Contents lists available at GrowingScience

International Journal of Industrial Engineering Computations

homepage: www.GrowingScience.com/ijiec

Part grouping and tool loading in versatile multi-tool machining centers

D. Ganeshwar Rao, C. Patvardhan and **Ranjit Singh**

Department of Mechanical Engineering, Faculty of Engineering, Dayalbagh Educational Institute, Dayalbagh, Agra-282110 (U.P.), India

ARTICLEINFO

Article history:
Received 15 July 2011
Received in revised form
September, 20, 2011
Accepted 22 September 2011
Available online
24 September 2011

Keywords:
Multi-tool machining centre
Tool-Part grouping
Stochastic Search
Simulated Annealing
Genetic algorithm

ABSTRACT

A central problem of tool management in Versatile Multi-tool machining centres is to decide how to batch the parts to be produced and what tools to allocate to the machine in order to maximize utilization of these expensive machines. Various authors have proposed heuristics and/or mathematical models to minimize the batches of parts to be manufactured in a production period. There is no comprehensive study reported to compare the number of actual batches (stoppages) formed with and without processing time considerations. In this paper, the sequential deterministic heuristics (SDHs) are appropriately adapted to include processing time of operations in the formation of groups. The modified heuristics are more realistic in reducing machine stoppages due to tools. Some stochastic search techniques have also been adapted to compute the number of groups. The results are compared with those obtained from SDHs and standard search techniques. The results indicate that the adapted search techniques are powerful approaches for forming optimum number of batches of parts and tools.

1. Introduction

One of the critical factors in economic justification of the high investment versatile multi-tool magazine CNC machines (VMCs) is high utilization. Efficient tool management is very essential to ensure this. The limited capacity of the tool magazine at the machine limits the number of different tools that can be mounted on the tool magazine, simultaneously. Therefore, parts requiring a variety of tools cannot be processed in a single tool setup. An automatic tool changing device enables the machine to change tools very quickly in a few seconds. This fast tool changing capability avoids costly machine stoppages while producing with the tools available in the magazine, and is an essential feature of VMCs designed for high utilization. When it becomes necessary to replace tools in the tool magazine to allow new operations, the machine sometimes has to be shut down while the tools are changed and then the machine may resume production. This replacement is time consuming and may take up to two hours on certain types of machines (Crama & Oerlemans, 1994). The utilization of the VMCs may therefore be considerably boosted by reducing the occurrences of these stoppages. Therefore, a central problem of tool management is to decide how to batch the parts to be produced and what tools to allocate to the machine in order to maximize parts produced in a single setup. This can be seen as a loading problem in a shop consisting of VMCs. Similar problem of

* Corresponding author. Fax.: +0562-2801226
E-mail: dgrao.dei@gmail.com (D. Ganeshwar Rao)

doi: 10.5267/j.ijiec.2011.09.005

allocation of operations and cutting tools to the limited capacity tool magazine has been identified by Stecke (1983) in the context of flexible manufacturing systems. Various authors have proposed heuristics and/or mathematical models to minimize the batches of parts to be manufactured in a production period. Stecke (1983) has defined and mathematically formulated both the grouping problem (how to best partition the machines into groups) and the loading problem, as nonlinear integer programming problems. Related problems in process planning are also studied by Kusiak (1985a, 1985b), Finke and Kusiak (1987) and Bard and Feo (1989). Crama and Oerlemans (1994) have implemented a column generation approach to solve a linear relaxation of the set covering formulation of job grouping problem.

Rajagopalan (1985, 1986) and Tang and Denardo (1988) have observed that partitioning jobs into a minimum number of batches can be seen as packing the jobs into a minimum number of bins with fixed capacity. It follows that the bin packing problem is a special case of job grouping problem, and hence, the latter is NP-hard. This can be seen as a loading problem in a shop consisting of VMCs. Effort has been made to develop efficient heuristics for this.

2. Literature review

Most of the heuristics available in the literature are based on a two-step approach (Rajagopalan, 1985, 1986; Tang & Denardo, 1988). In the first step, a part is picked to be used as a seed. Unless explained otherwise, always a part that requires the highest number of tools is picked. Then a selection rule is used to add jobs to the group until the tool magazine capacity constraint prohibits the addition of any other part to this group (i.e., until a maximal feasible group is obtained). The two-step procedure is repeated until all jobs are assigned to some group. For selecting the next part to be assigned to a group (in Step 2), a number of different rules as given below are considered. For a group S and a part i not belonging to S, let:

t_i = Number of tools required by part i,

b_i = Number of tools required both by part i and by some part already in S.

(1) MIMU Rule (Tang & Denardo, 1988)

The part having the largest number of tools in common with the jobs already in the group is selected. In case of a tie, the part, which requires the smallest number of additional tools is selected. The procedure is called maximal intersection minimal union. (Maximize b_i and in case of a tie minimize t_i)

(2) MI Rule

Only the first part of the MIMU rule is used and the ties are arbitrarily broken (Maximize b_i).

(3) MU Rule

Only the minimal union criterion is used i.e. the part that requires a minimal number additional tools is selected (Minimize t_i - b_i).

(4) Whitney and Gaul Rule (Whitney & Gaul, 1985)

The jobs that require a large number of versatile tools are selected. (Maximize $(b_i+1)/(t_i+1)$).

(5) Rajagopalan Rule (Rajagopalan, 1985)

Each tool receives a weight a_k equal to the number of jobs that requires tool *k* among the jobs that still have to be assigned to any group. Then, the priority of part *i* is calculated by summing the weights a_k of the tools that must be added to the tool magazine in case part *i* is assigned to the group. The part with the largest priority is selected first. For this rule, the first part in each group (seed) is also selected according to the same criterion.

