

PERFORMANCE ANALYSIS OF COMERCIAL SIMULATION-BASED OPTIMIZATION PACKAGES: OPTQUEST AND WITNESS OPTIMIZER

Hamidreza Eskandari
Ehsan Mahmoodi
Hamed Fallah

Christopher D. Geiger

Tarbiat Modares University
Advanced Simulation Lab
Dept. of Industrial Engineering
Tehran, 14117, IRAN

University of Central Florida
Dept. of Industrial Engineering and
Management Systems
Orlando, FL 32816, USA

ABSTRACT

The objective of this study is to evaluate and compare two commercial simulation-based optimization packages, OptQuest and Witness Optimizer, to determine their relative performance based on the quality of obtained solutions in a reasonable computational effort. Two well-known benchmark problems, the pull manufacturing system and the inventory system, are used to evaluate and compare the performance of OptQuest and Witness Optimizer. Significant validation efforts are made to ensure that simulation models developed in Arena and Witness are identical. The experimental results indicate that both optimization packages have good performance on the given problems. Both packages found near-global optimal (or satisfactory) solutions in an acceptable computation time.

1 INTRODUCTION

In general, simulation-based optimization is the process of searching for the best set of model specifications, *i.e.*, input parameters and structural assumptions, where the objective value is the output performance of simulation model for the underlying system (Swisher et al., 2000; Fu 2002; Olafsson and Kim 2002). Simulation-based optimization has been widely used in a variety of applications including manufacturing systems, project management (April et al., 2004), operations scheduling (Zeng and Yang 2009), inventory systems (Köchel and Nieländer 2005; Alferaei and Diabat 2009) and supply chain (Grewal 2010).

In the simulation context, an optimization algorithm should be robust enough to handle the challenges of randomness and uncertainty involved in the estimated objective functions of the simulated model. In this case, the existing uncertainty and noise might hinder the optimization algorithm trying to move into improving directions. It is worthy to note that there is a considerable gap between the approaches proposed in the research literature and those that are employed by commercial software packages for practical use (Fu 2002; Olafsson and Kim 2002). There are a few optimization packages typically based on heuristic search algorithms. However, one of the important phases of problem solving for decision makers and practitioners is to choose an efficient and affordable optimizer among different existing optimization packages.

The literature of performance comparison of commercial simulation optimization packages is not so rich, but comparison of heuristic search algorithms has been an active area for many researchers. Jafferli et al. (2005) compared two commercial optimizers OptQuest and SimRunner with respect to the quality of results and computational time for a family of manufacturing system problems. They found that SimRunner produced better results when the computation time is limited while OptQuest produced comparable, sometimes superior results when computation time is considerable.

As mentioned above, performance evaluation and comparative analysis of commercial simulation-based optimization packages have rarely been considered in the literature despite its importance and viability. The objective of this study is to evaluate and compare two commercial optimization packages, OptQuest and Witness Optimizer, to determine their relative performance based on the quality of obtained solutions in a reasonable computational effort. To perform this evaluation and comparative analysis, two generic real world problems, the pull manufacturing system and the inventory system, are modeled using Arena and Witness simulation software packages.

The remainder of this paper is as follows. In Section 2, we have a review on available commercial simulation-based optimization packages and OptQuest and Witness Optimizer are briefly introduced. In Section 3, two generic test problems, the pull manufacturing system and the inventory system, are described. The experimental design is explained in Section 4. Section 5 presents the experimental results and discussion and finally conclusion is given in Section 6.

2 SIMULATION OPTIMIZATION PACKAGES

Contemporary simulation software products employ optimization packages embedded in them for making the optimization of the stochastic simulation models possible. According to Law (2007) current optimization packages include AutoStat®, Extend Optimizer®, OptQuest®, SimRunner2® and Witness Optimizer® which use different search strategies (Table 1). In this study, we investigate the performance of OptQuest and Witness Optimizer packages for a couple of simulation problems.

