

A METHOD FOR GENERATING RANDOM CUTTING-TOOL REQUIREMENT MATRICES FOR MANUFACTURING SYSTEMS SIMULATION

Dharmaraj Veeramani
Moshe M. Barash

School of Industrial Engineering
Purdue University
West Lafayette, Indiana 47907

James R. Wilson

Department of Industrial Engineering
North Carolina State University
Raleigh, North Carolina 27695

ABSTRACT

In detailed shop-floor simulation applications, data concerning the cutting-tool requirements of the parts being machined in the manufacturing system are often necessary to drive the simulation model. While there are merits associated with the use of real data, such data are sometimes not readily available, and it is therefore desirable to have a method for generating hypothetical, yet realistic, tool requirement data. In this paper, we describe a two-stage procedure for generating random tool requirement matrices (part type vs. tool type 0-1 incidence matrices) on the basis of "expert" opinions about the characteristics of such data. In the first stage, we sample row-sums and column-sums of the tool requirement matrix from two user-specified multivariate normal distributions subject to the feasibility condition that the row- and column-sums yield the same grand total. In the second stage, we determine the binary values of the matrix elements by means of a heuristic. The resultant matrix reflects the desired characteristics, such as part type similarities and tool usage dependencies.

1 INTRODUCTION

The management of cutting-tools in advanced manufacturing systems, particularly Flexible Manufacturing Systems (FMSs), is becoming an increasingly studied area in both academia and industry. Simulation is often used to evaluate the impact of tool management strategies on the performance of manufacturing systems. In such simulation applications, information concerning the tool requirements for the family of parts being machined in the manufacturing system is necessary to drive the simulation model.

One approach for providing such input to the simulation model is to sample from an empirical dis-

tribution which has been constructed using data collected from existing process plans. However, this approach has several drawbacks as noted by Law and Kelton (1991). First, it is often very difficult and time consuming to collect complete and accurate data on tool requirements for each operation and each part which is machined in the manufacturing system. Second, this approach cannot be applied when data are unavailable (for example, in the case of new/projected systems). Third, the data that are collected constrain the range of the distribution. Random variables which fall outside this range will never be generated. These shortcomings of using empirical distributions become very relevant when the user desires to consider alternative (often speculative) scenarios for evaluating tool management strategies.

The alternative approach to modeling the tool requirement data is to sample from theoretical distributions that display the characteristics of "real" data. In this paper, we describe a two-stage procedure for generating tool requirement matrices based on the latter approach. In the first stage of the procedure, we sample row-sums and column-sums from two user-specified multivariate normal distributions so that the sum of the row-sums equals that of the column-sums. In the second stage, we determine the binary values of the matrix elements by means of a heuristic that takes into consideration both inter-row and inter-column correlations. This facility to generate "realistic" tool requirement matrices provides the user with a great deal of flexibility and ease in developing tooling related input models for manufacturing system simulation applications.

In the next section, we examine some of the characteristics of cutting-tool requirement matrices that illustrate why the generation of such matrices is not trivial. In §3 and §4, we describe the two stages of our procedure. The paper concludes with a discussion in §5.

2 CHARACTERISTICS OF CUTTING-TOOL REQUIREMENT DATA

We are primarily interested in generating 0-1 matrices having p rows and t columns in which each column corresponds to a particular type of cutting tool and each row corresponds to a particular part type. (For example, one of the columns might represent a 1/4" jobbers length, general purpose, high speed steel, straight shank, twist drill.) If part type i requires tool type j for processing, then element x_{ij} of the matrix has value 1; otherwise, x_{ij} equals 0. We denote the row-sums R_i , where $R_i = \sum_{j=1}^t x_{ij}$ for $i = 1, 2, \dots, p$. Similarly, the column-sums are denoted by C_j , where $C_j = \sum_{i=1}^p x_{ij}$ for $j = 1, 2, \dots, t$. Clearly, $\{C_j\}$ and $\{R_i\}$ are integers; and we must have $\sum_j C_j = \sum_i R_i$ for a feasible assignment of tool types to part types.

