Design of Intelligent Simulation Software with Capability of Optimization

A. Azadeh, M. Tabatabaei and A. Maghsoudi

Department of Industrial Engineering, Department of Engineering Optimization Research and Center of Excellence for Intelligent Based Experimental Mechanics, College of Engineering, University of Tehran, Iran

Abstract: This paper presents new object-oriented simulation software for optimization of queuing systems. It has a flexible structure both in simulation and optimization cores. In recent years several simulation packages have been introduced that use mathematical techniques for optimizing performance of production system. The major drawback of these optimizing packages is lack of open source structure which causes further studies and sensitivity analysis. The proposed simulation software has open source structure and allows researchers to easily deploy their optimization algorithm. The results of this software have shown good accuracy for predicting and optimizing the performance of queuing systems and by development the structure of model it can be used as a flexible tool for optimizing the performance of all kind of production systems.

Key words: Simulation; Optimization; Queuing Systems; Algorithm; Production Systems

INTRODUCTION

Optimization based on simulation is one of the most important technologies in the recent years. In the previous methodologies, integration simulation and optimization demanded to perform significant and difficult changes in the model that it was not economically possible for real cases especially for problems with a large response surface. One step on proceeding to increasing accuracy of the process was utilization of a special kind of algorithms that tested all possible situations. Although these algorithms guaranteed optimal solution, but there have been few problems. The matter was especially important whether the response surface was too large that it demands enormous time for solving these kinds of problems and so the efficiency of these algorithms was not considerable. Recent developments in the optimization methods have been caused to introduce number of intelligent searching processes (namely heuristic methods) that they can acquire the optimum or near optimum solution of large sophisticated problems by testing only a few possible solutions. In these new methods a set of constraints (that can be finite or infinite) are used in the problem that helps the optimizing algorithm to find the final solution. A number of most important methods that are available in the simulation packages are as below:

Evolution strategies (Back and Schwefel, 1993), Neural networks (Bishop, 1995; Haykin, 1999), Scatter search (Glover, 1999), Simulated annealing (Kors and Aarts, 1989) and Tabu search (Glover and Laguna, 1997; Dengiz and Alabas, 2000; Finke and Medeiros, 2002; Glover and Laguna, 1997). Also Genetic algorithm (Michalewicz, 1996) is another heuristic method that is used for optimization.

Nevertheless, several studies for improving the efficiency of these methods are performed by researchers of the filed of simulation optimization. One of the problems of these methods is the necessity of presenting a simulation software that accommodates its approach with regard to the optimization algorithm and indeed presentation of an optimization package based on simulation.

In recent years several simulation packages have been provided that use some mathematical techniques for optimizing performance of the problem. Major drawback of these optimizing packages is that it is not possible to study the manner of optimizing algorithms or change the structure of these algorithms to perform some researching work because these packages commonly don’t have an open-source structure. Also some of these simulation packages provide near optimum solutions that we can’t be sure about their optimality (Ahmed and Alkhami, 2002; Bowden and Hall, 1998; Fu, 2002; Pinto et al., 2000; Schriber and Brunner, 1997; Swisher et al., 2000). The objective of these routines is to search the best configuration of user defined input
parameters to optimize related criterion but in the current packages user doesn’t have any confidence about optimality of final solution. Indeed many of simulation optimization packages have just been presented from recent millennium (Law and Kelton, 2000; Law and McComas, 2002). For example AUTOSAT optimization package has been presented for AUTOMOD simulation software by AutoSimulation corp. this package uses genetic algorithm to reach the optimal solution. OptQuest is a well-known optimization packages that it can be linked to ARENA simulation software. OptQuest uses Tabu Search and neural networks for optimization. OPTIMIZ is another optimization software from VISUAL THINKING INTERNATIONAL corp. and uses artificial neural networks for optimization. Some other optimization softwares are SimRunner by PROMODEL, Extend Optimizer by PROMODEL, OptQuest by Imagine That and Optimizer by Lanner.

In this article we have proposed an optimizing software for queuing systems that has an object oriented structure and it has flexibility whether in simulation core an optimization core. The proposed simulation software has open source structure and allows researchers to easily deploy their optimization algorithm. Linking with this package is possible without encountering to redesign of new simulation software for analyzing the performance of algorithm. For designing this software, an optimization algorithm based on Tabu Search has been used to find the optimal or near optimal solution. The proposed software has a Graphic User Interface (GUI) that simplifies the problem definition and optimization processes of simulation model for the users. The results of this software have shown good accuracy for predicting and optimizing the performance of queuing systems and by development the structure of model it can be used as a flexible tool for optimizing the performance of all kind of production systems (Fu et al., 2000; Gehlsen and Page, 2001; Glover et al., 1996; Le Riche et al., 2003).