(6) Modified Rajagopalan Rule

Weight a_k for each tool *k* is defined as the number of jobs that requires tool *k* among the jobs already selected in the group. The priority of a part is the sum of the weights of the tools, which are needed for that part. The part with the highest priority is selected.

(7) Marginal Gain Rule (Crama & Oerlemans, 1994)

The addition of a part *i* to a group usually requires that extra tools be loaded in the tool magazine. This new tool configuration may in turn allow the execution of other (not yet selected) jobs. The number of such jobs is denoted by p_i. Part *i* that maximizes p_i is selected.

These seven heuristics are henceforth referred to as sequential deterministic heuristics (SDHs). The SDHs do not consider the stoppages occurring due to the tool wear and its replacement. In most of the cases except for a few (Zhang & Hinduja, 1995; Goswami Mohit et al., 2006), it is assumed that each tool has sufficient tool life for complete processing of all the parts of the batch and no tool changing is required while processing the batch (Bard, 1988; Kwasi, 1994; Crama et al., 1994). However, it is very important to consider processing times of various operations that the parts have to undergo and available tool lives while grouping the jobs, because the life of one or more tools may exceed the warning limit signifying that on further usage the tool may fail. Thus, the machine may have to be stopped for tool change. This particularly happens when the batch size is large and a tool is shared to perform many operations in a batch. Therefore, in methods that do not incorporate consideration of processing times in forming the groups, the actual stoppages may be much more than that for changing the tool setup for a new group. Further, if tools may have to be changed because of tool wear, it will be difficult to know how many copies of each tool type may possibly be required. If tools are allocated to the batch considering all possible tool requirements in the worst case, the inventory cost will become very high. Otherwise, costly machine stoppages may occur.

Bard (1988) has considered the tool wear factor as a secondary aspect in his model. He has pointed out that the importance of this factor becomes magnified as the batch sizes grow. Carrie and Perera (1986) have reported that for the system they investigated, tools had to be changed ten times more frequently for wear than for part variety. Zhang and Hinduja (1995) have considered the frequency of machine stoppages due to tool wear and have discussed the concept of sister tools. They have proposed that identical copies (i.e. sister tools) of heavily used tools should be mounted in the magazine of turret. Aktürk and Özkan (2001) have proposed a multistage algorithm to solve the scheduling problem in a flexible manufacturing system by considering the interrelated sub-problems of processing time control, tool allocation and machining conditions optimization. Denizel (2003) has addressed the parts-grouping problem with an integer programming formulation and developed a lower bounding procedure using Lagrangean decomposition. Goswami et al. (2006) have proposed an integrated approach to solve the sub-problems like tool-part grouping, job allocation on machines and minimization of make-span in a FMS. Mgwatu (2011) has demonstrated the importance of incorporating and solving the machining optimisation problem jointly with part selection and machine loading problems in order to avoid unbalanced workload in the FMS.

3. Problem definition

The above survey reveals that several authors have stressed the need to consider the processing times of operations in forming the batches of parts and tools but there is no comprehensive study reported to compare the number of actual batches (stoppages) formed with and without these considerations.

In this study, the SDHs are appropriately adapted to include processing time consideration in the formation of groups. Extensive computational experience with a large number of randomly generated tool part matrices is presented to quantify the actual difference in the number of groups formed due to these considerations and the reduction in the total stoppages (i.e. due to group change and tool wear). These results indicate that the modified heuristics are more realistic in reducing machine stoppages due to tools.

In recent years, stochastic search techniques have gained prominence for the solution of combinatorial optimization problems. Solving such a problem is to find the best or optimal solution out of a finite or countably infinite number of alternative feasible solutions. The stochastic search techniques have been applied successfully to such problems in many areas and have been proved to provide near optimal solutions in reasonable computational times. In the present research, the techniques genetic algorithms (GA) (Davis & Ritter, 1987; Goldberg, 1989), simulated annealing (SA) (Kirkpatrick et al., 1983), SA with first move (SAF) and SA with best move (SAB) (Ishibuchi et al., 1995) and guided evolutionary simulated annealing (GESA) (Yip and Pao, 1995) have been explored. These techniques provide a general optimization framework. However, several implementation details need to be suitably selected for providing efficient adaptation for a specific problem. Therefore, these techniques have been implemented with the appropriate modifications and their performance on various matrices has been compared with that of the heuristics mentioned above and their modified forms. The stochastic search techniques explicitly optimize the number of groups and hence outperform the other heuristics by providing smaller number of feasible groups. The difference is especially noticeable in large matrices. Another advantage of these techniques is that they are extremely general and can include the considerations of cost also, if desired.

4. Methodology

Given a set of VMCs and the tool part matrix of the parts to be manufactured on these machines, the grouping of the parts could be performed by utilizing the techniques referred above viz. (i) SDHs and (ii) GA, SA, GESA etc. The 'realistic' tool-part matrices have been generated randomly and appropriately, following the method suggested by Crama and Oerlemans (1994) and extensive simulations have been carried out to validate the above mentioned ideas. The real-world instances are more likely to exhibit subsets of 'similar' jobs, characterized by 'similar' tool requirements. The set of random instances has been designed to capture this type of features.