Matrices of this type often mirror salient features of the manufacturing system. For example, it is common for some part types to have processing requirements similar to those of other part types. Such similarity in processing requirements is often used as a basis for formation of part families. In our context, this implies that parts belonging to a family have a high degree of tool commonality (that is, they require a similar number of tools of similar types). This implies that the rows of the matrix are correlated. If R_a and R_b denote the row-sums corresponding to row a and row b , then similarity between these two part types (a and b) would be reflected in a high degree of correlation between R_a and R_b .

Another feature of cutting-tool requirement matrices relates to dependencies among the usage of tools of different types. For example, in order to produce a threaded hole, it is necessary (in the simplest case) to first drill the hole and then tap it. Let columns c and d denote a drill and a tap (of corresponding size) respectively. For the part types under consideration, if it is known that the holes of the dimension produced by the drill are usually threaded, then $x_{ic} = 1$ would imply that it is very likely that $x_{id} = 1$. Extrapolating this argument, the number of times, C_c , that the tap is used is strongly related to C_d , the number of times that the drill is used. Therefore, there exists a correlation between the columns of the matrix; and this is reflected in the correlation structure between the column-sums C_c and C_d .

Similarities between processing requirements for related parts and dependencies between usages of different tools appear as "clusters" in the tool requirement matrix. The presence of tool clusters in real tool requirement matrices is common (see Steckle (1989)

for instance) and corresponds to the partitioning of the parts into part families.

To summarize, the scheme for generating a tool requirement matrix should yield random row-sums (that is, tool counts for each part type) whose correlations accurately represent similarities (probabilistic dependencies) between part types; simultaneously, the generation scheme should yield random column-sums (that is, part counts for each tool type) whose correlations accurately represent similarities (dependencies) between tool usages. These considerations form the basis for the matrix generation scheme described in the next two sections

3 GENERATING ROW- AND COLUMN-SUMS

In the first stage of the procedure for generating a tool requirement matrix, the objective is to generate a $p \times 1$ random vector $\underline{R}$ of row-sums and a $t \times 1$ random vector $\underline{C}$ of column-sums

$$\underline{R} = [R_1, \dots, R_p]' \quad \text{and} \quad \underline{C} = [C_1, \dots, C_t]'$$

We assume that $\underline{R}$ (respectively, $\underline{C}$) is normally distributed with user-specified mean vector $\underline{\mu}_R$ (respectively, $\underline{\mu}_C$) and covariance matrix $\underline{\Sigma}_R$ (respectively, $\underline{\Sigma}_C$). We must sample $\underline{R}$ and $\underline{C}$ subject to the feasibility condition $\sum_{j=1}^t C_j = \sum_{i=1}^p R_i$. We write $\underline{R} \sim N_p(\underline{\mu}_R, \underline{\Sigma}_R)$ and $\underline{C} \sim N_t(\underline{\mu}_C, \underline{\Sigma}_C)$; and we let T denote the componentwise grand total for each of these vectors.

3.1 Preliminaries

Our approach to the problem of generating $\underline{R}$ and $\underline{C}$ is based on a linear transformation

$$\underline{R}^* = \underline{H}\underline{R}, \quad (1)$$

where $\underline{H}$ is any nonsingular $p \times p$ matrix whose last row consists of 1's. For simplicity, we will consider

$$\underline{H} = \begin{bmatrix} \underline{I}_{p-1} & \underline{0}_{p-1} \\ \underline{1}'_{p-1} & 1 \end{bmatrix} \Rightarrow \underline{H}^{-1} = \begin{bmatrix} \underline{I}_{p-1} & \underline{0}_{p-1} \\ -\underline{1}'_{p-1} & 1 \end{bmatrix}. \quad (2)$$

First we seek to determine the distribution of $\underline{R}^*$ in terms of the analogous properties of $\underline{R}$.