Table 1: Commercial optimization packages embedded within simulation software products (Law 2007)

Optimization Package	Vendor	simulation software	Search strategy
AutoStat	Brooks Automation	AutoMod, AutoSched	Evolution strategies
Extend Optimizer	Imagine That	Extend	Evolution strategies
OptQuest	OpTek, Inc	Arena, Flexsim, Promodel, Quest, Simul8	Scatter search, tabu search and neural networks
SimRunner2	Promodel Corp	Promodel	Evolution strategies, genetic algorithms
Witness Optimizer	Lanner Group Inc	Witness	Simulated annealing and tabu search

2.1 OptQuest

The OptQuest package employs three well-known search heuristics, scatter search (SS) as its main search strategy, tabu search (TS) as secondary method and neural networks (NN) as the last method. SS applies heuristic processes to generate a starting set of solution vectors and designate a subset of best vectors to be reference solutions. Then the algorithm forms the linear combination of subsets of current reference points and generates new points. In the next step, the SS algorithm selects a combination of the best solutions use them as starting points for a new application of the heuristic processes and repeat these steps until a specified number of iteration or reach stopping criteria. TS uses adaptive memory to prohibit the search from reinvestigating solutions that have already been evaluated and to guide the search to a globally optimal solution. Neural network is used to screen out solutions that are likely to be poor without allowing the simulation to evaluate them. The neural network is used as a prediction model to help the system accelerate the search by avoiding the need for evaluating objective function for a newly created reference point, in situations where the objective value can be predicted to be of low quality.

As described by Law (2007) OptQuest uses three stopping rules:

- a) Run for a user-specified maximum number of configurations (*MNC*).
- b) Run until there is no improvement in the value of the objective function for 100 consecutive configurations (called *automatic stop* in OptQuest).

- c) Combination of both above rules; in this case optimization run until the number of nonimproving configurations equals 5 percent of *MNC*.

OptQuest has the following features (Law 2007, page 661):

- Linear and nonlinear constraints on decision variables and on output random variables.
- An option to allow a variable number of replications for each configuration depending on its mean and variance estimates.
- Ranking and selection method to provide more confidence on the selection of the best solution reported by OptQuest

2.2 Witness Optimizer

Witness Optimizer uses Adaptive Thermo-statistical Simulated Annealing (ATSA) algorithm (Debusse et al. 1999) which is based on the simulated annealing (SA) and elements of tabu search are incorporated into the SA process which, by learning from its experience of the problem domain and modifying its search strategy accordingly. ATSA employs an adaptive cooling schedule which is an encapsulation of expert advice on practical optimization using the SA.

Witness Optimizer uses three stopping rules:

- a) Run for a user-specified maximum number of configurations (*MNC*).
- b) Run until the number of consecutive configurations for which there is no improvement (*NCI*) in the value of the objective function reached
- c) Run all feasible configurations.

Note that the first stopping rule is exactly and the second one is somehow the same as OptQuest. However, the last stopping rule is useful when the number of feasible configurations is reasonable or when we are interested in finding a global optimum solution. Witness Optimizer has the following features (Law 2007, page 661):

- Allows linear constraints on decision variables.
- Calculates the number of feasible configurations for an optimization problem, even when constraints have been specified.
- Provides a mechanism for evaluating the replication-to-replication variability of the objective function for a particular configuration.

3 PROBLEMS DESCRIPTION

Two well-known benchmark problems, the pull manufacturing system with seven decision variables and the inventory system with two decision variables, are used to compare the performance of OptQuest and Witness Optimizer.

3.1 Pull Manufacturing System

Consider a manufacturing line consisting of four workstations, three finite size buffers, and an infinite supply of blank parts, as shown in Figure 1. Each station does a specific process on a part and then the part goes to the next station until it exits the system. The processing time for each machine in the station 1 to 4 is exponentially distributed with the mean of 0.333, 0.500, 0.200, 0.250 hours respectively. When a machine at station 1, 2 and 3 finishes its current part, it will push it to the next buffer, unless all positions in the buffer are occupied.

There are seven decision variables in this problem: the number of machines in each work station and the number of positions in each buffer. Let v_i be the number of machines in station i (for $i=1, 2, 3$ and 4) and b_i the number of positions in buffer i (for $i=1, 2$ and 3). We are interested in determining the number of v_i and b_i to maximize the expected profit for a 30 day period. The company receives \$200 for each part that it sells. It costs \$25000 to use a machine at a station and \$1000 for a buffer position for 30 days. The company wants to decide about buying between 1 to 3 machines for each station and 1 to 10 buffer

positions for each buffer. Thus, there are $3^4 \cdot 10^3 = 81,000$ different combinations of seven decision variables.

