

University of Nebraska - Lincoln

DigitalCommons@University of Nebraska - Lincoln

---

Industrial and Management Systems Engineering  
Faculty Publications

Industrial and Management Systems Engineering

---

Summer 7-10-2008

# Randomly Generating Manufacturing Flow Line Models Using Mathematica

Paul Savory

University of Nebraska at Lincoln, psavory2@gmail.com

Follow this and additional works at: <http://digitalcommons.unl.edu/imsefacpub>

 Part of the [Industrial Engineering Commons](#), [Operational Research Commons](#), and the [Other Operations Research, Systems Engineering and Industrial Engineering Commons](#)

---

Savory, Paul, "Randomly Generating Manufacturing Flow Line Models Using Mathematica" (2008). *Industrial and Management Systems Engineering Faculty Publications*. 18.

<http://digitalcommons.unl.edu/imsefacpub/18>

This Article is brought to you for free and open access by the Industrial and Management Systems Engineering at DigitalCommons@University of Nebraska - Lincoln. It has been accepted for inclusion in Industrial and Management Systems Engineering Faculty Publications by an authorized administrator of DigitalCommons@University of Nebraska - Lincoln.

**RANDOMLY GENERATING MANUFACTURING FLOW LINE MODELS USING MATHEMATICA**

Paul Savory, Ph.D.

Department of Industrial and Management Systems Engineering  
 University of Nebraska - Lincoln  
 Lincoln, Nebraska 68588-0518

**ABSTRACT**

To test heuristic algorithms and techniques, researchers need numerous datasets so as to measure effectiveness and improve approaches. This paper discusses using *Mathematica*, a mathematical programming language, for randomly generating the specifications for manufacturing flow line models. Important issues include determining an arrival rate to a flow line, the number of flow line stations, the number of parallel servers for each production station, and specifying the service time distributions and their associated parameters. The paper concludes with a discussion on generating more general types of simulation models.

**1 INTRODUCTION**

Consider a manufacturing flow line system as a series arrangement of a finite number of production stations or resources. Production flow line (or flow shop) systems of this type are widely used in industry to represent situations in which parts arrive to a service area, obtain the service they require, and then move on to the next service area or leave the system (Carmichael 1987). This relationship is illustrated in Figure 1 (based on Pinedo 1982). A description of the basic notation is given in Table 1.

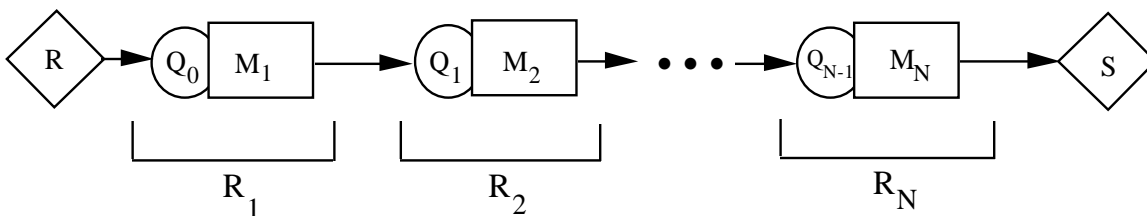


Figure 1: A flow line has N production steps, where each resource or production step consists of a parallel machine and associated waiting or buffer area.

Table 1: Description of flow line system

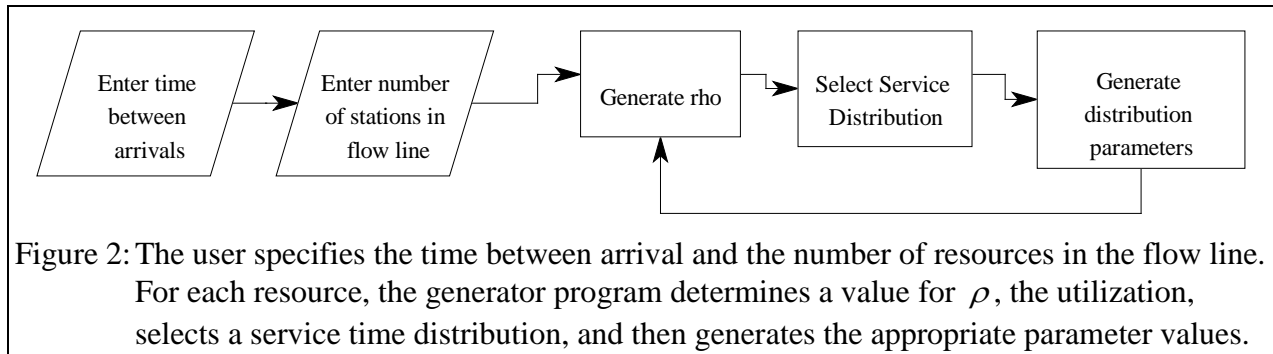
- R Receiving area
- S Shipping area
- N Number of production steps to produce a part
- $R_i$  A resource or production step consisting of a queue and associated machine ( $i = 1$  to  $N$ )
- $M_i$  Machine  $i$  ( $i = 1$  to  $N$ ) with  $j$  number of parallel servers
- $Q_j$  Infinite Queue or buffer proceeding  $M_{j+1}$  ( $j = 0$  to  $N-1$ )

