased tool, given by: $PT\_\_OPC_k = PT\_OT_k * CO$

RT_OPC_k Total cost in Rs. for operation using reground tool, given by: $RT\_OPC_k = RT\_OT_k * CO$

PT_TCC_k Total carrying cost in Rs. for purchased inventory of tool k, given by: $PT\_TCC_k = (PT\_BR_k + PT\_N_k)*PT\_CC_k$

PT_TCC_k Total carrying cost in Rs. for reground inventory of tool k, given by: $RT\_TCC_k = (RT\_BR_k + RT\_N_k)* RT\_CC_k$

6. REFERENCES

[1] V.P. Rovito, and L. Hankins, "The Impact of Tooling in Flexible Manufacturing Systems", Proceedings of the Advanced Manufacturing Systems Exposition and Conference, 1986, pp. 249-271.

[2] F. Mason, "Computerized Cutting Tool Management", American Machinists & Automated Manufacturing, Vol. 130, No. 5, 1986.

[3] C.H. Chung, "Planning Tool Requirements for Flexible Manufacturing Systems", Journal of Manufacturing Systems, Vo. 10, No. 6, 1991, pp. 446-483.

[4] A.E. Gray, A. Seidmann and K.E. Stecke, "A Synthesis of Decision Models for Tool Management in Automated Manufacturing", Management Science, Vol. 39, No. 5, 1993, pp. 549-567

[5] S.K. Khator and I.C. Leung, "Intermediate Tool Requirement Planning for FMS", Journal of Manufacturing Systems, Vo. 13, No. 1, 1993, pp. 9-19.

[6] H.A. ElMaraghy, "Automated Tool Management in Flexible Manufacturing", Journal of Manufacturing Systems, Vol. 4, No. 1, 1985, pp. 1-13.

[7] R. Bell and DeSouza, "The Management of Tool Flows in Highly Automated Flexible Machining Installations", Proceedings of the Second International Conference on Computer Aided Production Engineering, University of Edinburg, 1987, pp. 233-238.

[8] J. Zeleny, "Flexible Manufacturing Systems with Automatic Transport of Tools", Annals of CIRP, Vol. 30, No. 1, 1981, pp. 349-352.

[9] G.J. Gaalmann and L.W. Nawijin, "Tool Sharing in an FMS - A Feasibility Study", Engineering Cost and Production Economics, Vol. 12, 1987, pp. 107-115.

[10] S. Hinduja and G. Barrow, "Sits - A Semi-Intelligent Tool Selection System for Turned Components", Annals of the CIRP, Vol. 42, No. 1, 1993, pp. 535-539.

[11] Yeong-Dae Kim, Geun-Cheol Lee, Seong-Kook Choi and Seung-Kil Lim, "Tool Requirements Planning in a Flexible Manufacturing System with an Automatic Tool Transporter", IEEE Transactions on Robotics and Automation, Vol. 17, No. 6, 2001, pp. 795-804.

[12] Martin Noël, Manbir S. Sodhi and Bernard F. Lamond, "Tool planning for a lights-out machining system", Journal of Manufacturing Systems, Volume 26, Issues 3-4, 2007, pp.161-166.

[13] D.E. Goldberg, "Genetic Algorithms in Search, Optimization and Machine Learning", Addison-Wesley, 1989.

