Evaluating Optimization Model Based Decision Support Systems in the Framework of a System Dynamics Based Game

Biswajit Mahanty and Pratap K J Mohapatra
Department of Industrial Engineering and Management
Indian Institute of Technology
Kharagpur721302, India

Abstract

Evaluating optimization model based decision support systems is a complex task. Once an optimization model is built, one is not sure how to compare the effectiveness of two competing optimization models. One is also not sure how the model results will fare when they are actually implemented in practice. This paper lays out a procedure for evaluating optimization models in the framework of a system dynamics based game. Using the suggested procedure, a number of optimization model based decision support systems are evaluated. Investigations are then carried out on the effect of such decision support systems on the performance of the game participants. The proposed procedure opens up the possibilities of developing realistic and credible optimization models by testing them for their effectiveness in the context of a specific problem situation.

Evaluating Optimization Model Based Decision Support Systems in the Framework of a System Dynamics Based Game

Introduction

Optimization models form the core of the management science approach to managerial decision making. The rationality they depict, the objectivity they assume, the optimality they purportedly reach, and the exactitude they preach have made them very popular to the decision makers.

However, evaluation of optimization models is a subjective task. Once an optimization model is developed, one is not sure how the model results will fare when they are actually implemented in practice. One is also not sure how to compare the effectiveness of two competing optimization models.

Traditionally, evaluation of optimization models is carried out by embedding them in DSS's (Mehrez et al. 1988). However, such an approach does not give an "a-priori" opportunity to test the model for its appropriateness before it is put to use in a DSS environment.

This paper details a procedure in which the effectiveness of optimization models can be tested prior to its implementation in the framework of a system dynamics based game that creates a DSS-like environment. A number of optimization models, developed as decision support systems for the game participants, are evaluated by using the suggested procedure. The effect of such decision support systems on the performance of the game participants is then reported.

The Evaluation Procedure

System dynamics models have been used in the past as data generators to test the effectiveness of econometric models (Senge 1975; Mass and Senge 1977). To test the effectiveness of optimization models, system dynamics models will not only generate the data but also implement the optimal decisions and simulate them. To accept the optimal decisions and simulate them intermittently at regular intervals requires that the system dynamics model be converted into its game form.

A system dynamics model, in its game form, provides excellent opportunity to test the effectiveness of optimization models in updated parameter environment just as in a DSS (Mohapatra 1993). An optimization model provides the optimal decisions valid for a small time period. A system dynamics model accepts these decisions in its game mode and simulates the model in time for the stipulated plan period. As time advances with the simulation, new parameter values emerge. The optimization model now accepts these parameter values and gives new optimal decisions for the next plan period. The process continues.

Just as a manager modifies the recommended optimal decisions to take care of real life exigencies, a system dynamics model treats the optimal decisions as desired values of policy variables and modifies them in the light of local constraints to obtain realistic values of policy variables.

In essence, therefore, the optimization model and the system dynamics model must interact with each other in a cyclic fashion (Figure 1).

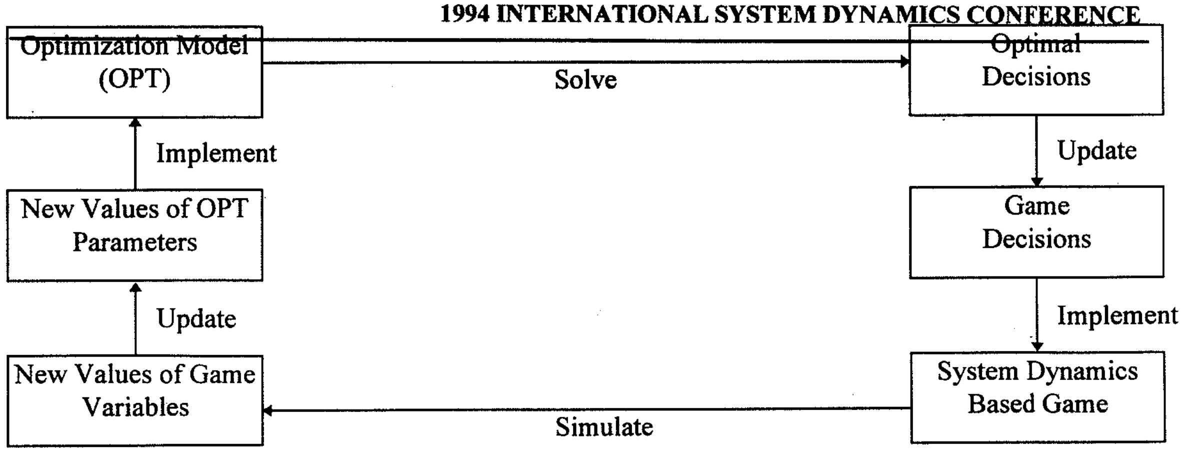

Figure 1: Evaluating Optimization Models

The Game

The authors have developed a computer-based game depicting quality fluctuations of products manufactured by a company. The game is named "QGAME", an acronym for Quality GAME. The quality fluctuations arise out of the hiring and production policies of the company.

