Computers & Industrial Engineering 60 (2011) 555–565

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

A Lexicographic Nelder–Mead simulation optimization method to solve multi-criteria problems ☆

Glenn W. Kuriger [a,*], F. Hank Grant [b]

[a] Department of Mechanical Engineering – Center for Advanced Manufacturing and Lean Systems, University of Texas at San Antonio, One UTSA Circle, San Antonio, TX 78249, USA
[b] Department of Industrial Engineering, University of Oklahoma, 100 East Boyd, Suite R-208, Norman, OK 73019, USA

ARTICLE INFO

Article history:
Received 6 October 2009
Received in revised form 27 August 2010
Accepted 21 December 2010
Available online 30 December 2010

Keywords:
Simulation optimization
Goal programming
Multi-criteria optimization
Simulation

ABSTRACT

Simulation optimization is an approach that is used to solve problems involving stochastic components. This paper presents a Lexicographic Nelder–Mead (LNM) based simulation optimization (LNM-SO) method to solve multi-criteria simulation optimization problems. The method is designed to be relatively easy-to-implement and be applicable to a wide range of problem domains. To effectively evaluate the overall performance of this method, a Time-Quality Estimator (TQE) was developed to evaluate the performance of LNM-SO in terms of both quality of solution and computational speed. Computational results of five different test problems showed that the method was highly effective.

1. Introduction

Real-world problems frequently involve variables that exhibit some form of random behavior. Traditional optimization techniques often cannot adequately solve these problems. One approach to solve problems that involve stochastic coefficients in optimization components is by incorporating a simulation model within the optimization tool. The simulation model is used to address this stochastic nature of the system and to find feasible and representative values for the coefficients of the model, while optimization is used to find an optimal or near-optimal solution. This process is referred to as *simulation optimization*. For problems where multiple and usually conflicting objectives are to be optimized, the process is referred to as multi-criteria simulation optimization.

Simulation optimization essentially combines simulation with an optimization technique or heuristic to determine the combination of input parameters that produces a near-optimal solution for one or more output measures. The optimization problem can involve stochastic responses in the objective function coefficients, the constraint coefficients, or both. Typical examples of stochastic responses include arrival rates, customer demand rates and quantities, time between failures, lead times, and throughput rates.

This research proposes a Lexicographic Nelder–Mead (LNM) method (Kuriger & Ravindran, 2005) based simulation optimization method to solve multi-criteria simulation optimization problems. This method was developed by adapting and extending the LNM method to incorporate simulation and goal programming. The proposed methodology has been applied to solve five different test problems, representing five different problem domains and the results were compared to the solutions obtained from two competing simulation optimization methods.

The main objectives of this research were to:

- Develop a simulation optimization method based on the LNM optimization technique to solve multi-criteria problems involving stochastic components.
- Validate the effectiveness and applicability of the proposed LNM simulation optimization (LNM-SO) methodology by ensuring a high quality solution can be obtained as fast as possible for a wide range of problems.
- Identify a performance measure that can be used to simultaneously evaluate the quality of solution and the computational speed.
- Establish a series of test problems to evaluate the effectiveness of LNM-SO.

The following sections describe the techniques, methodology, and results of the computational analysis of the LNM-SO method. Section 2 provides a review of the related literature and a description of the underlying problem. Section 3 develops the mathematical models used in the problem and presents the methodology that was used to implement the research. Section 4 provides an overview of the LNM-SO method developed in this research.

☆ This manuscript was handled by Area Editor Paul Savory.

* Corresponding author. Tel.: +1 210 458 6792; fax: +1 210 458 6504.
E-mail address: glenn.kuriger@utsa.edu (G.W. Kuriger).

0360-8352/$ - see front matter
doi:10.1016/j.cie.2010.12.013

Section 5 details the proposed performance evaluation criteria, the comparison methods, and the test problems. Section 6 presents the results of the computational analysis. Finally, Section 7 presents the general conclusions that can be drawn from this research and offers recommendations for further research.

2. Background

Simulation optimization can be used to determine the system configuration (i.e., the set of input parameters) to provide a near-optimal if not an optimal solution (Law & McComas, 2002). Further discussion on simulation optimization formulation can be found in Law and Kelton (2000). For a complete introduction to typical simulation optimization methods, see Tekin and Sabuncuoglu (2004), Olafsson and Kim (2002), Fu (1994, 2001), Swisher, Hyden, Jacobson, and Schruben (2000), Azadivar (1999), Andradottir (1998), and Carson and Maria (1997).

Real problems often require that several objectives should be optimized. Multi-criteria problems involving conflicting objectives are difficult in themselves to solve. Combining multi-criteria optimization with simulation optimization adds the extra complexity of stochastic components into the problem (Tekin & Sabuncuoglu, 2004). Evans, Stuckman, and Mollaghesemi (1991) describe some issues unique to multi-criteria simulation optimization. These include: the relationship between the input and output variables is not of a closed form, the output variables may be random or stochastic variables, and many local optima may be present on the response surface.

