Proceedings of the 2000 Winter Simulation Conference
J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick, eds.

OPTIMIZING SIMULATIONS WITH CSIM18/OPTQUEST: FINDING THE BEST CONFIGURATION

Herb Schwetman

Mesquite Software, Inc.
PO Box 26306
Austin, TX 78755-0306, U.S.A.

ABSTRACT

In many cases, a simulation model of a system is used to evaluate alternative configurations of that system, with the goal of finding the configuration which maximizes (or minimizes) the value of an objective while meeting all of the stated requirements. The CSIM18/OptQuest package automates this kind of search for the best configuration by combining a powerful simulation engine, CSIM18, and a state-of-the-art optimization package, OptQuest. This paper describes this integrated package for doing optimization and simulation. The paper concludes with two examples: finding the best configuration for a job-shop, and finding the best configuration for a web server.

1 INTRODUCTION

The combination of a powerful simulation engine and a state-of-the-art optimization package opens up new areas of applications for systems analysis. In many situations, the goal of a modeling project is to predict which of several system configurations will best meet the goals for the system. In some cases, a new system is being designed for service in a demanding operating environment: there may be stringent performance requirements and, at the same time, the cost of the system is expected to fall below a specified level. Finding the system that best meets both the performance goals and the cost criteria is a challenge. With complex systems, the range of possible configurations is large and the search for the best configuration can be time consuming and expensive.

One approach to solving problems involving configuring systems is to devise a representation of the system and the search for the best configuration in a form suitable for solution using optimization techniques. Unfortunately, with complex systems, evaluating the function representing the performance of the system may be impossible, primarily because such a function may not exist.

Using a simulation model of a complex system can provide a viable approach to estimating the performance of the system, especially if a realistic model of the system can be implemented. However, with a simulation model, the search for the best configuration can be time consuming and, in some cases, ineffective. If there are a large number of variables defining a system configuration, it may be difficult to implement search procedures that investigate all of the feasible configurations.

CSIM18/OptQuest is an integrated package which combines a state-of-the-art optimization package with a robust and flexible simulation engine (Schwetman 2000). Analysts can use this package to implement models that can help find the best configuration for a system, where best is defined to be the configuration which meets all of the requirements and which has the best value of the objective found "so far". OptQuest can be viewed as a "run manager" for the model. OptQuest generates a new configuration and then invokes the model to simulate the operation of this configuration of the system. The model produces values for the objective and for all of the requirements and returns these values to OptQuest. OptQuest evaluates these values and then either terminates if the requested number of runs have been completed, or generates a new configuration and continues.

This paper describes the CSIM18/OptQuest package and then illustrates use of the package on two examples.

2 THE CSIM18 SIMULATION ENGINE

CSIM18 is a tool kit for building process-oriented, discrete-event simulation models of systems (Schwetman 1996, Schwetman 1999). An analyst writes a C++ (or C) program which uses the components of CSIM18 to easily implement a model of the system of interest. CSIM18 has provisions for:

- Creating and managing processes – these processes are the active entities of the model

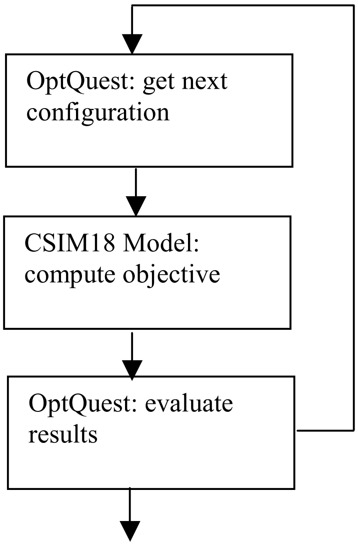

Figure 1: Optimization Process Flow

- Instantiating instances of facilities – facilities represent a class of resources in the system
- Instantiating instances of storages – storages represent another class of resources in the system
- Instantiating mailboxes and events – mailboxes are used to implement inter-process communications and events are used to implement process synchronization mechanisms
- Collecting and presenting statistical summaries on the operation of the model, and
- Generating streams of random values derived from specified probability distributions.