Flow line manufacturing systems have been widely studied in operations research as serial, series or tandem queueing systems (Hendricks 1992; Altioik 1989; McCormick, Pinedo, Shenker, and Wolff 1989; Graves 1986; Konig and Schmidt 1984; Hillier and Boling 1967; Ku and Niu 1986; Lee and Zipkin 1992; Maaloe 1973; Pinedo 1982; Brandwajin and Jow 1988; Shalmon and Kaplan 1984; Suresh and Whitt 1990; Whitt 1983, 1984; Wolff 1982; Wittrock 1988).

To test heuristic algorithms and techniques, researchers need numerous datasets so as to measure effectiveness and improve approaches. This objective of this paper is to discuss using *Mathematica*, a mathematical programming language, for generating the specifications for manufacturing flow line models. Section 2 explores the issues involved in generating random flow line models. Section 3 presents an example of a randomly generated model. Section 4 concludes with a discussion on generating different types of simulation models.

## 2 GENERATING RANDOM FLOW LINE MODELS

A *Mathematica* program has been developed for generating random flow line simulation models (Savory 1993). Figure 2 illustrates the process by which a model is generated. The remainder of this section discusses the specifics on randomly generating a model.



### 2.1 Specifying the time between arrivals and the number of stations

The program first prompts the user to specify the time between arrivals to the flow line ( $1/\lambda$ ) and the number of stations or resources in the flow line (N).

### 2.2 Utilization of each work station

For each of the resources in the flow line, the *Mathematica* program generates a value for  $\rho$ , the utilization of the resource. To ensure that the flow line remains balanced, the utilization ( $\rho$ ) is limited to a range between twenty and eighty percent. In *Mathematica*, the utilization is set using the following code:

$$\rho = \text{Random}[\text{UniformDistribution}[\.20, .80]]$$

This and subsequent *Mathematica* statements require that the *Mathematica* DiscreteDistributions package be loaded:

```
<< Statistics`DiscreteDistributions`
```

### 2.3 The number of parallel servers for each work station

For each of the resources, the program next generates the number of parallel servers performing the resource's task. The *Mathematica* code for accomplishing this is:

```
S = Random[DiscreteUniformDistribution[10]]
```

The variable S, the number of servers for the resource, is set to be an integer ranging from one to ten.

### 2.4 The service time distribution for each production station

For each resource or production station, a service time distribution must be specified. The generating program randomly selects (with equal probability) the distribution to be either the exponential, lognormal, normal, triangular, or uniform distribution. Using this specification, the generation program next estimates the service time parameters. The key to accomplishing this specification is to realize that since the mean time between arrivals, the number of servers, and the utilization of each production station are known, computing the mean for the generated parameter values must be such that  $\rho \leq 1$  to ensure that the flow line achieves steady-state.

### 2.5 Generating parameters for service time distributions

#### 2.5.1 exponential distribution

For an *exponential* service time distribution, the only parameter that needs to be estimated is its mean  $\mu$ . For any queueing system, a resource's utilization (*i.e.*,  $\rho$ ) is equal to:

$$\rho = \frac{\lambda\mu}{s}$$

Since the time between arrivals ( $1/\lambda$ ), the number of servers (S), and the resource utilization ( $\rho$ ) are known, solving for the service mean results in the following equation:

$$\mu = \rho \times S \times \frac{1}{\lambda}$$

Thus, the mean parameter is based in the resource utilization that was previously generated.

#### 2.5.2 normal and lognormal distributions

For the *lognormal* and *normal* service time distributions, the mean value is generated using the service time utilization relationship:

$$\mu = \rho \times S \times \frac{1}{\lambda}$$

The standard deviation is generated using its relationship to the coefficient of variation (the standard deviation divided by the service mean):

$$\sigma = \text{Random[UniformDistribution[.01,.10]]} \times \mu$$

Thus, a value between 0.10 and 0.10 is generated for the coefficient of variation and multiplied by the generated service time mean.

### 2.5.3 uniform distribution

For the *uniform* service time distribution, a mean and standard deviation are computed using the same logic as for the lognormal and normal distributions. To generate the specific parameter values (a minimum and a maximum) for the uniform distribution, the following relationship is used (Pritsker 1986):

$$\begin{aligned}\text{minimum} &= \text{mean} - \sqrt{3} \times (\text{standard deviation}) \\ \text{maximum} &= \text{mean} + \sqrt{3} \times (\text{standard deviation})\end{aligned}$$

Using the generated mean and standard deviation, the minimum and maximum parameter values are computed.

