Pergamon International Journal of Machine Tools & Manufacture 40 (2000) 1735–1753

INTERNATIONAL JOURNAL OF
MACHINE TOOLS & MANUFACTURE
DESIGN, RESEARCH AND APPLICATION

Optimization of hole-making operations: a tabu-search approach

Farhad Kolahan, Ming Liang *

Department of Mechanical Engineering, University of Ottawa, Ottawa, Ontario, Canada

Received 10 May 1999; received in revised form 3 March 2000; accepted 17 March 2000

Abstract

This paper reports a tabu-search approach to minimize the total processing cost for hole-making operations. Four issues, namely, tool travel scheduling, tool switch scheduling, tool selection, and machining speed specification have been simultaneously addressed in this study. The total processing cost consists of tooling cost, machining cost, non-productive tool travelling cost, and tool switching cost. This problem has a structure similar to the Travelling Salesman Problem (TSP) and hence is NP-complete. In addition, the problem under consideration is more complex since the cost associated with each operation is both sequence-dependent and position-dependent. To provide an efficient solution procedure, a tabu search approach is used. To improve the search performance two new neighbourhood generation and move selection policies have been proposed and tested. The decisions on the above issues can be made simultaneously based on the output of the proposed algorithm. The results obtained from computational experiments show that the total processing cost can be significantly reduced within a reasonable search time. The effects of some search parameters and diversification strategies on the search performance have also been investigated. © 2000 Elsevier Science Ltd. All rights reserved.

Keywords: Hole-making; Optimization; Tabu search

1. Introduction

Hole-making operations such as drilling, reaming, and tapping account for a large portion of machining processes for many industrial parts such as dies and moulds. For instance, a typical plastic injection mould could have over 100 holes of different diameters, depths, tolerance and surface specifications, representing various tool requirements and a large number of tool switches. Due to the point-to-point machining feature in hole-making, tool travel takes a significant amount

* Corresponding author. Tel.: +1-613-5625800-6269; fax: +1-613-5625177.
E-mail address: liang@eng.uottawa.ca (M. Liang).

Nomenclature

i	tool type index in ascending order according to the tool diameters, i=1,...,I
j,k,l	hole index, j=1,...,J k=1,...,J l=1,...,J
i_j	index for the last tool to be used on hole j
s	sequence index, denoting a specific permutation of operations
a	cost per unit non-productive travelling distance
b	cost per unit tool switch time
C	machining cost per unit time
$\hat{C}_{ij}$	combined tool and machining costs when tool type i is used on hole j
d_i	diameter of tool i
D_j	final size of hole j
e_{ij}	depth of cut when tool type i performing a cutting operation on hole j
f_i	recommended feed rate for tool type i
I_j	set of tools that can be used to drill hole j to its final size
L_j	depth of hole j, including the clearance
N_j	number of tools in I_j
p_{jk}	non-productive travelling distance between hole j and hole k
$q_{ii''j}$	tool switch time between current tool type, i'', and tool i required by hole j
Q_i	cost of tool type i
$t^*_{ij(s,l)}$	optimum processing time of operation ij located in lth position of sequence s
t_{ij}	machining time required by tool i for hole j
T_{ij}	life of tool type i associated with cutting operation on hole j
$T^*_{ij(s,l))}$	optimum life of tool type i associated with cutting operation on hole j in lth position of sequence s
U	total number of possible operations
V_{ij}	cutting speed of tool i associated with an operation on hole j
V^*_{ij}	optimum cutting speed for tool i operating on hole j
W_s	a 0-1 integer variable, W_s =1 if sequence s is selected; 0, otherwise
$x_{ii'i''ljk}$	a 0-1 integer variable, $x_{ii'i''ljk}$=1 if tool i replaces tool i'' to drill hole j which is located in the path between holes l and k and has been drilled by tool i'; 0, otherwise
$G(s)$	total cost associated with operations in sequence s
$Cbest$	the best objective function value found so far
$Mmax$	maximum allowed number of moves for the entire search process
$Nmax$	number of allowed moves in each search phase
M_ctr	a counter used to record total number of moves made so far
N_ctr	a counter used to record number of moves made so far in a search phase
$N(s)$	set of job sequences in the neighbourhood of s
S_best	the best job sequence found so far
T_list	tabu list
T_size	size of tabu list
$s*$	the best sequence in the current neighbourhood
$s**$	the best sequence found in the immediate previous neighbourhood

of time. A considerable amount of the processing time is spent on switching tools and moving the table from one drilling location to another. The survey by Merchant [1] has shown that tool and part movements take on average 70% of the total time in a manufacturing process. In addition, to drill a given hole to its final size, different sets of tools with different cutting speeds may be used which directly influences tooling and machining costs.

For a part with many holes, a particular tool may be required by several holes and also tools of different diameters may be used to drill a single hole to its final size. To reduce tool traverse, it may be suggested that the spindle should not be moved until a hole is completely drilled using several tools of different diameters. This however will lead to excessive tool switches. By the same token, though tool switches can be reduced by completing all operations on all the holes that require the current tool, the travel time will be increased. Furthermore, the amount of tool movement and the number of tool switches will depend on which set of tools are to be used to drill each hole to its final size. The machining cost and tool cost are affected by the selection of tool combination for each hole and the machining speeds. Hence, the proper determination of the operations sequence and the corresponding machining speed used to perform each operation are crucial in reducing the total cost of production.

Surprisingly the above problem has not been addressed directly in literature. The studies on similar problems in punching operations are also scarce. In this direction, Walas and Askin [2] and Chauney et al. [3] proposed heuristic algorithms based on the travelling salesman problem to minimize total tool travel distance in punching operations. Using an artificial intelligence approach, Ssemakula and Rangachar [4] proposed a method to generate an operation sequence applicable to a variety of manufacturing processes. Roychoudhury and Muth [5] examined several heuristic techniques for NC punch press operation sequencing.

Luong and Spedding [6] are among the few who addressed process planning in hole making operations. They developed a generic knowledge based system for process planning and cost estimation in hole making. Based on input data, the proposed system can recommend the appropriate tools, tool sequence, and machining conditions for each individual hole. The manufacturing cost is then calculated based on the recommended process plan. A similar approach was taken by Khoshnevis and Tan [7] to develop rule-base modules for hole making. Their system can be used to provide a process plan found from all possible operation sequences for a given feature. Taiber [8] presented a search procedure to minimize the number of tool changes, non-cutting tool path, machining time, and tool cost for prismatic workpieces. At each iteration, a good solution is found for each of these criteria using different algorithms. A threshold accepting heuristic method is then used to navigate the search through the solution space. His computational experiments showed a 15% cost reduction over 25 min of search time for a workpiece consisting of 48 manufacturing features and 31 tool types. Usher et al. [9] presented an object-oriented approach to generate and rank alternative tool sets for the part process plan. The system is designed as a part of a dynamic process planning system and is able to extract the required data from CAD files.