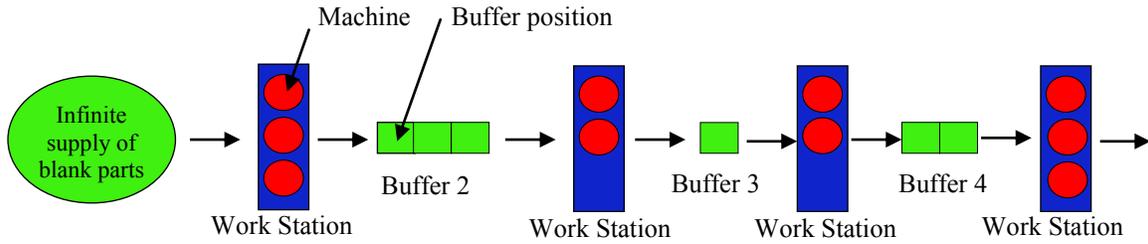


Figure 1: Manufacturing system with four work stations and three buffers (Law and McComas 2002)

The profit for a 30 day period, $R(v_1, \dots, v_4, b_1, b_2, b_3)$, is calculated as follows:

$$R(v_1, \dots, v_4, b_1, b_2, b_3) = 200N - 25000 \sum_{i=1}^4 v_i - 1000 \sum_{i=1}^3 b_i, \tag{1}$$

where N is the number of output parts. We want to maximize expected value of $R(v_1, \dots, v_4, b_1, b_2, b_3)$, subject to range constraints:

$$\begin{aligned} 1 &\leq v_i \leq 3 \\ 1 &\leq b_i \leq 10. \end{aligned}$$

In this problem, the objective function is actually to maximize the average profit which is a function of the number of parts produced (the throughput), the total number of machines and the total number of buffer positions. As reported by Law and McComas (2002) it is *apparently* possible to determine the true optimal solution for this problem, since it would appear that 3, 3, 2, and 2 machines at stations 1, 2, 3, and 4, respectively, are optimal. Therefore, the number of machines in each work station is fixed and the number of positions for each of the three buffers can be determined. Witness Optimizer is an appropriate tool for this purpose, since it allows one to do an exhaustive enumeration of all possible configurations. Our experimentation indicates that 8, 9, and 4 positions at buffers 1, 2, and 3, respectively, are optimal, which results in the average profit of about \$588,000 making 1,000 independent replications.

3.2 Inventory System

For the second problem, consider an inventory system with a single item using stationary (s, S) policy (see Law 2007). If the inventory level reaches to reorder level s , we should put an order up to a maximum inventory level S ($s < S$). The lead-time has a uniform distribution between 0.5 and 1. The demand is 1, 2, 3, and 4 with the probabilities 0.137, 0.333, 0.333, and 0.137. The reorder level varies between 0 and 99 and the maximum inventory level varies between 1 and 100 led to 5,500 feasible configurations. Every time an order is placed, there is a cost incurred of \$32 regardless of the order quantity, plus \$3 per item ordered. Whenever there are items actually physically in inventory, a holding cost is incurred of \$1 per item per day, and whenever we are in a backlog, a shortage cost of \$5 per item per day is incurred.

For the inventory model, the objective is to minimize the average total cost, which is the sum of average holding cost, order cost and shortage cost. Having employed the Witness Optimizer, all 5,050 feasible configurations are evaluated, and our results indicate that the optimal solution is (23, 59) policy which results in the average total cost of \$118.26 making 1,000 independent replications.

4 EXPERIMENTAL DESIGN

This section describes the procedure employed to perform a fair comparative analysis between the performance of two optimization packages, OptQuest and Witness Optimizer. To have a fair performance

comparison, we need to make sure that not only both simulation models developed in Arena and Witness is the same, but also the optimization setting is identical for both optimization packages.

4.1 Simulation Models Development

The most challenging task of this study was to develop identical simulation models in Arena® and Witness® software products. Because of the difference in the modeling aspects and features of two simulation software products, this task is very challenging and time consuming. For each simulation model we need to check many replication and setup settings including warm-up period, replication length, time units, system reinitialization, and random number generation scheme which may have different definitions in Arena and Witness. For example, in Arena, warm-up period is part of the replication length, but in Witness, it is not.

After all replication and setup settings are checked in both simulation software products, for each model the output statistics of several configurations obtained from Arena and Witness are compared. There should not be any significant statistical difference between their output statistics when each model is run for a large number of replications. For better understanding, consider the pull manufacturing system with three different configurations as described in Table 2. Having been performed two-sample- t approach (with $\alpha=0.05$), it is concluded that with 95% confidence there is no significant difference between the throughput (number of parts produced) and the average total profit statistics obtained from Arena and Witness. Our careful investigation, described in Section 6.1, indicates that the small difference between their output statistics is due to randomness effect caused by generation of different random number streams.

Table 2: Comparison of the pull manufacturing simulation models with 1,000 replications.