To clarify the development, we introduce some additional notation that emphasizes the dependence of the random vectors under consideration on their respective dimensionalities. Let us rewrite $\underline{R}$ as

$$\underline{R}(p) = [\underline{R}'(p-1), R_p]'$$

where $\underline{R}'(p-1) = [R_1, R_2, \dots, R_{p-1}]$. Similarly we define corresponding mean vectors

$$\underline{\mu}_{R(p)} \equiv E[\underline{R}(p)] = [\underline{\mu}'_{R(p-1)}, \mu_p]'$$

where $\underline{\mu}'_{R(p-1)} \equiv E[\underline{R}'(p-1)] = [\mu_1, \mu_2, \dots, \mu_{p-1}]$. We also define the corresponding covariance matrices

$$\underline{\Sigma}_{R(p)} \equiv \text{cov}[\underline{R}(p)] = \begin{bmatrix} \underline{\Sigma}_{R(p-1)} & \underline{\gamma}_p \\ \underline{\gamma}'_p & \sigma_{p,p} \end{bmatrix},$$

where $\underline{\gamma}_p = [\sigma_{1,p}, \sigma_{2,p}, \dots, \sigma_{p-1,p}]'$ and

$$\underline{\Sigma}_{R(p-1)} \equiv \text{cov}[\underline{R}(p-1)] = \begin{bmatrix} \sigma_{1,1} & \sigma_{1,2} & \dots & \sigma_{1,p-1} \\ \sigma_{2,1} & \sigma_{2,2} & \dots & \sigma_{2,p-1} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{p-1,1} & \sigma_{p-1,2} & \dots & \sigma_{p-1,p-1} \end{bmatrix}.$$

For the transformed random vector (1), we define the analogous quantities $\underline{R}^*(p) = \|\underline{R}_i^*\|$, $\underline{\mu}_{R^*(p)} = \|\underline{\mu}_i^*\|$, $\underline{\Sigma}_{R^*(p)} = \|\sigma_{h,i}^*\|$, and $\underline{\gamma}_p^* = [\sigma_{1,p}^*, \sigma_{2,p}^*, \dots, \sigma_{p-1,p}^*]'$.

We derive the relevant properties of $\underline{R}^*(p)$ in terms of those of $\underline{R}(p)$. In view of Equations (1) and (2),

$$\underline{R}^*(p) = \underline{H}\underline{R}(p) = \left[\underline{R}'(p-1), \sum_{i=1}^p R_i \right]' \quad (3)$$

is p -variate normal so that its distribution is completely determined by its mean vector and covariance matrix. Recall that $T \equiv \sum_{j=1}^t C_j = \sum_{i=1}^p R_i$. Taking expectations in Equation (3), we have

$$\underline{\mu}_{R^*(p)} = \underline{H}\underline{\mu}_{R(p)} = [\underline{\mu}'_{R(p-1)}, \mu_T]' \quad (4)$$

where $\mu_T \equiv E[T] = \sum_{i=1}^p \mu_i$. Computing the covariance matrix corresponding to Equation (3), we have

$$\begin{aligned} \underline{\Sigma}_{R^*(p)} &= \underline{H}\underline{\Sigma}_{R(p)}\underline{H}' \\ &= \begin{bmatrix} \underline{\Sigma}_{R(p-1)} & \underline{\Sigma}_{R(p-1)}\underline{1}_{p-1} + \underline{\gamma}_p \\ \underline{1}'_{p-1}\underline{\Sigma}_{R(p-1)} + \underline{\gamma}'_p & \sigma_T^2 \end{bmatrix}, \end{aligned} \quad (5)$$

where $\sigma_T^2 \equiv \text{var}(T) = \sum_{h=1}^p \sum_{i=1}^p \sigma_{h,i}$ and

$$\underline{\Sigma}_{R(p-1)}\underline{1}_{p-1} + \underline{\gamma}_p = \left[\sum_{i=1}^p \sigma_{1,i}, \sum_{i=1}^p \sigma_{2,i}, \dots, \sum_{i=1}^p \sigma_{p-1,i} \right]'$$

We make the key observation that given a fixed value of $R_p^* = T$, the conditional distribution of