QGAME is based on a well-known text book system dynamics model ("Future Electronics Model" by Jarmain 1974). The original model is appropriately modified to include the production-inventory dynamics. The use of a system dynamics model has lent to the game the transparency of the underlying model situation due to the use of a causal loop diagram (Figure 2).

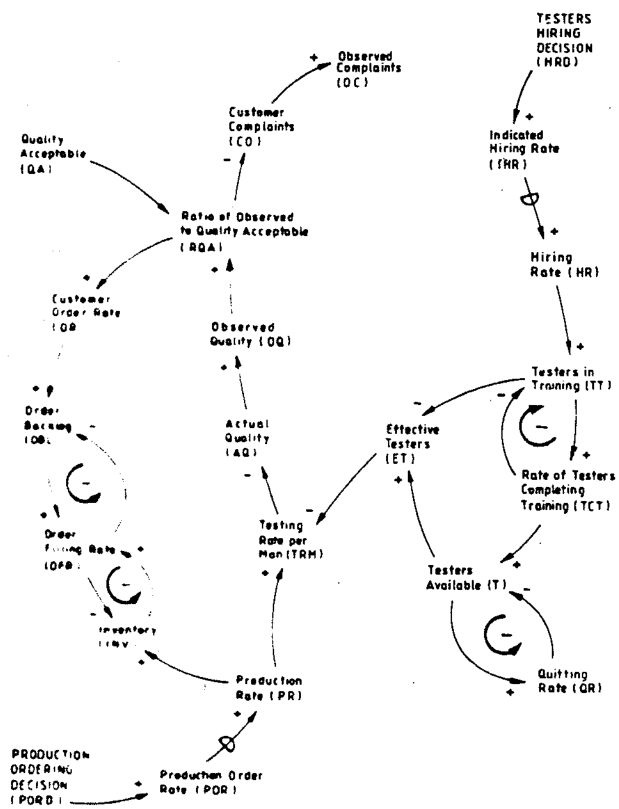

Fig 2 Causal Loop Diagram of the Game System Dynamics Model.

Two decision variables are selected for the QGAME. A participating team makes these decisions for a plan-period of one year from the second plan-period onwards. These are:

(a) **annual production ordering decision**, and
(b) **annual testers hiring decision**.

The original model equations are run for the first plan-period starting with initial steady state values. A step rise of 20% is given to the customer order function at the beginning of the model run. The system shows unsteady values after the first plan-period. The first plan-period results are provided to the game participants to form the background material.

Two evaluative variables are designed in QGAME. The **plan performance** is defined as the total earnings minus the total cost in a plan-period. The **value of the game** is the average value of the plan performances. The game participants' goal is to attain a steady state of operation with a high value of the game.

QGAME is an interactive game. Each participating team comprises two members. The two members jointly make the production and the hiring decisions for the next plan-period. Each team is provided with a personal computer along with a copy of the game software, the first plan-period results in the form of a plan report, a performance sheet, monthwise plots of the key variables, and a decision sheet. In the debriefing session that follows the game play, each team presents the game results, the decision-making strategies followed by them, and the effectiveness of such strategies. The teams also relate their experience with the real world.

Performance benchmarks, that provide absolute measures to compare and evaluate individual team results, were also developed in QGAME by making use of best policy runs of the system dynamics model. These are depicted in the Table 1:

Table 1: Performance Benchmark Values

The Performance Measures	Benchmark Results	
	Steady-state	All-plan Average
Value of the Game	1078	1001
Reaching Steady-State	After 5 plans	-
Observed Quality	1.1	1.07
Testers Available	18	17
Inventory	1320	1310
Order Backlog	1320	1641

QGAME was played by groups of industrial engineering students and research scholars, administrative officers, and practising managers of India. Altogether, 132 persons, divided into 66 teams, played the game. **The Optimization Models**

The different optimization models evaluated by the authors are actually variations of a basic optimization model considering different sets of the constraints and the objective functions.

The basic optimization model is developed by considering 12 periods of one month duration each for a plan-period of one year. The decisions and the input values are read for the entire plan-period. Other variables, important in the model, are represented by a set of equations in such a way that they can, ultimately, be expressed as functions of the two decision variables and the input variable values.