Configuration	Decision Variable							Output Statistics			
	M1	M2	M3	M4	B1	B2	B3	Throughput		Average Profit	
								Arena	Witness	Arena	Witness
Config. 1	3	2	2	3	3	1	2	2829.85 ± 3.05	2828.31 ± 3.25	309969.8 ± 610.5	309961.5 ± 648.2
Config. 2	2	2	2	2	6	6	6	2854.95 ± 2.97	2852.40 ± 3.25	352990.4 ± 594.7	352480.4 ± 649.4
Config. 3	3	3	2	2	6	6	6	4270.03 ± 3.72	4268.96 ± 3.84	586005.2 ± 743.8	585791.4 ± 767.7

4.2 Experimental Settings

According to Barr et al. (1995), the factors affecting the performance of the optimization algorithm are divided into three categories: 1) problem nature, 2) algorithm setting, and 3) environmental factors. Problem nature is characterized by the problem response surface shape, and the number and type of the decision variables (Boolean, discrete, integer or real). Algorithm setting consists of algorithm parameters such as population size, crossover rate and mutation rate for genetic algorithms, stopping criterion and initial solution. Environmental factors include hardware factors such as CPU speed, processor type and memory and software factors such as operating system. To provide a fair comparison, all algorithm factors and environmental factors are kept the same and we investigate how each optimization algorithm performs for two simulation problems.

Unlike the performance analysis of optimization algorithms in deterministic problems context, we add the level of random noise as a new influential factor affecting the performance of the optimization algorithms. In the simulation context, the objective functions do not take deterministic values and at each replication due to the randomness effect, they take random output. This random output makes the task of the optimization algorithm more difficult. Each algorithm in the simulation-based optimization process typically has its own approach to deal with the noisy objective values. Thus, it is an important aspect to

investigate how robust each optimization algorithm is against different levels of noise. For this purpose, we evaluate each optimization algorithm for different number of replications (10, 50 and 100) to produce different levels of noise. To make the levels of noise identical for both algorithms, in all experimentation the number of replications is kept the same for both models. In other words, each simulation model in Arena and Witness is run for the same number of replications as clearly reported in the following tables and figures.

5 EXPERIMENTAL RESULTS AND DISCUSSION

Several experimental runs are performed to compare the performance of OptQuest and Witness Optimizer packages, and the performance measures are tabulated for pull manufacturing and inventory systems. Since we are dealing with computationally expensive problems in simulation context, the computational CPU time is not considered in this study. It is very important to understand the random output problem typically encountered within simulation experimentation. Thus, let us briefly review the random output problem before we illustrate the experimental results.

5.1 Random Outputs

We know that because of randomness effect of stochastic processes within a simulation model, the output statistics are typically random. The phenomenon of random outputs makes the task of fair comparison among different configurations of a simulation model or validating the possession of the same simulation model in different software products very difficult. In output analysis, we know that when the number of replications in the simulation model execution is increased, the mean value of outputs may change (usually becomes closer to the true mean) and the variability degree of outputs (half-width) is decreased. For better understanding, the results of running the inventory model with near-optimal solution ($s = 24$ and $S = 63$) for different number of replications in both Arena® and Witness® is shown in Figure 2. It is seen that for each comparison with the given number of replications, we have no idea how much the mean values obtained from Arena and Witness are close to the true average total cost of \$118.26 (shown in blue dashed line). All we can say is that the confidence intervals become smaller and the mean values usually move closer to the true cost of \$118.26 when the number of replications is increased.

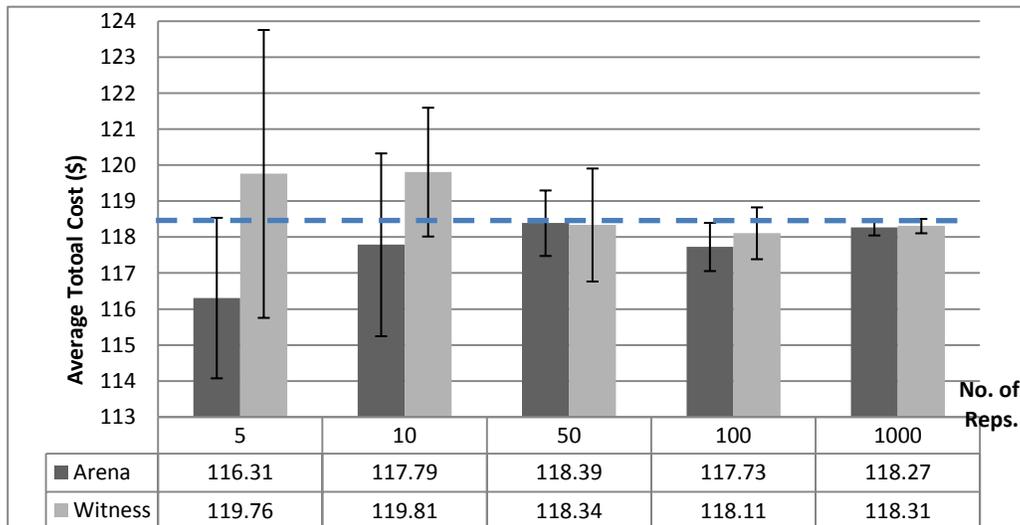


Figure 2: Average total cost of inventory system given by Arena and Witness for different number of replications.