$\underline{R}^*(p-1)$ is $(p-1)$ -variate normal with mean

$$E[\underline{R}^*(p-1) | R_p^* = T] = \underline{\mu}_{R^*(p-1)} + \underline{\gamma}_p^*(T - \mu_p^*)/\sigma_{p,p}^*; \quad (6)$$

see pp. 135–136 of Johnson and Wichern (1982). We can rewrite Equation (6) elementwise in terms of the original quantities $\{\mu_i\}$ and $\{\sigma_{h,i}\}$ as follows:

$$E[R_i^* | R_p^* = T] = \mu_i + \left(\sum_{m=1}^p \frac{\sigma_{i,m}}{\sigma_T^2} \right) (T - \mu_T)$$

for $i = 1, 2, \dots, p-1$. In addition, the conditional covariance matrix of $\underline{R}^*(p-1)$, given $R_p^* = T$, is

$$\text{cov}[\underline{R}^*(p-1) | R_p^* = T] = \underline{\Sigma}_{R^*(p-1)} - \underline{\gamma}_p^*(\underline{\gamma}_p^*)' / \sigma_{p,p}^*. \quad (7)$$

In terms of the original covariances $\{\sigma_{h,i}\}$, we can rewrite Equation (7) as

$$\text{cov}[R_h^*, R_i^* | R_p^* = T] = \sigma_{h,i} - \frac{1}{\sigma_T^2} \left(\sum_{m=1}^p \sigma_{h,m} \right) \left(\sum_{m=1}^p \sigma_{i,m} \right)$$

for $h, i = 1, 2, \dots, p-1$.

3.2 Algorithm for generating row- and column-sums

To generate each row-sum R_i ($i = 1, 2, \dots, p$) and each column-sum (C_j for $j = 1, 2, \dots, t$), we perform the following steps.

1. Generate $\underline{C}(t) \sim N(\underline{\mu}_C, \underline{\Sigma}_C)$.
2. Round off the column-sums $\{C_j : j = 1, \dots, t\}$ to the nearest nonnegative integer and compute $T = \sum_{j=1}^t C_j$.
3. Given T , generate $\underline{R}^*(p-1)$ from a $(p-1)$ -variate normal distribution whose mean vector and covariance matrix are given by Equations (6) and (7) respectively.
4. Round off the variates $\{R_i^* : i = 1, \dots, p-1\}$ to the nearest nonnegative integer values and recover the desired row-sums from Equation (2) as follows:

$$R_i = R_i^*, \quad i = 1, \dots, p-1; \quad R_p = T - \sum_{i=1}^{p-1} R_i^*.$$

4 GENERATING THE TOOL REQUIREMENT MATRIX

In the second stage of the procedure, our objective is to assign 0's and 1's to the tool requirement matrix $\|\text{matrix}(i,j)\|$ so that (a) $\text{matrix}(i,j)=1$ when part type i requires tool type j ; and (b) the row-sums and column-sums of $\|\text{matrix}(i,j)\|$ respectively match the values of $\{R_i\}$ and $\{C_j\}$ that were generated by the procedure given in §3.2. Ideally, the user-specified

correlation $\text{corr}(C_j, C_k)$ between C_j and C_k should also be the correlation between the j th and k th elements of each row of the tool requirement matrix; similarly, $\text{corr}(R_h, R_i)$ should be the correlation between the h th and i th elements of each column of the tool requirement matrix.

4.1 The Assignment Heuristic

The following heuristic begins by assigning the target row-sums and column-sums to the corresponding rows and columns respectively. On each iteration, a row, **ibestrow**, is chosen. For this row, a column, **ibestcol**, is selected, and the element **matrix(ibestrow, ibestcol)** is assigned a 1; and the corresponding row- and column-sums, R_{ibestrow} and C_{ibestcol} , are each decremented by 1. If the resultant R_{ibestrow} is positive, then another column is assigned to **ibestcol**. This procedure of assigning 1's to the elements of row **ibestrow** in the tool requirement matrix continues until R_{ibestrow} has been decremented to zero. Then, the next iteration begins with the assignment of another row to **ibestrow**. The procedure terminates when all the row-sums have been reduced to zero. In the algorithmic statement given below, "s.t." is an abbreviation for the phrase "such that" and $\#(E)$ denotes the number of elements in the set E .

Input: The sampled row-sums and column-sums and the user-specified inter-row-sum and inter-column-sum correlations.

Output: The tool requirement matrix $\|\text{matrix}(i, j)\|$.

Step (0): [Initialize.]

Initialize $\{R_i\}$ and $\{C_j\}$ to the sampled row- and column-sums.