Using these components, analysts are able to develop models which accurately mimic the behavior of large, complex systems.

Because a model is a program, it is straightforward to embed a model in another application and to have a model operate with other programs such as database management systems. Also, CSIM18 models (programs) are very efficient; as a result, it is practical to develop models of large systems, with many resources and many processes. It is also possible to represent the behavior of almost any kind of system.

3 THE OPTQUEST OPTIMIZER

OptQuest (Glover, Kelly and Laguna 1999) is a general purpose optimizer designed to work with simulation models of systems. As indicated above, OptQuest generates a succession of configurations which are used, one by one, by the model to estimate the performance of the system using the configuration. OptQuest uses each of the performance estimates to suggest another configuration. In effect, OptQuest conducts a search through the space of configurations of the system, to find the best feasible configuration. The feasibility of a configuration is determined by whether or not the system meets all of the requirements specified for the system. The goodness of a configuration is determined by the value of the objective specified for the system.

OptQuest uses a combination of scatter search, tabu search and a neural network to guide the search for better configurations. Scatter search is related to tabu search (Glover 1977). These techniques are used to develop successive sets of configurations and results to build a set of "reference points" that constitute good solutions. Experience has shown that OptQuest finds better solutions (feasible configurations with improving values of the objective function) in a very efficient manner. The interested reader is referred to (Glover, Kelly and Laguna 1999) for more information.

4 EXAMPLE 1 – CONFIGURING A JOB-SHOP

A job-shop is a collection of workstations, where each workstation has one or more machines which perform a function (Law and Kelton 1991). Jobs (pieces to be manufactured or assembled) arrive and are routed through the workstations. When a job is completed, it leaves the area. There are typically different types of jobs. Jobs of the same type visit the same workstations in the same order, and the times spent at the stations are from the same probability distributions.

A major issue in the design of a job-shop is specifying the number of machines in each workstation so that the jobs all flow through the job-shop in an efficient and timely manner. However, each of these machines can be very expensive, so there are significant constraints on the number of machines that can be installed.

A CSIM18/OptQuest model of the job-shop described in (Law and Kelton 1991) was implemented, and used to find and evaluate different configurations for the machines in the workstations. The parameters for this model are given in Appendix A. The optimization problem was setup as shown in Table 1:

Table 1: Optimization Constraints and Cost Vector

Workstation	1	2	3	4	5
Minimum machines	1	1	1	1	1
Maximum machines	5	5	5	5	5
Cost per machine	1	2	10	1	6

The objective of this problem was to minimize the cost of the entire system with the requirement that the average delay for all parts processed is less than 10 hours.

This CSIM18/OptQuest program was setup so that a configuration (a vector consisting of the number of machines in each workstation) is the input to one execution (run) of the simulation model. The outputs from one run of the model are the cost of the configuration and the average delay per job processed. A model of a particular configuration executes until the average job delay as computed in the model has "converged" to a 90% confidence interval with a relative error of +/- 5%. CSIM18 has a feature which automatically controls the execution of each run (Schwetman and Brumfield 1997). Many of the configurations of this system do not have enough capacity to "keep up" with the arrival rate of the jobs. These insufficient configurations cause an error ("too many active processes in the model"). This error is detected and the run for that configuration is aborted; an average delay of one plus the maximum allowable delay is returned. The optimizer marks each of these insufficient configurations as infeasible and they are not considered any further.

The first three configurations proposed by OptQuest for evaluation are as shown in Table 2.