Nevertheless, there are two main drawbacks in existing literature on this type of the optimization problem. First, in punching operations the main concern is to minimize those costs related to non-cutting tool travel only. It was also assumed that for each operation the cutting speed is fixed and the problems of tool selection and tool assignment do not arise [5]. Second, every feature (hole) is treated separately and global optimization of the overall process plan is considered.

In reality, the planning decisions in hole making operations involve several issues. They are:

(a) tool travel routing; (b) tool switch scheduling; (c) tool-hole grouping; and (d) selection of cutting speed for each tool-hole combination (operation). These issues are interlocked and concurrently contribute to the total processing cost. For these reasons, we suggest that the above four issues be addressed concurrently. The objective is to minimize production cost which consists of machining cost, tool cost, tool travel cost, and tool switch cost.

2. Problem statement

2.1. Problem description

As mentioned earlier, this paper attempts at developing an efficient solution method to determine the best sequence of operations and associated machining speeds in hole-making operations so that the total processing cost is minimized. Generally, a part, e.g. a plastic injection mould, may have many holes of various diameters, surface finishes, tolerances, and possibly different depths. Depending on the hole diameter, tool geometry, and surface quality specifications, a hole may or may not be completed using a single tool. If the diameter of hole is relatively large, a pilot hole may have to be drilled using a tool of smaller diameter and then enlarge to its final size with a larger tool, possibly followed by reaming or tapping when necessary. A hole can be made using alternative *sets* (*or combinations*) of tools. Each set of tools could include a pilot tool, a final tool (a tool with the same diameter as that of the final hole), and one or more intermediate tools. Fig. 1 shows a schematic representation of different set of tools that could be used to drill three holes on a part.

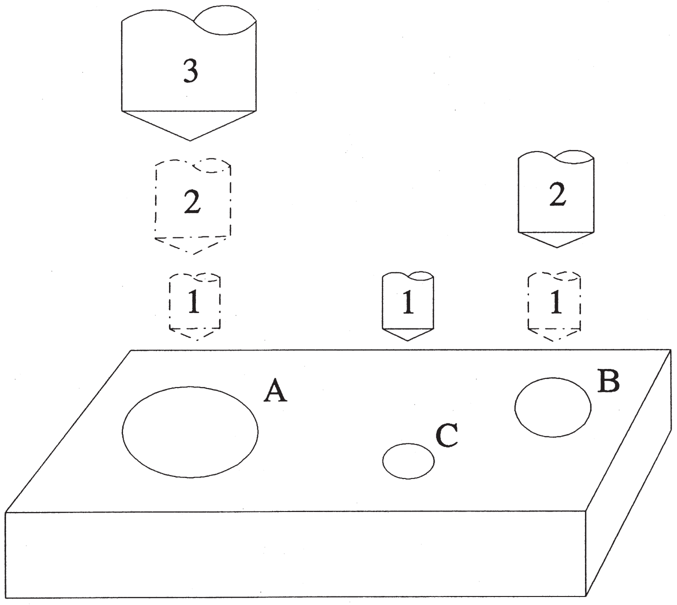

Fig. 1. A schematic representation of alternative sets of tools for hole making.

For each hole in Fig. 1 the largest tool, shown by solid lines, has to be used to drill the hole to its final size. Some pilot or intermediate tools, shown by dashed lines, may also be used. For instance, for hole A, there could be four different sets of tools; {1,2,3}, {2,3}, {1,3}, and {3}. The selection of tool set for each hole directly affects the optimum cutting speeds, required number of tools switches, and tool travel distance.

The problem under consideration is similar to the Travelling Salesman Problem (TSP) in which each node (operation) in a tour (sequence of all operations) must be visited only once. However, unlike the TSP, the cost of visiting each node is determined by its position in the sequence of operations (constructed tour), i.e. certain tools may not be used for some holes even though such operations are feasible (such nodes are visited at no cost). This adds more complexity to the problem as the cost involved in performing a cutting operation in a drilling site is affected by the historical background pertaining to that particular site. That is, the selection of cutting speed and the schedule of tool switches depend upon the depth of cut and hence the current status of the concerned hole which, in turn, is determined by whether or not other tools have already been used on that hole.

The problem is now to select a set of operations along with the optimum cutting speed and sequence those operations in such a way that the total processing cost is minimized. The cost components considered in this paper include:

a) *Tool travel cost*: This is the cost of moving the tool from its previous location to the current drilling position. Tool travel cost is proportional to the distance required for the spindle to move between two consecutive drilling locations, p_{jk}. The cost per unit time is usually lower for non-cutting motions than for the actual cutting processes due to the lower power consumption.

b) *Tool switch cost*: This cost occurs whenever a different tool is used for the next operation. If for operation ij tool type i is not available on the spindle, then the required tool must be loaded on the spindle prior to performing operation ij. In CNC machining each tool is usually mounted on the tool magazine with its own collet. For any tool switch, the tool magazine has to rotate so that the tool changing device can reach the required tool. The tool switch time depends on the location of the required tool on the magazine. If for operation ij the location of replacing tool i on the tool magazine is far from the current tool, i', then the magazine has to rotate a longer distance before the tool switch can take place. This causes a longer tool switch time, $t_{ii'}$, and hence a higher tool switch cost.

c) *Tool and machining costs*: The tool cost consists of the new tool cost and the cost of machine down time required to replace the tool. The operating cost and the machine overhead cost are the major components of machining cost. Both tool and machining costs are affected by machining parameters such as depth of cut, feed rate, and cutting speed. The combined tool and machining costs when tool type i is used on hole j can be written as:

$$\hat{C}_{ij}=\frac{t_{ij}}{T_{ij}}Q_i+Ct_{ij} \tag{1}$$

where the machining time, t_{ij}, is calculated by:

$$t_{ij}=\frac{\pi d_i L_j}{1000V_{ij}f_i} \tag{2}$$

and the general tool life expression, T_{ij}, is as follows [10]:

$$T_{ij}=\left(\frac{A_1 d_i^{A_2}}{V_{ij} e_{ij}^{A_3} f_i^{A_4}}\right)^{A_5} \tag{3}$$

The above expression is an empirical formula in which T_{ij} is the life of tool type i performing a cutting operation on hole j with the speed of V_{ij}, d_i is the tool diameter, and f_i is the recommended feed rate for tool type i which is determined by tool and part materials, required surface finish, and tool geometry. In the case where an existing hole is enlarged by drilling, reaming, or tapping, the depth of cut, e_{ij}, is the difference between the tool radius and the current hole radius. The values of A_1, A_2, A_3, A_4 and A_5 will depend on the type of operation (drilling, reaming, tapping, etc.).