Set **matrix**(i, j) $\leftarrow 0$ for $i = 1, \dots, p; j = 1, \dots, t$.

Step (1): [Select current row and column.]

ibestrow $\leftarrow i$ s.t. $R_i = \max \{R_k \mid k = 1, 2, \dots, p\}$

ibestcol $\leftarrow j$ s.t. $C_j = \max \{C_m \mid m = 1, 2, \dots, t\}$

matrix(ibestrow, ibestcol) $\leftarrow 1$

$R_{\text{ibestrow}} \leftarrow R_{\text{ibestrow}} - 1$

$C_{\text{ibestcol}} \leftarrow C_{\text{ibestcol}} - 1$

If ($R_{\text{ibestrow}} > 0$) then

 go to Step (2)

Else

 go to Step (3)

Endif

Step (2): [Select next column for current row.]

If (there is **nextcol** s.t. $\text{corr}(C_{\text{nextcol}}, C_{\text{ibestcol}}) = \max \{\text{corr}(C_m, C_{\text{ibestcol}}) \mid m \neq \text{ibestcol}, C_m > 0\}$

& $\text{corr}(C_{\text{nextcol}}, C_{\text{ibestcol}}) \geq 0.5$

& **matrix(ibestrow, nextcol)** = 0) then

ibestcol $\leftarrow \text{nextcol}$

[In case of a tie, choose the candidate column with the largest current column-sum.]

Else

nextcol $\leftarrow j$ s.t.

$C_j = \max \{C_m \mid \text{matrix}(\text{ibestrow}, m) = 0\}$

ibestcol $\leftarrow \text{nextcol}$

Endif

matrix(ibestrow, ibestcol) $\leftarrow 1$

$R_{\text{ibestrow}} \leftarrow R_{\text{ibestrow}} - 1$

$C_{\text{ibestcol}} \leftarrow C_{\text{ibestcol}} - 1$

If ($R_{\text{ibestrow}} > 0$) then

 go to Step (2)

Else

 go to Step (3)

Endif

Step (3): [Select next row for assignment.]

If ($R_i = 0$ for $i = 1, 2, \dots, p$) then

 go to Step (5)

Endif

—— Check the exception condition:

Cmax $\leftarrow \max \{C_j \mid j = 1, 2, \dots, t\}$

Rnum $\leftarrow \# \{R_i \mid R_i > 0, i = 1, 2, \dots, p\}$

If (**Cmax** > **Rnum**) then

 go to Step (4)

Endif

If (there is **nextrow** s.t. $\text{corr}(R_{\text{nextrow}}, R_{\text{ibestrow}}) = \max \{\text{corr}(R_k, R_{\text{ibestrow}}) \mid R_k > 0\}$

& $\text{corr}(R_{\text{nextrow}}, R_{\text{ibestrow}}) \geq 0.5$) then

iprevrow $\leftarrow \text{ibestrow}$

ibestrow $\leftarrow \text{nextrow}$

 [In case of a tie, choose the candidate row with the largest current row-sum.]

 If (there is j s.t. $C_j = \max \{C_m \mid C_m > 0; \text{matrix}(\text{iprevrow}, m) = 1\}$)

 then

ibestcol $\leftarrow j$

 Else

ibestcol $\leftarrow l$ s.t. $C_l = \max \{C_m\}$

 Endif

matrix(ibestrow, ibestcol) $\leftarrow 1$

$R_{\text{ibestrow}} \leftarrow R_{\text{ibestrow}} - 1$

$C_{\text{ibestcol}} \leftarrow C_{\text{ibestcol}} - 1$

 If ($R_{\text{ibestrow}} > 0$) then

 go to Step (2)

 Else

 go to Step (3)

 Endif

Else

 go to Step (1)

Endif

Step (4): [Eliminate exception condition.]

Undo the assignment of values to the elements of **ibestrow**—that is, set all elements in the row to zero

and reinstate the row-sum and column-sums to the values prior to this row's iteration.

