

Capacity Planning Using a Simulation Model

George Tompkins, SAS Institute Inc., Cary, NC

ABSTRACT

Discrete event simulation models coupled with statistical analysis allow you to select a system design or resource configuration with a high degree of certainty. This paper introduces discrete event simulation and demonstrates the use of response surface methodology to select a resource configuration. The example system is kept intentionally generic to emphasize the applicability of this methodology in many different industries.

INTRODUCTION

Simulation models allow us to model complex systems that would not otherwise yield to analytical methods. With an accurate, validated simulation model, we find a tool for providing direction for complex questions such as "how many?", "when?", even "what policy?". The analysis of a simulation model draws heavily on statistical methods; an analyst should always bear in mind that the analysis of the model will probably require almost as much time as its construction. In this paper, we will use a simulation model to estimate the resources required to achieve the maximum system throughput. The example used is intentionally generic; it could represent anything from a machine shop to a hospital. The intent is to demonstrate the general usefulness of a simulation model and the statistical techniques used in its analysis. The following sections will describe the type of simulation model used, the particular example found in this paper, a general analysis methodology, and the results for our example.

SIMULATION MODELS

The phrase *simulation model* is very general. This paper focuses on discrete event simulation models. The discrete event description applies to any system whose state changes at discrete points in time, not continuously. For example, the kinetic energy of a pendulum, or the amount of gasoline in your fuel tank change continuously over time. Contrast this with a bank teller. The teller is idle until someone appears at the window for service and then the teller remains busy until the person leaves. The state of the teller (busy or idle) changes only at particular, or discrete, moments in time. Fortunately, this is true of many systems: a machine tool begins and ends its operation on a part at discrete times, phone calls begin and end at discrete times, etc.

To model these discrete event systems, the user must enumerate the states of the system and specify the conditions which lead to a change in state. This is a very general approach that is quite flexible, but it can be cumbersome. The modeling process is often simplified by picturing it in the framework of a queue. Transactions arrive periodically, if the server is idle then service can be commenced. However, if the server is busy or blocked, the transaction must wait in line for its turn. The queue can be described with two state variables, the number of transactions in the queue and the status of the server; both of which can change only at particular moments in time. Even though it is simpler, this framework is still quite flexible; no mention was made of exactly what a transaction represents. It could be a physical entity such as a wafer or machined part, or it could be an abstraction like a phone call or the occurrence of a failure event. Often, a discrete event simulation is visualized as a network of connected queues.

Specialized languages have been developed for modeling systems of queues, often called network or queuing simulations. SAS/OR® Software has added a tool called QSIM for this purpose. QSIM was available in beta form under release 6.12 and is production in version 7. Note: this paper does not focus on model building in QSIM. Instead, it looks at the analysis of a model built in QSIM.

EXAMPLE

The simulation model used in this paper was adapted from an example by Law and Kelton[1]. It represents a system with six stations: five workstations and one holding area. Transactions arrive in a holding area and must be processed through the workstations, or a subset of them, in a particular sequence. A transportation mechanism exists to move the transactions between the stations; it operates under a first come, first served policy. Three types of transactions exist, each requiring a unique processing sequence. The processing times at a station also depend on the transaction type. Our goal is to determine the number of servers needed at each station to maximize the number of transactions processed through the system in an 8-hour day. The statistical details of transaction mix and the distributions governing transaction arrival and service times have already been determined. This information is available in the appendix.

The original problem represented a jobshop manufacturing system where a forklift was used to move parts between workstations. Each workstation represented a cell of like machine tools, and the processing sequence modeled the part's process routing. All these concepts still apply, but this example could represent many other systems, such as patients waiting to see doctors or the flow of paper work in insurance claim processing. The goal remains the same: given certain assumptions about arrival rate and processing times, how many resources are required to provide a reasonable level of service? In our example there are six decision variables, the number of servers at each of the five workstations and the number of transportation servers. If each of these variables can take values of 1 through 7, for instance, then there are 117,649 possible combinations. These combinations or coordinates will be represented in a vector form; [# Transportation servers, # of Station 1 servers, # of Station 2 servers,... # of Station 5 servers]. Clearly, we must use an efficient approach to explore this space.

