Optimization of simulated systems

Kleijnen, J.P.C.; Wan, J.

Published in:
Simulation Modelling Practice and Theory

Publication date:
2007

Simulation for the Optimization of (s, S) Inventory System with Random Lead Times and a Service Level Constraint by Using Arena and OptQuest

(Date: 18 June, 2006)

Jie Wan[1, 2], and Jack P.C. Kleijnen[3]

[1] Operation Research and Logistics Group, Wageningen University and Research Centre, Hollandseweg 1, 6706 KN Wageningen, Netherlands

[2]Department of Industrial Engineering, School of Management, Hebei University of Technology, 300130, Tianjin, China

[3]Department of Information System and Management/Center for Economic Research, Tilburg University, Postbox 90153, 5000 LE Tilburg, Netherlands

Abstract In this paper, we consider the simulation of constrained optimization problem, the (s, S) inventory system with stochastic lead time and a service level constraint. We allow the orders to cross in time which makes the problem more complicated. Bashyam and Fu (1998) first present this problem and obtained the answer by using Perturbation Analysis. Angun, Gurken, Hertog and Kleijnen (2006) studied the same question by using Response Surface Method. The motivation for our work comes from the difference answers between them for the same model under the same situations. We establish the (s, S) inventory model by using Arean and find the estimators by OptQuest. We try to solve several issues: what the true optimal values of (s, S) are in this specified conditions; whether the OptQuest can find the optimal values more efficiently; how we can prove these different outcomes are the estimators of true optimal values and which one is better. In order to identify the best estimator, we test their KKT conditions by applying two methods: small sample size procedure and large sample size procedure. In our conclusion, we give the true optimal estimator of (s*, S*) pairs estimated by Brute Force and prove that OptQuest can be used in solving the stochastic constrained optimization problem and find the near optimum effectively. Further, we point out that the outcome obtained from Bashyam and Fu is the estimator near the true optimal value, but not as close as the one gained by OptQuest, while the result of Angun et al (2006) gained is far away from the optimum. Furthermore, we also prove that the rejection probability for each null-hypothesis obtained by the KKT testing procedure with large example size is more obvious than that of small example size.

Key words simulation, (s, S) inventory systems, random lead times, service level constraint, Arena, OptQuest, KKT test for multiresponse

1 INTRODUCTION

Many inventory systems are controlled by an (s, S) stocking policy, which is characterized by a reorder point s, and an order-up-to level S. This paper presents a simulation model and statistical testing of Krush-Kuhn-Tucker (KKT) optimality conditions for the computation of (s, S) values which minimize expected costs with a service constraint . We consider the constrained optimization problem of (s, S) inventory policy with random demand and stochastic lead times subjected to a service level requirement. A major assumption same as Bashyam and Fu (1998) in the analysis of (s, S) inventory system is that orders are not received in the same sequence as they placed which complicated the analysis. We propose to establish the Arena model and estimate the optimal value of s and S with OptQuest, the add-on parts of Arena.

An (s, S) inventory policy has been focusing on by researchers for a long time. Many approaches have been taken to compute optimal or near-optimal inventory levels for the (s, S) inventory system with different environment of interests. For example, The review time could be continuous or periodic, the demand size might be deterministic or stochastic; the lead time could be zero, constant or random; research object could be one item or multi-items, consideration of the service level constraint or not; it is also possible to consider the different penalty cost of stock-out due to the complication of estimating unsatisfied demand; it could be lost sale or backlogging; whether it is allowed for the crossing sequence of delivering, and so on.

Generally, depending on the different assumptions, the huge body of literature can be classified into two types. Some literatures treat the inventory system as a 'white box'; i.e., built a mathematical model and examine it to get an exact, analytical solution to see how it can be used to answer the questions. Examples of these studies include: Archibald (1978), Ehrhardt (1979, 1981), Porteus (1979, 1985), Freeland and Porteus (1980), Tijms and Groenevelt (1984), and Federgruen and Zipkin (1984), Cohen, Kleindorfer and Lee (1988), Hu (1992), Fu and Hu (1993), Fisher and Hornstein (2001).

When some analytical solutions become extraordinarily complex, some methods treat the inventory system as a 'black box', study it by means of simulation, i.e., numerically exercising the model for inputs in question to see how they affect the output measures of performance. There are many methods for optimizing simulated systems (see Fu (2002) and Spall (2003)). Some of these optimization methods assume that no gradient information is available. Examples are the many metaheuristics (ant colony optimization, genetic and evolutionary algorithms, scatter search, simulated annealing, tabu search), simultaneous perturbation stochastic approximation (Spall (2003)), and Response Surface Methodology (RSM). Other methods estimate the gradients from a single simulation run—best known are perturbation analysis and the score function method. Our method to optimizing simulated system is to use OptQuest, which combines the metaheuristics of Tabu search, neural networks, and scatter search into a single, composite search algorithm to provide maximum efficiency in identifying new scenarios (Kelton, Sadowski and Strurrock (2004)). OptQuest uses search outputs as self-learning aids to seek the next set of alternatives. If an alternative in its search space does not fit the constraints defined, it is automatically eliminated, and better alternatives that are more likely to match needs are explored. It allows to explicitly define integer and linear constraints (such as budget limits, space restrictions, and workforce allocations), as well as boundaries on the output functions. It also includes logical conditions to better refine the search.

But as we know, although there are a huge body of literature that has been developed on searching the optimal value of s and S, the literature on constrained optimization is more sparse (Bashyam and Fu1998). A penalty cost, the cost to losing customer goodwill, is used to make the problem becoming unconstrained one. A review of a large number of optimal and approximately optimal algorithms to carry out this unconstrained optimization can be found in Porteus (1988). However, even in the unconstrained problem, in order to achieve some degree of tractability, almost all analytical models have assumed that lead times for orders do not cross in time, i.e., the orders arrive in the same sequence as they are placed. This assumption is not always being hold in the real world when the lead time is random, especially when multiple suppliers are involved.

Due to the difficult assessing of unsatisfied demand, more and more people tend to replace penalty cost by service level measures in practise. Under this environment, the constrained problem becomes to determine the s, S pairs to minimize a cost function, defined only in terms of setup and holding costs, subject to prescribed level of service level. Unfortunately, it is still difficult to define the distribution of backlogged demand in an analytical viewpoint.

A review of the literature on this issue shows that there are two most relevant works: Bashyam and Fu (1998) and Angun, Gurken, Hertog and Kleijnen (2006). Bashyam and Fu determine values of s and Q=S-s that minimize the average setup cost and holding cost per period, subject to the constraint that the fill-rate is above a prescribed level. Due to the analytic intractability caused by the order crossing, they adopt a simulation-based approach to the problem. The perturbation analysis method is applied to estimate the gradients of the costs and fill rate with respect to s and Q, and the feasible directions method from nonlinear programming is used to search for optimum. By contrasting their outputs of computational experiments, within 5% of optimality in 95%, with that of analytical methods, 10% for over 75% of the cases, they believe that they offer a computationally viable algorithm to handle the case of order crossing while analytical models perform poorly.

Angun, Gurken, Hertog and Kleijnen (2006) investigate simulation-based optimization problems with a stochastic objective function and stochastic constraints, besides deterministic input box constraints. More specifically, it generalizes RSM to account for stochastic constraints as well as the search direction. They also derive a heuristic that uses this search direction iteratively. Further they illustrate their heuristic by applying it to the optimization of an (s, S) inventory system under the same conditions as Bashyam and Fu (1998).

The motivation for our work comes from the difference answers between Bashyam and Fu (1998) and Angun, Gurken, Hertog and Kleijnen(2006) for the same model under the same situations. There are several issues to consider: what the true optimal values of (s, S) in this specified conditions are; whether the OptQuest can find the optimal values more efficiently; how we can prove these different outcomes are the estimators of true optimal values and which one is better.

In order to compare these different outcomes obtained from different papers, we test their KKT conditions in statistical method. Gill, Murray, and Wright (2000) present the KKT first-order optimality conditions in deterministic nonlinear mathematical programming. Karaesman and Van Ryzin (2004) present an unconstrained optimization algorithm to check the KKT conditions for a single stochastic response. They used the estimated gradient of the goal function, including a score function estimator. But the constrained (s, S) inventory system with random lead times includes two discrete inputs and two stochastic responses. To check KKT conditions in random simulation models with multiple responses, both Angun and Kleijnen (2005) and Bettonvil, Castillo and Kleijnen (2005) derive procedures. The former proposes an asymptotic statistical procedure with a large sample size, while the latter solves the same issue by assuming expensive simulation, namely small number of replications per scenario.

In this paper, we establish an Arena model for the (s, S) inventory policy and search the best estimator for s, S pairs by using OptQuest. We also try to find the true estimator of the optimum by applying brute force method in order to know the gap between them. So considering the results of Bashyam and Fu and Angun et al, totally there are 4 outcomes already, namely the output of Brute Force, the outcome of OptQuest, (1040, 1065) from

Bashyam and Fu, and (1160, 1212) from Angun et al. In our experiments, we test their KKT optimality conditions by using above two different procedures: small sample size one provided by Bettonvil, Castillo and Kleijnen (2005), and large sample size one from Angun and Kleijnen(2005).

In our conclusion, we give the true optimal estimator of (s*, S*) pairs estimated by Brute Force and prove that OptQuest can be used in solving the stochastic constrained optimization problem and find the near optimum effectively. Further, we point out that the outcome obtained from Bashyam and Fu is the estimator near the true optimal value, but not as close as the one gained by OptQuest, while the result of Angun et al (2006) gained is far away from the optimum. Furthermore, we also prove that the rejection probability for each null-hypothesis obtained by the KKT testing procedure with large example size is more obvious than that of small example size. And it is reasonable in theory.

The rest of the paper is organized as follows. In section 2, we formally specify the model and describe and validate the corresponding Arena model. In section 3, the OptQuest conditions are illustrated. We describe the outcomes of Bashyam and Fu (1998) and Angun, Gurken, Hertog and Kleijnen(2006) and analyze our outcomes in section 4. Section 5 gives the method to test KKT conditions statistically for each of those outcomes. The specified (s, S) inventory model is described according to the model in Beettonvil, Castillo and Kleijnen(2005). In section 6, we apply the statistical test for KKT conditions with a large sample size. Section 7 contains concluding remarks and directions for further research.