There is a set of machining parameters with which the total tooling and machining costs can be minimized. In drilling operations the depth of cut is fixed. It is also a common practice to keep the feed as a constant rate of cutting speed. Therefore, the optimum cutting speed, V_{ij}^{*}, for the constant feed rate can be found by solving the following differential equation:

$$\frac{d\hat{C}_{ij}}{dV_{ij}}=0 \tag{4}$$

The cutting speed obtained from Eq. (4) minimizes the sum of machining and tool costs for a single operation (single tool-hole combination). However, it can not be used to find total processing cost without knowing which set of operations is to be selected to complete each hole. Therefore, the cutting speeds obtained by solving Eq. (4) are used only as input data to the search algorithm.

Since the problem in hand involves a single part, the planning decision has to be made frequently and quickly. For this purpose, a tabu search approach is developed to solve the combined tool travel scheduling, tool switch scheduling, tool selection, and machining speed specification problem for hole-making. The details are given in the following sections.

3. Problem formulation

To minimize the production cost, the following model can be formulated:

$$MinG(s)=Min\sum_{i\in I_j}\sum_{i'\in I_j}\sum_{i''\in I}\sum_{l=1}^{J}\sum_{j=1}^{J}\sum_{k=1}^{J}x_{ii'i''ljk}\left[a\left(\frac{p_{lj}+p_{jk}}{2}\right)+bq_{ii''j}+\frac{t_{ii'j}}{T_{ii'j}}Q_i+t_{ii'j}C\right] \tag{5}$$

subject to:

$$\sum_{i'\in\bar{I}_j}\sum_{i''\in I}\sum_{l=1}^{J}\sum_{k=1}^{J}x_{i_ji'i''ljk}=1\ \forall j \tag{6}$$

$$x_{ii'i''ljk}+x_{ii'i''kjl}\leq 1\ \forall\{l,j,k,i,i',i''\}l\neq j,k\neq j,i\in I_j,i'\in I_j,i''\in I \tag{7}$$

The objective is to minimize the total cost of processing a job with J holes. The 0-1 decision variables, $x_{ii'i''ljk}$, simultaneously determine the sequence of holes to be processed as well as the sequence of tools to be used to process each hole. It is noted that to find out the tool path at each instance at least three points, i.e. starting, end point and a point in between, should be known. That is the reason why the decision variable includes three indices for three holes l, j, and k with j denoting current position of the spindle. Since the distance between two adjacent drilling sites will be counted twice, one for the current path and the other for the next path, the total tool travel distance in the objective function is divided by two to represent the actual tool travel distance. The indices i, i', and i'' in the 0-1 decision variable are used to determine the proper tool switch order during the operation. Although the tool switch order for the current hole j can be found by the first and third indices, i and i'', it is important to know which tool has performed the immediate previous operation on hole j so that the proper cutting speed and tool life for the current operation can be determined accordingly. This is achieved using notation i'. Constraint set (6) ensures that each hole is drilled to its final size. The last set of constraints states that backward movement of spindle is not allowed unless a tool switch is needed. This prevents redundant spindle movement that may occur when the tool travel cost is negligible.

This model has a large number of 0-1 decision variables. Unfortunately, for medium or large problems, solving this model requires an excessive amount of computational time. To provide an efficient solution procedure, a tabu search approach is presented.

4. Tabu search

4.1. Tabu search definition

Tabu search is an optimization technique used to solve combinatorial optimization problems. This method was introduced by Glover [11]. In a broad sense, tabu search is a high-level iterative procedure that provides a framework for a neighbourhood search to escape from local optima. It involves the exploration of a problem's solution space through the iterative investigation of solution neighbourhoods. The search process starts from a feasible solution and moves stepwise towards a neighbouring solution so that after a number of moves an optimal or near-optimal solution is obtained. To make a move, a set of neighbouring solutions around the current solution is generated and evaluated according to the problem specifications and its constraints. Then, a move is made to the best allowable solution in the neighbourhood. However, unlike other descent search techniques, such a move may not necessarily improve the objective function.

Another important feature of tabu search is the use of tabu list to keep track of solutions selected in the past iterations. Tabu list contains a pre-defined number of previous moves which are not allowed at the current iteration. This can be thought of as a short term memory and is used to avoid returning to the portion of solution space that has already been searched. In addition to the tabu list, many tabu search applications make use of diversification strategies (long term memory) to re-route the search to new regions. These features are designed to prevent cycling and to expand the search area. The basic components of tabu search are explained below.

a) *Starting point*: The search has to commence from an initial feasible solution. This could be any feasible solution, say s, that satisfies problem specifications and its constraints.

b) *Neighbourhood*: For a give solution, s, the neighbourhood $N(s)$ is a set of feasible solutions which is directly generated by performing one transition in the current solution, s, within the feasible range. In the scheduling literature, *pairwise interchange* is probably the most widely used operator to make such a transition. In this method, a solution is obtained by switching the jobs in positions i and j. The complete pairwise interchanges of a J-job problem leads to $|N(s)|=J(J-1)/2$ neighbours.

c) Move: A move is the transition from the best solution, s^{**}, in the previous neighbourhood to the best permissible solution, s^*, that has the lowest $G(s)$ value in the current neighbourhood. The stepwise transition from one solution to another allows the search to reach an optimal or a close-to-optimal solution after a number of moves. However, a single move, by itself, may not necessarily improve the current value of objective function. This distinguishes tabu search from other traditional techniques such as hill climbing that require each move to be an improving step. Throughout the search, the best solution found so far, *Cbest*, and its corresponding sequence, *Sbest,* will be recorded and updated.

d) Tabu list: One of the important features of tabu search is its ability to avoid being trapped in local optima by constructing a list of tabu moves. Tabu list, *T_list*, includes a certain number of previous moves which are not allowed at the current iteration. Once a move from s^{**} to s^* is made, s^{**} is added to the top of tabu list and the oldest member of *T_list* is removed. Thus, returning back to this s^{**} is forbidden for the next *T_size* iterations. This can exclude, to some extent, those moves which lead to possible cycling. The size of tabu list can affect the search performance. Although a longer list may prevent cycling, it requires more computer scanning and may limit the search domain. The best tabu list size appears to be problem dependent and there is no fixed rule to follow in determining tabu-list size so far.

e) Termination criteria: The last element necessary for tabu search is termination criterion. In general, search can be stopped after a certain number of iterations, *Mmax*, is completed, after a pre-defined of computational time, *Tmax,* is reached, or when no improvement can be obtained in a specific number of moves.