Let $E = \{C_j \mid C_j \text{ satisfied exception condition in Step (3)}\}$

```

If (  $\#(E) > R_{\text{ibestrow}}$  ) then
  Find  $i$  s.t.  $R_i = \max\{R_k \mid k \neq \text{ibestrow}\}$ 
   $\text{ibestrow} \leftarrow i$ 
   $\text{ibestcol} \leftarrow j$  s.t.  $C_j = \max\{C_m \mid m = 1, \dots, t\}$ 
   $\text{matrix}(\text{ibestrow}, \text{ibestcol}) \leftarrow 1$ 
   $R_{\text{ibestrow}} \leftarrow R_{\text{ibestrow}} - 1$ 
   $C_{\text{ibestcol}} \leftarrow C_{\text{ibestcol}} - 1$ 
  If (  $R_{\text{ibestrow}} > 0$  ) then
    go to Step (2)
  Else
    go to Step (3)
  Endif
Else
  Do for every  $q$  s.t.  $C_q \in E$ 
     $\text{matrix}(\text{ibestrow}, q) \leftarrow 1$ 
     $R_{\text{ibestrow}} \leftarrow R_{\text{ibestrow}} - 1$ 
     $C_q \leftarrow C_q - 1$ 
     $\text{nextcol} \leftarrow q$ 
  Endo
  If (  $R_{\text{ibestrow}} = 0$  ) then
    go to Step (3)
  Else
    Let  $\text{iprecrow}$  denote the row that preceded  $\text{ibestrow}$  in assignment procedure.
    If (  $\text{corr}(R_{\text{iprecrow}}, R_{\text{ibestrow}}) \geq 0.5$  ) then
      If ( there exists  $j$  s.t.
         $C_j = \max\{C_m \mid C_m > 0;$ 
         $\text{matrix}(\text{iprecrow}, m) = 1\}$  )
      then
         $\text{ibestcol} \leftarrow j$ 
      Else
         $\text{ibestcol} \leftarrow l$  s.t.
         $C_l = \max\{C_m \mid m = 1, \dots, t\}$ 
      Endif
       $\text{matrix}(\text{ibestrow}, \text{ibestcol}) \leftarrow 1$ 
       $R_{\text{ibestrow}} \leftarrow R_{\text{ibestrow}} - 1$ 
       $C_{\text{ibestcol}} \leftarrow C_{\text{ibestcol}} - 1$ 
      If (  $R_{\text{ibestrow}} > 0$  ) then
        go to Step (2)
      Else
        go to Step (3)
      Endif
    Else
       $\text{ibestcol} \leftarrow \text{nextcol}$ 
      go to Step (2)
    Endif
  Endif
Endif

```

Step (5): [Exit the procedure.]

Return $\|\text{matrix}(i, j)\|$.

The only source of complication in this heuristic is the **exception condition**. This arises when, at the beginning of any iteration, the number of unfilled rows remaining is less than the largest (current) column-sum. This implies that some element in each column having the largest column-sum will need to be assigned a value greater than 1 in order to satisfy the column-sum constraints. This problem is avoided in step (4) of the heuristic by giving these columns a higher priority for assignment over other columns than would otherwise be more desirable.

4.2 Sample Results

Figures 1a, 1b, and 1c illustrate three sample 20×25 matrices generated on the basis of the user-specified data shown in Table 1. The inter-row-sum correlations $\{\text{corr}(R_h, R_i)\}$ for $1 \leq h, i \leq 10$ and for $11 \leq h, i \leq 20$ were specified as 0.9. All other inter-row-sum correlations were set at 0.1. The inter-column-sum correlations $\{\text{corr}(C_j, C_k)\}$ for $1 \leq j, k \leq 12$ and for $13 \leq j, k \leq 25$ were specified as 0.8, and all other inter-column-sum correlations were set at 0.1.

The row-sums and column-sums were generated based on this input by the first stage of the procedure that is described in §3. The values of the matrix elements were determined by the heuristic in § 4.1.

All the three sample matrices reflect the desired tool commonality among rows 1-10 and rows 11-20. The desired tool usage dependencies are also evident in these matrices. These examples illustrate that, subject to the above input conditions, the desired matrix may have more than one form of clustering (compare Figure 1a and 1c). Also note that deviations from perfect cluster formation is sometimes necessary to satisfy the row-sum and column-sum constraints (See Figure 1a rows 15 and 20, for example).