2 SIMULATION MODEL

2.1 Arena Model of (s, S) Inventory System

We consider an infinite horizon continuous review inventory system with continuous-valued i.i.d. (independent and identical distribution) demands and full backlogging for a single item. Figure 1 is the description of sequence of events in any period: demand, review, order and delivery. The details can be stated as follows:

1) At the beginning of the period, the demand $\mathrm{D_n}$ is arriving;
2) Fulfil the demand if on-hand inventory $V_n \geq D_n$
3) If $V_n < D_n$, fulfil the demand as much as V_n. Note that, any unfilled demand in a period is entirely backlogged to be satisfied by future deliveries. Then we calculate the size of backorder: $B_n = D_n - V_n$.

4) Review the inventory position $\mathrm{I_n} = V_n + \sum_{l=n-v}^{n-1} q_l - \sum_{i=n-m}^{n-1} B_i$, where $\sum_{l=n-v}^{n-1} q_l$ shows the total outstanding orders placed but not delivered until this period, $\sum_{i=n-m}^{n-1} B_i$ is the sum of all backorders which have not been satisfied before this period. Whenever the $\mathrm{I_n}$ is less or equal to reorder point s, an order of size $\mathrm{S - I_n}$ is placed;
5) The delivery arriving time is determined by the lead times, L_n which are assumed to be integer valued i.i.d. stochastic variables. The arrived products are first used to fulfil the backorders happened until this period before replenish the on hand inventory.

The performance of the system will be evaluated by a cost function and a service level measure, where the measure of cost considers only the setup and holding costs. We use h presents the holding cost per period, K denotes the setup cost per order, and u is the order cost per unit. So the cost for a single period can be expressed as

$$C_n = h(V - \sum_{i=n-m}^{n-1} B_i)^+ + I\{I_n < s\}\left(K + u\left(S - I_n\right)\right) \tag{1}$$

where $I\{.\}$ means the indicator function, so we calculate the expected average cost per period $C_N = E\left(\frac{C_n}{N}\right)$ during the infinite period N . When letting $N \to \infty$, we can get the infinite horizon measures of cost.

Note that, instead of a shortage cost, we shall consider a stock-out constraint which is defined as the fraction of total demand that is not satisfied from on hand stock, The lack of the fill-rate measure considered in Tijms and Groenevelt (1984), given by,

$$\gamma = E\left(\frac{total\ number\ of\ backorder}{total\ number\ of\ demand}\right) = E\left(\frac{\sum_{n=1}^{N} B_n}{\sum_{n=1}^{N} D_n}\right) \tag{2}$$

This constraint will be referred to as $\gamma -$service level, which is also used in the assumption of Bashyam and Fu (1998).

In order to describe the (s, S) inventory policy clearly and logically, we defined 11 variables totally. The variables and their initial values are shown in the Figure2. By changing the initial values for s and S, the different scenarios could be generated. For example, in Figure 2, we assume that the initial value of s and S are 500 and 1000 respectively.

Insert Figure 1: The Arena Model of (S, S) Inventory System

Insert Figure 2: The Original Conditions and the State Variables

2.2 Verification of the Simulation Model

Verification is concerned with determining whether the conceptual simulation model (model assumptions) has been correctly translated into a computer "program"(Law and Kelton 2000). Obviously, it is more difficult to debug a large-scale simulation program than a small one. Although our model is not very complex, we still verify it by three techniques.

Firstly we run the (s, S) simulation under a variety of settings of the input parameters to check if we get the reasonable outputs. Especially the extremely situation: initial inventory, s and S are all 0. The corresponding results include γ equal to 1. That means the service level is 0. No demand is satisfied during the whole 30,000 periods. It means our model got the practical reasonable outcomes. We also calculate the different average costs and $\gamma-$service levels by increasing s and S respectively. We found that the average cost is increasing with the $\gamma-$service level goes down, i.e., the cost goes up with the service level increasing. Furthermore, the $\gamma-$service level tends to decrease with the values of s and S being up. On the contrary, under the same conditions the average cost tends to be up. Those trends are consistent with the real practise.

In order to verify the expression and logic of our model, we use one of the most powerful techniques, "trace", to debug our discrete-event simulation program. Given that the original inventory is 500, reorder point is 500, order-up-to point is 1000, an exponential distribution demand with mean 100 units, and lead time is Possion distribution with parameter 10 days, and we simulate our program for 30,000 periods. Figure 3 is the partial data listed in the Output Analyzer in Arena. Use 71th period as an example, at the beginning of period, the demand 95.4 is arriving. The inventory at the end of 70th period is 0, so the demand can not be satisfied. Then 95.4 backorder generated, and the total backorder increased to 4.06E+003(=95.4+3.97E+003). Because the inventory position (inventory plus on order minus sumbackorder = 0 + 1.65e+003 - 770) is greater than 500, no order was placed. At the end of period, the delivery of order placed at 64th period was arriving, the delivery quantity is 543. But they are used to fulfil the previous backorders first, so the next period inventory is still 0, while the sumbackorder is 323 (= 770 + 95.4 - 543). During this period, there is no order placed, no on hand inventory holding, thus no cost spending (equal to 0). The $\gamma -$service level is 0.602 (= 3.97E+003 / 4.06E+003).

We also check some special situations, for example, the period which has the placing order event and two or three deliveries. By comparing with hand calculations, we draw a conclusion that the system is operating as intended.

Finally, we use the "trace" technique to prove that the replenishment of order in our model is crossing. The underline data in Figure3 shows that at the 71th period, the order placed at 64th period was replenished earlier than that placed at 61th which is delivered at 74th period. We call this situation as the order crossing.

Insert Figure3

3 SETTING OF OPTQUEST CONDITIONS

Recent developments in the area of optimization have allowed for the creation of intelligent search methods capable of finding optimal or near optimal solutions to complex problems involving elements of uncertainty. Often, optimal solutions can be found among large sets of possible solutions even when exploring only a small fraction of them. OptQuest, from OptTek System Inc, is the result of implementing these search technologies in combination with simulation models built for Arena. It uses heuristics known as tabu search, neural networks, and scatter search into a single, composite search algorithm to move around intelligently in the input-control space and try to converge quickly and reliably to an optimal point (Kelton, Sadowski and Strurrock (2004)).

After the establishing of Arena model, the OptQuest takes over the execution of the Arena model. Once the optimization problem is described by means of selecting controls (inputs), the objective, and possibly imposing constraints and requirements, Arena is called every time a different set of control values to be evaluated. The optimization method used by OptQuest evaluates the responses from the current simulation run, analyzes and integrates these with responses from previous simulation runs, and determines a new set of values for the controls, which are then evaluated by running the Arena model. This is an iterative process that successively generates new sets of values for the controls, not all of them improving. The process continues until some termination criterion is satisfied—usually expressed as a limit on the amount of time devoted to the search in Arena. It depends on the analyst's experience instead of accuracy scientific standard. The test of KKT could be used as a stopping rule for simulation process.

The setting of the controls, constraints, and objective/requirements can be done easily by using OptQuest configuration screens, see figure 4 to 8. The procedure needs 4 steps in our model:

Step1: Select two variables, order-up-to point S and reorder-point s, as the control variables and give them the lower and upper bounds and suggested values. Note we defined them as the integers (because the number of the products is integer). For example, as shown in Figure 4, the search area for both s and S are from 0 to 2000 and the suggested values are 1000 and 500 respectively. The suggestion value is very important for OptQquest to find the better solution in a relatively short time. Its selection is tricky. What we choose for them here is arbitrary at the first time. After trial-and error test, we intend to use the coarse solution gained from the previous experiments as the suggestion values.

Insert Figure 4

Step2, Define the inputs constraints. Our model only has one constraint need to be considered: "S-s>= 0"

Insert Figure 5

Step3: Fill in the objective of this optimal issue: minimum the average total cost. And give the requirement, the $\gamma-$service level, the lower and upper bound. Here we just need the upper bound 0.10, which is same as Bashyam and Fu's setting. See figure 6.

Insert Figure 6

Step 4: Select the optimization options. The Options window (see figure 7) lets us set options for controlling the optimization process. We control how long to run the optimization by the time tab. In the precision tab, we choose "Vary the Number of Replications" to allow OptQuest to test for the statistical significance between the current value of the objective function and the best value found so far. If the test determines that the mean of the objective value obtained with the current solution is statistically inferior to the best mean objective value known, it will stop the current simulation (i.e., no more replications are performed). Also we choose "Stop when 95% confidence interval half-width is within 5 % of the mean" to control the number of replications based on the size of the objective function's confidence interval. The percent error (e.g., 5%) and the confidence level (which is fixed at 95%) determine the precision of the interval. The stopping rules in OptQuest could depend on the predetermined time or the numbers of simulation. It also provides the automatic rule: stop after some nonimproving solutions. Van Beers and Kleijnen (2006) use the similar approach to select the number of replications. Also see Law and Kelton (2000)

Insert Figure 7

4 EXPERIMENTS AND OUTCOMES

4.1 Bashyam and Fu's Outcomes

Bashyam and Fu (1998) defined two variables in their (s, S) inventory system: reorder point ***s*** and the reorder quantity ***Q***. and simulated one of their experiments under the following conditions:

- Demands have an exponential distribution with mean 100.
- Lead times have a Poisson distribution with mean 6, which represents the relative large probability of order crossing.
- The $\gamma -$service level is 0.10 (they also give the flexible limit: 0.11);
- The holding cost $h = 1$, the unit cost is $u = 2$ and the setup cost is $K = 36$.

Bashyam and Fu estimated the ***true*** optimal by means of a Brute Force simulation with 5×5 grid initially, namely changing the s, S value by 5 each time to get the new combination, and then 1×1 grid in the neighbourhood of the coarse solution. They simulated with $N = 30{,}000$ periods and average over 10 replications. When the feasibility limit for $\gamma -$service level is 0.11(target is 0.10), they got the ***true*** optimal cost estimator is 702.76, and the corresponding $\gamma -$service level is 0.1099, which does slightly exceed the target one. They didn't give the corresponding values of s and Q in the paper.

By using Perturbation Analysis, they estimated the gradients of the average cost and $\gamma -$service level with respect to s and Q, and feasible directions to search for optimum. Under the same conditions as Brute Force using: 30,000 periods and averaging over 10 replications, the final estimators were 708.20 and 0.1076 for average total cost and $\gamma -$service level respectively. The corresponding values of s and Q obtained in private communication are1040 and 25, respectively. And the gap between the estimator gained by Brute Force and that obtained by Perturbation Analysis is around 5(=708.20-702.76).

4.2 Angun's Outcomes

In Angun, Gurken, Hertog and Kleijnen (2006)'s paper, they use the same conditions as Bashyam and Fu. Their conclusion of the Brute Force simulation experiments is that (1160, 1212), which has the average cost of 647.1495 with a standard error of 8.5531 and the average fill rate of 0.8948(namely, the $\gamma -$service level is 0.1052) with a standard error of 0.0100, is the "best" estimate of (s□, S□).