The constraints are derived from the considerations of the minimum quality acceptable, maximum inventory acceptable, maximum order backlog, and replacement hiring.

A number of objective functions are considered for the development of the model. The objectives functions, finally included in the model, are maximization of the plan performance, maximization of the quality, and a combination of the two.

The model is nonlinear in nature and it is solved by a grid-search technique. The details of the model development are available in Mahanty (1993).

Variations of the Optimization Model

Four variations of the basic optimization model are developed in the form of decision support systems. **DSS-1** and **DSS-2** attempt to maximize the plan performance for a plan-period. **DSS-3** attempts to maximize the average of actual quality, and **DSS-4** attempts to maximize the plan performance (with 40 per cent weight) as well as the average of the actual quality (with 60 per cent weight).

Excepting for DSS-1, all the other DSSs consider all the constraints including that of replacement hiring. The constraint of replacement hiring is considered because it is a part of the revised testers hiring policy (Jarmain 1974). The game participants, however, may not be aware of any such policy, and inadvertently may not consider it. Such an environment is created by not considering the constraint in the formulation of DSS-1.

Evaluation of the Optimization Models

All the four optimization models are evaluated by simulating the optimal decisions in the QGAME. The game results are shown in Table 2 in the form of all-plan average values of the observed quality, the value of the game, the inventory, the order backlog, the testers, and the testers in training. The corresponding performance benchmark results are also included.

Table 2: Evaluation of DSSs: All-Plan Average Values

sl.	DSS	Testers Avl.	Observed Quality	Inventory Position	Order Backlog	Value of the Game
1.	DSS-1	7.60	0.84	1284	1821	643
2.	DSS-2	14.02	0.99	1473	1942	844
3.	DSS-3	20.51	1.07	1259	4471	871
4.	DSS-4	16.13	1.03	1420	1741	923
5.	Benchmark	16.94	1.07	1310	1641	1001

The results show that DSS-1 is the least effective. DSS-1 does not consider replacement hiring. Therefore the average number of testers are always less than adequate. That has resulted in low quality, low customer order, and low value of the game. The values of the inventory and the order backlog, however, compare well with the benchmarks.

The results obtained for the DSS-2 are far better than those for the DSS-1. Since DSS-2 considers replacement hiring, the number of testers available has been adequate on the average. Thus, compared to the DSS-1, the quality level is good, customer orders are high, and the overall performance is better. The inventory and the order backlog are also close to the benchmarks.

The results obtained from the DSS-3 are also good. A quality maximization view has led to a good performance in the DSS-3 run. The value of the game is high and the average quality level is same as that of the benchmark run. Average inventory is low, but average order backlog is slightly higher. However, more stress on the quality has led to more hiring than required, and because of this, the performance has suffered to a certain extent.

The best results are obtained for the DSS-4 run for which the value of game, the average level of the quality, testers available, the inventory, and the order backlog are all comparable to those for the benchmark run. A combined objective function of maximizing plan performance and maximizing quality has led to the achievement of such near-benchmark results.

Use of Decision Support Systems in Game Related Investigations

DSS-1 and DSS-2 are selected as decision aids for the game participants. The experience, obtained during the model development exercise with groups of students, was useful in making the choice. Most of the students had selected an objective function of maximizing the profit. Since both DSS-1 and the DSS-2 attempted to maximize the plan performance (a surrogate for profit), these decision supports appear to approximate the wishes of the participants the most. It was also thought that they would appear most appealing to the game participants. The results of the DSS-1, the DSS-2, and the benchmark runs are depicted in Figure 3 in terms of the planwise actual values. Figure 3 shows that the performance has gradually deteriorated in the case of DSS-1. The final value of the observed quality was as low as 0.6. Figure 3 also shows much better performance in the case of DSS-2. Although there are quality fluctuations, the behaviour has improved with time. The DSS-2 model is different from the DSS-1 in only one count. It considers replacement hiring which can be considered as an "anchor" to the testers hiring decision. The use of "anchoring and adjustment" as a decision-making strategy is well known. It was, therefore, thought to be important to study the effect of the DSS-1 and DSS-2 on the performance of the game participants.

Effect of the DSS-1 on the Game Performance

10 teams during the game play were provided with optimal decisions from DSS-1. The teams appeared to have not performed well. Only two teams scored a value of game above 800, three teams scored between 700 and 800, four between 600 and 700, and the lowest score was 597. All the teams were characterized by low observed quality (less than 1.0 quality unit) and small number of testers (less than 15). The teams, however, managed the inventory and the order backlog well. All the teams had the average inventory and order backlog positions less than 4000 units.