Table 2: Results for First Three Configurations

Run	1	2	3	4	5	Cost	Avg. Delay
1	3	3	3	3	3	60.0	6.209
2	1	1	1	1	1	20.0	11.00
3	5	5	5	5	5	100.	2.998

The first run is the average of the minimum and maximum number of machines in each workstation. The second run uses the minimum number of machines in each workstation; this is the least expensive configuration, but it is not feasible, because the average delay exceeds the requirement (10 hours). In fact, the average delay of 11.0 indicates that this configuration is insufficient. The third run uses the maximum number of machines in each workstation. This configuration is the most expensive and also the "fastest" (has the minimum average delay of the configurations evaluated).

The best configuration found by the CSIM18/OptQuest model is as shown in Table 3.

Table 3: Best Configuration Found in 312 Trials

Run	1	2	3	4	5	Cost	Avg. Delay
308	3	2	3	3	1	46.00	7.973

The OptQuest optimizer was setup to select 312 configurations (10 per cent of the total of 3125 possible configurations). Modeling (evaluating) these 312 configurations required 142 seconds on a 300 MHz Pentium II processor. Of these 312 configurations, 256 were "insufficient" and were quickly rejected, leaving 56 to be run until the average delay time converged.

This job shop problem can be completely analyzed by evaluating each of the 3125 configurations of the machines in each workstation. Thus, the optimal configuration can be determined. The model was revised so as to iterate through all of the possible configurations. Running all of the configurations required about 1181 seconds of cpu time on the 300 MHz Pentium II processor. Of the 3125 possible configurations, 2585 were "insufficient" to handle the load, leaving 540 to be simulated until the average delay time converged. It turns out that the best configuration found by the CSIM18/OptQuest package (see Table 3) was indeed the optimal configuration, as determined by this complete analysis.

5 EXAMPLE 2 – CONFIGURING A WEB SERVER

A web server is a system connected to the Internet which responds to requests for files. A user accesses a web server through a browser, a program executing on the user's desktop system. The browser fetches html pages from the server and renders these pages on the user's system. Typically, in the process of rendering a page, other pages and files must be fetched before the initial page is completely rendered.

A web server provides service to users all over the world. If a web site located on a server becomes "hot", the rate of page fetches can rise to high levels. As the page request rate increases, service at the server can degrade. Furthermore, if too many browsers are accessing the server, subsequent requests will be rejected. The performance of the server can have a significant impact on the level of satisfaction of users accessing the web server.

A CSIM18/OptQuest model of a typical web server was implemented, to help configure the server. In particular, the server needs to be configured so as to deliver good service as the request rate for pages increases to levels projected for future operations.

The web server is modeled as a client-server system. Transactions are generated with exponentially distributed interarrival intervals. A new transaction must obtain a connection to the system. If a connection cannot be obtained within the time-out interval the transaction is "rejected".

A transaction has to obtain a file and send it back to the client. A file is on a disk unit. In some cases, the file is also in the file cache and the disk unit is not accessed. In other cases, the file is not in the cache and the disk unit must be accessed. The "cache hit rate" specifies the probability of finding a file in the file cache. Finally, the file is sent to a router via the local area network (LAN).

A configuration for this system model consists of the following:

- One or more CPU's; with additional CPU's, the instruction processing rate of each CPU decreases, because of interference between the CPU's in accessing main memory,
- One or more disk units,

- Main memory, the amount of main memory impacts the file cache hit rate, and
- A local area network, of a specified speed.

The parameters of this model are summarized in Appendix B.

The objective of the optimization was to find the system configuration with minimal cost subject to two requirements:

1. The overall throughput rate of completed transactions was greater than or equal to 15 transactions per second and
2. The number of transactions rejected (due to time-outs waiting to get connections to the server) must be less than five per cent of the number of transactions arriving at the server.

The first three configurations selected by the optimizer were as shown in Table 4.

Table 4: Results for First Three Configurations

Run	Cpu	Dsk	Hit	Net	Cost	Tput	Rej
1	3	9	0.85	1	21750	19.6	0.0
2	1	1	0.00	0	10750	16.8	0.9
3	4	16	0.99	2	29000	20.1	0.0

The second run is the least expensive configuration, but it is infeasible because over 90 per cent of the transactions arriving at the server timed out and were rejected.