The details of this technique are well documented in [11–14].

4.2. Tabu search algorithm for hole making problem

The following definitions are provided to facilitate the development of the proposed algorithm.

Possible operation: A possible operation is defined as the operation performed by tool type i on hole j in which (a) $i \in I_j$ and (b) the tool diameter, d_i, is not greater than the final size of the hole to be drilled, D_j. For a problem involving J holes (drilling sites) and I tool types, the maximum number of feasible tool-hole combinations or *possible operations* is given by:

$$U=\sum_{j=1}^{J}N_j \tag{8}$$

Required operation: Operation ij is called a required operation if (a) it is a possible operation, and (b) no tool with the diameter greater than that of tool type i has been already used for hole j. Hence, for a given operation sequence, s, an operation could be either a possible or a required operation.

For the problem under consideration, a solution is defined as a sequence of all possible operations in which some operations, depending on their position in the sequence, are required ones. The starting sequence should include all possible tool-hole combinations. This will ensure that all possible operations are available for the search, though some of such operations may be redundant in the final solution, i.e. some tools may not be used on some holes. This point is illustrated in the following example. Consider the part shown in Fig. 1 in which three holes, A, B, and C are to be drilled to their final sizes D_A, D_B, and D_C respectively. Let us assume I_A ={1,2,3}, I_B ={1,2}, and I_C ={1} where $d_3 = D_A$, $d_2 = D_B$, $d_1 = D_C$ and $d_3 > d_2 > d_1$. Then one of the possible starting solutions for the search is s_0 ={1C, 1B, 2B, 1A, 2A, 3A}. During the search process some operations may become redundant due to the changed sequence. For instance, after n iterations sequence s_0 may be change to s_n ={1C, 2B, 1B, 1A, 3A, 2A}. In sequence s_n operations 1B and 2A are unnecessary since holes B and C have already been enlarged to the sizes greater than the diameters of tool types 1 and 2 respectively. Therefore only the costs associated to performing operations 1C, 2B, 1A, and 3A contribute to the total processing cost. These are the set of required operations which will be executed on the machine to complete holes A, B, and C to their final size.

The complete description of the proposed tabu-search procedure is presented below:

Step 1. Initialize the search

1. Read the input data D_j, d_i, Q_i, $p_{jj'}$, $q_{ii'}$, for j=1,..., J, i=1,..., I, and I_j,C, a, b
2. Read the search parameters, *T_size*, *Mmax*, *Nmax*
3. Calculate total number of possible operations, k, and set *T_list*={∅}, *S_best*={∅}, *M_ctr* =0, *N_ctr*=0, and *C_best*=*M_big*
4. Find s^{**}, a feasible starting sequence including all possible operations, i.e. a sequence with U "required" operations. Compute $G(s^{**})$

Step 2. Search

```
WHILE $M\_ctr < Mmax$ DO
set $G(s^*) = M\_big$;
 DO $mm$ =1 to $U$-1
  DO $nn = mm+1$ to $U$
  generate a new neighbour $s$ for $s^{**}$ by swapping the operation in position $mm$ with the
  operation in position $nn$.
    IF $s \in T\_list$, discard $s$, and continue;
    ELSE compute the sum of processing costs for all "required" operations in $s$, $G(s)$.
      IF $G(s) < G(s^*)$, set $G(s^*) \rightarrow G(s)$ and $s^* \rightarrow s$;
      ELSE discard s, and continue;
      ENDIF
    ENDIF
   ENDDO
  ENDDO
```

```
Step 2.1 Move
  Set $G(s^{**}) \leftarrow G(s^*)$, $s^{**} \leftarrow s^*$ and update T_list;
    IF $G(s^{**}) < Cbest$, set $Cbest \leftarrow G(s^{**})$, Sbest $\leftarrow s^{**}$, $M\_ctr \leftarrow M\_ctr$ +1, and $N\_ctr \leftarrow$
    $N\_ctr$ +1;
    ELSE set $M\_ctr \leftarrow M\_ctr$ +1, $N\_ctr \leftarrow N\_ctr$ +1;
    ENDIF

Step 2.2 Diversify
  .
    IF $N\_ctr \geq Nmax$, set $T\_list = \{\emptyset\}$, $N\_ctr \leftarrow 0$, and diversify the search path using one of
    the following strategies:
    (1) Restart from S_best, the best sequence found so far.
    (2) Restart from a randomly selected sequence.
    ELSE continue;
    ENDIF
  ENDWHILE
Stop
```

The diversification policies are optional and may be used to enhance the search performance.

5. Case study

The proposed algorithm was applied to determine the set of tools, sequence of operations, and cutting speeds for the upper base of a plastic injection mould shown in Fig. 2. This figure includes the data about the distances between the holes, type of operations required, and depth of each hole.