4.1 Generation of realistic matrix

An instance of type (*M, N, C*) where *M* is the number of tools, *N* the number of parts and *C*, the magazine capacity is generated as follows. The number of tools that are required by a part is bounded by two parameters ‘Min’ and ‘Max’ selected, appropriately. The number of parts that would be placed in a group is bounded by two parameters ‘Minjob’ and ‘Maxjob’ selected, appropriately. First, a number N_1 is drawn uniformly between Minjob and Maxjob, and a subset of tools M_1 of size exactly *C*, is randomly chosen. Then, N_1 ‘similar’ jobs are created by making sure that these jobs use only the tools in M_1 (and hence, form a feasible group). When N_1 jobs have been defined, the procedure is repeated to produce N_2, N_3 ….... additional jobs. This process stops after *k* iterations, when almost all columns of the incidence matrix have been generated, specifically, when:

$$\sum_{(i=1)}^{k} \mathrm{N_i} \geq \mathrm{N\text{-}Maxjob}$$

The remaining columns are filled independently to each other.

4.2 Grouping with SDHs including processing time considerations

As described in section 3, it is important to include processing time considerations while forming the tool-part groups. In this section, suitable modifications of the SDHs have been investigated. The computational results of both the original SDHs and the modified ones have been compared. The minimum number of groups formed and the number of stoppages due to both group change and tool wear are presented in each case of the original SDHs. In the modified heuristics, the number of groups (the same as number of stoppages) formed have been presented.

4.2.1 Modification of the SDHs

In each of the heuristics (SDHs), the step 2 is modified as follows. When a part is selected according to the rule provided in the heuristic, the remaining processing times in the various tools used by this part are checked to find out if the warning limit of some tool would be exceeded by the inclusion of the part in this group. If so, another copy of that tool (sister tool) is provided if spare magazine capacity is available. If not, the part is not placed in this group and another part is selected. The modified MIMU heuristic has been presented in the form of pseudo code below. The other heuristics have also been similarly modified.

Pseudo code of Modified MIMU heuristic

```
Step 1:   Let i=1; Count = 0; Number of Parts = P; Capacity of Magazine = C.
Step 2:   Pick a part that requires the largest number of tools. Assign it to Gr[i];
          Store the life of tools remaining after processing this part.
          Count = Count+1.
Step 3:   Select the next part having largest number of tools in common with the part(s) already
          assigned to Gr[i]. In case of a tie, select the part requiring smallest number of additional tools.
Step 4:   Check if the number of tools exceed the capacity of tool magazine on assigning the part to Gr[i].
          If yes then leave this part and go to step 7 else go to step 5.
Step 5:   Check if the tools have sufficient life remaining to process this part.
          If yes then go to step 9 else go to step 6.
Step 6 :  Count the number of toots exceeding their life on processing the part.
          Check if the total number of tools exceeds the magazine capacity on adding these tools (as sister tools) to
          magazine.
          If yes then leave this part and go to step 7 else go to step 9.
Step 7:   Check if all the parts have been checked for assigning to Gr[i].
          If no then go to step 3 else go to step 8.
Step 8:   i=i+1;
          Go to step 2.
Step 9:   Assign the part to Gr[i].
          Store the remaining life of tools after processing this part.
          Count = Count+1.
Step 10:  If Count = P then go to step 11 else go to step 3.
Step 11:  Output the number of groups formed and the assignment of parts to those groups.
```

4.2.2 Computations with standard and modified SDHs

The tool-part matrices of various sizes have been generated as mentioned in section 4.1. Extensive simulations have been carried out on these matrices.

The seven SDHs - MIMU, MI, MU, Whitney and Gaul rule, Rajagopalan rule, Modified Rajagopalan rule and Marginal Gain rule (named in the present work as MIMU, MI, MU, W&G, RG, RGM, and MG respectively) have been used to find the number of groups. The same heuristics have been modified to consider the processing times. The modified heuristics have been named as MIMU_P, MI_P. MU_P, W&G_P, RG_P, RGM_P and MG_P respectively.

Table 1

Number of stoppages using various SDHs with and without considering processing times

MS	NoS	DS	Number of groups without processing time (i.e. with Standard Heuristics)							Number of groups with processing time (i.e. with Modified Heuristics)						
			MIMU	MI	MU	W&G	RG	RGM	MG	MIMU_P	MI _P	MU_P	W&G_P	RG_P	RGM_P	MG_P
40 x 40	20	1	9(9)	10(10)	10(10)	10(10)	13(12)	12(11)	11(10)	9	10	10	10	11	10	10
		2	11(9)	10(9)	10(9)	11(8)	13(11)	13(10)	12(10)	9	9	9	9	10	10	10
		3	10(8)	11(8)	12(8)	11(7)	13(10)	12(10)	11(9)	9	9	9	8	10	9	9
		4	12(8)	12(8)	12(8)	12(8)	13(9)	13(9)	12(9)	9	9	9	9	10	10	10
40 x 30	20	1	8(8)	8(8)	9(8)	8(8)	11(10)	10(9)	9(9)	8	8	9	8	10	9	9
		2	9(8)	7(7)	8(8)	10(10)	12(11)	11(10)	11(10)	8	7	8	10	11	10	10
		3	8(7)	8(7)	7(6)	8(7)	9(8)	9(8)	9(8)	7	7	7	7	9	9	8
		4	10(10)	11(11)	13(12)	10(10)	13(12)	13(12)	13(12)	10	11	12	10	12	12	12
50 x 50	25	1	10(10)	11(11)	10(10)	10(10)	13(13)	11(11)	12(11)	10	11	10	10	13	11	12
		2	10(9)	11(10)	14(11)	12(10)	13(13)	13(12)	13(12)	10	11	11	10	13	12	12
		3	10(8)	15(8)	10(8)	10(9)	13(10)	12(9)	13(9)	9	10	9	9	12	11	11
		4	13(11)	13(11)	12(9)	12(9)	14(13)	15(12)	14(12)	11	11	10	11	13	13	12
60 x 60	30	1	17(8)	16(9)	16(11)	17(9)	16(12)	16(11)	17(11)	11	11	11	10	13	12	13
		2	17(10)	15(10)	19(10)	18(10)	18(13)	19(12)	18(12)	11	11	12	11	13	12	12
		3	14(12)	14(12)	12(11)	12(11)	14(14)	14(13)	14(13)	12	12	12	11	14	13	13
		4	13(11)	13(11)	16(13)	15(11)	16(14)	16(14)	15(12)	12	12	13	12	14	13	13