Then they apply their heuristic to (s, S) inventory system. The heuristic indeed reaches the neighbourhood of their true optimum at the point (1185, 1230.7) with the estimated cost of

671.3 (within three standard error of their true cost, 647.1495). They didn't give the corresponding estimator of average fill rate in that paper.

4.3 Our outcomes

As for getting the comparable results, we use the same assumptions for our model as Bashyam and Fu (1998) did. But we made the assumption that the initial inventory on hand is 1000 units, while Angun uses the order-up-to point as the initial inventory. In order to eliminate the influence of initial inventory level, we run the simulation with 300 warm-up periods for each of iterations consisted of 30,000 periods.

We use the brute force method to get our true optimal estimator of (s*, S*) by the following procedure:

Step1: we simulated the case for 30,300 periods (300 warm-up periods) with the initial inventory level is 1000. We average the costs and the $\gamma-$service level 10 replications over (s, S) plane, where $0 \le s \le 3000$ and $0 \le S \le 3000$. We increased each time both s and S by 100 and keep the constraint $S \ge s$ (Angun's model uses $S > s$). After evaluating 325 combinations of (s, S), we concluded that an estimate of (s*, S*) is restricted to $900 \le s \le 1250$ and $1050 \le S \le 1250$.

Step 2: we repeat the simulation (30,300 periods and 10 replications) over this restricted area: $900 \le s \le 1250$ and $1050 \le S \le 1250$. We increase both s and S this time by 10($<< 100$) and keep $S \ge s$. After evaluating 546 combinations, we gained the more restricted range $1010 \le s \le 1030$ and $1070 \le S \le 1090$.

Step 3, because we consider the integer input, there are 441(=21*21) combinations for the area $1010 \le s \le 1030$ and $1070 \le S \le 1090$ to be simulated using 30,300 periods and 50 replications. The outcomes are shown in Figure 8. The average $\gamma-$service level per point is amplified for 7000 times in order to identify its trend clearly.

Our conclusion of this brute-force simulation experiments is that some combinations in this area could be estimator of (s*, S*); the average $\gamma-$service level is from 0.0999 to 0.1001 (that is, the service level is in 0.9001 to0.8999) and the average total cost is from 623.739 to 625.514 (See Table 1).When s is 1020 and S is 1075, the corresponding average total cost estimator is 623.739 and $\gamma-$service level is 0.1001, with standard deviation 2.3577 and 0.004298, respectively. We call it the best true estimator of (s*, S*).

Insert Table 1.

To estimate the optimal value of (s*, S*) in QptQuest, we first search the area where $0 \le s \le 3000$ and $0 \le S \le 3000$ with the suggested value 1000 for both s and S and 30,000 periods with 10 replications. (In the OptQuest, it provides the function to control the precision by changing the number of replications. We choose to vary number of replications for 10 to 100 stopping if an interior solution is found and stop when 95% C.I. half-width is with 10% of the mean). And we give the automatic stop rule as stopping after 300 non-improving solutions. It took about 3 hours to find the best solution (996, 1116) so far, which is called coarse solution, under this specified condition.

Then we search the more restricted area $800 \le s \le 1200$ and $800 \le S \le 1200$ with the suggestion values for s, S are 996 and 1116 respectively. This search has the same conditions: 30300 periods, average 10 replications and initial inventory level 1000. But we defined different automatic stop rule: stopping after 500 non-improving solutions. After around 150 minutes, the simulation stopped with the solution s is 1021 and S is 1077, the corresponding average total cost is 624.869 (Std. Dev. = 3.7885), feasible Requirement: $\gamma-$service level is 0.1002 (Std. Dev. = 0.005001).

The average total cost gap between OptQuest and the best true solution obtained by Brute-force is about 1.

Note that this search process depends on the size of search area and the original suggestion value closely. If the search area is huge and no good initial suggested value, it might need long time to find an approximately solution.

So far, some different outcomes obtained by using different methods are summarised in Table2.

Insert Table2.

In Table 2, we list these different estimators of average cost and service level with their standard deviations. Some data are obtained from original paper, such as, those of point D, some of them came from private communication, point C, for example. Point A and B are our model's outcomes. Since we can not get more information about other points, these 4 points are selected as representative points. In order to estimate those KKT test procedure's power function, we add on an obviously non-optimal point E, (985, 1188), which is obtained in our searching process for the optimum. We are sure that it is far away the true optimal value. In figure 9 and 9a, we illustrate the 5 test points and some estimators of true (s*, S*) by using Brute force.

Insert Figure 9.

Insert Figure9a.

In the following sections, we will try to compare them and evaluate which one is the best estimators for (s*, S*).

5 PROCEDURE AND RESULTS OF KKT TEST BY USING SMALL SAMPLE SIZE METHOD

We formalize our (s, S) inventory system as follows according to the Bettonvil, Castillo and Kleijnen (2005) and Angun and Kleijnen (2005) in order to follow their KKT testing procedures. We let d_j denote the original (non-standardized) input $j(j = 1,\ldots,k)$, namely the control variables. An (s, S) policy model has $k = 2$ inputs: the reorder point s and order-up-to point S. The responses can be described as $W_{h'}(h' = 1,\ldots,z-1)$, which z is the number of response, here equal to 2. The objective output is to minimize the average total cost, denoted as W_0. Then the mathematical programming formulation of (s, S) inventory system can be described as following:

$$\begin{array}{ll} \textit{Minimize} & E(W_0(\mathbf{d},\mathbf{r})) \\ \textit{Subject to} & E(W_1(\mathbf{d},\mathbf{r})) \le 0.1 \end{array} \tag{3}$$

where $\mathbf{r}$ denotes the pseudo-random number (PRN) and W_1 is the $\gamma-$service level with right-hand-side is 0.1. Our model has two inputs and two outcomes, so it falls into the constrained, nonlinear, random multiresponses optimization problem.

The KKT conditions for deterministic problem can be shown as

$$\boldsymbol{\beta}_{-0:0} = \mathbf{B}_{-0:J}\lambda \tag{4}$$

where $\boldsymbol{\beta}_{-0:0}$ denotes the gradient of the goal function, $\mathbf{B}_{-0:J}$ is the $k \times J$ matrix with the gradients of the J binding constraints, and λ is the corresponding non-negative Langrange multipliers. In our model the $\mathbf{B}_{-0:J}$ can be expressed as $\mathbf{B}_{-0:1}$, a 2×1 matrix.

We cannot get any more improvement if for the optimal point there is not a vector existing being both a descent direction and a feasible direction. So if the test point is optimal value, the $\boldsymbol{\beta}_{-0:0}$ and $\mathbf{B}_{-0:1}$ of this point should be in the (roughly) same direction.

Bettonvil, Castillo and Kleijnen (2005) formalize a constrained nonlinear random optimization problem and its KKT conditions by focusing on 'expensive' simulations. They use Ordinary Least Squares (OLS) to locally fit a second-order polynomial per response (when the first-order polynomial gives lack of fit), using a Central Composite Design (CCD) with the m replications for the centre point, m is greater than $z+1$. The classic (univariate) lack-of-fit test combined with Bonferroni's inequality is presented. The KKT testing procedure is to find whether the central point of the local area satisfies the optimality conditions. The student's t test is used to check whether the simulation response is feasible and whether any constraints are binding (active). Because of the expensive simulation, they drive a bootstrap procedure to test the linear relationship between $\boldsymbol{\beta}_{-0;0}$ and $\mathbf{B}_{-0;1}$. They also check the number of non-negative Lagrange multipliers to test the other optimality conditions.

As we state above, what we need to check are 5 points labeled as A, B, C, D and E in Figure 9. Following the testing procedure in Bettonvil, Castillo and Kleijnen (2005), we first design experiments by using the most popular design (see Kleijnen (1987) and Myers and Montgomery (2002)), CCD, which consists of a Resolution-5 (R-5) design, the 2k axial points, and the centre point. We realize that this procedure could be influenced deeply by the local area. In order to reduce the effects of the local area size on the final results, we experiment with three types of situations to obtain the other 4 points of R-5 design. In the first situation, they are found by (arbitrarily) changing the coordinates of the central points by 2, about 0.25 %,(a total change 4 in the local area size). We call it the "Local Area Size 4" situation. The second way is to change the coordinates of the central points by 5, about 0. 5 %,(a total change 10 in the local area size), so-called the "Local Area Size 10". The third one is to change by 10, about 1 %,(a total change 20 in the local area size), so-called the "Local Area Size 20" this three type describe as Small, Middle and Large respectively in Table 3. For example, as for the point C, the central point is (1040, 1065), changing the coordinates by 2, we found the other 4 design points, namely (1038, 1063), (1042, 1063), (1038, 1067), (1042, 1067). The other 4 axial points of CCD are determined with the constant $\sqrt{2}$. The selection of constant is discussed in the classic DOE literature. See Kleijnen (1987). Further, we replicate the central point for 3 times, namely $m = 3$, while $m = 1$ for other 8 design points.

Insert Table 3.

Then we estimate the coefficients of a second-order polynomial by

$$\hat{\boldsymbol{\beta}}_{h'} = \left(X^T X\right)^{-1} X^T W_{h'} \tag{5}$$

where $h'(= 0,1)$ represents the number of responses, X denotes the matrix of explanatory (regression) variables, which is completely determined by standardized design matrix with elements of R-5 design.

Next we test the lack of fit by using the classic lack-of-fit f test, calculated by the following formula,

$$F_{n-q,N-n}(h') = \frac{\sum_{i=1}^{n} m_i (\overline{W}_{i,h'} - \hat{y}_{i,h'})^2 / (n-q)}{\sum_{i=1}^{n} \sum_{r=1}^{m_i} \left(W_{i,h',r} - \overline{W}_{i,h'}\right)^2 / (N-n)} \tag{6}$$

where $\overline{W}_{i,h'}$ denotes the average response for the number of replications per design point, $n(n = 1,2,\cdots,9)$ denotes the number of design points and N is the total number of replications for all design points, which equal to $m_1 + m_2 + \cdots + m_n = 8 + 3 = 11$. $\hat{y}_{i,h'} = X_{h'}\hat{\boldsymbol{\beta}}$ denotes the vector with regression estimators.