To machine the holes on this part, three types of operations, drilling, reaming, and tapping are required. The tool life expressions for these operations are as follows [10]:

$$T_{ij}=\left(\frac{8d_i^{0.4}}{V_{ij}f_i^{0.7}}\right)^5 \text{ for drilling a new hole} \tag{9}$$

$$T_{ij}=\left(\frac{18.4d_i^{0.4}}{V_{ij}e_{ij}^{0.2}f_i^{0.5}}\right)^5 \text{ for enlarging a hole by drilling} \tag{10}$$

$$T_{ij}=\left(\frac{12.1d_i^{0.3}}{V_{ij}e_{ij}^{0.2}f_i^{0.65}}\right)^{2.5} \text{ for enlarging a hole by reaming or tapping} \tag{11}$$

The following optimum cutting speeds can be obtained respectively by solving differential Eq. (4) with above tool life equations:

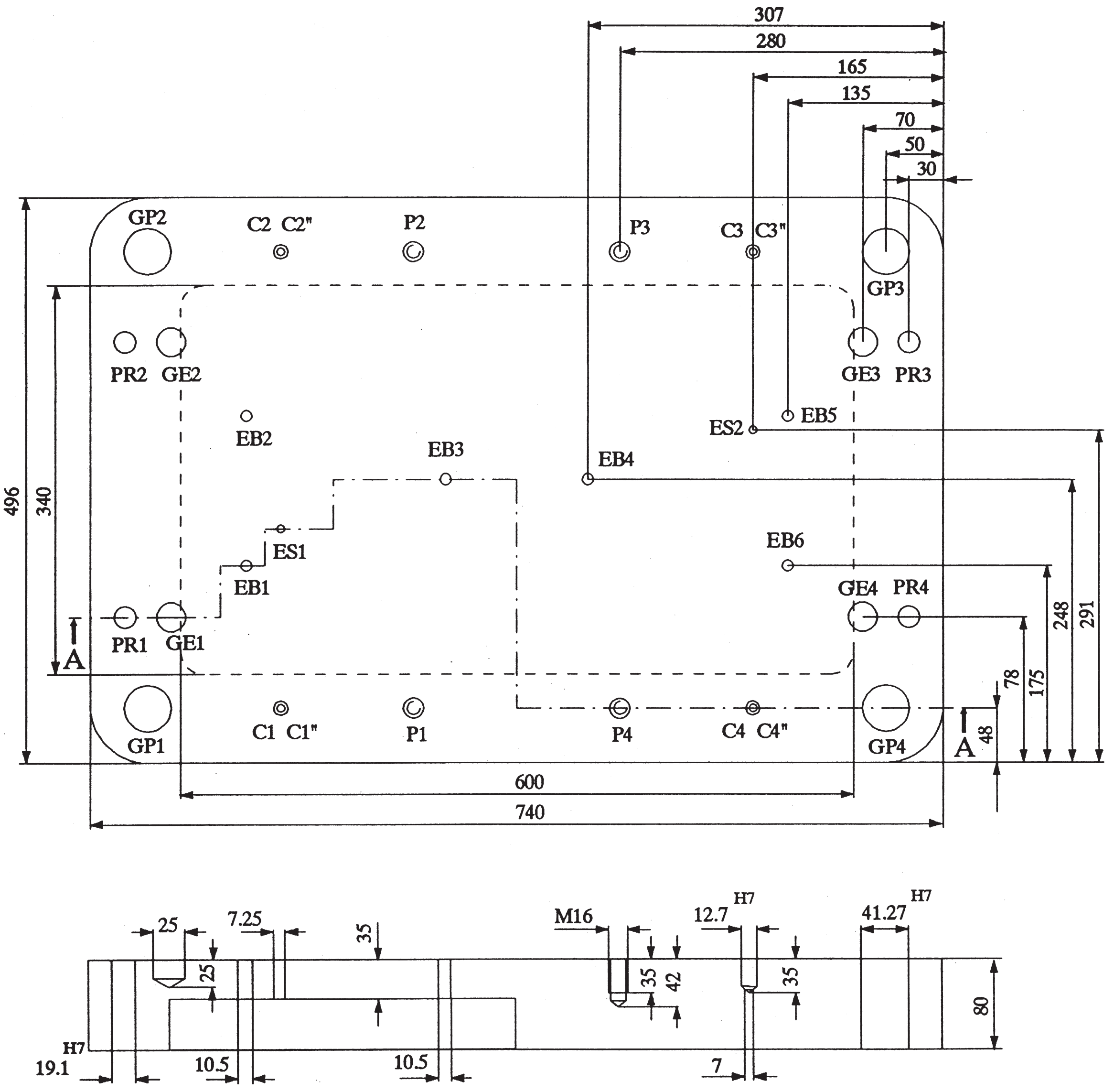

Fig. 2. Upper holder of the plastic injection mould (courtesy of Komtech Plastics Corp.).

$$V_{ij}=6\sqrt[5]{\frac{Cd_i^2}{Q_i f_i^{3.5}}} \tag{12}$$

$$V_{ij}=13.9\sqrt[5]{\frac{Cd_i^2}{Q_i e_{ij} f_i^{2.5}}} \tag{13}$$

Table 1
Tool diameter, cost, and specified feed rate data

Tool type i	Drill								Reamer			Tap
	1	2	3	4	5	6	7	8	9	10	11	12
f_i (mm/rev)	0.12	0.10	0.12	0.15	0.20	0.20	0.18	0.15	0.50	0.80	0.80	1.50
d_i (mm)	7.0	7.25	10.5	12.5	13.0	19.0	25.0	41.0	12.7	19.1	41.2	16.0
Q_i ($)	10	12	15	15	14	20	26	50	55	70	85	45

$$V_{ij}=10.3\sqrt[2.5]{\frac{Cd_i^{0.75}}{Q_i e_{ij}^{0.5} f_i^{1.65}}} \tag{14}$$

During the search, these speeds are used to calculate the tool and machining costs for each operation. The information about tools is given in Table 1. Table 2 shows the tool switch times which are asymmetric and dependent on the location of tools on the tool magazine.

Since each hole can be made using alternative tool sets, a feasible set of tool-hole combinations should be known to construct the initial sequence of possible operations. Such a set of feasible operations used in this example is listed in Table 3.

The remaining process parameters assumed for this problem are: C=\$1/min, a=\$0.0008/mm, and b=\$1/min.