MS: Matrix size, NoS: Number of Slots, DS: Data Sets

The Table 1 shows the number of actual stoppages due to both group change and tool wear, with actual groups formed in brackets, in case of SDHs. The number of actual stoppages using modified heuristics is also shown. 16 tool-part matrices of 4 different sizes have been used for the comparison.

The results in Table 1 clearly indicate the advantage of the modifications implemented in the SDHs. The number of actual stoppages (due to both group change and tool wear) is reduced in most of the examples. In the others, they are equal.

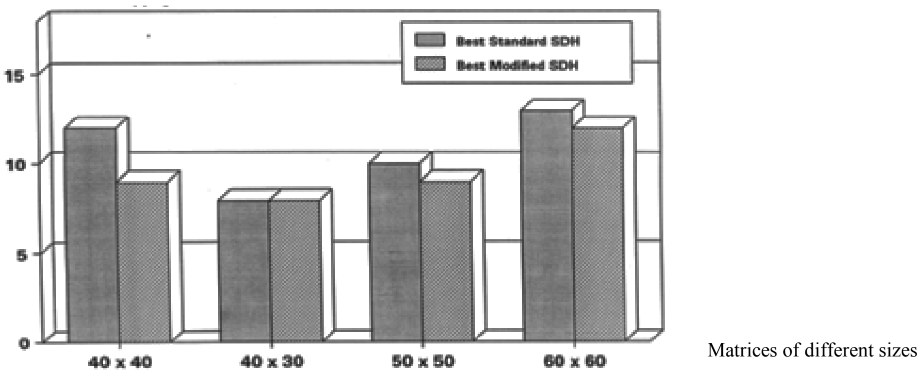


Fig. 1. Comparison of actual stoppages obtained from standard and modified SDHs

4.3 Adaptation of stochastic search techniques for grouping

Several suitable modifications have been made in the stochastic search techniques to obtain efficient adaptations for the problem of tool-part grouping. The details of these implementations are described in the next section. The objective of the present work is not only to solve the job grouping problem but also to study these various stochastic search techniques and their behaviour on this problem.

The inputs required for application of these techniques are as follows:

(i) Total number of tools, parts and the number of slots in the tool magazine,

(ii) The minimum and maximum number of tools to process each part,

(iii) A tool-part incidence matrix of size m x n, where *m* is the number of tools and *n* the number of parts. The entry a, in the matrix is 1 if part *j* is processed by the tool *i* and 0, otherwise,

(iv) The processing times of each operation performed by the tool and tool life of each tool, the objective is to minimize the number of groups or batches of parts taking into consideration the tool life and processing time factors. The number of groups is the fitness or objective function.

The following assumptions have been made:

(1) The allocation of the parts to various machines is already done.

(2) The tool magazine on the machine can accommodate any combination of tools provided number of slots required is not exceeded.

(3) A tool can take one slot or more than one slot depending on its type. In case of multi-slot tool, the partially covered slot cannot partially accommodate another tool.

(4) The parts and the tools required for various operations are known.

Several features have been introduced to enhance the convergence rates of the four heuristics at hand.

The modified implementations including these features are referred to as SAFM, SABM and GESAM instead of SAF, SAB, GESA respectively. The following features included in the adapted implementations are described for all four heuristics studied in this investigation.

(i) Representation of search space

The search space is represented in the form of strings of integers. The string Part[i] is a permutation of integers from 1 to *j*, where *j* is the total number of jobs. The string denotes the order in which each part is assigned to a group. For example, the string [2 5 4 3 1] indicates that the part number 2 is assigned to group number 1, then the part number 5 will be considered for allocating to group number 1 considering the magazine capacity constraints and then the part number 4 will be considered and if capacity constraint is violated, the part will be assigned to the next group i.e. 2nd group. The last part assigned to this group will be part number 1. The string is considered to be "valid" if it contains a permutation of all the part numbers.

(ii) Generation of Initial Solutions

Since SAFM and SABM 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 and GESAM 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*=10 has been taken as a reasonable compromise.

(iii) Neighbourhood and Crossover

In case of SAFM, SABM and GESAM, the next solution is generated from the neighbourhood of the current solution. The shift neighbourhood is defined as the set of strings obtained by removing any part from one position and putting it at another position. The shift operator has been employed by Osman and Potts (1989). Their simulation results indicate that the shift neighbourhood is better than the interchange neighbourhood defined as exchanging two parts. Twist operator can be defined as swapping the parts between two positions.

In the present work, both shift and twist operators have been employed. If the solution is not improved for a certain number of iterations by application of the shift operator, the algorithm is deemed to be stuck in a local minimum. It becomes difficult to come out of it merely by applying shift operator within the limits of the stopping criterion. In such cases, the twist operator is applied to begin search for a solution in a different region.