ANALYSIS METHODOLOGY

A discrete event simulation model is like a scale model of a system in your computer. If the model is well written, it should reflect the same broad behaviors as the original system. However, all performance measures for the simulation model will be subject to statistical variation, just like any performance measures for the original system. Therefore, statistical methods must be used to compare and analyze the data generated by a simulation model.

A statistical procedure often used to find optimal settings or conditions for a system is Response Surface Methodology (RSM). RSM attempts to describe the response algebraically and then determine the settings that will maximize or minimize the response. In our case the response is the number of transactions processed in a day, and the objective is to maximize this value.

RSM involves three phases: 1) determining a direction of movement (estimate the gradient), 2) moving along this gradient until a region containing the maximum (or minimum) is found (steepest ascent), and 3), building an algebraic model of the response in this region (fitting a regression model) and estimating the coordinates of the

maximum (or minimum). We will apply this methodology to our capacity problem.

ESTIMATING THE GRADIENT

Typically in RSM, a factorial screening design is used to estimate the gradient. However, in this example, that is not necessary. As more servers are added, the wait times will be reduced and more transactions will be processed in a given time. Thus, the direction of increased throughput points to increasing the number of servers.

STEEPEST ASCENT

There is no fixed procedure for deciding how large the steps along the gradient should be. In this example the starting point was [1,1,1,1,1,1], and an increment of 2 was used. Thus, the first four combinations examined were [1,1,1,1,1,1], [3,3,3,3,3,3], [5,5,5,5,5,5] and [7,7,7,7,7,7]. Five independent replications were made at each of these points. A scatter plot of the data is shown in Figure 1.

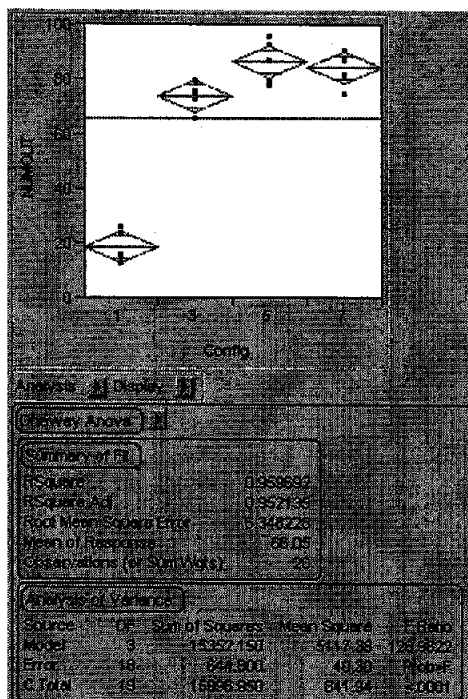


Figure 1

Clearly, there is no statistically significant difference in the throughput between 5 and 7 servers. We now have evidence of the region where the optimum lies.

REGRESSION MODEL FITTING

If the simulation is itself a model, what benefit is there to fitting a second model to the simulation? Hopefully, the algebraic equation provided by a regression model will succinctly capture the behavior of throughput and make it easier to derive the coordinates of an optimum. The standard response surface model is a quadratic equation of the form

$$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4 + \beta_5 X_5 + \beta_6 X_6 + \beta_7 X_1^2 + \beta_8 X_2^2 + \dots$$

This form provides good flexibility for approximating a response, while at the same time being fairly simple.

The challenge often lies in selecting the experimental design to provide data for fitting the model. A screening design is inexpensive, in terms of the number of design points required, but does not allow us to fit curvature. The design can be augmented with center points, but this will provide only a general measure of the curvature. Better alternatives include central composite and Box-Behnken designs. A

drawback in this example is size; with 6 factors a central composite and Box-Behnken designs will require, respectively, 53 and 54 runs. In addition, both designs would require numerous runs at levels which do not exist – for example, 3.375 servers.