The data given in Table 3 lead to 80 ($U=\Sigma_j N_j$) possible operations. The size of neighbourhood

Table 2
Tool switch times (min)

Successor tool	Predecessor tool											
	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.6	0.2	0.4	0.4	0.9	0.6	1.0	0.8	1.4	0.4	1.0
2	0.6	0.0	0.8	1.2	0.4	0.8	1.2	0.5	0.6	0.6	1.2	1.4
3	0.2	0.8	0.0	0.6	1.4	1.2	1.4	1.0	0.4	1.0	1.9	1.2
4	0.4	1.2	0.6	0.0	0.4	0.7	1.0	1.6	0.8	1.4	0.4	0.6
5	0.4	0.4	1.3	0.5	0.0	0.8	0.6	1.0	0.8	1.8	1.6	1.5
6	0.5	0.5	1.2	0.2	0.8	0.0	0.2	1.5	1.2	1.3	0.4	1.9
7	0.6	1.2	1.4	1.0	0.6	0.4	0.0	1.0	0.8	0.8	1.2	0.6
8	1.0	0.6	0.6	0.4	0.4	0.7	0.6	0.0	0.8	1.5	0.4	0.5
9	0.2	0.9	1.1	0.6	1.5	1.2	1.4	1.0	0.0	1.2	2.0	1.2
10	0.4	1.2	0.7	1.5	0.5	0.3	1.0	0.4	0.7	0.0	0.4	0.6
11	0.4	1.2	2.0	0.4	1.2	0.8	0.6	1.2	0.8	1.0	0.0	0.2
12	1.0	0.7	0.3	0.4	0.4	0.8	0.6	1.4	1.0	1.5	0.5	0.0

Table 3
Possible tool-hole combinations used in the example problem

j	GP1-GP4	GE1-GE4	PR1-PR4	C1-C4	C'1-C'4	P1-P4	EB1-EB6	ES1-ES2
I_j	1-6-7-8-11	3-6-7	3-6-10	1-4-9	1	1-5-12	3	2

is $U(U-1)/2=3160$ for this problem. These neighbouring solutions must be evaluated before a move can be made.

A set of computations was carried out on 10 different starting solutions, i.e. 10 different initial sequences. For each initial sequence, the search was terminated after 100 moves. To investigate the effects of diversification strategies and tabu-list size on search performance, the following search strategies were used.

Diversification strategy	Phase length (*Nmax*)	Tabu-list size (*T-size)*
(1) Restart from *S_best*	20	10,15
	50	10,20,40
(2) Restart from a random sequence	20	10,15
	50	10,20,40
(3) No diversification	100	20,40,50

The purpose of diversification strategies is to further reduce the chance of cycling and to enhance search performance by increasing the chance of resuming the search from a good starting solution or increasing the search domain by clearing the tabu list. In the first strategy, tabu-list is cleared after a pre-determined number of moves has been completed and search resumes from the best solution found so far. The same principle applies for the second diversification strategy with the exception that the restarting point is selected randomly. In the third strategy, there is no diversification and all moves are performed within a single phase. It is noted that tabu-list size should not exceed the phase size since otherwise all moves become tabu. For example the longest tabu-list for a 20-move phase is 15 and the longest tabu list for 50-move phase is 40 and so on.

The computational experiments showed that in most cases diversification strategies 1 and 3 result in the same final solutions and are superior to strategy 2. The results for the best combination of starting sequence, tabu-list size and diversification strategy are summarized in Table 4.

The final solution consists of 56 required operations (reduced from 80 possible operations given in initial solution). The total processing cost for the final solution is \$64.8 including \$45.2 machining and tool costs, \$11.0 non-productive travelling cost, and \$8.6 tool switch cost. This result indicates a 47% cost reduction over the initial solution. The operation sequence, tool switch sequence, cutting speed, and tool sets can all be determined simultaneously based on the output shown in Table 4. For example, tool type 6 is used first to drill 12 holes (4 GPs, 4 GEs, and 4 PRs) with the cutting speed of 34 m/min. The tool is then switched to tool type 8 to enlarge 4 GP holes to 41 mm in diameter. The same holes are then reamed to their final sizes (41.2 mm) using tool type 11 and so on.

Table 4
Operation sequence and associated cutting speeds

ij	6-GP4	6-PR4	6-GE4	6-GE3	6-PR3	6-GP3	6-GP2	6-PR2	6-GE2	6-GE1	6-PR1	6-GP1
V^*	34	34	34	34	34	34	34	34	34	34	34	34
ij	8-GP1	8-GP4	8-GP3	8-GP2	11-GP2	11-GP1	11-GP4	11-GP3	10-PR3	10-PR2	10-PR1	4-C1
V^*	45	45	45	45	9	9	9	9	10	10	10	36
ij	1-C1’	1-C4’	3-EB6	1-C3’	1-C2’	3-EB2	3-EB1	3-EB3	3-EB4	3-EB5	2-ES2	2-ES1
V^*	37	37	39	37	37	39	39	39	39	39	40	40
ij	7-GE1	7-GE2	7-GE3	7-GE4	10-PR4	4-C4	4-C3	4-C2	5-P2	5-P3	5-P4	5-P1
V^*	50	50	50	50	10	36	36	36	30	30	30	30
ij	12-P1	12-P2	12-P3	12-P4	9-C4	9-C3	9-C2	9-C1				
V^*	4	4	4	4	11	11	11	11				

6. Discussion

6.1. Solution

The computational results revealed the following findings that could be useful for day to day shop floor planning decisions:

a) *There is a trade off between tool travel cost and tool switch cost*. The solution found to the example problem neither minimizes the tool switch cost nor the tool travel cost. In fact, the tool switch cost in the final solution is 18% more than the one for the same problem when the objective function does not include non-productive travelling cost; i.e. a=0. Also the length of tool travelling is almost four times higher than the case when there is no tool switch cost; i.e. b=0. In the same manner, none of the above scenarios will minimize the total processing cost. It is also found that the relative significance of tool travelling cost and tool switch cost results in notable changes in the final sequence of operations. This is illustrated in Table 5 which corresponds to the best sequence found for the example problem when the tool switch times in Table 2 are reduced by 50%.

This sequence has a total processing cost of $60.2 from which $45.2 is the tool cost and machining cost, $10.1 tool switch cost, and $4.9 tool travel cost. A comparison between the

Table 5
Operation sequence and associated cutting speeds for the example problem when tool switch costs are reduced by 50%

ij	3-EB3	3-EB4	3-EB6	3-EB5	4-C3	6-GP3	6-PR3	6-GE3	7-GE3	1-C3’	8-GP3	2-ES2
V^*	39	39	39	39	36	34	34	34	50	37	45	40
ij	5-P2	5-P3	12-P3	12-P2	1-C2’	4-C2	6-GE2	7-GE2	6-PR2	6-GP2	8-GP2	11-GP2
V^*	31	31	4	4	37	36	34	50	34	34	45	9
ij	10-PR2	9-C2	3-EB2	3-EB1	6-GE1	7-GE1	6-PR1	6-GP1	8-GP1	11-GP1	10-PR1	4-C1
V^*	10	11	39	39	34	50	34	34	45	9	10	36
ij	1-C1’	9-C1	2-ES1	5-P4	5-P1	12-P1	12-P4	1-C4’	4-C4	6-GE4	6-PR4	6-GP4
V^*	37	11	40	30	30	4	4	37	36	34	34	34
ij	9-C4	8-GP4	7-GE4	11-GP4	10-PR4	10-PR3	11-GP3	9-C3				
V^*	11	45	10	9	50	50	10	11				

results in Tables 4 and 5 indicates that, when tool switch cost is reduced, the search attempts to reduce the tool travel cost even though the number of tool switches may increase.

b) *Tools selection and spare provision can be simultaneously determined based on computational results*. With reference to Table 3, although several tools may be available to drill a hole to its final size, only a few of them will be actually used (see Table 4). For instance, there is no need to use tool types 1 and 7 on holes GP1-GP4 though they are provided in the initial solution. Furthermore, the work load of some tools may exceed the life of those tools and therefore extra copies are needed. This is the case for tool type 6 which has been used as the pilot drill for 12 holes.