5 DISCUSSION

In this paper, we have described a two-stage method for generating cutting-tool requirement matrices based upon user-specified information on the distributions of the row-sums and column-sums. The matrices that are generated not only satisfy the row-sum and column-sum constraints but also reflect part similarities and tool usage dependencies that are characteristic of real tooling data. This facility to generate random, yet realistic, cutting-tool input data will be of use in detailed shop-floor simulation applications, such as for the evaluation of cutting-tool management strategies.

Table 1: Target row-sums and column-sums

Row-Sum R_i			Column-Sum C_j		
i	μ_i	σ_i	j	μ_j	σ_j
1	10.0	1.0	1	10.0	2.0
2	12.0	1.0	2	10.0	2.0
3	10.0	2.0	3	8.0	1.0
4	10.0	1.0	4	8.0	1.0
5	8.0	1.0	5	10.0	2.0
6	10.0	1.0	6	10.0	2.0
7	12.0	2.0	7	8.0	1.0
8	10.0	2.0	8	8.0	1.0
9	10.0	1.0	9	10.0	2.0
10	8.0	1.0	10	10.0	2.0
11	10.0	1.0	11	8.0	1.0
12	12.0	2.0	12	8.0	1.0
13	10.0	2.0	13	10.0	2.0
14	10.0	1.0	14	10.0	2.0
15	8.0	1.0	15	8.0	1.0
16	10.0	1.0	16	8.0	1.0
17	12.0	2.0	17	10.0	2.0
18	10.0	2.0	18	10.0	2.0
19	10.0	1.0	19	8.0	1.0
20	8.0	1.0	20	8.0	1.0
			21	10.0	2.0
			22	10.0	2.0
			23	8.0	1.0
			24	8.0	1.0
			25	10.0	2.0

```

1 1 1 1 1 1 1 1 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 10
1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 12
0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 10
1 1 1 0 0 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 11
0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 8
1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 11
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 14
0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 11
1 1 1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 10
0 0 0 0 1 1 1 0 0 0 0 0 0 0 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 9
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 12
0 1 0 0 1 1 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 15
1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 14
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 12
0 0 0 1 1 1 0 0 0 0 0 0 0 0 1 1 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 10
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 12
1 1 1 0 1 0 0 0 0 1 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 17
0 0 0 0 1 0 0 0 0 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 15
0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 13
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 11

```

```

0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 0 1 1 0 0 1 1 0 0 1
8 9 8 7 0 8 7 7 7 9 7 8 7 4 0 9 4 3 8 9 3 3 9 9 4

```

Figure 1b: Cutting-tool requirement matrix #2

```

1 1 1 1 1 1 1 1 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 10
1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 12
0 0 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 9
1 1 1 1 1 1 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 9
0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 8
1 1 1 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 9
0 0 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 11
1 1 1 1 1 1 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 10
1 1 0 0 1 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 10
1 1 1 1 1 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 8
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 1 1 1 0 1 1 1 0 1 10
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 12
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 1 1 0 0 1 1 0 0 1 9
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 10
0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 8
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 1 1 1 0 1 1 1 0 1 10
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 12
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 1 0 1 1 1 1 1 1 1 1 1 1 10
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 10
0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 5

```

```
7 7 8 7 9 8 8 8 8 8 8 8 9 8 6 7 7 8 6 6 9 9 8 6 9
```

Figure 1a: Cutting-tool requirement matrix #1

```

0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 11
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 13
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 12
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 11
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 0 0 1 1 1 1 1 1 1 1 1 9
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 11
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 13
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 11
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 11
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 1 1 1 1 1 0 0 1 1 0 0 1 9
1 1 1 1 1 0 1 1 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 11
1 1 1 1 1 1 1 1 1 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 12
1 1 0 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 12
1 1 0 0 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 11
0 0 0 0 0 0 1 0 1 1 1 1 1 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 8
1 1 1 1 1 1 1 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 10
1 1 1 1 1 1 1 1 1 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 12
1 1 1 1 1 1 1 1 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 11
1 1 1 1 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 10
0 0 0 0 1 1 0 1 0 0 0 0 0 1 0 0 1 0 0 0 1 0 1 1 1 1 1 1 1 1 9

```

```

0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 1 1 0 0 1
8 8 7 6 8 7 8 8 7 9 6 8 0 2 9 9 2 9 8 8 1 0 9 9 1

```

Figure 1c: Cutting-tool requirement matrix #3

One direction for future research in this area is the incorporation of information related to processing times within the tool-requirement matrix. This is pertinent in applications where information on the processing times associated with each cutting-tool for every part is required. Processing times, thus, would add another dimension to this variate generation problem.