Note that because of different random number streams used in Arena and Witness, there is considerable difference between their mean values, particularly when the number of replications is small (5 and 10). Here, we observe that there is about 2\$ difference between the average total cost obtained from Arena and Witness for the same solution ($s = 24$ and $S = 63$). Therefore, it is important to note that when we compare the performance measure values of different near-optimal solutions obtained from OptQuest and Witness Optimizer, we could not realize whether the difference between the performance values is due to the random error or it is because of any algorithm outperformance over the other one. Now, the question is how we can conduct a fair comparison between two optimization algorithms while there is considerable noise (error) in the outputs obtained from Arena® and Witness®.

To tackle the random output problem encountered within experimentation, we reevaluate different near-optimal solutions obtained from OptQuest and Witness Optimizer for a large number of replications (1000 replications in this study) using one of the simulation software products (Arena in this study). This approach helps us significantly reduce the disturbance of random output and remove the potential nuisance factors of using different software products.

5.2 Pull Manufacturing System Results

Figure 3 illustrates the search progress of two optimization packages, OptQuest and Witness Optimizer, for the pull manufacturing problem. Both algorithms start from almost the same point using initial solution with profit of about \$353,000 and they stop after 100 consecutive configurations for which there is no improvement in the average profit. Both algorithms are so competitive within the whole search process. It is observed that both have a sudden profit increase at the early stage of the search. In Figure 3 (a) (search with 10 replications), it is observed that OptQuest and Witness optimizer find the best solution at configurations 117 and 137, respectively. It can be seen from Figure 3 (b) (search with 100 replications) that average profit does not increase much beyond configurations 22 and 57, using OptQuest and Witness optimizer, respectively. For manufacturing problem, we observe that Witness optimizer takes more aggressive search, but in less than 150 configurations evaluation both packages become very close to the true optimal average profit of about \$588,000.

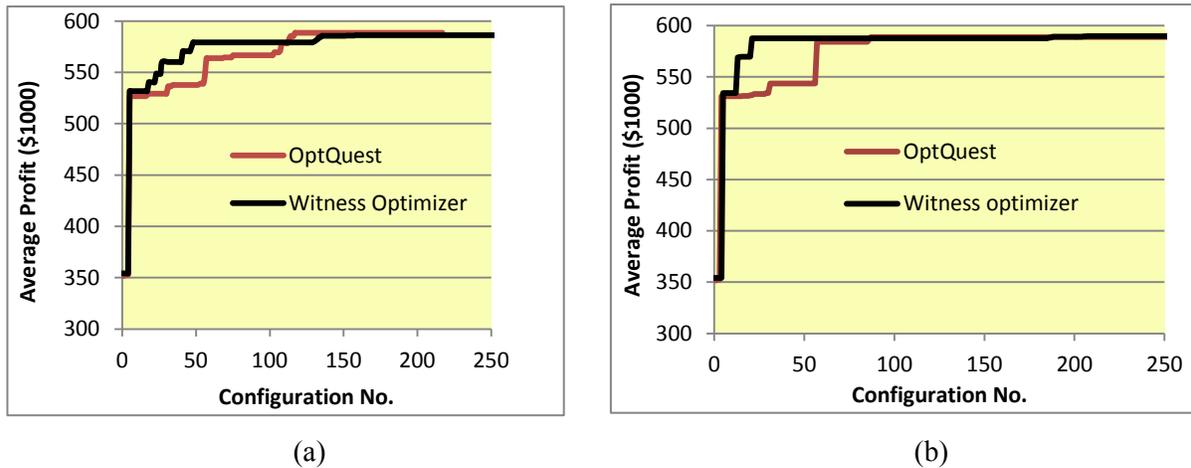


Figure 3: Maximum average profit at configuration i for the pull manufacturing system,(a) 10 replications, (b) 100 replications.