According to Bettonvil, Castillo and Kleijnen (2005)'s paper, we need to test three null-hypotheses obtained from the optimality conditions of KKT in deterministic constrained optimization problem as follows:

(i) Test whether the current solution is feasible and whether the constraint is binding. By testing the representative centre point of the current local area, we test the following null-hypothesis,

$$\mathbf{H}_0^{(1)} : E\left(W_1\left(\tilde{\mathbf{d}} = 0\right)\right) = 0.1 \tag{7}$$

To test the hypothesis, we use the classic Student t test:

$$t_{m-1} = \frac{W_1\left(\tilde{\mathbf{d}} = 0\right) - 0.1}{\hat{\sigma}_h / \sqrt{m}}, \tag{8}$$

where both the numerator and the denominator use the $m = 3$ replicated simulation outputs at the centre point of CCD, and $\hat{\sigma}_h$ is the 'pure error' standard deviation following from Equation 6.

(ii) Test the null-hypothesis which generated by replacing all deterministic quantities in the original KKT conditions by their (random) estimators; i.e.

$$\mathbf{H}_0^{(2)} : E\left(\hat{\boldsymbol{\beta}}_{-0:0}\right) = E\left(\hat{\mathbf{B}}_{\mathbf{-0:1}}\hat{\lambda}\right) \tag{9}$$

(iii) Test that the Lagrange multipliers estimated,

$$\mathbf{H}_0^{(3)} : E\left(\hat{\lambda}\right) \geq 0 \tag{10}$$

To get an accurate estimate of the test procedure, we run 500 macro-replicates of our experiment. We got our testing results in Table 4, which displays results for each of the 5 locations and following two factors: Local area size and noisy. Local area size is described above. In order to compare the effects of different noisy on the outcomes, we repeated the simulations with 3,000 periods and 30,000 periods, which determine the resulting noise, respectively.

Insert Table 4.

We select the probability of type-I error rate $\alpha = 0.1$ (as Angun et al. (2004) do), the observed value is denoted by $\hat{\alpha}$, which is binomially distributed. Identifying the relationship between $\hat{\alpha}$ and α, we can draw a conclusion described as follows:

1). the point A (1020, 1075) obtained by brute force has the best fit for most of test conditions. Its test procedure rejects the null-hypothesis (7), (9) and (10) with the probability of type-I error rate $\hat{\alpha}$ less than 0.1. It can be called the true approximately optimal point in both local area 10 and 20.

2). the point B (1021, 1077) can be regarded as the point very close to the optimum. Its procedure accepts the null-hypothesis in (7) implying a binding constraint and the null-hypothesis (9) with the probability $\hat{\alpha}$ less than 0.1 and (10) with the probability $\hat{\alpha}$ slightly greater than 0.1 when the local area is small. But as we know, we select $\alpha = 0.1$ arbitrarily by considering conservative Bonferroni inequality, it can be accepted as the optimal one when we ignore it.

3). the point C (1040, 1065) is a little bit further than point B. Its probability $\hat{\alpha}$ of the null-hypothesis (10) is greater than 0.1 in all situations. Especially the local area is 20 with small noisy. So it is also can be regarded as one of the alternatives of the close optimal estimators when we ignore that slight greater amount.

4). the point D (1160, 1212) obtained by Angun et al (2006) gives all results of 500 replications an inactive (the slack is positive) service constraint!

5). the point E (980, 1188) is further away from the optimum than point A and B. It has a higher probability to reject the null-hypothesis (7) or (10) in each situation.

6). the different noisy doesn't seem making obvious difference for the outcomes under this specific case.

6 PROCEDURE AND RESULTS OF KKT TEST BY USING LARGE SAMPLE SIZE METHOD

Angun and Kleijnen (2005) derive an asymptotic statistical procedure for testing the KKT first-order necessary optimality conditions in random simulation models with multiple responses. They assume the number of replications is large, so the asymptotic statistical properties are applied. The statistical theory on Design of Experiments (DOE) proves that the best design to estimate a first-order polynomial is an Response-3(R-3) design, which requires 'few' input combinations, namely $k+1$ rounded upwards to the next multiple of four. So they use it, and augmented with a centre point to estimate the gradients. The Roy's largest root test and the classic F test combined with Bonferroni's inequality are discussed in their paper for multivariate responses to check the estimator's lack-of-fit. They also use student's t test for slack vector, which has an asymptotic multivariate normal distribution, to check the binding constraints at central point. In order to know whether the estimator of gradient of the objective can be expressed as a nonnegative linear combination of the gradients of the binding constraints, both a simple form of the Delta method and a generalized version of Wald's statistic are used in the procedure. Delta method is used to show that nonlinear statistics are asymptotically multivariate normally distributed. The generalized form of Wald's statistic is used to test composite hypotheses of optimality conditions of KKT. We refer to Angun and Kleijnen (2005) for detailed explanation.

In our testing experiments we estimated the gradient starting with an R-3 design augmented with a central point for each point listed in the Table 2 in our Arena model. For each point, we find the other four local design points by changing the coordinates of the central points by 0.25%, around 2 (4 totally), same as we did in above CCD. See them marked with bold in Table 3. Argument with a central point, the total design points is 5. Differently, we repeat 20 times for all design points to get a large example size.

Then we fit linear regression metamodels by the same way as we did in small example size. Next we test these metamodels for lack-of-fit using Roy's test by using the formulas (numbered (7) and (8) in Angun and Kleijnen (2005)'s paper).

If there is no lack-of-fit, then we test for binding constraints at central point through t statistic. (Refer to formula (10) in Angun and Kleijnen (2005)'s paper)

If we find the binding constraint, then we test for KKT conditions at central point through Wald's generalized test. (Refer to formula (20) in Angun and Kleijnen (2005)'s paper)

We repeat our procedure for 100 macro-replications in order to estimate the type-I error probability and compare it with the prescribed significant level $\alpha = 0.1$. We use the same significant level for each of above three statistics: testing lack-of-fit, binding constraints and KKT conditions. Table 5 summaries our results.

Insert Table 5.

We offer the following comments.

1). It is accepted for point A to be regarded as the true estimator of s, S pairs because all three null-hypothesises with the rejected probabilities of type-I error rate α are less than 0.1.

2). Both Point C and point B are not the true optimum. Their rejection probabilities of the third null-hypothesis, 0.22, 0.28 respectively, exceed the nominal 0.1.

3). Point C is further away from the true optimum than point B. It has a higher rejection rate for the third null-hypothesis than point B. In general, the power of Wald's generalized test increases as the input combination tested moves away from the true optimum.

4). It is obvious that point D is furthest away from the true optimum among those 5 points.

7 CONCLUSIONS AND FUTURE RESEARCH

In this paper, we establish a simulation model of (s, S) inventory policy with random lead times and a service level constraint by using Arena. OptQuest is used for the computation of (s, S) values which minimize the total expected average costs. We apply two statistical testing procedures of KKT optimality conditions into 4 different results which obtained by using different optimization algorithms in different papers. In order to check the power of these two procedures, an obvious non-optimal point was added on.

Our outcomes give the answers about our original questioned that the true optimal estimator of (s*, S*) pairs could be any combination in the area of the average $\gamma-$service level is from 0.0999 to 0.1001 (that is, the service level is in 0.9001 to 0.8999) and the average total cost is from 625.514 to 623.739. When s is 1020 and S is 1075, the corresponding average total cost estimator is 623.739 and $\gamma-$service level is 0.1001, with standard deviation 2.3577 and 0.004298, respectively.

We also prove that OptQuest can be used in solving the stochastic constrained optimization problem and find the near optimum effectively.

The outcome, (1040, 1065) obtained from Bashyam and Fu, is also the estimator near the true optimal value, but not as close as the one gained by OptQuest. The result of Angun et al (2006) gained is far away from the optimum.

Besides the above conclusions, we also find that the rejection probability obtained by the KKT testing procedure with large example size is more obvious that that of small example size. And it is reasonable.

For the further research, we try to investigate the optimum through white box methods. We also have great interest on the searching of robust solution for (s, S) inventory system by using OptQuest and Arena. Robust is of great importance in practice: a solution that is optimal for a given scenario is not practically relevant if that solution breaks down as soon as the environment changes. We try to derive values for the controllable factors when making a strategic decision about (s, S) inventory policy—accounting for the randomness of the environmental factors.

Acknowledge

The authors acknowledge the financial support of the China Scholarship Council. They also thank Professor Jack G.A.J. van der Vorst, Logistics and Operation Research Group, Wageningen University, for his comments and support. And they are grateful to E. del Castillo, E. D. Angun for their programs of KKT testing.

References

Angun, E., D. den Hertog, G. Gürkan, and J.P.C. Kleijnen (2006), Response surface methodology with stochastic constrains for expensive simulation. Working Paper (download from http://center.kub.nl/staff/kleijnen/papers.html)

Angun, E., and J.P.C. Kleijnen (2004), An asymptotic stopping rule for Response Surface Methodology with stochastic constraints. Working Paper (download from ttp://center.kub.nl/staff/kleijnen/papers.html)

Angun, E., and J.P.C. Kleijnen(2005), An Asymptotic Test of Optimality Conditions in Multiresponse Simulation-Based Optimization. Working Paper (download from ttp://center.kub.nl/staff/kleijnen/papers.html)

Archibald B. E. Silver (1978), (s, S) Policies under Continous Review and Discrete Compound Passion Demand, Management Science, 24, pp 899-909

Bashyam S. and M.C. Fu (1998), Optimization of (s, S) Inventory Systems with Random Lead Times and a Service Level Constraint. Management Science, 44, no, 12, Part 2, p. 243-256

Bettonvil, B., E. del Castillo, J.P.C. Kleijnen (2005), Statistical Testing of Optimality Conditions in Multiresponse Simulation-based Optimization. Working paper, Department of Information Systems and Management, Tilburg University, Tilburg, Netherlands.

Cohen, Kleindorfer and Lee (1988). Service Constrained (s, S) Inventory Systems with Priority Demand Classes and Lost Sales, Management Science, Vol. 34, No. 4, pp 492-499.