Another extension of this problem is related to the notion of processing flexibility. In the machining context, it is often possible to process a part with alternate tool types (not necessarily with the same efficiency). For example, a part may require tool types 1, 3, 4, and one of 5, 6, and 7. Therefore, this part can be processed with three possible tool sets. A method is necessary to capture this feature of tool type substitutability when generating cutting-tool requirement data. An additional complication arises in the case of a tool type that requires two or more tool types to substitute it. For example, if the shape generated by tool type 4 can also be generated by using both tools 7 and 9, then we are faced with a situation where one tool type can be substituted by two different tool types. Thus, not only can a particular part type have more than one set of (alternative) tool requirements, but also these sets may contain different number of tools.

In conclusion, the problem of generating realistic cutting-tool requirement data is not trivial. It is hoped that the methodology presented in this paper will stimulate more research in this area that will lead to the development of robust input modeling techniques for detailed manufacturing simulation applications.

ACKNOWLEDGMENTS

The authors would like to thank Professor Bruce Schmeiser, Professor Vijaya Chandru, and Vispi Bulsara for their helpful suggestions and observations. This research was supported by the National Science Foundation under grant CDR 8803017 to the Engineering Research Center for Intelligent Manufacturing Systems at Purdue University. It was also partially based upon work supported by the National Science Foundation under Grant No. DMS-8717799.

REFERENCES

- Johnson, R. A. and D. W. Wichern. 1982. *Applied multivariate statistical analysis*. Englewood Cliffs: Prentice-Hall, Inc.
- Law, A. M. and W. D. Kelton. 1991. *Simulation modeling and analysis*. 2d ed. New York: McGraw-Hill Book Company.

- Stecke, K. E. 1989. Algorithms for efficient planning and operation of a particular FMS. *International Journal of Flexible Manufacturing Systems* 1: 287-324.

AUTHOR BIOGRAPHIES

DHARMARAJ VEERAMANI is a Ph.D. candidate in the School of Industrial Engineering at Purdue University. He holds a Bachelor's degree in Mechanical Engineering from the Indian Institute of Technology, Madras, India, and a Master's degree in Industrial Engineering from Purdue University. His research interests include the design, analysis, and control of manufacturing systems, concurrent engineering, and applications of operations research to manufacturing. He is a member of SME, ORSA, TIMS, and IIE. His membership in honor societies includes Tau Beta Pi, Alpha Pi Mu, Omega Rho, and Phi Kappa Phi. He will be joining the University of Wisconsin at Madison in January 1992 as Assistant Professor of Industrial Engineering.

JAMES R. WILSON is a Professor in the Department of Industrial Engineering at North Carolina State University. His current research interests are focused on the design and analysis of simulation experiments, including modeling and generation of probabilistic input processes, output analysis, variance reduction techniques, and simulation optimization. He also has an active interest in applications of operations research techniques to all areas of industrial engineering. He currently serves as Associate Editor of *IIE Transactions*, Departmental Editor of *Management Science* for Simulation, and Associate Program Chairman of the 1991 Winter Simulation Conference.

MOSHE M. BARASH is Ransburg Professor of Manufacturing and Professor of Industrial Engineering at Purdue University. He received the B.Sc. and Dipl.Ing. degrees in electrical engineering at the Technion, Israel Institute of Technology, Haifa, and the Ph.D. degree in mechanical engineering at the University of Manchester, England. He had design and research experience in industry, and his academic experience includes positions at the University of Manchester Institute of Science and Technology and Purdue University. Dr. Barash has published numerous research and technical papers in design, manufacturing processes, process modeling and automatic planning, manufacturing systems and other topics.