Table 3 shows the optimal solutions obtained from the pull manufacturing system for different number of replications. We see solutions identified by both optimization packages are very close to the true optimal solution, and the solutions have the same values for the first four decision variables (number of machines at each station). However, each package for each number of replication find slightly different values for the last three decision variables (number of buffer positions). Recall that for this problem the

number of machines at each station is more important decision variable. Here, we observe that due to random output problem the average profit obtained by Arena with 10 replications (\$591,920) exceed the true optimal value for more than \$3900.

Table 3: The best solutions obtained for the pull manufacturing system at 10, 50 and 100 replications.

Optimization Package	No. of Reps.	Optimal Solutions							Average Profit (\$) (run in Arena)
		M1	M2	M3	M4	B1	B2	B3	
OptQuest	10	3	3	2	2	8	7	4	591,920 ± 6,211
	50					6	10	4	588,752 ± 4,066
	100					8	9	4	588,764 ± 2,539
Witness Opt.	10	3	3	2	2	7	7	5	579,540 ± 11,488
	50					8	7	4	587,564 ± 3,388
	100					7	6	7	585,668 ± 2,039

As described in Section 6.1, to get rid of random output problem resulting in a more reliable comparative analysis, Figure 4 depicts the average profit of the manufacturing system optimal solutions run in Arena for 1,000 replications. It is interesting to note that the best solutions obtained by OptQuest and Witness Optimizer have very close average profit with considerable overlap between their confidence intervals using 10 and 50 replications. The two-sample-*t* test with $\alpha = 0.05$ indicates that there is no difference between them. However, the two-sample-*t* test, as it can be seen in Figure 5, indicates that the best solution found by OptQuest has higher average profit than that found by Witness Optimizer.

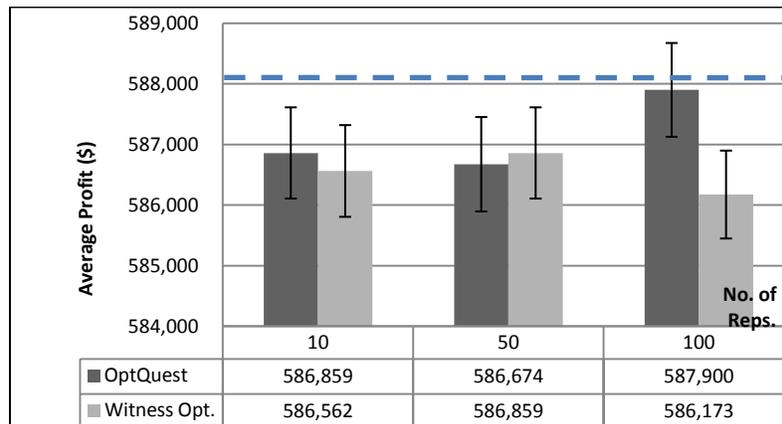


Figure 4: Average profit of the pull manufacturing system optimal solutions run in Arena for 1,000 replications.

5.3 Inventory System Results

Figure 5 illustrates the search progress of two optimization packages for the inventory problem. Both algorithms start from the same initial solution ($s=20$ and $S=40$) with average total cost of about \$125 and they stop whenever there is no improvement in the 100 consecutive configurations. In Figure 5 (a), search with 10 replications, it is observed that OptQuest and Witness optimizer find identical best solution ($s=23$ and $S=64$) at configurations 38 and 77, respectively. It can be seen from Figure 5 (b), search with 100 replications, that OptQuest and Witness optimizer find their best solutions at configurations 131 and 63, respectively. Similar to the manufacturing problem, we observe that Witness optimizer here takes more aggressive search, but in less than 100 configurations evaluation both packages become very close to the true optimal average total cost of about \$118.3. Note that in neither Figure 5 (a) nor Figure 5 (b) we can conclude that one algorithm is superior than the other, since the small difference between their average total costs could be due to the random output. In addition, for the search with 10 replications, both

algorithms find the same best solution (look at Table 4) so the difference of about \$0.8 between their costs shown in Figure 5 (a) is definitely because of random output.