Ehrhardt, R.(1979), The Power Approximation for Computing (s, S) Inventory Policies. Management Science. 25, pp 777-786

Ehrhardt, R. (1984), Policies for a Dynamic Inventory Model with Stochastic Lead Times. Operations Research. 32, pp 121-132

Federgruen A. and Zipkin P (1984), An Efficient Algorithm for Computing Optimal (s, S) Policies. Operations research, Vol 32, No 6, pp 1268-1285

J.D. M. Fisher and A. Hornstein (2000), (S, s) Inventory Policies in General Equilibrium, The Review of Economic Studies, Vol. 67, No. 1, pp. 117-145

Freeland J. R. and Porteus E. L. (1980) Evaluating the effectiveness of a new method for computing approximately optimal (s, S) inventory policies, Operations Research, Vol.28. No. 2, March-April, pp 353-364

Fu, M.C. (2002), Optimization for Simulation: Theory Vs. Practice. INFORMS Journal on Computing, 14, pp. 192-215

Fu, M.C. and and Hu J. Q. (1994), (s, S) Inventory Systems with Random Lead Times: Harris Recurrence and Its Implications in Sensitivity and Analysis, Probability in the Engineering and Information Sciences, Vol.8, No.3, pp335-376

Gill, P. E., W. Murray, M. H. Wright (2000). Practical Optimization, 12th edition. Academic Press, London

H. C. Tijms and H. Groenevelt (1984), Simple Approximations for the Reorder Point in Periodic and Continuous Review (s, S) Iinventory Systems with Service Level Constraints, European journal of operational research, 17, pp 175-190

Hu, J.Q., Nananuku, S., and Gong, W.B (1994), A New Approach to (s, S) Inventory Problems, Journal of Applied Probability, Vol. 30, pp. 898-912.

Karaesman, I and G. van Ryzin (2004), Overbooking with Substitutable Inventory Classes, Operations Research, 52, No. 1, pp. 83-104.

Kelton W. David, Sadowski P. Randall, David T. Strurrock (2004), Simulation with Arena, 3rd Ed., McGraw-Hill, New York.

Kleijnen, J.P.C. (1987), Statistical tools for Simulation Practitioners. Marcel Dekker, New York

Law M.Averill and Kelton W. David, simulation modelling and analysis: third edition, McGraw-Hill series in Industrial Engineering and Management Science, Bonston.

Myers, R. H. and D. C. Montgomery (2002), Response Surface Methodology: Process and Product Optimization Using Designed Experiments; second edition. John Wiley & Sons, New York

Porteus E. L. (1979) An adjustment to the Norman-White approach to approximating dynamic programs. Operations Research, Vol. 27, No 6, pp 1203-1207

Porteus E. L. (1985) Numerical Comparisons of Iinventory Policies for Periodic Review Systems, Operations Research, Vol 33, No 1, pp134-152

Porteus E. L. (1988), Stochastic Inventory Theory. D. Heyman, M. Sobel Eds. Handbooks in Operations Research and Management Science, Vol. II: stochastic models, North-Holland, NY.

Rao, C. R. (1967), Least Squares Ttheory Using an Estimated Dispersion Matrix and its Application to Measurement of Signals. Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, I, pp. 355-372

Spall J. C. (2003), Introduction to Stochastic Search and Optimization; Estimation, Simulation, and Control. Wiley, Hoboken, New Jersey.

Schneider H. and J. L. Ringuest (1990), Power Approximation for Computing (s, S) Policies Using Service Level, Management Science. Vol. 36, No. 7 pp 822-834

Wim C.M. van Beers and Jack P.C. Kleijnen(2006), Customized Sequential Designs for Random Simulation Experiments: Kriging Metamodeling and Bootstrapping, working paper, http:// center.uvt.nl/staff/kleijnen

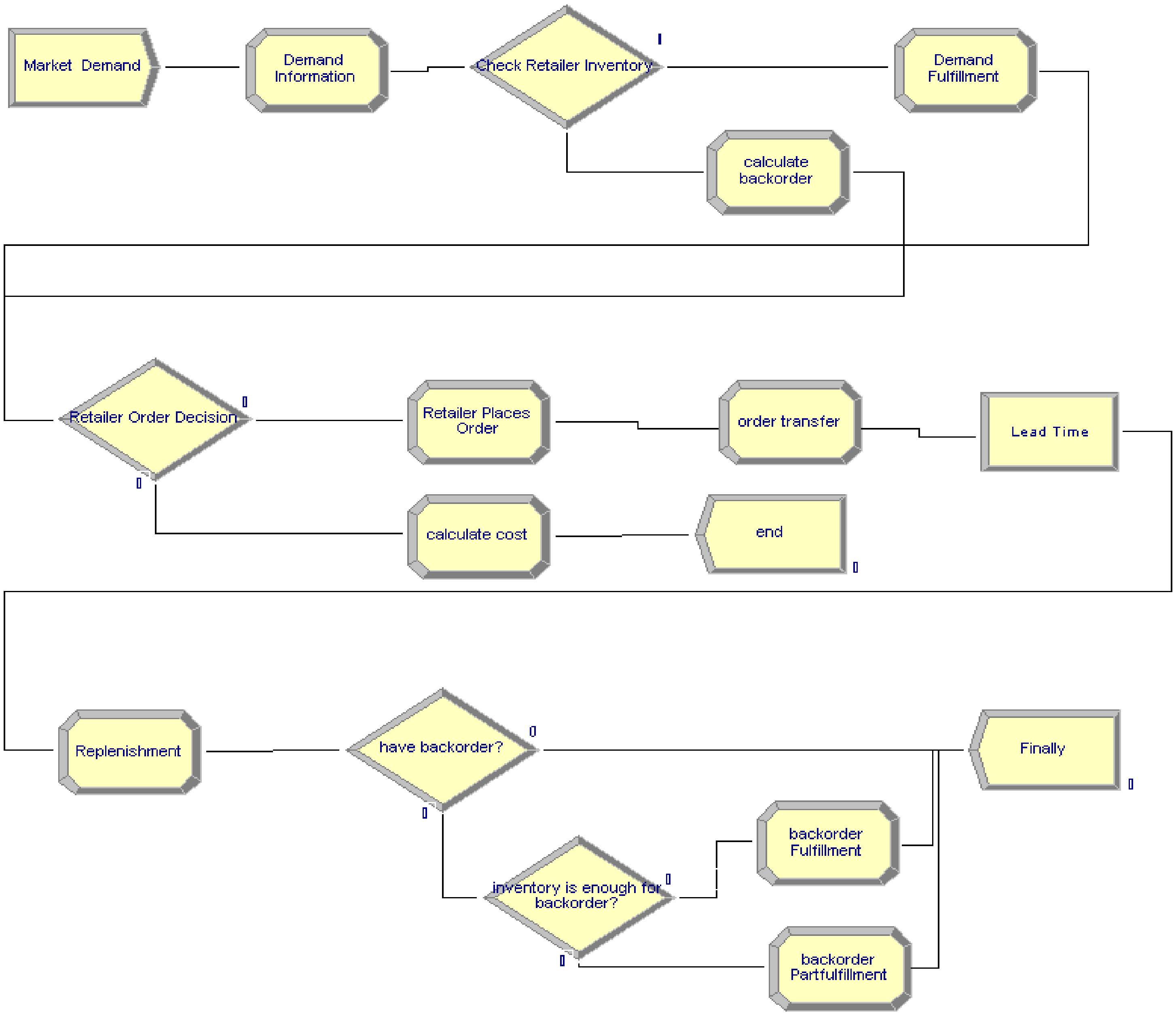
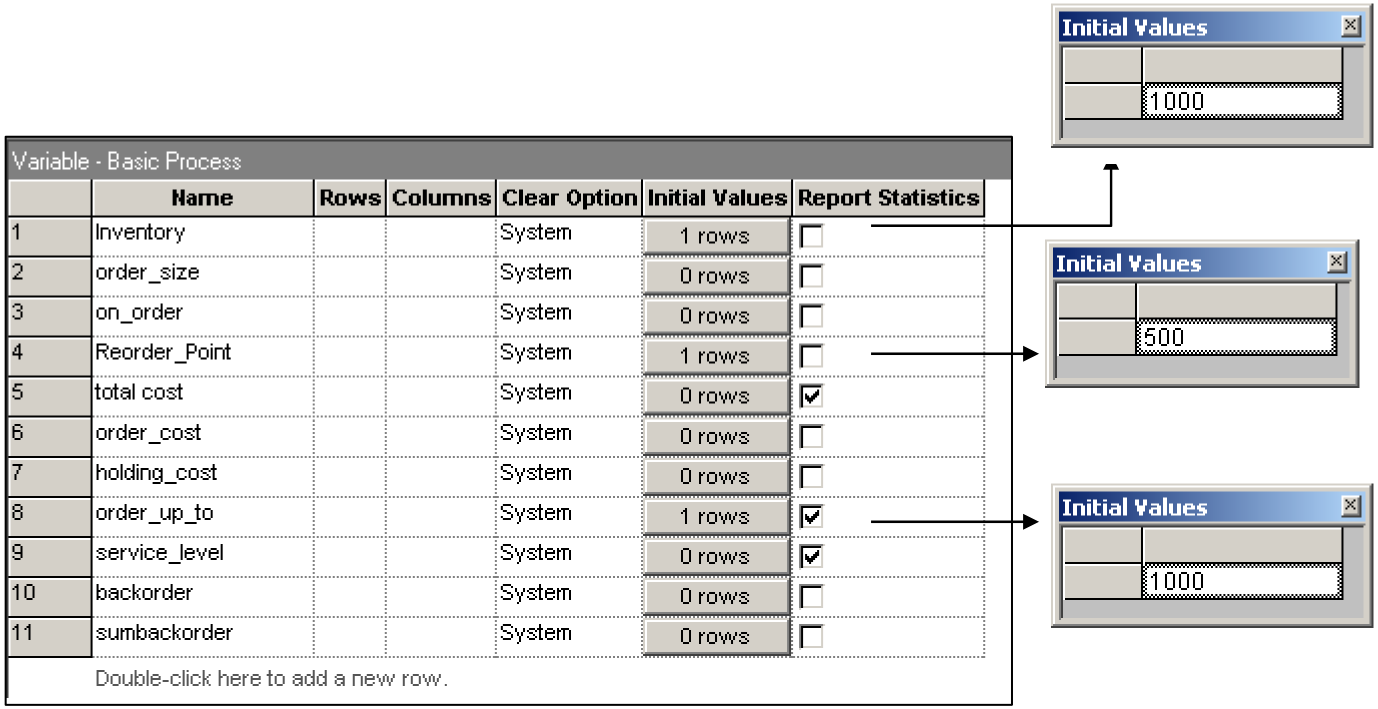


Figure1: the Arena model of (s, S) inventory policy



Variable - Basic Process

	Name	Rows	Columns	Clear Option	Initial Values	Report Statistics
1	Inventory			System	1 rows	☐
2	order_size			System	0 rows	☐
3	on_order			System	0 rows	☐
4	Reorder_Point			System	1 rows	☐
5	total cost			System	0 rows	☑
6	order_cost			System	0 rows	☐
7	holding_cost			System	0 rows	☐
8	order_up_to			System	1 rows	☑
9	service_level			System	0 rows	☑
10	backorder			System	0 rows	☐
11	sumbackorder			System	0 rows	☐

Double-click here to add a new row.