6.2. Search performance

1) *Accuracy*: The accuracy of the solutions obtained using the proposed search method was verified by comparing with optimal solutions. In doing so, five sets of 10-operation problems were generated and solved by enumerating all possible (10!=3 628 800 each) solutions. These problems were also solved using the proposed algorithm. The search was terminated after 1.0 min for every problem. The comparison showed that the average error is less than 1.7%. This is sufficiently accurate for practical planning purpose.

2) *Effects of diversification strategies and tabu-list size*: Throughout the computational experiments two aspects of diversification policies, namely number of phases and starting sequences, were examined. In addition, each strategy was tested with different tabu-list sizes to investigate the effect of tabu-list size. It should be mentioned that the total number of moves was fixed for all problems for fair comparison. It is found that a tabu-list with a size less than or equal to 15 may lead to cycling while tabu lists with sizes 20 to 40 perform equally well in terms of solution quality. For the majority of starting solutions, diversification strategies 1 and 3 yield similar results whereas strategy 2 appears to be the least efficient one. In summary, the search is more sensitive to the initial solution than the diversification strategy.

7. Improvement of the tabu search algorithm

Generally, computational time is an important issue in solving the day to day shop floor planning problems. The study carried out in this paper involves solving a number of planning problems for a single part and hence it becomes even more important to obtain a good solution in a short period of search time.

In the proposed tabu search algorithm, the neighbourhood size is the product of two parameters: the number of holes and the number of available tools. As the value of either parameter increases, the number of tool-hole combinations, and consequently the neighbourhood size, grows very rapidly. Consequently, the search time for large size problems may be too long for real applications. In this section, it is attempted to improve the search performance by reducing the computational time without sacrificing the solution quality. This can be achieved through adapting different neighbourhood generation mechanisms and move selection policies. The details are given in the following.

As mentioned earlier, in the basic tabu search all candidate moves are generated by the pairwise interchange mechanism and then a move is made toward the best non-tabu solution in the neighbourhood. For a drilling process with U possible operations, the lower and upper limits of transposition range are 1 and U-1 respectively. This approach is sometimes called steepest descending. The adjacent pairwise interchange, or fastest descending method, is another neighbourhood generation scheme in which only the adjacent jobs are switched to generate the candidate moves. Since only the adjacent jobs are exchanged, the transposition range for this method is equal to 1 resulting in a neighbourhood of size U-1 candidate moves. Although adjacent pairwise interchange does not guarantee a best solution at each iteration, its neighbourhood size is much smaller than that of pairwise interchange and hence the search can progress much faster in the solution space. In the following, two approaches will be introduced which combines the above policies in order to improve the search performance.

Partial Neighbourhood (PN): This approach is proposed to achieve a compromise between the quality of candidate moves given by pairwise interchange and the search speed resulting from adjacent pairwise interchange mechanism. The objective is to find a transposition range that provides the best combination of these parameters. Therefore, the upper boundary of transposition range, RA, can be set between 1 and U-1 which is fixed for the entire search process. The best range is then selected based on the computational results for a given search time. Usually, the best transposition range (and hence the neighbourhood size) is between those of pairwise and adjacent pairwise interchange mechanisms.

Dynamic Neighbourhood (DN): This method aims to take advantage of both adjacent pairwise interchange and simple pairwise interchange mechanisms by dynamically changing the transposition range in different stages of the search process. The idea is to guide the search as fast as possible toward the unsearched areas where the optimum solution may be located and then find such a solution by complete evaluation of the entire neighbourhood. To achieve this, the adjacent pairwise interchange is used to speed up the search process at the early stages. As the search progresses, the neighbourhood size is enlarged by increasing the transposition range which, in turn, improves the solution quality at each iteration. Finally, in the later iterations, the pairwise interchange mechanism is employed to generate the neighbourhood. This will maximize the possibility of finding the best solution in the current neighbourhood. To this end, we propose the following linear function to determine the transposition range at each iteration.

$$RA(n)=\begin{cases}1 & \text{if } Z\leq 1\\ Z & \text{if } 1<Z<U-1\\ U-1 & \text{if } Z\geq U-1\end{cases} \tag{15}$$

where

$$Z=A+Bn \tag{16}$$

In the above expression, $RA(n)$ is the upper limit of transposition range at the nth iteration, and A and B are the constant values determined experimentally. It should be mentioned that the above methods are based on the size of generated neighbourhood (not the location of the exchanged operations) which can be controlled by changing either the upper or the lower boundary. Therefore, in Eq. (15) only the upper bound of transposition range is determined and its lower boundary can be set to 1 throughout the search.

For comparison, the computations were repeated for 10 min of search time for each of the three neighbourhood generation and move selection policies, namely, pairwise interchange (PI), partial neighbourhood (PN), and dynamic neighbourhood (DN). The following values are found to be best suited for our computations: *RA =U/2* for PN and A=4, B=1.5 for DN. Fig. 3 illustrates the convergence curves for different neighbourhood generation and move selection mechanisms.

As illustrated in Fig. 3, all the three curves tend to converge to the same solution in the long run. However, the DN method converges much faster towards the final solution and most of the cost reduction is achieved in the first 5 min of search time. This has a significant impact on the search performance specially for the large size problems or when there is a limited search time available. The main advantage of the proposed PN and DN methods is that they can greatly reduce the computational time with little tuning.