### 2.5.4 triangular distribution

The *triangular* service time distribution requires estimating three parameters: a maximum, a mode, and a minimum. To estimate these, a maximum mean value is computed using the utilization relationship:

$$\mu = \rho \times S \times \frac{1}{\lambda}$$

The program next randomly generates a maximum value, a mode (a value between zero and the generated maximum), and a minimum (a value between zero and the generated mode):

$$\begin{aligned}\text{maximum} &= \text{Random}[\text{UnifromDistribution}[0, 1/\lambda]] \\ \text{mode} &= \text{Random}[\text{UnifromDistribution}[0, \text{maximum}]] \\ \text{minimum} &= \text{Random}[\text{UnifromDistribution}[0, \text{mode}]]\end{aligned}$$

The program computes the mean for these three parameter estimates:

$$\mu^* = \frac{\text{maximum} + \text{mode} + \text{minimum}}{3}$$

If the new mean ( $\mu^*$ ) is less than the originally generated maximum mean ( $\mu$ ), then these three values represent the respective parameters of a triangular distribution. Otherwise, the program generates three more estimates of the maximum, mode, and minimum. This procedure continues until the mean of the three parameters ( $\mu^*$ ) is less than the maximum mean ( $\mu$ ).

## 3 EXAMPLE MODEL

Running the *Mathematica* program for a flow line consisting of six resources and a time between arrivals of 100 seconds results in the flow line described in Figure 3. The resulting description includes such information as each resource's arrival mean, arrival rate, arrival variability, arrival coefficient of variation, and arrival squared coefficient of variation. It also specifies the number of parallel resources for the resource, the service distribution and its parameters, the service mean, the service variability, the service time standard deviation, the service time coefficient of variation, the service time squared coefficient of variation, and the resource utilization.

## Technical Paper: July 2008

<p><b>Resource #1 of the Flow Line</b>            Arrival Mean is 100            Arrival Rate is 0.01            Arrival Variability is 10000            Arrival COV is 1.            Arrival SQCOV is 1.            Number of Parallel Resources is 1            Service Distribution is Uniform (75, 85)            Service Mean is 80.            Service Variability is 8.33333            Service SD is 2.88675            Service Time is COV is 0.0360844            Service Time SQCOV is 0.00130208            Resource Utilization is 0.8</p>	<p><b>Resource #4 of the Flow Line</b>            Arrival Mean is 100            Arrival Rate is 0.01            Arrival Variability is 10000            Arrival COV is 1.            Arrival SQCOV is 1.            Number of Parallel Resources is 4            Service Distribution is Normal (320, 25)            Service Mean is 320.            Service Variability is 625.            Service SD is 25.            Service Time is COV is 0.078125            Service Time SQCOV is 0.00610352            Resource Utilization is 0.8</p>
<p><b>Resource #2 of the Flow Line</b>            Arrival Mean is 100            Arrival Rate is 0.01            Arrival Variability is 10000            Arrival COV is 1.            Arrival SQCOV is 1.            Number of Parallel Resources is 2            Service Distribution is Normal (130, 15)            Service Mean is 130.            Service Variability is 225.            Service SD is 15.            Service Time is COV is 0.115385            Service Time SQCOV is 0.0133136            Resource Utilization is 0.65</p>	<p><b>Resource #5 of the Flow Line</b>            Arrival Mean is 100            Arrival Rate is 0.01            Arrival Variability is 10000            Arrival COV is 1.            Arrival SQCOV is 1.            Number of Parallel Resources is 1            Service Distribution is Triangular (32, 43, 60)            Service Mean is 45.            Service Variability is 33.1667            Service SD is 5.75905            Service Time is COV is 0.127979            Service Time SQCOV is 0.0163786            Resource Utilization is 0.45</p>
<p><b>Resource #3 of the Flow Line</b>            Arrival Mean is 100            Arrival Rate is 0.01            Arrival Variability is 10000            Arrival COV is 1.            Arrival SQCOV is 1.            Number of Parallel Resources is 2            Service Distribution is Triangular (120, 150, 180)            Service Mean is 150.            Service Variability is 150.            Service SD is 12.2474            Service Time is COV is 0.0816497            Service Time SQCOV is 0.00666667            Resource Utilization is 0.75</p>	<p><b>Resource #6 of the Flow Line</b>            Arrival Mean is 100            Arrival Rate is 0.01            Arrival Variability is 10000            Arrival COV is 1.            Arrival SQCOV is 1.            Number of Parallel Resources is 1            Service Distribution is Uniform (64, 80)            Service Mean is 72.            Service Variability is 21.3333            Service SD is 4.6188            Service Time is COV is 0.06415            Service Time SQCOV is 0.00411523            Resource Utilization is 0.72</p>

Figure 3: Output resulting from running the generation program. This is a six station production flow line in which the time between part arrivals to the flow line was specified to be 100 seconds.