Figure2: the Arena variable table for original conditions and state in (s, S) model

periods	Table : demandsize (1)	Inventoryonhand (1)	ordersize (1)	ordertime (1)	onorder (1)	sbackorder (1)	sumboder (1)	totalbackorder (1)	deliveryquantity (1)	deliitime (1)	tcost (1)	rservicelevel (1)
50	14.6	0	0	0	1.41e+003	14.6	796	2.28e+003	0	0	0	0.492
51	8	0	0	0	1.41e+003	8	881	2.37e+003	0	0	0	0.501
52	157	0	624	52	2.04e+003	157	1.04e+003	2.53e+003	624	0	1.28e+003	0.517
53	29.7	0	0	0	2.04e+003	29.7	1.07e+003	2.55e+003	0	0	0	0.52
54	17.4	329	864	39	624	0	0	2.57e+003	864	54	0	0.522
55	62.5	266	0	0	624	0	0	2.57e+003	0	0	266	0.515
56	9.28	257	0	0	624	0	0	2.57e+003	0	0	257	0.514
57	18.6	239	0	0	624	0	0	2.57e+003	0	0	239	0.513
58	112	127	0	0	624	0	0	2.57e+003	0	0	127	0.501
59	147	0	0	0	624	20.4	20.4	2.59e+003	0	0	0	0.491
60	100	0	0	0	624	100	120	2.69e+003	0	0	0	0.501
61	55.8	0	552	61	1.18e+003	55.8	176	2.75e+003	552	0	1.14e+003	0.506
62	218	0	0	0	1.18e+003	218	394	2.97e+003	0	0	0	0.525
63	10.1	0	0	0	1.18e+003	10.1	404	2.98e+003	0	0	0	0.526
64	316	0	543	64	1.72e+003	316	719	3.29e+003	543	0	1.12e+003	0.551
65	45.5	0	624	52	1.1e+003	45.5	141	3.34e+003	624	65	0	0.554
66	150	0	0	0	1.1e+003	150	291	3.49e+003	0	0	0	0.565
67	141	0	0	0	1.1e+003	141	432	3.63e+003	0	0	0	0.575
68	29.2	0	0	0	1.1e+003	29.2	461	3.66e+003	0	0	0	0.577
69	188	0	553	69	1.65e+003	188	649	3.85e+003	553	0	1.14e+003	0.589
70	122	0	0	0	1.65e+003	122	770	3.97e+003	0	0	0	0.596
71	95.4	0	543	64	1.11e+003	95.4	323	4.06e+003	543	71	0	0.602
72	117	0	0	0	1.11e+003	117	440	4.18e+003	0	0	0	0.609
73	255	0	589	73	1.7e+003	255	695	4.43e+003	589	0	1.21e+003	0.623
74	160	0	552	61	1.14e+003	160	302	4.59e+003	552	74	0	0.631
75	131	0	0	0	1.14e+003	131	433	4.73e+003	0	0	0	0.638
76	100	0	0	0	1.14e+003	100	534	4.83e+003	0	0	0	0.643
77	97.2	0	0	0	1.14e+003	97.2	631	4.92e+003	0	0	0	0.647
78	61	0	549	78	1.69e+003	61	692	4.98e+003	549	0	1.13e+003	0.65
79	72.7	0	0	0	0	72.7	764	5.06e+003	0	0	0	0.653
80	219	0	0	0	0	219	983	5.27e+003	0	0	0	0.663

Figure3: the Output Analyzer partial outcomes table for simulation with s=500, S=1000, initial inventory level is 500.