Another alternative is a D-optimal design. A D-optimal design is tailored to a particular model. It selects a set of design points that minimize the variance of the parameter estimates. D-optimal designs are often used when a traditional factorial design is too large or infeasible. For this example, a D-optimal design using 32 runs was created (Figure 2). Inspection of scatter plots indicated that the design covered a large portion of the space. Recall that the goal is to explore the behavior of throughput across a fairly large design space as efficiently as possible. The D-optimal response surface design will help us do this in a systematic manner. Note that the original design had 17 runs with non-integer values; these were adjusted to integer values.

dopt							
8 Cols							
	trn	s1	s2	s3	s4	s5	
1	3	3	3	5	3	3	
2	2	2	2	2	5	2	
3	2	2	2	2	5	5	
4	5	2	2	3	5	3	
5	2	2	5	3	3	5	
6	5	2	5	5	5	5	
7	2	2	5	5	5	2	
8	2	3	2	5	5	5	
9	2	5	5	5	2	2	
10	5	5	2	5	2	3	
11	2	5	5	2	5	2	
12	5	2	2	2	2	5	
13	2	3	5	2	2	5	
14	3	2	4	2	5	5	
15	5	5	2	5	5	5	
16	5	4	5	3	5	4	
17	2	5	2	2	5	5	
18	5	3	5	5	2	5	
19	5	5	5	5	5	2	
20	2	5	2	5	5	2	
21	3	2	2	5	2	2	
22	3	5	2	3	2	5	
23	5	5	5	2	2	2	
24	5	2	5	2	5	2	
25	3	2	5	2	2	3	
26	5	5	5	2	4	5	
27	5	2	2	5	4	2	
28	2	3	2	2	2	2	
29	5	5	2	2	5	2	
30	2	5	3	2	2	4	
31	5	2	3	3	2	2	
32	2	5	5	5	5	5	

Figure 2

Five independent replications were carried out at each design point and the throughput was recorded. The summary statistics for the fitted response surface model are shown in Figure 3.

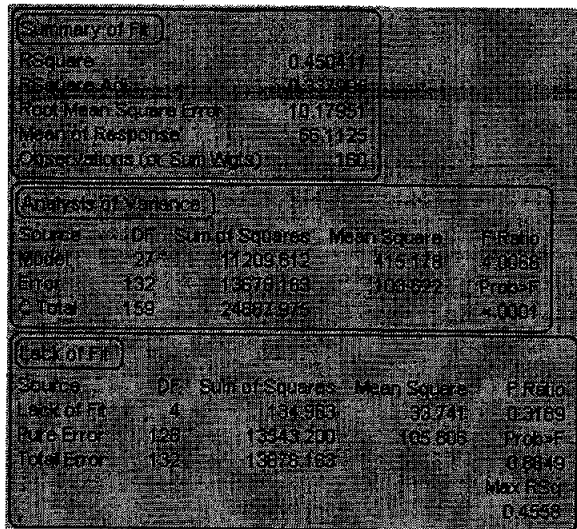


Figure 3

The summary statistics indicate the model is statistically significant and does not suffer from a significant lack of fit. The rsquare statistic is smaller than we would hope for; however, the Lack of Fit report indicates that the vast majority of the error is due to pure error. The estimated standard deviation of 10.18 explains this. Apparently, there is a fairly large inherent variability in daily throughput.

The desirability function can help us search for an optimal combination of server resources (Figure 4).

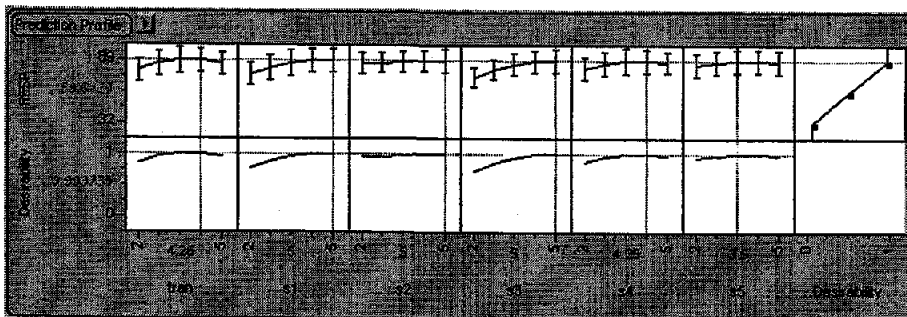


Figure 4

The desirability function allows us to select a target for the response (maximization in this case), and the software will suggest settings to achieve this target. Notice that it should be possible to achieve our goal without using five servers at each station. Also notice the indication that throughput is quite insensitive to the number of servers at station 2. We will test the suggested solution, [4,5,5,5,4,3], in the simulation model, and compare it to a solution with number of servers at station 2 reduced, [4,5,3,5,4,3].

VERIFICATION

Fifty independent replications were made at both conditions. The summary statistics for the suggested solution are shown in Figure 5 and those for the alternative solution in Figure 6.