In case of GA adaptation, two common genetic operators ‘crossover’ and ‘mutation’ are used. A crossover operator involves combining the elements from two parent strings into one or more child strings. Mutation typically works with a single string leaving the parent intact with the population. Goldberg's partially mapped crossover (PMX) operator (Goldberg, 1989) has been used in the present work. The procedure of Goldberg's PMX operator is as follows.

Step1 : Choose an interval, which is a set of consecutive positions in a string, from each of the two parent strings,

Step2 : Determine mappings of the elements in the two selected intervals,

Step 3 : Swap the selected intervals in the two parent strings,

Step 4 : Exchange the mapping elements which are determined in Step 2 and do not lie in the selected intervals in the parent structures.

(iv) Computation of the fitness function

The number of groups for each string is calculated according to the following pseudo code:

```
Step 1 : Set i=1, Part=1,
Step 2 : Assign first part of the string to group $Gr_i$ .
Step 3 : Pick next part of the string. Check if the part can be accommodated without violating the tool magazine
         capacity constraint. If yes, go to step 4 else go to step 8,
Step 4 : Check if any of the tools is exceeding its warning limit on processing this part. If yes, then go to step 5 else go
         to step 6,
Step 5 : See if the capacity constraint is violated on adding one sister tool, If yes then go to step 8 else go to step 6,
Step 6 : Part=Part+1; Assign part to group $Gr_i$,
Step 7 :  If all jobs are assigned to some group then output the number of Groups else go to step 3,
Step 8 :  Part=Part+1; i=i+1;
           Assign part to group $Gr_i$;
          Go to step 3.
```

(v) Cooling Schedule

In SAFM and SABM, *c* 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 (1986) has been employed:

$$c_{i+1} = c_i / (1 + \beta c_i),$$

where $i = 1, 2, ...N-1$ and β is a constant whose value is specified as $\beta = (c_1-c_N)/(c_1c_N(N-1))$; 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, 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 optimum (Laarhoven & Aarts, 1987).

A modification has been implemented in the present work. As the temperature goes down, the possibility of choosing a bad move goes down. If the algorithm is stuck in a local minimum at lower temperature, further improvement in the solution becomes very difficult. In such cases instead of cooling, heating has been applied. The following schedule has been followed for heating:

$$T=T*(1+(I-I\_last)/I\_last),$$

where *T* is the temperature, *I* is the number of iteration and *I_last* is the number of iteration when the last best solution was found. Once the sequence comes out of the local minimum, again cooling is started.

(vi) Stopping criterion

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 after 10,000 evaluations in most of the cases but 30,000 evaluations has been used as termination criterion to avoid any possibility of getting stuck in local minimum.

Table 2
Pseudo Code of GA for Tool-Part Grouping

Step 1:	*Randomly and independently generate N strings of parts to form the initial population,*
Step 2:	*Determine feasible number of groups for these strings,*
Step 3:	*Calculate the selection probability for each string of population, where the selection probability is defined as P[i]=GP[i]/TGP; where P[i]is the probability of string i, GP[i] is the number of groups for string i and TGP is the total number of groups for 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.* $P_{cu}>\rho$*); where, ρ is a 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.* $P_{cr}>\rho$*); apply PMX 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 ρ (i.e.* $P_m>\rho$*) and apply mutation operator,*
Step 8:	*Repeat steps 3 to 7 until the stopping criterion is reached.*

Since the number of evaluations is the same for all the four algorithms, the basis of comparison is only the solution quality i.e. the number of groups formed.

The search heuristics described above can be easily understood from the pseudo codes given in Table 2 to Table 5.

Table 3
Pseudo Code of SAFM for Tool-Part Grouping

Step 1 :	*Let i=0. Set temperature at the maximum,* *Randomly generate a string of parts, Seq(x), Determine number of feasible groups x, for this string,*
Step 2 :	*Randomly and independently generate K part strings from the neighbourhood of the current string Seq(x), Determine number of groups for K strings in random order, Accept the solution if the number of groups is better than that of the current string, If there is no string that improves seq(x), then find the best string, seq(y) out of the K strings and store the number of groups y obtained from this string,*
Step 3:	*Replace seq(x) by seq(y) and x by y, either if(x>y) or if(exp(x-y)/t>ρ), where, x is the number of groups obtained from the current part string and y the number of groups obtained from the next generated part 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 4
Pseudo Code of SABM for Tool-Part Grouping

Step 1:	*Let i=0. Set temperature at the maximum.* *Randomly generate a string of tools for magazine, Seq(x). Determine number of feasible groups (batches of parts) x for this string,*
Step 2:	*Randomly and independently generate K part strings from the neighbourhood of Seq(x), the current string. For each string, determine the number of feasible groups. Let the best part string among the generated K strings be Seq(y) and the number of groups for this string be y. Let i=i+K,*
Step 3:	*Replace seq(x) by seq(y) and x by y, either if(x>y) or if(exp(x-y)/t>ρ), where, x is the number of groups obtained from the current part string and y the number of groups obtained from the next generated part 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,*

5. Computational results with search techniques and SDHs

The tool-part matrices of various sizes have been generated and extensive simulations have been carried out. The techniques MIMU, MI, MU, W&G, RG, RGM, and MG have been used to find the number of stoppages due to both batch change and tool wear. The batches have also been computed using the techniques MIMU_P, MI_P, MU_P, W&G_P, RG__P, RGM_P and MG_P. The standard techniques GA, SAF, SAB, GESA and adapted techniques SAFM, SABM and GESAM have also been used to compute the optimum number of groups in each case. Tables 6, 7 and 8 present the number of groups obtained by these techniques for tool-part matrices of sizes 30x30, 40x40, and 50x50 respectively. These tables show (i) the number of actual stoppages due to both group change and tool wear (with actual groups formed in brackets) in case when processing times are not considered and (ii) the number of actual stoppages considering the processing times, in case of all the methods (i.e. SDHs and stochastic search methods, both standard and modified).