**Definition of the Problem:**

The general form of a sample of optimization problems based on simulation is as below:

Propose that \( V_1, V_2, ..., V_k \) are decision making variable for a simulation model. Also \( f(v_1, v_2, ..., v_k) \) is the value of random output function for simulation model with regard to the values of \( V_1 = v_1, V_2 = v_2, ..., V_k = v_k \).

Now the considered optimization problem is generally as below:

\[
\max E[f(v_1, v_2, ..., v_k)] 
\]

As \( p \) linear constraint as following is been:

\[
a_{11}v_1 + a_{12}v_2 + ... + a_{1k}v_k \leq c_1 \\
a_{21}v_1 + a_{22}v_2 + ... + a_{2k}v_k \leq c_2 \\
... \\
a_{p1}v_1 + a_{p2}v_2 + ... + a_{pk}v_k \leq c_p 
\]

So we want to maximize the value of objective function \( E[f(v_1, v_2, ..., v_k)] \) ( \( E \) is the mathematical mean of random variable \( f(v_1, v_2, ..., v_k) \) on the possible values that is confirmed in the existed constraints (the coefficients in the constraints are constants). The symbols of “Greater than” or “Lower than” and “Maximization” or “Minimization” of the objective function could be changed with regard to the purpose of the problem.

In the general, \( n \) different runs of the simulation should be performed for the situation of the system and the mean of \( f \) as an estimation of \( E[f] \) should be calculated and this is for the sake that \( f(v_1, v_2, ..., v_k) \) is a random variable.

**Introduction of the Software:**

The presented software in this article is an optimization software based on simulation for queuing systems. In this software the relationship with user is via a graphic user interface (GUI) in the environment of Windows that requested model is designed in this environment and then unrestrained variables are defined. Also related constraints for applying on the variables and the objective function for optimization is specified. Then the software via to consecutive and controlled runs of simulation finds suitable values of unrestrained variables that optimize the quantity of the related objective function. This approach is controlled by the optimization algorithm (Microsoft Corporation, 2003).

The parochial structure of different sections of the software and their relationships have been presented in Fig. 1.
GUI is a graphic environment that is used by user for creating required queuing model via graphic forms for nodes and networks of the model. Also the required information of each node and information related to variables, objective function and constraints are entered to the program from number of standard windows.

The control unit of the program receives the information from GUI unit and initializes the simulation core and optimization section with regard to this information. The operations of these units are controlled by control unit and finally the results are presented to the user by GUI as the outputs of the program.

The simulation core section performs the operation of queuing simulation by using defined classes in it that includes the classes of nodes, approaches and entities.

Section of creating random numbers, performs the task of creation of random number required by simulation core section.

Section of optimizing operation, directs the simulation operation by using the optimization algorithm that have been used in it and uses different simulation runs based on method of the algorithm for finding optimum solution.

The operational structure of optimization based on simulation in the software has been presented as Fig. 2.

As it is seen, in each stage of simulation operation, the simulation unit receives one simulation model and simulates it with regard to structure and parameters of the model. Finally the acquired results of a user requested output are presented (e.g. time of the system). Now the task of optimization algorithm is to receive the acquired results from the simulation unit and create a new model based on related algorithm and represent it to the simulation unit. This process is repeated until the optimization unit detects that the related model has been presented a satisfactory solution.

Several simulation optimization algorithms differs from each other in method of creating subsequent simulation models (method of moving in the response space) and also in method of stopping. In the presented software, most of the parameters have discrete structure so our selection is been from the methods that are used for discrete input parameters and with a large response surface. In this software the optimization algorithm is written with regard to the approach of Tabu Search optimization method. As mentioned earlier, the utilized algorithm in this software has capability of changing and converting to the other simulation optimization algorithms without any contrast in other sections of the system(Schriber and Brunner, 1997; Swisher et al., 2000).
Fig. 3: Main screen of the program

Fig. 4: Remove menu and changing the node in the network

Fig. 5: The window of adding and removing constraints
Fig. 6: The window of node specifications for Facility

**Capability for Using in the Future Researchs:**

The most important objective of designing this software is not only to present a simulation optimization package for queuing systems but also to preset a suitable field for testing different simulation optimization algorithms. For this reason, entire section related to optimization algorithm has been located in a separate class in the software that is connected to other sections of the program via only a few limited and general functions. Performing different changes in the optimization algorithm or replacing the entire algorithm is also possible by changing only one class. Hence this software can be considered as a suitable lab for using by researchers in the field of simulation-based optimization.

**Conclusion:**

Presented software in this article provides a suitable extent for researchers in the field of intelligent simulation with the vision of optimization based on simulation. Flexible structure and capability of changing optimization process in this software can help the researchers to improve searching methods. Moreover the software can be used as an appropriate optimizing tool for the users that want to improve the quality of their systems.

**REFERENCES**