The example problem in the previous case study and comparison contains 80 possible operations. To examine the proposed approach for larger problems and to compare the three methods in a broader range, an additional 40 test problems have been generated with 10 problems for each of the four problem sizes: *U* =50, 100, 150, and 200 possible operations. These problems were then solved using each of the three neighbourhood generation mechanisms, namely PI, PN and DN. The search was performed in a single phase for 10 min for the 50-operation and 100-operation

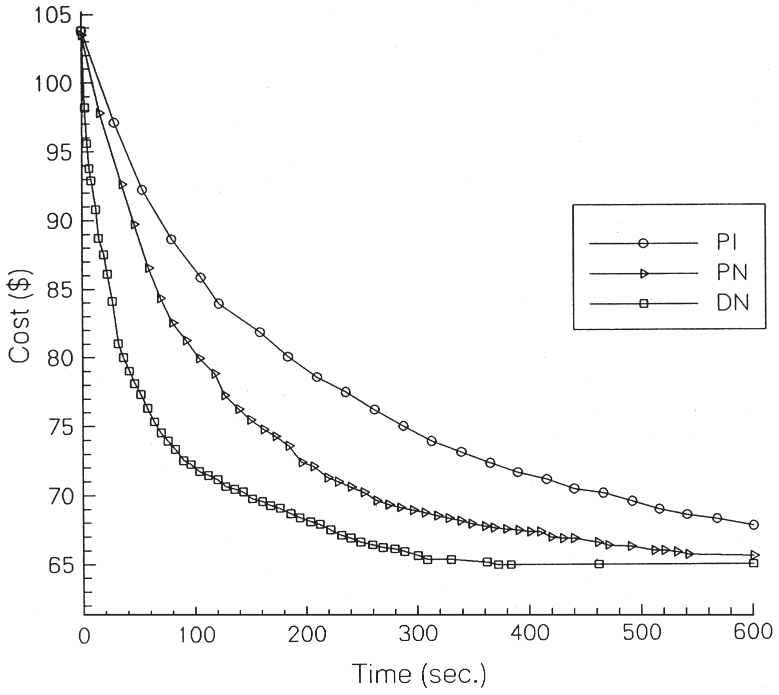

Fig. 3. Convergence curves for different neighbourhood mechanisms.

problems and 15 min for the problems with 150 and 200 operations. For fair comparisons, the same starting points were used for all the runs. Also, only the results obtained using the best tabu-list size are presented and compared. The average cost reductions for each problem set are listed in Table 6.

As shown in Table 6, all three methods provide comparable results for small to medium sized problems, i.e. U =50, 100. However, as the problem size grows there is a significant gap among the final cost reductions produced by different neighbourhood generation mechanisms. For instance, after 15 min of search time the PI method gives an average of 32.2% cost reduction for the problem set 3 (U =150) while for the same amount of search time the PN method results in 53% improvement over the same initial solutions. The DN method provides even better results with about 64% improvement, almost twice as that of PI. This trend becomes even more evident for the problems with a larger number of tool-hole combinations, e.g., U =200, where the best performer, DN, gives a 64.8% improvement compared to only 20% and 32% cost reductions provided by the PI and PN methods respectively.

In summary, as indicated by the above computations, it is sufficient to use regular tabu search with PI method for small size problems. However, it may be beneficial to employ more sophisticated neighbourhood generation mechanisms such as DN for the large size problems since they can greatly improve the final solutions within a given run time. Therefore, the limited parameter tuning needed for such mechanisms can well be justified.

8. Conclusion

In this paper, a tabu-search approach has been proposed to minimize the total processing cost in hole making operations. The objectives considered in formulating the problem stated in this research are different from those in literature and the associated model gives a more realistic presentation of the planning problems in hole making. The proposed tabu search is quite straightforward and easy to implement. A PC and a C compiler will suffice for computations. The input data may also be obtained from manufacturing database if available. Our computational results show that the total production cost can be significantly reduced within a reasonable search time.

Table 6
Comparison of different neighbourhood generation methods

Problem set[a]	No. of possible operations U	Run time (min.)	Average cost reduction (%)[c]		
			PI	PN	DN
1	50	10	31.6[b]	31.2	31.4
2	100	10	48.4	52.9	54.3
3	150	15	32.2	53.0	63.9
4	200	15	20.0	32.0	68.4

[a] Each set contains 10 randomly generated problems.
[b] Average of 10 randomly generated problems.
[c] Cost reduction = (initial cost − final cost)/initial cost.

To further improve the search performance, two partial neighbourhood search schemes were proposed. It is shown that, particularly for the large size problems, these methods can considerably reduce the computational time.

References

[1] R.L. Merchant, World trends and prospects in manufacturing technology, International Journal for Vehicle Design 6 (1985) 121–138.

[2] R.A. Walas, R.G. Askin, An algorithm for NC turret punch press tool location and hit sequencing, IIE Transactions 6 (1984) 280–287.

[3] F. Chauney, R.R. Loulou, E. Wagneur, Sequencing punch operations in an FMS: a three-dimensional space filling curve approach, INFOR 25 (1987) 26–45.

[4] M.E. Ssemakula, R.M. Rangachar, The prospects of process sequence optimization in CAPP systems, Computers and Industrial Engineering 16 (1989) 161–170.

[5] B. Roychoudhury, J. Muth, Tool path optimization procedures for machine tools, International Journal for Computers and industrial Engineering 28 (1995) 367–377.

[6] L.H.S. Luong, T. Spedding, An integrated system for process planning and cost estimation in hole making, International Journal of Manufacturing Technology 10 (1995) 411–415.

[7] B. Khoshnevis, W. Tan, Automated process planning for hole making, Manufacturing Review 8 (2) (1995) 106–113.

[8] J.G. Taiber, Optimization of process sequencing considering prismatic workpieces, Advances in Engineering Software 25 (1996) 41–59.

[9] M. Usher, J. Fernandes, S. Dhage, G. Sharma, Object-oriented implementation of tool selection for dynamic process planning. Proceedings of IIE Industrial Engineering Research Conference, Norcross, GA, USA, 1997, pp. 895–900.

[10] R. Zhao, Handbook for Machinists, Shanghai Science and Technology Press, China, 1992.

[11] F. Glover, Tabu search — Part I, ORSA Journal of computing 1 (3) (1989) 190–206.

[12] F. Glover, Tabu search — Part II, ORSA Journal of computing 2 (1) (1990) 4–32.

[13] F. Glover, Tabu search: a tutorial, Interfaces 20 (1990) 74–94.

[14] A. Hertz, D. De Werra, The tabu search metaheurestic: how we used it, Annual Mathematics in Artificial Intelligence 1 (1991) 111–121.