Table 5
Pseudo Code of GESAM for Tool-Part Grouping

```
Step 1:  Set initial temperature at the maximum,
Step 2:  Randomly select N strings of parts. Determine the number of feasible groups (batches of parts) for these strings. Call these strings as parents,
Step 3:  Randomly and independently generate K strings, each from the neighbourhood of these parent strings. Call them children. Determine number of feasible groups for these strings,
Step 4:  Find the best child of each parent. (The child string with minimum number of groups),
Step 5:  Find the parents for the next generation for each family; accept the best child as the parent for next generation if($Y1>Y2$) or ($exp(Y2-Y1)/t>\rho$), where, Y1 is the number of groups for the best child, Y2 the number of groups for its parent; t is the temperature coefficient and ρ a random number uniformly distributed between 0 and 1,
Step 6:  Find the number of children that will be generated from the parents of next generation.
         (second level competition: steps 7 to 12)
Step 7:  Repeat steps 8 to 10 for each family,
Step 8:  Set count=0,
Step 9:  If the number of groups of the child is lower than that of its parent, increase the count by 1, if not then increase the count by 1 if($exp(Y2-Y1)/t >\rho$), where, Y1 is the number of groups for the child and Y2 is the lowest number of groups ever found. Repeat this step for each child,
Step 10: Consider final count as the acceptance number for the family,
Step 11: Sum up the acceptance numbers of all the families,
Step 12: For each family, calculate the number of children to be generated as per the formula: $M=T*A/S$; where, M is the number of children that will be generated for that family; T the total number of points; A the acceptance number and S the sum of all acceptance numbers,
Step 13: Decrease the temperature according to the cooling schedule,
Step 14: Repeat steps 3 to 13 until the number of evaluations as specified is completed.
```

From Tables 6, 7 and 8 it is evident that consideration of processing times always results in fewer or the same actual number of stoppages, particularly as the size of tool-part matrix increases. Therefore the performance of the four stochastic search heuristics is compared with the modified SDHs. Table 9 shows the number of groups obtained by various methods on a number of tool-part matrices of different sizes. The results indicate that SAFM and SABM consistently outperform the seven SDHs and also the other stochastic search heuristics.

Fig. 2 shows the convergence of GA, SAF, SAB and GESA and Fig. 3 shows the convergence of GA, SAFM, SABM and GESAM for a tool-part matrix of size 40 x 40.

The convergence graphs clearly indicate that SAFM and SABM converge almost equally and the lowest value of groups is obtained within 20000 evaluations. Furthermore, it is observed that the minimum number of groups is obtained in case of SAFM, SABM and GESAM. The convergence SAF Vs SAFM, SAB Vs SABM and GESA Vs GESAM for a tool-part matrix of size 40 x 40 has been shown in Fig. 4, Fig. 5 and Fig. 6, respectively. The figures indicate that the convergence of SABM and SAFM is much better as compared to other techniques.

Table 6 Optimum number of groups obtained for a tool-part matrix of size 30 x 30 (for a 15 slots tool magazine) using various techniques

	MIMU	MI	MU	W&G	RG	RGM	MG
Without considering Processing Times	7(7)	8(7)	8(7)	6(6)	9(10)	9(9)	9(9)
	MIMU_P	MI_P	MU_P	W&G_P	RG_P	RGM_P	MG_P
Considering Processing Times	7	7	7	6	10	9	10
	GA	SAF	SAB	GESA	SAFM	SABM	GESAM
Without considering Processing Times	9(9)	6(6)	6(6)	6(6)	6(6)	6(6)	6(6)
Considering Processing Times	8	6	6	6	6	6	6

Table 7
Optimum number of groups obtained for a tool-part matrix of size 40 x 40 (for a 20 slots tool magazine) using various techniques

	MIMU	MI	MU	W&G	RG	RGM	MG
Without considering processing times	10(9)	10(9)	9(8)	9(8)	11(10)	10(10)	10(10)
	MIMU_P	MI_P	MU_P	W&G_P	RG_P	RGM_P	MG_P
Considering processing times	9	9	8	8	10	9	10
	GA	SAF	SAB	GESA	SAFM	SABM	GESAM
Without considering processing times	10(10)	8(8)	8(8)	8(8)	7(8)	7(9)	7(8)
Considering processing times	10	8	8	8	7	7	7

Table 8
Optimum number of groups obtained for a tool-part matrix of size 50 x 50 (for a 25 slots tool magazine) using various techniques

	MIMU	MI	MU	W&G	RG	RGM	MG
Without considering Processing Times	12(9)	12(9)	14(8)	11(8)	13(10)	12(10)	13(10)
	MIMU_P	MI_P	MU_P	W&G_P	RG_P	RGM_P	MG_P
Considering Processing Times	9	10	9	9	12	10	12
	GA	SAF	SAB	GESA	SAFM	SABM	GESAM
Without considering Processing Times	13(11)	12(9)	10(9)	12 (11)	9(8)	11 (9)	9(8)
Considering Processing Times	10	9	9	9	8	9	9

Note: In the tables 6, 7 and 8, the best values obtained out of ten runs have been presented in case of stochastic search heuristics.