The mean absolute deviations for the production ordering decisions varied from around 2000 units/year in the initial periods to 3000 to 4000 units/year in the later plan-periods, and, the mean absolute deviations for the hiring decisions rose steadily from near zero to almost five testers. This suggests that the teams almost totally disregarded the optimal decisions provided to them. In fact, the correlation co-efficients (r) between the optimal and the actual decisions were found to be very low,

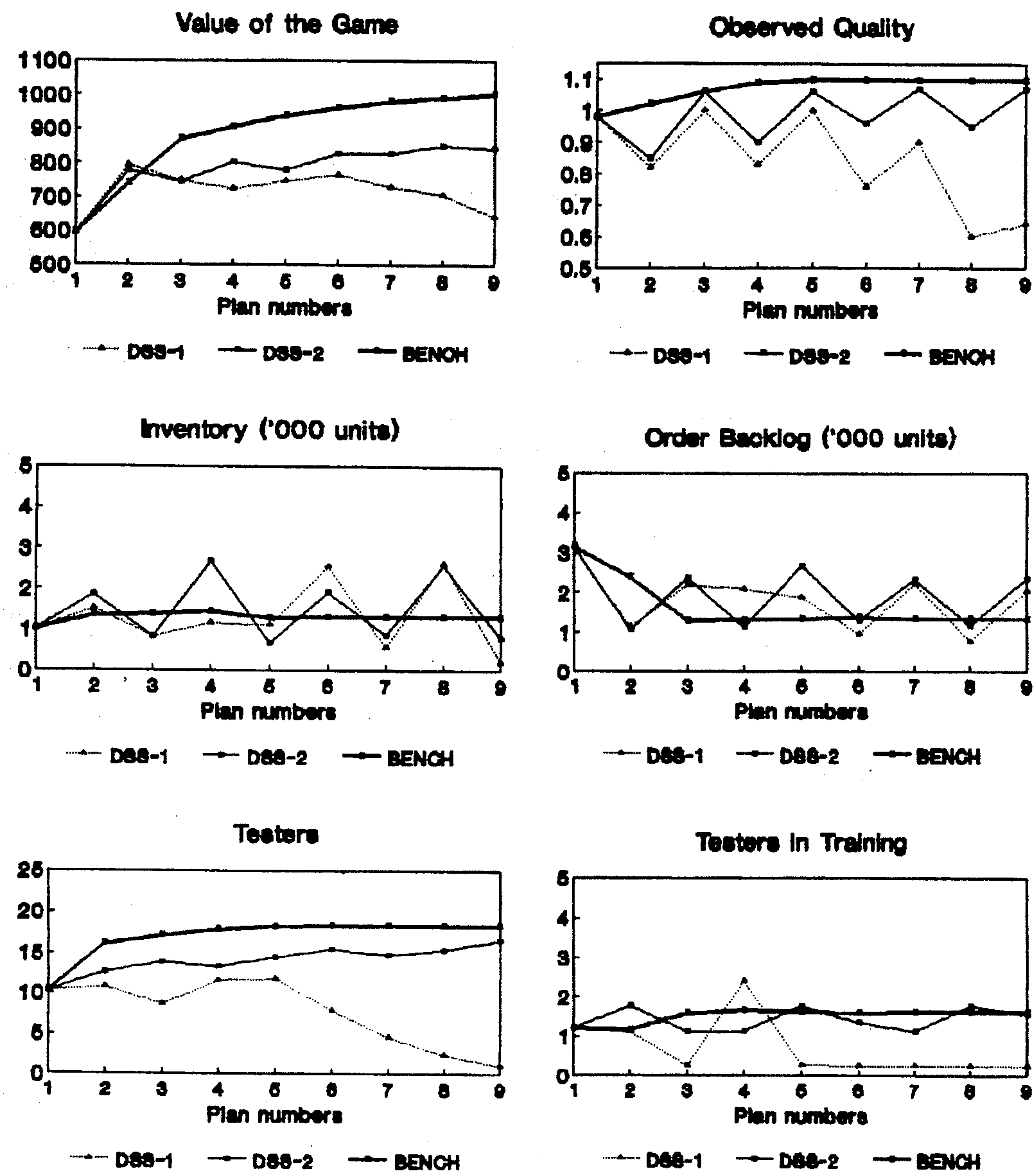

Figure 3: DSS-1, DSS-2, and Benchmark Results: Planwise Actual Values

0.4384 for the production ordering and -0.1440 for the testers hiring, confirming that the teams did not use DSS-1 recommendations.