The best configuration found in 50 trials is as shown in Table 5.

Table 5: Best Configuration Found in 50 Trials

Run	Cpu	Dsk	Hit	Net	Cost	Tput	Rj
33	2	16	0.00	0	15500	19.6	0.0

The optimizer correctly focused on trading disk drives for main memory (and a higher cache hit rate). Because of the cost factors for these components, the "best" system turned out to be the one with 16 disks and the smallest amount of main memory (with a cache hit rate of 0.0). The cost of the best system found ($15500) is 53 % of the cost of the most expensive system. This was found by evaluating 6.5 % of the 768 possible configurations.

Each configuration was evaluated using a simulation model of the system. Each trial ran until the mean transaction response times converged to a 90% confidence interval of width +/- 5 per cent of the average value (Schwetman and Brumfield 1997). The cpu time required for the 50 trials was 255.156 seconds on a 300 MHz. Pentium II processor.

The CSIM18/OptQuest program was also utilized to analyze the system for an increasing sequence of transaction arrival rates. The results are summarized in Table 6. It can be seen that as the arrival rate increases, the costs of configurations which can meet the performance requirement (less than 5 % rejects) also increases. This type of information can be of assistance as the managers of the server develop plans for dealing with increased demands for service.

Table 6: Best Configuration for Five Arrival Rates

Arr. rate	Cpu	Dsk	Hit	Net	Cost
10 tps	2	8	0.00	10	13500
20 tps	2	16	0.00	10	15500
30 tps	3	13	0.50	10	18250
40 tps	3	15	0.50	10	18750
50 tps	4	12	0.85	10	21500

6 SUMMARY

CSIM18/OptQuest is an integrated package, combining CSIM18, a powerful and flexible simulation engine, with OptQuest, an efficient, state-of-the-art optimizer. OptQuest guides the search for better system configurations, using a CSIM18 model to "evaluate" each of these configurations. For each model, a set of configurations, a set of requirement variables and an objective are specified. For each iteration, OptQuest selects a configuration and invokes the model to evaluate that configuration. The model returns to OptQuest the value of the objective and the values of the requirement variables. OptQuest determines whether or not the configuration is feasible (all of the requirements are met). If the configuration is feasible, then it is ordered among the other feasible configurations based on the value of the objective. OptQuest uses heuristics to select future configurations based on the ones evaluated so far,

Experience has shown that the sequence of configurations generated by OptQuest are much more efficient than random walks and exhaustive search. In many cases, evaluating less than 10 % of the complete space of configurations leads to configurations which are very close to the optimal configuration.

CSIM18 models usually execute quickly. Furthermore, the automatic run length feature in CSIM18, guarantees that a model runs for only a long as is necessary to reach pre-specified levels of confidence and accuracy. The availability of CSIM18/OptQuest means that a wider spectrum of problems for configuring complex systems can now be attacked.

ACKNOWLEDGMENTS

CSIM is copyrighted by Microelectronics and Computer Technology Corporation (MCC). CSIM18 is supported and marketed by Mesquite Software, Inc. under license from MCC. Dr. Jeff Brumfield developed the data collection and presentations functions including the run-

length control algorithm in CSIM18. OptQuest is a product of OptTek Systems, Inc. It is available for use with CSIM18 from Mesquite Software, Inc.

APPENDIX A: JOB SHOP PARAMETERS

The parameters of the model of the job shop are as shown in Table A-1.

Table A-1: Parameters for Job Shop Model

Number of workstations	5
Number of job types	3
Routings:	Workstations visited
Job type 1	3, 1, 2, 5
Job type 2	4,1,3
Job type 3	2,5,1,4,3
Service time distributions:	
2-Erlang, means are:	
Job type 1	0.50, 0.60, 0.85, 0.50
Job type 2	1.10, 0.80, 0.75
Job type 3	1.20, 0.25, 0.70, .0.90, 1.00
Relative frequencies	
Job type 1	0.3
Job type 2	0.5
Job type 3	0.2
Interarrival times	
Exponential, mean	0.275

APPENDIX B: WEB SERVER PARAMETERS

Some of the configuration parameters for the web server model are as shown in Tables B-1 – B-5.