There have been numerous methods proposed in the literature that use simulation optimization to address a multi-criteria problem. Mollaghesemi and Evans (1994) proposed a simulation optimization method to solve multi-criteria problems that is a modification of the STEP method. The method is interactive, that is, the decision-maker is asked to provide input after each iteration to guide the algorithm. Teleb and Azadivar (1994) also propose an interactive algorithm to solve multi-criteria simulation optimization problems. The method is a modification of Box's complex search method (Box, 1965) and assumes that the stochastic objectives and constraints follow a normal distribution. Several other multi-criteria simulation optimization methods were found in the literature that vary primarily on the specific optimization method they use. These optimization methods include: genetic algorithms (Al-Aomar, 2002; Baesler & Sepulveda., 2000, 2001; Eskandari, Rabelo, & Mollaghasemi, 2005), simulated annealing (Avello, Baesler, & Moraga, 2004), and response surface methodology (Boyle & Shin, 1996; Rees, Clayton, & Taylor, 1985; Yang & Tseng, 2002; Kleijnen, 2008; Dellino, Kleijnen, & Meloni, 2010).

Other simulation optimization methods vary primarily on how they handle the multiple objective aspect of the problem. One common approach is to use a criterion model. A criterion model is a means of combining the output of multiple performance measures and ranking the outcome of an experiment based on the preferences of the decision-maker (Anderson, Evans, & Biles, 2006). These include Szidarovszky and Eskandari (1999) and Anderson et al. (2006). Another common approach is to use goal programming. These include Yang and Tseng (2002), Baesler and Sepulveda (2000), Baesler and Sepulveda (2001), Clayton, Warren, and Taylor (1982), and Rees et al. (1985).

The majority of the methods found in the literature were developed to solve, or at least demonstrated using, one specific example. It is not often clear how to implement these methods to other problem domains. This paper seeks to present a method that is relatively easy-to-implement and that is applicable to a wide range of problems.

3. Methodology

This research focused on problems that involved multiple objectives. Pre-emptive goal programming (Ravindran, Phillips, & Solberg, 1987) was the approach that was used to address the multiple criteria. In pre-emptive goal programming, it is assumed that the decision-maker wants to satisfy higher priority goals much more than lower priority goals. The decision-maker must rank the goals in order of decreasing importance. Additionally, the decision-maker would need to set target values for each goal. If a goal satisfies its target value, then it is said to have been achieved. It is important to note that for the case of conflicting objectives, the only way to improve the state of a lower priority goal is if alternate optima exist for higher priority goals. Thus, setting appropriate target values is very important.

In order to make each goal more comparable to the other goals, the deviation from the target value was normalized by dividing it by the absolute value of the corresponding target value. The objective function, Z, for each test problem was calculated by taking the sum of the normalization of the deviation from each goal and multiplying it by the corresponding priority factor. Thus, Z was calculated using the following equation:

$$Z = \sum_{i=1}^{g} P_i \left(\frac{d_i^+ + d_i^-}{|T_i|} \right) \tag{1}$$

where Z is the objective function to be minimized; g are the number of goals, T_i are the target values for the goals, P_i are the priority factors or penalties set by the decision-maker, d_i^+ are the positive deviational variables, and d_i^- are the negative deviational variables.

To ensure that the higher priority goals were satisfied first, it was important that the priority penalty be weighted heavily in favor of the higher priority goals (i.e., $P_1 \gg P_2 \gg \ldots \gg P_{g-1} \gg P_g$). To quantify this, the following formula was used for this study:

$$P_i = 100^{(g-i)}, \quad i = 1, \ldots, g \tag{2}$$

where g is the number of goals and i represents the current goal. The value of one hundred was chosen to ensure that there was essentially no chance that a lower priority goal could be improved at the expense of a higher priority goal. Thus, the objective function was penalized much more for not satisfying higher priority goals.

The basic structure of the goal programming model used in this research can then be expressed as:

$$\text{Minimize} \quad Z = \sum_{i=1}^{g} P_i \left(\frac{d_i^+ + d_i^-}{T_i} \right) \tag{3}$$

$$\text{Subject to} : (\max / \min G_i(\mathbf{X})) : E(\mathbf{Y}(\mathbf{X}, \omega)) + d_i^- - d_i^+ = T_i \tag{4}$$

$$X_j^l \le X_j \le X_j^u, \quad j = 1, \ldots, n \tag{5}$$

$$d_i^-, d_i^+ \ge 0, \quad i = 1, \ldots, g \tag{6}$$

$$X_j \in \Re, \quad j = 1, \ldots, n \tag{7}$$

where G_i are the goals; j is the number of parameters that are fed into the simulation model; X_j are the input parameters, forming the vector $\mathbf{X}$, that are fed into the simulation model; ω represents the stochastic effects; $E(\mathbf{Y}(\mathbf{X}, \omega))$ is the expected value of the vector of output parameters, $\mathbf{Y}$, determined from the simulation; and X_j^l and X_j^u are the lower and upper bounds, respectively, of the X_j input parameter.

The multi-criteria simulation optimization method developed in this research is used to determine the best combination of input values that provide a near-optimal solution to the multi-objective problem. Specifically, it uses the interaction of an optimization program and a simulation model to solve multi-criteria simulation optimization problems. The multi-criteria component of the procedure is primarily handled in the formulation of the problem. This