DSS-1 used an optimization model which was known for its inferior optimal solutions. An attempt to maximize the plan performance in the short-term led the teams to ignore the impact of such decisions on the long term health of the company. The teams ignored the level of the observed quality, the number of testers available, the customer orders received, and other important variables in the initial periods. The teams apparently rejected the model recommendations after the first two plan-periods and could not perform well. However, as is evident from their debriefing session comments, the teams could realize the importance of the long term policy considerations in the later stages of the game.

Thus, one may say that optimization models, deficient in some important respect, even when residing in DSSs, are likely to be rejected by the decision makers because implementation of the recommended optimal results may not give the expected results.

Effect of the DSS-2 on the Game Performance

11 teams during the game play were provided with optimal decisions from DSS-2. These teams showed consistently good performance. Although only one team scored above 900, seven teams scored above 800 and the lowest value of the game was 697. The average observed quality level was higher than 1.0 quality unit for six teams, and around 0.95 quality unit for the rest. The teams had more testers (4 teams had above 15 men). They also maintained their inventory and the order backlog well (less than 4000 units). As many as three teams of this group reached a steady state of operation at the highest level of quality.

The teams of this group agreed that they followed the optimal decisions to a large extent. This was supported by high values of the correlation co-efficients (r) between the optimal and the actual decisions of these teams ($r = 0.6915$ for the production ordering decisions and $r = 0.4297$ for the testers hiring decisions).

That the teams followed the optimal decisions to a large extent was also supported by the values of mean absolute deviations (MAD) between the actual and the optimal decisions for all the teams. The mean absolute deviations in the production ordering decisions varied from 1000 to 3000 units/year. The mean absolute deviations for the hiring decisions remained within one to two testers per year.

In the debriefing session, the teams said that their strategies had involved making corrections for the inventory and the order backlog in the light of the previous year's production orders for arriving at the new production ordering decisions. The strategies also involved compensating for the leaving rate of testers while making the testers hiring decisions.

The DSS-2 optimization model was based on a hiring strategy that considered an "anchor" in the form of a constraint that ensured replacement hiring and that was known to give superior results on implementation. The model, therefore, provided useful decisions to the game participants. The teams appeared to follow the recommended solutions of DSS-2 and performed well. However, little could be inferred on the actual understanding of these teams on the decision situation of the QGAME.

Thus, one may say that when an optimization model used for decision support to a manager yields good result, it gets credibility and acceptance from the managers. However, the decision-making ability of a manager may not be enhanced by the use of such an optimization model.

Conclusions

In this paper, a procedure is laid out for evaluating a number of optimization model based decision support systems in the framework of a system dynamics based game on quality fluctuations (QGAME). Four optimization models were successfully evaluated by using the proposed procedure.

Two of the optimization models were integrated with the game to provide decision support to the game participants. It was revealed that an optimization model, that excludes an important consideration, is likely to result in inferior performance during its evaluation in the framework of system dynamics based games. Such inferior optimization models run the risk of being rejected by the real-world managers. Potentially superior optimization models, that perform well in a game environment, are very likely to be adopted as decision aids. In fact, how the decision support systems fare in the first few plan-periods after they are implemented holds the key to the credibility it will earn for itself and to the likelihood of its acceptance by the decision makers.

The authors stress that a major contribution of system dynamics based games is in their use in evaluating alternative optimization models. The proposed procedure opens up the possibilities of developing realistic and credible optimization models by testing them for their effectiveness in the context of a specific problem situation.

References

Jarmain, E. 1974. Future Electronics Model. In Study Notes in System Dynamics, ed. M.R. Goodman, 349-364. Massachusetts: MIT Press.

Mahanty, B. 1993. A System Dynamics Gaming Study of Decision-Making Behaviour. Unpublished Ph.D. Dissertation, Indian Institute of Technology, Kharagpur, India.

Mass, N.J., and P.M. Senge. 1977. Alternative Tests for the Selection of Model Variables. Working Paper D-2636-1. MIT, USA: System Dynamics Group, MIT.

Mehrez, A., G. Rabinowitz, and A. Reisman. 1988. A Conceptual Scheme of Knowledge Systems for MS/OR. Omega 16(5): 421-428.

Mohapatra, P.K.J. 1993. Building and Evaluating Optimization Models in System Dynamics Framework. In System Dynamics for Management Support, ed. Sushil, 98-104. New Delhi: Wiley Eastern Ltd.

Senge, P.M. 1975. Testing Estimation Techniques for Social Models. Working Paper D-2199-4. MIT, USA: System Dynamics Group, MIT.