Table B-1: Web Server Configuration Parameters

Component	CPU	MEM	DISK	NET
Min value	1	0	1	0
Max value	4	3	16	2
Cost – base	0	0	0	5000
Cost – incr.	1000	2500	250	2000

The file cache hit rate is a function of the amount of main memory in the system; these hit rates are as shown in Table B-2.

Table B-2: File Cache Hit Rates

Memory size	0	1	2	3
File cache hit rate	0.00	0.50	0.85	0.99

The cpu speeds degrade as more cpu's are attached to a system; this is caused by competition for access to main memory between the multiple cpu's. The cpu speeds, expressed in terms of millions of instructions per second (MIPS) are as shown in Table B-3.

Table B-3: CPU Speeds

Number of cpu's	1	2	3	4
MIPS per cpu	250	240	225	200

There are three networks which can be used; the properties of these are as shown in Table B-4.

Table B-4: Network Properties

Model number	0	1	2
Speed (Mbits/sec)	10	100	1000
Block size (bytes)	1500	1500	64000

Other parameters of the model are as summarized in Table B-5.

Table B-5: Other Parameters

Disk block time (sec)	0.015
Disk block size (bytes)	512
Trans. interarrival time (sec)	0.05
Connection time out (sec)	10.0
Number of connections	32
Blocks per trans – min.	1
Blocks per trans – mode	10
Blocks per trans. – max	100
Million cpu instrcs./block	0.5

REFERENCES

Glover, F. 1977. *Tabu Search*, Kluwer Academic Publishers.

Glover, F, J. Kelly and M. Laguna. 1999. New advances for wedding optimization and simulation, *Proceedings of the 1999 Winter Simulation Conference*, P. Farrington, H. Nembhard, D. Surrock and G. Evans (Eds.), 255 – 259. Piscataway, NJ: Institute of Electrical and Electronics Engineers.

Law, A. and D. Kelton. 1991. *Simulation Modeling & Analysis*, New York: MacGraw-Hill.

Schwetman, H. 1996. CSIM18 – the simulation engine, *Proceedings of the 1996 Winter Simulation Conference,* J. Charnes, D. Morrice, D. Brunner, and J. Swain (Eds.), 515 – 521. Piscataway, NJ: Institute of Electrical and Electronics Engineers.

Schwetman, H. and J. Brumfield. 1997. Data analysis and automatic run-length control in CSIM18. In *Proceedings of the 1997 Winter Simulation Conference*. Ed. S Andradottir, K. Healy, D. Withers, and B. Nelson, 687 - 692. Piscataway, NJ: Institute of Electrical and Electronics Engineers.

Schwetman, H. 1999. "Model, then build": a modern approach to systems development using CSIM18, *Proceedings of the 1999 Winter Simulation Conference*, P Farrington, H. Nembhard, D. Surrock and G. Evans (Eds.), 249-254. Piscataway, NJ: Institute of Electrical and Electronics Engineers.

Schwetman, H. 2000. Finding the best system configuration: an application of optimization and simulation, *Proceedings of the 2000 European Simulation Multiconference*, R.Van Landeghem (Ed.), 589-592, San Diego: Society for Computer Simulation.

AUTHOR BIOGRAPHY

HERB SCHWETMAN is founder and president of Mesquite Software, Inc. Prior to founding Mesquite Software in 1994, he was a Senior Member of the Technical Staff at MCC from 1984 until 1994. From 1972 until 1984, he was a Professor of Computer Sciences at Purdue University. He received his Ph.D. in Computer Science from The University of Texas at Austin in 1970. He has been involved in research into system modeling and simulation as applied to computer systems since 1968. He can be contacted at <hds@mesquite.com>.