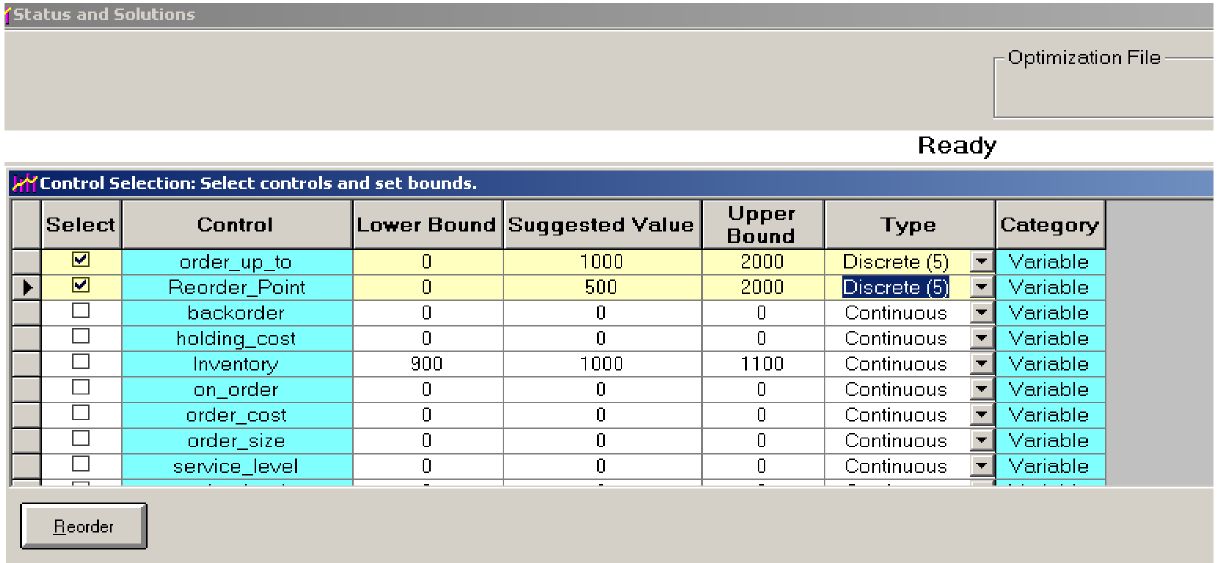


Figure4: the control selection screen in OptQuest

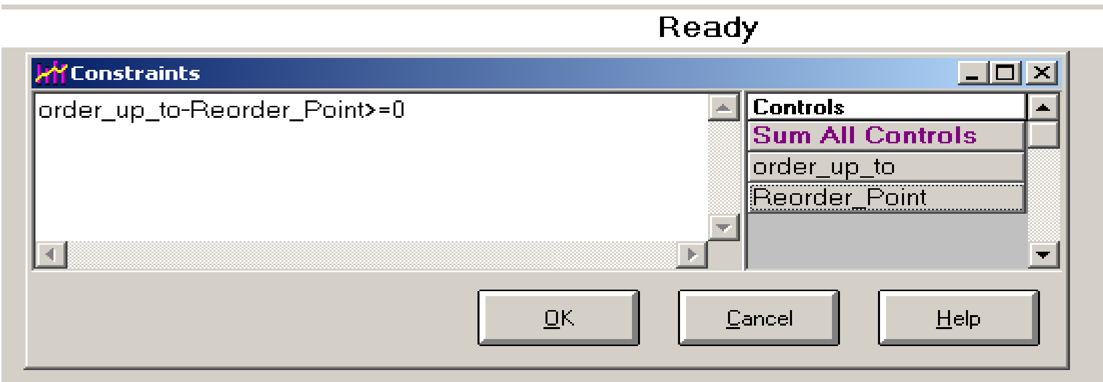


Figure5: the constrains window in OptQuest

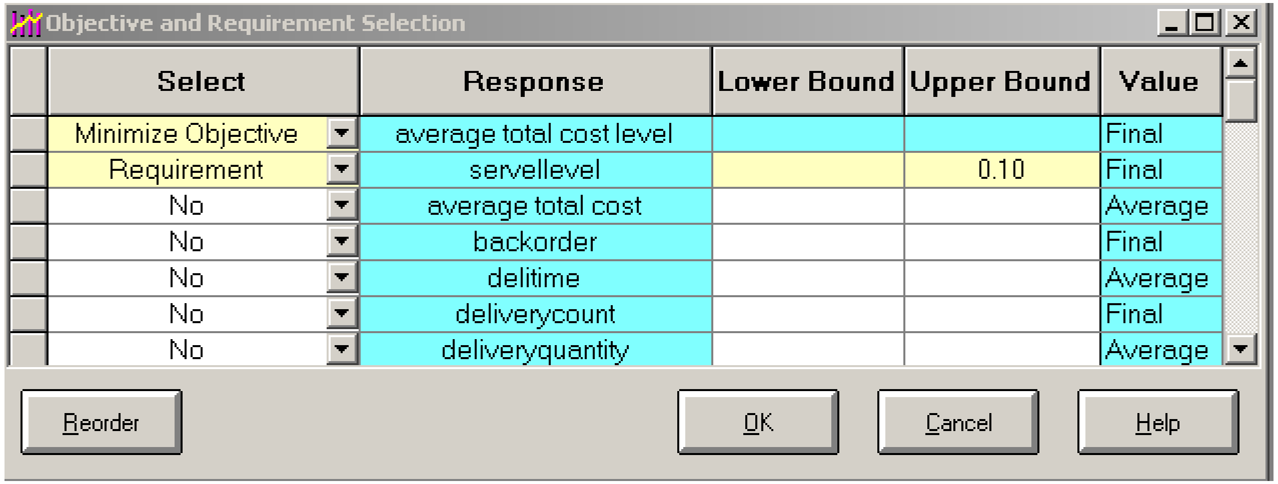


Figure6: the objective and requirement selection window in OptQuest

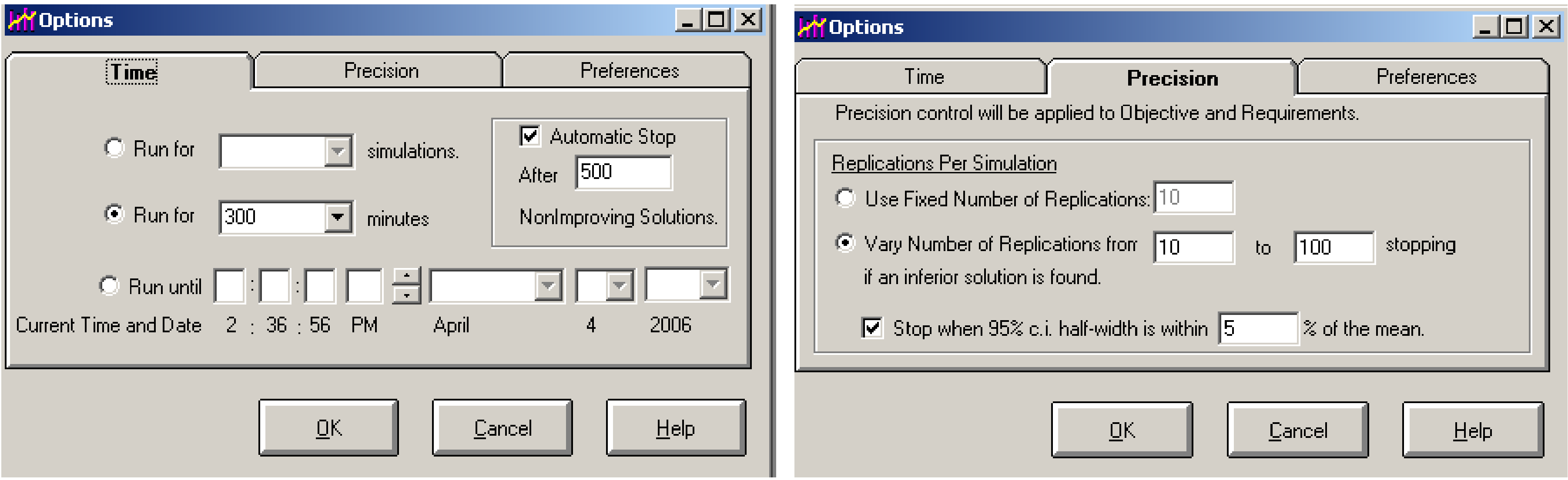


Figure7: the options window in OptQuest

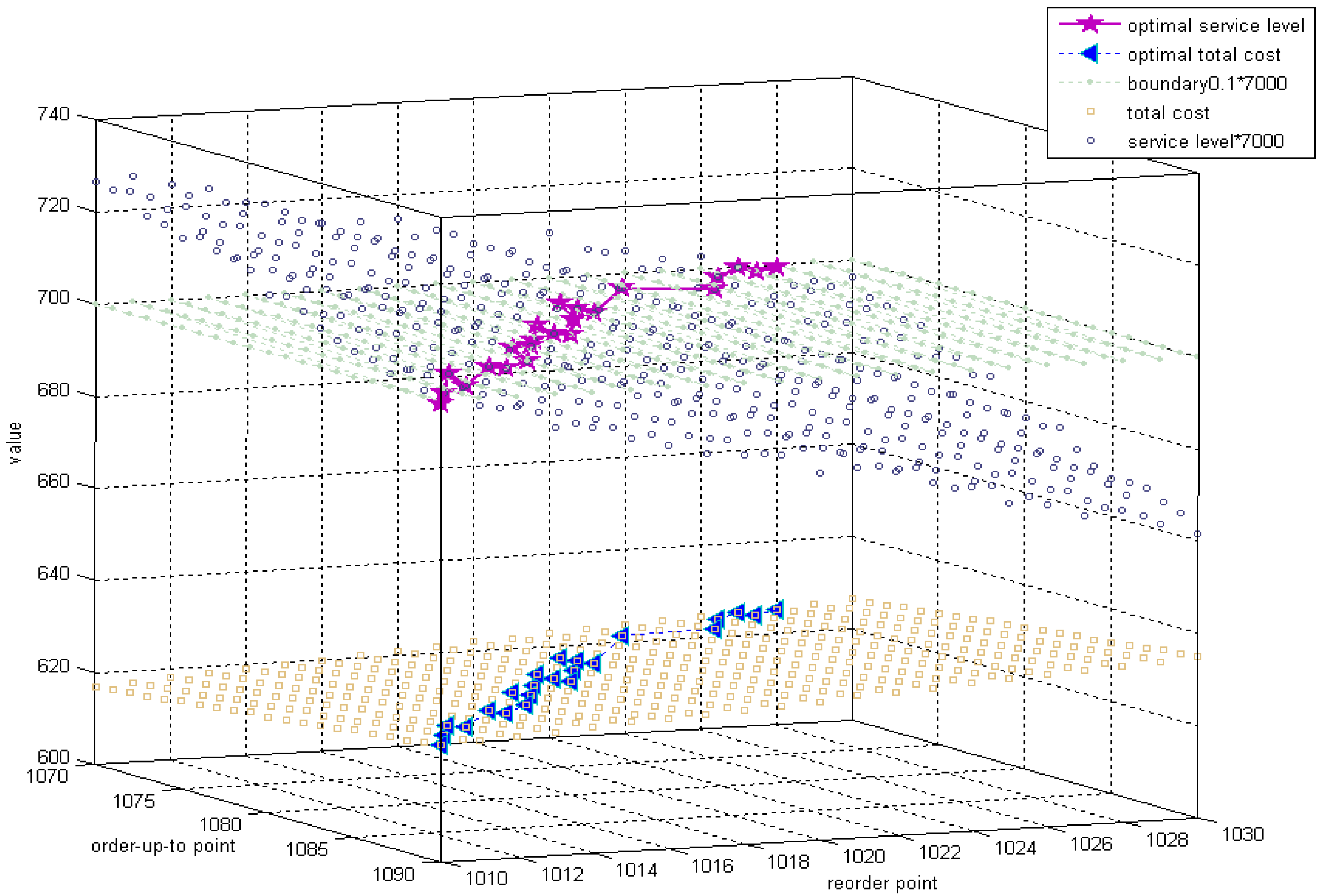


Figure 8: the average cost and $\gamma-$service level (amplified 7000 times) estimators of all combinations of s and S in the area

$$1010 \le s \le 1030 \text{ and } 1070 \le S \le 1090$$

Table 1: the estimators of (s*, S*) with γ−service level between 0.0999 to 0.1001 obtained by Arena in Brute Force

No.	(s, S)	cost	S.D. of cost	$\gamma-$service level	S.D. of $\gamma-$service level
1	1020,1075	623.739	2.3577	0.1001	0.004298
2	1023,1072	623.941	2.2963	0.0999	0.004535
3	1018,1078	624.129	2.3910	0.1	0.004293
4	1020,1076	624.188	2.5075	0.1001	0.004582
5	1017,1080	624.341	2.1866	0.0999	0.004375
6	1018,1079	624.373	2.2716	0.0999	0.004050
7	1016,1084	624.469	2.2361	0.0999	0.004519
8	1014,1084	624.473	2.3412	0.0999	0.004312
9	1012,1086	624.477	2.4472	0.1001	0.004667
10	1019,1078	624.610	2.0771	0.1001	0.004529
11	1020,1077	624.615	2.3386	0.1001	0.004075
12	1016,1082	624.803	2.1465	0.1001	0.004099
13	1018,1080	624.851	2.2445	0.1	0.004007
14	1011,1088	624.884	2.3114	0.0999	0.004223
15	1014,1085	624.926	2.0629	0.1	0.004050
16	1026,1071	624.955	2.3246	0.0999	0.004159
17	1027,1070	625.026	2.4936	0,1	0.004959
18	1028,1070	625.164	2.1842	0.0999	0.004263
19	1012,1087	625.239	2.3905	0.0999	0.004428

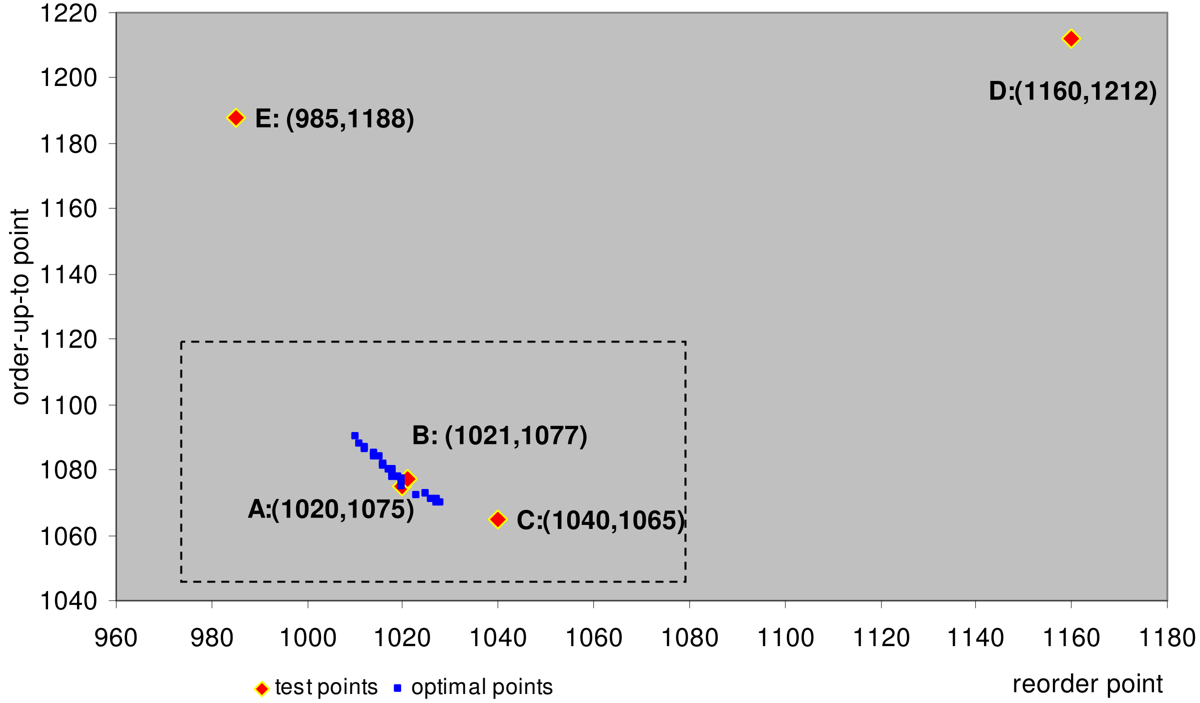
20	1010,1090	625.290	2.1919	0.0999	0.003820
21	1015,1084	625.328	2.4606	0.1	0.004044
22	1027,1071	625.422	2.4601	0.1	0.004006
23	1025,1073	625.514	2.4478	0.0999	0.004107