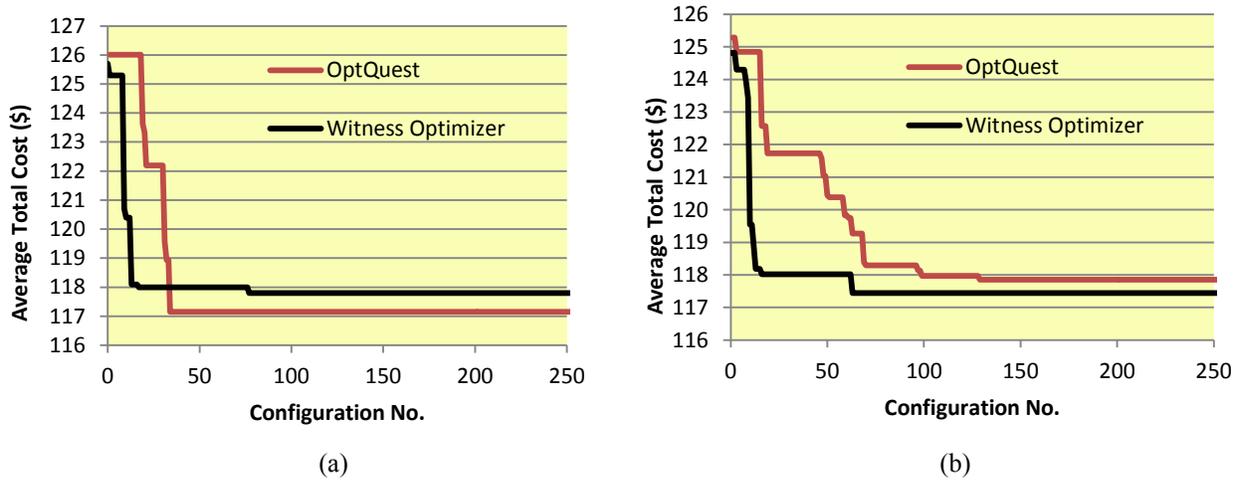


Figure 5: Minimum average total cost at configuration i for the inventory system,(a) 10 replications, (b) 100 replications.

Table 4 presents the best solutions obtained from OptQuest and Witness optimizer for the inventory system at 10, 50 and 100 replications. We see solutions identified by both optimization packages are very close to the true optimal solution, and the solutions have very close values for both decision variables. The average total cost statistics, given in Table 4, are obtained from their corresponding solutions run in Arena for the given number of replications. However, when these solutions are run in Arena for 1,000 replications, more precise statistics for their average total cost are provided as shown in Figure 6. As the considerable overlaps between the solutions’ confidence intervals shows, the two-sample- t test indicates that there is no difference between the obtained best solutions.

Table 4: The best solutions obtained for the inventory system at different number of replications

Optimization Package	No. of Reps.	Decision Variables		Average Total Cost (\$) (run in Arena)
		Little s	Big S	
OptQuest	10	23	64	116.94 ± 1.94
	50	24	63	118.39 ± 0.91
	100	23	59	117.89 ± 0.55
Witness Opt.	10	23	64	116.94 ± 1.94
	50	25	61	117.85 ± 0.75
	100	24	63	117.73 ± 0.67

6 SUMMARY AND CONCLUSIONS

The performance of two optimization packages, OptQuest and Witness Optimizer, are compared solving two benchmark problems. A pull manufacturing problem with different number of machines and buffers at each work station, and an inventory system with different values for its ordering policy were considered. Each optimization algorithm was evaluated for different number of replications to investigate how robust each algorithm is against different levels of noise. To tackle the random output problem in simulation context, near-global optimal solutions obtained from OptQuest and Witness Optimizer were reevaluated for 1,000 replications using Arena software.

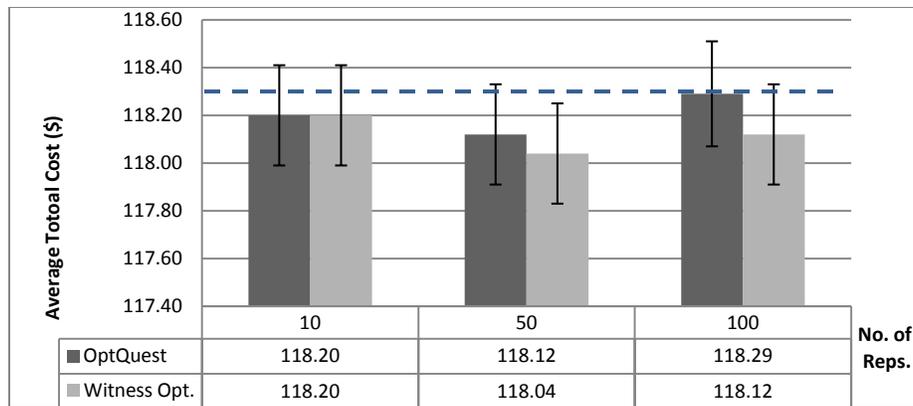


Figure 6: Average total cost of best solutions for the inventory system run in Arena for 1,000 replications.