Table 9 Optimum number of groups obtained for different tool-part matrices using search techniques considering processing times

Matrix Size	Number of slots in the magazine	Best value obtained out of seven SDHs	Number of groups using stochastic search techniques (Best out of 10 runs)						
			GA	SAF	SAB	GESA	SAFM	SABM	GESAM
20x30	10	5	5	5	5	5	5	5	5
20x20	10	5	5	5	5	5	5	5	5
30x30	15	6	7	6	6	6	6	6	6
30x30	12	8	9	8	8	8	7	7	7
40x40	20	9	11	9	8	9	8	8	9
40x40	25	7	8	7	7	7	7	6	7
50x50	25	9	10	8	9	9	8	8	8
50x50	30	8	9	8	7	8	7	7	7

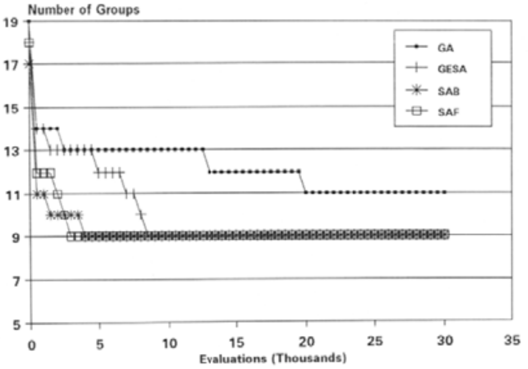


Fig. 2. Convergence of four standard stochastic search techniques for a tool part matrix of size 40 x 40

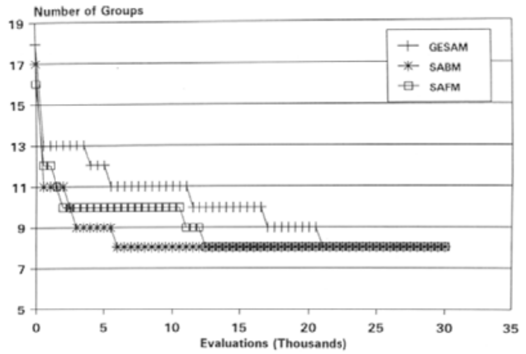
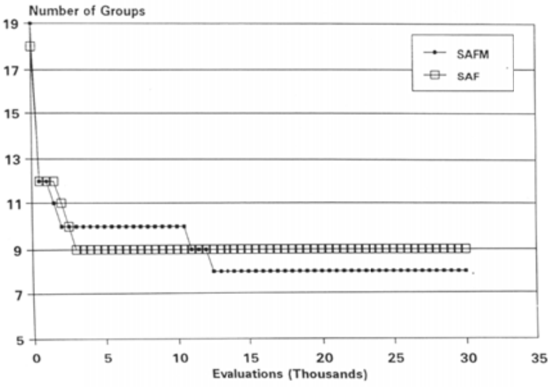


Fig. 3. Convergence of three adapted stochastic search techniques for a tool part matrix of size 40 x 40



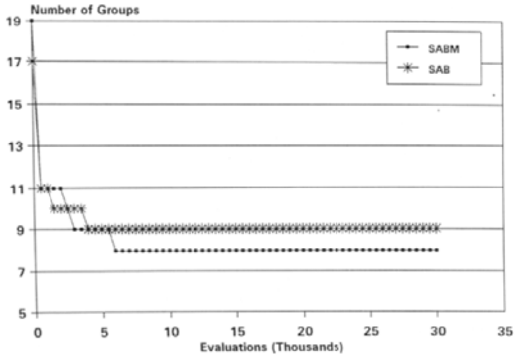


Fig. 4. Convergence of SAF Vs SAFM for a tool part matrix of size 40 x 40

Fig. 5: Convergence of SAB Vs SABM for a tool part matrix of size 40 x 40

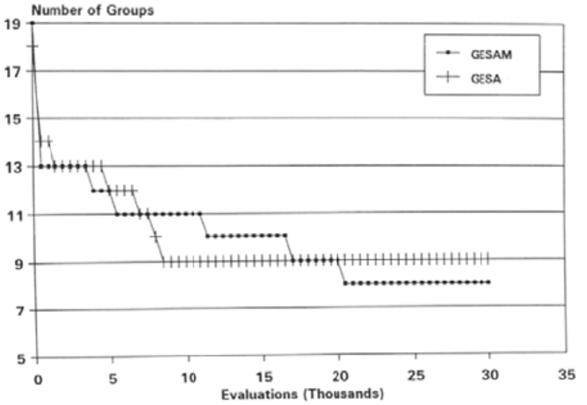


Fig. 6. Convergence of GESA Vs GESAM for a tool part matrix of size 40 x 40

6. Conclusions

In this paper, the number of groups formed for different tool-part matrices using various sequential deterministic heuristics (SDHs) has been computed. The SDHs are appropriately adapted to include processing time consideration in the formation of groups. Extensive computational experience with a large number of randomly generated tool part matrices is presented to quantify the actual difference in the number of groups formed due to these considerations and the reduction in the total stoppages (i.e. due to group change and tool wear). These results indicate that the modified heuristics are more realistic in reducing machine stoppages due to tools.