## 4 EXTENSIONS

The objective of this paper has been to demonstrate using the mathematical routines of *Mathematica* to generate random simulation models to use for testing algorithms or techniques.

The manufacturing flow line discussed in this paper requires that all parts pass through the same sequence of resources with no feedback, scrap, or rework. In addition, it assumes that resources are 100% reliable and never experience a breakdown. These types of characteristics can be added to the generation program. For example, one approach to generating flow line models with scrap is for the program to generate a probability that after finishing processing at a resource, the part is scrapped or continues to the next stations. In discussing networks of queues, Molloy (1989) and Mehdi (1991) remark that one needs to adjust the arrival rate to the subsequent resource in the flow line (since some of the arrivals are “lost” to scrap). As for breakdowns, a similar approach of determining a time between breakdowns (either deterministic or probabilistic) and a breakdown repair distribution.

## REFERENCES

- Hendricks, K. B., 1992. “The Output Processes of Serial Production Lines of Exponential Machines With Finite Queues.” *Operations Research*. 40: 1139-1147.
- Altiok, T., 1982. “Approximate Analysis of Exponential Tandem Queues and Blocking.” *European Journal of Operations Research*. 11: 390-398.
- Altiok, T., 1989. “Approximate Analysis of Queues in Series with Phase-Type Service Time and Blocking.” *Operations Research*. 37: 601-610.
- Brandwajin, A. and Y. L. Jow, 1988. “An Approximation Method for Tandem Queues With Blocking.” *Operations Research*. 36: 73-83.
- Carmichael, D. G., 1987. *Engineering Queues in Construction and Mining*. West Sussex, England: Ellis Horwood Limited.
- Graves, S. C., 1986. “A Tactical Planning Model for a Job Shop.” *Operations Research*. 34: 522-533.
- Hillier, F. S. and R. W. Boling, 1966. “The Effect of Some Design Factors on the Efficiency of Production Lines with Variable Operation Times.” *Journal of Industrial Engineering*. 17: 651-658.
- Konig, D. and V. Schmidt, 1984. “Relationship Between Time/Customer Stationary Characteristics of Tandem Queues Attended by a Single Server.” *Journal of the Operations Research Society of Japan*. 27: 191-204.
- Ku, P. S. and S. C. Niu, 1986. “On Johnson’s Two-Machine Flow Shop with Random Processing Times.” *Operations Research*. 34: 130-36.
- Lee, Y. J. and P. Zipkin, 1992. “Tandem Queues With Planned Inventories.” *Operations Research*. 40: 936-947.
- Maaloe, E., 1973. “Approximations Formulae for Estimation of Waiting-Time in Multiple-Channel Queueing System.” *Management Science*. 19: 703-710.
- McCormick, S. T., M. L. Pinedo, S. Shenker, and B. Wolf, 1989. “Sequencing in an Assembly Line with Blocking to Minimize Cycle Time.” *Operations Research*. 37: 925-935.
- Medhi, J., 1991. *Stochastic Models in Queueing Theory*. New York: Academic Press, Inc.

## Technical Paper: July 2008

- Molloy, M. K., 1989. *Fundamentals of Performance Modeling*. New York: Macmillan Publishing Company.
- Pinedo, M., 1982. "Minimizing the Expected Makespan in Stochastic Flow Shops." *Operations Research*. 30: 148-162.
- Pritsker, A. A. B., 1986. *Introduction to Simulation and Slam II*. West Lafayette, Indiana: Systems Publishing Corporation.
- Savory, P. A., 1993. "A Robust Aggregation Approach to Simplification of Manufacturing Flow Line Models." Ph.D. Dissertation, Arizona State University.
- Shalmon, M. and M. A. Kaplan, 1984. "A Tandem Network of Queues with Deterministic Service and Intermediate Arrivals." *Operations Research*. 32: 753-773.
- Suresh, S. and W. Whitt, 1990. "Arranging Queues in Series: A Simulation Experiment." *Management Science*. 36: 1080-1091.
- Whitt, W., 1983. "Approximating a Point Process by a Renewal Process: Two Basic Methods." *Operations Research*. 31: 125-147.
- Whitt, W., 1984. "Departures from a Queue with Many Busy Servers." *Mathematics of Operations Research*. 9: 534-544.
- Wittrock, R. J., 1988. "An Adaptive Scheduling Algorithm For Flexible Flow Lines." *Operations Research*. 36: 445-453.
- Wolff, R. W., 1982. "Tandem Queues with Dependent Service Times in Light Traffic." *Operations Research*. 30: 619-635.