Our experimental results indicate that the performance of both optimization packages is good for these two problems. Both packages found near-global optimal solutions in an acceptable computation time. Witness Optimizer appears to take more aggressive search resulting in higher convergence speed and it could therefore provide a better quality solution under limited number of solutions evaluations. However, OptQuest is capable of providing slightly better quality solution when a large number of solutions evaluations is allowed. One should take into consideration that we could not make any conclusion from these results about the superiority of one algorithm over the other in optimizing different simulation models.

REFERENCES

- Alferaei M.H., and A.H. Diabat. 2009. A Simulated Annealing Technique for multi-objective simulation optimization. *Applied mathematics and computation* 215: 3029-3035.
- April, J., M. Better, F. Glover, and P.J. Kelly. 2004. New advances and applications for marrying simulation and optimization. In *Proceedings of the 2004 Winter Simulation Conference*, edited by R. G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, 80-86. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Debuse, J.C.W., V.J. Rayward-Smith, and G.D. Smith. 1999. Parameter Optimization for a Discrete Event Simulator. *Computers & Industrial Engineering* 37:181-184.
- Fu, M. 2002. Optimization for Simulation: Theory and Practice, *INFORMS Journal on Computing* 14(3): 192-215.
- Grewal, C.S., P. Rogers, and S.T. Enns. 2010. Performance evaluation of inventory replenishment strategies in a capacitated supply chain under optimal parameter settings. *Int. J. Value Chain Management* 4(3): 195-212.
- Jafferli, M., J. Venkateshwaran, and Y.J. Son. 2005. Performance Comparison of Search-Based Simulation Optimization Algorithms for Operations Scheduling. *International Journal of Simulation and Process Modelling* 1: 58-71.
- Köchel, P., and U. Nieländer. 2005. Simulation-Based Optimization of Multi-Echelon Inventory Systems. *International Journal of Production Economics* 93: 505-513.
- Law, A.M. 2007. *Simulation Modeling and Analysis*. 4rd Edition. McGraw-Hill, New York.
- Law, A.M., and M.G. McComas. 2002. Simulation Optimization: Simulation-Based Optimization. In *Proceedings of the 2002 on Winter Simulation Conference*, edited by E. Yücesan, C. H. Chen, J. L. Snowdon, and J. M. Charnes, 41-44. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

- Olafsson, S. and J. Kim. 2002. Simulation optimization. In *Proceedings of the 2002 Winter Simulation Conference*, edited by E. Yücesan, C. H. Chen, J. L. Snowdon, and J. M. Charnes, 79-84. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Swisher, J.R., P.D. Hyden, S.H. Jacobson, and L.W. Schruben. 2000. A Survey of Simulation Optimization Techniques and Procedures. In *Proceedings of the 2000 Winter Simulation Conference*, edited by J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick, 119-128. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Zeng, Q., and Z. Yang. 2009. Integrating simulation and optimization to schedule loading operations in container terminals. *Computers and Operation Research* 36: 1935-1944.

BIOGRAPHY

HAMIDREZA ESKANDARI is an Assistant Professor of Industrial Engineering and director of Advanced Simulation Lab at the Tarbiat Modares University, Tehran, Iran. He received his Bachelor's degree in Electrical Engineering from the University of Tehran (1998), his Master's degree in Socio-Economic Systems Engineering from the Iran University of Science and Technology (2001) and his Ph.D. in Industrial Engineering from the University of Central Florida (2006). His research interests include simulation modeling and analysis, simulation optimization, and evolutionary multi-objective optimization. His email address is eskandari@modares.ac.ir.

EHSAN MAHMOODI is MSc student of Industrial Engineering at the Tarbiat Modares University, Tehran, Iran. He received his bachelor's degree in Chemical Engineering from Mazandaran University (2008). His research interests are mathematical modeling, simulation modeling and simulation optimization. His email address is ehsan.mahmoodi@modares.ac.ir.

HAMED FALLAH is a PhD student of Industrial Engineering at the Tarbiat Modares University, Tehran, Iran. He received his M.Sc. and B.Sc. degrees in Industrial Engineering from Amirkabir University of Technology and Khaje Nasir Toosi University, respectively. His research interests include simulation optimization, simulation modeling and analysis, supply chain network. His email address is hamed.fallah@modares.ac.ir

CHRISTOPHER D. GEIGER is an Associate Professor in the Department of Industrial Engineering and Management Systems at the University of Central Florida in Orlando, FL. His research and teaching interests include simulation modeling and analysis, machine learning in production planning and scheduling and heuristic optimization. He is a member of IIE, INFORMS and SME. His e-mail address is cdgeiger@mail.ucf.edu.