The stochastic search techniques have been adapted to compute the number of groups. The results are compared with those obtained from SDHs and standard search techniques. The results indicate that the techniques SAFM, SABM and GESAM are powerful approaches for forming optimum number of batches of parts and tools as these heuristics consistently provide much better results. The conclusions can be summarized as follows:

1. Processing times of operations must be considered in tool-part grouping. This is amply demonstrated by the improved performance of the SDHs with the modification to include consideration of processing times.
2. The stochastic search techniques are effective in determining the tool-part groups while minimizing the number of groups formed considering tool magazine capacity constraint and processing times. The number of groups formed has been either equal or less than the number of groups formed by any of the modified SDHs in all the examples tried.

3. The convergence of the stochastic search techniques becomes better and the number of groups obtained are improved by the incorporation of the modifications to obtain SABM, SAFM and GESAM.

References

Bard, J.F., & Feo, T. (1989). The cutting path and tool selection problem in computer aided process planning. *Journal of Manufacturing System*, 8, 17-26.

Bard, J. F. (1988). A heuristic for minimizing the number of tool switches on a flexible machine. *HE Transactions*, 20, 382-391.

Carrie, A.S. and Perera, D.T.S. (1986). Work scheduling in FMS under tool availability constraints. *International Journal of Production Research*, 24, 1299-1308.

Crama, Y., & Oerlemans, A.G. (1994). A column generation approach to job grouping for flexible manufacturing systems. *European Journal of Operational Research*, 78, 58-80.

Crama, Y., Kolen Antoon, W.J., Oerlemans, A.G. and Spieksma Frits, C.R. (1994). Minimizing the number of tool switches on a flexible machine. *International Journal of Flexible Manufacturing Systems*, 6, 33-54.

Davis, L., & Ritter, F. (1987), Schedule optimization with probabilistic search. *Proceedings of 3rd IEEE Conference on Artificial Intelligence*. 231 -236.

Fausett, L. (1994). *Fundamentals of Neural Networks*: Prentice Hall. Englewood. Cliffs. NJ.

Finke, G., & Kusiak, A. (1987). Models for the process planning problem in a flexible manufacturing system. *International Journal of Advanced Manufacturing Technology*, 2, 3-12.

Goldberg, D.E. (1989). *Genetic algorithms in search, Optimization and Machine Learning*: Addison-Wesley.

Goswami, M., Tiwari, M. K., & Mukhopadhyay S. K. (2006). An integrated approach to solve tool-part grouping, job allocation and scheduling problems in a flexible manufacturing System. *International Journal of Advanced Manufacturing Technology*, 35(11-12), 1145-1155.

Ishibuchi, H., Misaki, S., & Tanaka H. (1995). Modified Simulated Annealing Algorithm for the Flow-shop Sequencing Problem. *European Journal of Operational Research*, 81, 388-398.

Kirkpatrick, S., Gelatt, C.D., and Vecchi, M.P. (1983). Optimization by simulated annealing. *Science*, 220, 671-680.

Kusiak, A. (1985a). The part families problem in flexible manufacturing systems. *Annals of Operations Research*, 3, 279-300.

Kusiak, A. (1985b). Planning of flexible manufacturing systems. *Robotica*, 3, 229-232.

Kwasi, A. G. (1994). A comparative study of FMS tool allocation and part type selection approaches for a varying part type mix. *International Journal of Flexible Manufacturing Systems*. 6, 179-207.

Laarhoven, P.J.M., & Aarts, E.H.I. (1987). *Simulated Annealing: Theory and Applications*: Kluwar Academic Publishers.

Lundy, M., & Mees, A. (1986). Convergence of an Annealing Algorithm. *Mathematical Programming*, 34, 111-124.

Denizel, M. (2003). Minimization of the number of tool magazine setups on automated machines: A Lagrangean decomposition approach. *Operations Research*, 51, 309-320.

Mgwatu, M.I. (2011). Integration of part selection, machine loading and machining optimisation decisions for balanced workload in flexible manufacturing system. *International Journal of Industrial Engineering Computations*, 2(4), 913–930.

Osman, I.H., & Potts, C.N. (1989). Simulated annealing for the permutation flow-shop scheduling. *OMEGA*, 17, 551-557.

Rajagopalan, S. (1985). Scheduling problems in flexible manufacturing systems. *Working Paper, Graduate Schools Industrial Administration*, Carnegie-Mellon University, Pittsburgh, P.A.

Rajagopalan, S. (1986). Formulation and heuristic solution for parts grouping and tool loading in flexible manufacturing systems. *Proceedings of the Second ORSA/TIMS Conference on Flexible Manufacturing Systems*, 311-320.

Aktürk, M. S., & Özkan, M (2001). Integrated Scheduling and Tool Management in Flexible Manufacturing. *International Journal of Production Research*, 39, 2697- 2722.

Stecke, K.E. (1983). Formulation and solution of nonlinear integer production planning problems for flexible manufacturing systems. *Management Science*, 29, 273-288.

Tang, C.S., & Denardo, E.V. (1988). Models arising from flexible manufacturing machine, Part II: Minimization of the number of switching instances. *Operations Research*, 36, 778-784.

Whitney, C.K., & Gaul, T.S. (1985). Sequential decision procedures for batching and balancing in FMSs. *Annals of Operations Research*, 3, 301-316.

Yip, P.P.C., & Pao, Y.H. (1995). Combinatorial optimization with use of guided evolutionary simulated annealing. *IEEE Transactions on Neural Networks*, 6, 290-295.

Zhang, J.H., & Hinduja, S. (1995). Determination of the optimum tool set for a given batch of turned components. *Annals of CIRP*, 445-450.