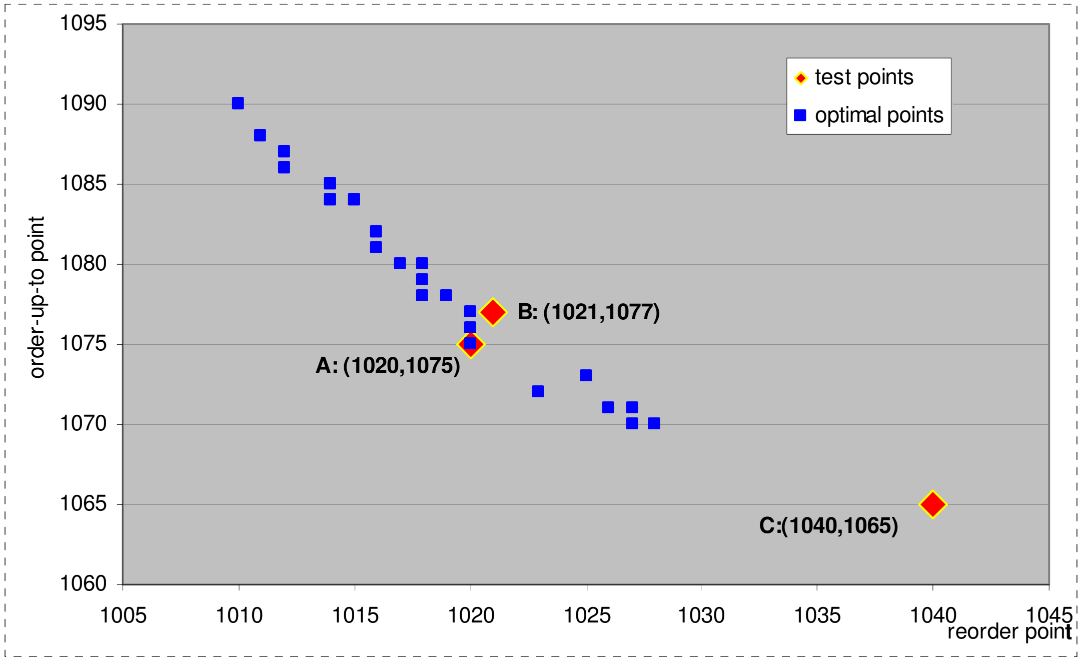
Table 2: the outcome obtained by using different optimization algorithm for the same (s, S) inventory system

Resources	Optimization Method	Denoted as	(s, S)	cost	S.D. of cost	Service Level	S.D. of Service Level
Wan and Kleijnen	Brute force	A	(1020,1075)	623.739	2.3577	0.1001	0.004298
	OptQuest	B	(1021,1077)	624.869	3.7885	0.1002	0.005001
Bashyam and Fu's(1998)	Brute force		*	702.76	*	0.1099	*
	Perturbation Analysis	C	(1040,1065)	708.20	*	0.1076	*
Angun et al (2006)	Brute force	D	(1160,1212)	647.149	8.5531	0.1052	0.0100
	RSM		(1185,1230.7)	671.3	*	*	*

* means that the information is not available.

→
Figure 9: 5 test points and the estimators of (s*, S*) by using Brute Force





← Figure 9a: the amplified part surrounded by dotted line in Figure 9

Table3: the CCD points (standard-nonstandard) for 5 test points

	No.	1	2	3	4	5	6	7	8	9 central points
	standard	-1,-1	1,-1	-1,1	1,1	$\sqrt{2}$,0	- $\sqrt{2}$,0	0, $\sqrt{2}$	0, - $\sqrt{2}$	0,0
	replications	1	1	1	1	1	1	1	1	3
A	Small 4	**1018,1073**	**1022,1073**	**1018,1077**	**1022,1077**	1022.83,1075	1017.17,1075	1020,1077.83	1020,1072.17	**1020,1075**
	Middle 10	1015,1070	1025,1070	1015,1080	1025,1080	1027.07,1075	1012.93,1075	1020,1082.07	1020.1067.93	1020,1075
	Large 20	1010,1065	1030,1065	1010,1085	1030,1085	1034.14,1075	1005.86,1075	1020,1089.14	1020,1060.86	1020,1075
B	Small 4	**1019,1075**	**1023,1075**	**1019,1079**	**1023,1079**	1023.83,1077	1018.17,1077	1021, 1079.83	1021, 1074.17	**1021,1077**
	Middle 10	1016,1072	1026,1072	1016,1082	1026,1082	1028.07,1077	1013.93,1077	1021,1084.07	1021,1069.93	1021,1077
	Large 20	1011,1067	1031,1067	1011,1087	1031,1087	1035.14,1077	1006.86,1077	1021,1091.14	1021,1062.86	1021,1077
C	Small 4	**1038,1063**	**1042,1063**	**1038,1067**	**1042,1067**	1042.83,1065	1037.17,1065	1040, 1067.83	1040, 1062.17	**1040,1065**
	Middle 10	1035,1060	1045,1060	1035,1070	1045,1070	1047.07,1065	1032.93,1065	1040,1072.07	1040,1057.93	1040,1065
	Large 20	1030,1055	1050,1055	1030,1075	1050,1075	1054.14,1065	1025.86,1065	1040,1079.14	1040,1050.86	1040,1065
D	Small 4	**1158,1210**	**1162,1210**	**1158,1214**	**1162,1214**	1162.83,1212	1157.17,1212	1160,1214.83	1160,1209.17	**1160,1212**
	Middle 10	1155,1207	1165,1207	1155,1217	1165,1217	1167.07,1212	1152.93,1212	1160,1219.07	1160,1204.93	1160,1212
	Large 20	1150,1202	1170,1202	1150,1222	170,1222	1174.14,1212	1145.86,1212	1160,1226.14	1160,1197.86	1160,1212
E	Small 4	**983,1186**	**987,1186**	**983,1190**	**987,1190**	987.83,1188	982.17,1188	985, 1190.83	985, 1185.17	**985,1188**

	Middle 10	980,1183	990,1183	980,1193	990,1193	992.07,1188	977.93,1188	985,1195.07	985,1180.93	985,1188
	Large 20	975,1178	995,1178	975,1198	995,1198	999.14,1188	970.86,1188	985,1202.14	985,1163.86	985,1188

A. (1020,1075) obtained by Wan and Kleijnen in Brute Force method

B. (1021,1077) obtained by Wan and Kleijnen in OptQuest

C. (1040,1065) obtained by Bashyam and Fu

D. (1160, 1212) obtained by Angun et al(2006) in RSM

E. (980, 1188) non-optimal point obtained by Wan and Kleijnen

Table 4: the KKT testing results for different points when using small sample size and 500 macro-replications

A. (1020,1075) obtained by Wan and Kleijnen in Brute Force method	Local Area Size 4		Local Area Size 10		Local Area Size 20	
	Small noisy	Large noisy	Small noisy	Large noisy	Small noisy	Large noisy
Reject Binding constraint	37/500=0.074	31/500=0.062	37/500=0.074	29/500=0.058	37/500=0.074	29/500=0.058
Polynomial lack-of-fit	33/463=0.071	37/469=0.079	29/463=0.063	36/471=0.076	31/463=0.067	37/471=0.079
Reject Linear KKT model	11/430=0.024	5/432=0.011	17/434=0.039	6/435=0.013	18/422=0.043	19/434=0.044
Negative Lagrange multipliers	44/430=**0.103**	45/432=**0.104**	32/434=0.074	41/435=0.094	4/422=0.009	43/434=0.099

B. (1021,1077) obtained by Wan and Kleijnen in OptQuest	Local Area Size 4		Local Area Size 10		Local Area Size 20	
	Small noisy	Large noisy	Small noisy	Large noisy	Small noisy	Large noisy
Reject Binding constraint	40/500=0.08	22/500=0.044	28/500=0.056	28/500=0.056	28/500=0.056	27/500=0.054
Polynomial lack-of-fit	29/460=0.063	41/478=0.086	36/472=0.076	31/472=0.066	29/472=0.061	35/473=0.074
Reject Linear KKT model	6/431=0.014	0/437=0	19/436=0.044	9/441=0.020	29/443=0.065	12/438=0.027
Negative Lagrange multipliers	47/431=**0.109**	51/437=**0.117**	34/436=0.078	40/441=0.091	5/443=0.011	47/438=**0.107**

C. (1040,1065) obtained by Bashyam and Fu	Local Area Size 4		Local Area Size 10		Local Area Size 20	
	Small noisy	Large noisy	Small noisy	Large noisy	Small noisy	Large noisy
Reject Binding constraint	33/500=0.066	27/500=0.054	32/500=0.064	30/500=0.060	32/500=0.064	30/500=0.060
Polynomial lack-of-fit	35/457=0.075	38/473=0.080	35/468=0.075	38/470=0.081	42/468=0.090	25/470=0.053

Reject Linear KKT model	21/432=0.049	6/435=0.014	44/433=**0.102**	11/432=0.025	87/426=**0.204**	21/445=0.047
Negative Lagrange multipliers	48/432=**0.111**	52/435=**0.120**	29/433=0.067	49/432=**0.113**	3/426=0.007	**55/445=0.124**

D. (1160, 1212) obtained by Angun et al(2006) in RSM	Local Area Size 4		Local Area Size 10		Local Area Size 20	
	Small noisy	Large noisy	Small noisy	Large noisy	Small noisy	Large noisy
Reject Binding constraint	500/500=**1**	500/500=**1**	500/500=**1**	500/500=**1**	500/500=**1**	500/500=**1**
Polynomial lack-of-fit	0	0	0	0	0	0
Reject Linear KKT model	0	0	0	0	0	0
Negative Lagrange multipliers	0	0	0	0	0	0

E. (980, 1188) non-optimal point obtained by Wan and Kleijnen	Local Area Size 4		Local Area Size 10		Local Area Size 20	
	Small noisy	Large noisy	Small noisy	Large noisy	Small noisy	Large noisy
Reject Binding constraint	92/500=**0.184**	43/500=0.086	92/500=**0.184**	43/500=0.086	92/500=**0.184**	43/500=0.086
Polynomial lack-of-fit	20/408=0.049	35/457=0.077	23/408=0.056	27/457=0.059	28/408=0.069	34/457=0.074
Reject Linear KKT model	3/388=0.008	4/422=0.009	7/385=0.018	8/430=0.019	6/380=0.016	9/423=0.021
Negative Lagrange multipliers	68/388=**0.175**	**60/422=0.142**	40/385=**0.104**	54/430=**0.126**	2/380=0.005	58/423=**0.137**

Table 5: the KKT testing results for different points when using large sample size

	A. (1020,1075) obtained by Wan and Kleijnen in Brute Force method	B. (1021,1077) obtained by Wan and Kleijnen in OptQuest	C. (1040,1065) obtained by Bashyam and Fu in PA	D. (1160, 1212) obtained by Angun et al (2006) in RSM	E. (980, 1188) non-optimal point obtained by Wan and Kleijnen
Polynomial lack-of-fit	0/100=0	0/100=0	1/100=0.01	1/100=0.01	1/100=0.01
Reject Binding constraint	9/100=0.09	4/100=0.04	4/99=0.04	99/99=**1**	33/99=**0.33**
KKT Rejected	6/91=0.066	21/96=**0.22**	27/95=**0.28**	0/0	21/66=**0.32**