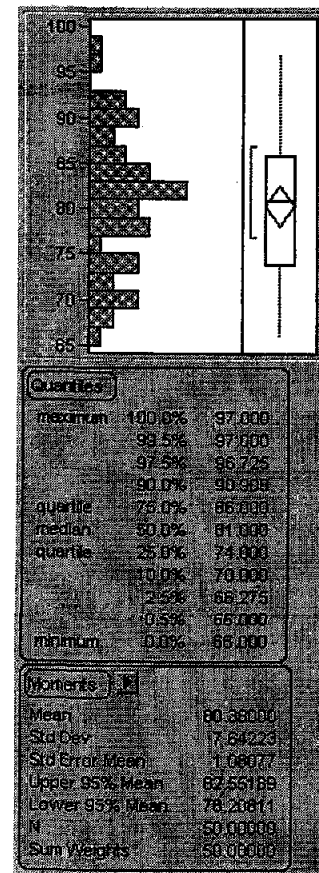


Figure 5

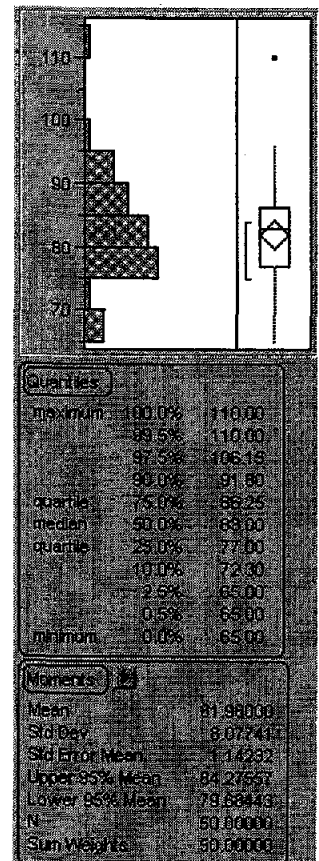


Figure 6

CONCLUSIONS

The histograms indicate throughput is normally distributed, and the overlap of the confidence intervals on the mean indicate there is no statistically significant difference in average throughput for these two solutions ([4,5,5,5,4,3], and [4,5,3,5,4,3]).

Using the general technique of response surface methodology, we have been able to easily identify a solution that yields high predicted throughput without using five servers at every station. Given the typical expense of resources (both people and equipment), the savings suggested by the D-optimal design easily justify its use. This is especially true since no changes to an actual system were required during the search.

FURTHER RESEARCH

It would be interesting to compare not only mean throughput, but variation as well. It may be possible to find a solution that has a reasonable throughput but reduced daily variability. Also, it would be interesting to observe the solutions suggested by the response surface model if server utilization were used as a response along with throughput. Finally, the effect on throughput of different service policies at the workstations and transporter could be explored.

These, and other questions, can be investigated at relatively small expense using discrete event simulation models coupled with statistical analysis.

SAS/OR is a registered trademark of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

APPENDIX

Transaction interarrival times are exponentially distributed with a mean of 1/15 hr.

Table 1 Transaction Mix

Type	Proportion
1	.3
2	.5
3	.2

Table 2 Transaction Routing

Type	Routing
1	3,1,2,5
2	4,1,3
3	2,5,1,4,3

Transportation times are uniformly distributed between .01 and .04 hours.

Transaction service times at all workstations follow gamma distributions with a shape parameter of 2. Tables 3 and 4 list the mean and scale parameter for each transaction type.

Table 3 Mean Service Times (hrs.)

Type	S1	S2	S3	S4	S5
1	.15	.1	.25		.3
2	.2		.3	.15	
3	.35	.15	.2	.2	.1

Table 4 Gamma Scale Parameters

Type	S1	S2	S3	S4	S5
1	.075	.05	.125		.15
2	.1		.15	.075	
3	.175	.075	.1	.1	.05

REFERENCES

[1] Law, A. M. and Kelton, W. D. (1991), *Simulation Modeling and Analysis, 2nd Edition*, New York: McGraw-Hill, Inc.

ACKNOWLEDGMENTS

I would like to thank Diane Tompkins for her help in the preparation of this paper.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

George Tompkins
SAS Institute Inc.
SAS Campus Drive
Cary, NC 27513
919/677-8000
george.tompkins@sas.com