

# Model-Based Optimization of a Hydraulic Backhoe using Multi-Attribute Utility Theory

Roxanne A. Conigliaro, Aleksandr A. Kerzhner and Christiaan J. J. Paredis

Georgia Institute of Technology

Copyright © 2009 SAE International

## ABSTRACT

Modeling and simulation are commonly used in all stages of the design process. This is particularly vital to the success of systems engineering projects where the system under consideration is complex and involves interactions between many interdisciplinary subsystems. In the refining stages of the design process (after concept selection), models and simulations can be used to refine and optimize a system with respect to the decision maker's objectives. In this paper, a dynamic model of a hydraulic backhoe serves as a test-bed for a large-scale sensitivity analysis and subsequent optimization of the most significant design parameters. The model is optimized under uncertainty with respect to a multi-attribute utility function that includes fuel consumption, cost of the key components, and machine performance. Since such an optimization can be costly in terms of time and computational resources, the objective of this paper is to provide a useful, cost-effective methodology for this type of problem. To perform the optimization, a parallel computing cluster is used in conjunction with a kriging surrogate model to reduce computation time.

## INTRODUCTION

Modeling and simulation have become increasingly important to the success of many system design endeavors. Although no model can ever perfectly emulate a physical system, models can be useful in design space exploration and subsequent design decisions, even in the presence of uncertainty.

The greatest challenges arise in the context of systems engineering where the interactions between several subsystems yield complex system dynamics [1, 2]. It has been shown that it is preferable to optimize the composite system, and not to decouple the system into individual subsystems and optimize each system sequentially [3]. This is so because one parameter in, for example, a hydraulic subsystem is dependent upon parameters in the mechanical or electrical system with which it is coupled. Modeling such a complex architecture of interacting subsystems often requires a relatively high fidelity model and large systems of linear and nonlinear equations. Even with a fast processor, simulating these models can be computationally expensive. Since the simulation must be run many times per iteration during the optimization and sampling processes, the complexity of the model and the number of design variables are constrained by the cost of computational resources.

In addition to the cost of simulating complex systems, it is often costly to formulate an objective function. In the presence of many competing objectives and system interactions, eliciting an objective function is non-trivial and the process is often overly simplified to conserve resources. If the design objectives are ambiguous or poorly defined, an optimization is likely to result in a solution that does not reflect the decision maker's preferences.

In this paper, we present a methodology for eliciting an objective function and recommendations for reducing the

computation time of the optimization process. In particular, multi-attribute utility theory is used to elicit preferences in the presence of competing design objectives, and the optimization is performed using a parallel computing cluster in conjunction with a kriging surrogate model [4]. The kriging model is an interpolation method which generates a polynomial surface in the design space so that during the optimization process, the simulation does not have to be run at every point in the design space.

This methodology is demonstrated on a hydraulic backhoe model. We first elicit our preferences and formalize them in an appropriate multi-attribute utility function. We then run a sensitivity analysis over a large number of model parameters and then select a few of the most significant parameters as design variables and uncertain variables. Finally, an optimization is performed on the model with the use of uncertain variables.

The remainder of this paper is organized in the following manner. In the next section, the problem background is surveyed, and then the overall approach to the problem is formulated. The approach includes setting up the problem, extracting the relevant modeling concepts prior to the modeling stage, and optimizing design parameters using an energy-based system model. This approach is demonstrated on a hydraulic backhoe model. Creation of the backhoe model is not the focus of this paper, though some of the assumptions and modeling tools for the hydraulic backhoe are delineated during the demonstration. Finally, the results are discussed, along with contributions and future work.

## BACKGROUND

The engineering design process involves the transformation of design requirements and objectives into a solution structure that is iteratively refined [5]. In systems engineering, there are several distinct stages to this process and many iterations. During conceptual design, system architectures are abstracted in terms of subunits and their interactions. The mere act of developing a basic solution structure is non-trivial, as the interactions of the various subsystems are vital to the success of the final design. There are several design strategies that address design concept selection in a systems engineering context, but these are not the focus of this paper. In this paper, we assume that the physical architecture of the solution structure is known, and that we are in the refinement stage of the design process.

Once the overall system architecture is known, modeling and simulation are invaluable aids in making the final parameter selections. Selection of the best parameter values completely depends on the decision maker's preferences. While these preferences may vary based on corporate or consumer objectives and differing use cases, maximizing the overall utility of the final product

requires understanding the tradeoffs that are being made.

This paper is written from a Decision-Based Design (DBD) perspective [6-8], in that we assume decisions are best made using mathematically sound methods derived from decision theory. In particular, we employ Multi-attribute utility theory (MAUT) [9], which is an extension of von Neumann and Morgenstern's utility theory [10]. MAUT allows for several competing design objectives to be taken into account.

The decision problem in utility theory is modeled as

$$a^* = \arg \max_{a \in A} E[u(\mathbf{x}_a)] \quad (1)$$

where  $A$  is the set of feasible decision alternatives,  $a \in A$  is a specific alternative,  $a^*$  is the most preferred alternative,  $\mathbf{x}_a \in \mathbb{R}^n$  is a random vector of attributes for alternative  $a$  having distribution function  $F_a$ ,  $u: \mathbb{R}^n \rightarrow \mathbb{R}$  is a suitably defined utility function, and  $E[\cdot]$  is the expectation operator. In this context, an attribute measures the degree to which an alternative meets a decision objective [9]. Since utility theory may seem complex on the surface, in this paper, we apply it in a simple scenario and attempt to demystify the process of eliciting such a function.

It is important to identify the sources of uncertainty that are relevant to the attributes and objectives under consideration. There are several methods for taking uncertainty into account in the modeling and optimization processes [11-13]. Uncertainty can be epistemic (knowledge-based) or aleatory (random or chance based), and depending on the model's limitations, it may be possible to account for both types of uncertainty during the optimization. With each uncertain variable comes additional complexity and the need for computational resources, but it is possible to pare down the number of uncertain parameters by performing a sensitivity analysis.

## DESIGN OBJECTIVES AND TRADEOFFS

**OBJECTIVES HIERARCHY** - The first step in making any design decision is to identify the objectives. The highest level objectives are called the fundamental objectives [9], and they reflect the overall goal of making this decision. The fundamental objectives can also set the scope of the design problem. For example, fundamental objectives might be to 'Maximize Safety' or 'Maximize This Quarter's Profits'. To move down the hierarchy, we provide more detailed definitions of what is implied at the echelon above. It is important to note that the tree does not provide the means to achieve these objectives. In addition, every objective at every level in the tree is considered fundamental in that no justification is required on the part of the decision maker. A generic objectives tree is shown in Figure 1.

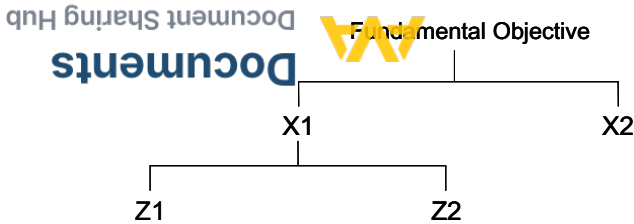


Figure 1: Generic Objectives Hierarchy

The objectives tree can grow quite large, particularly if the fundamental objectives selected reflect lofty goals, many of which exceed the domain of the available models. For example, if fundamental objective is to 'Maximize Profit', but the decision maker is an engineer seeking to improve a manufacturing process for a widget produced by his company, it is logical to prune the tree down to a sub-hierarchy containing the objectives that are within the authority of the decision maker.

**INFLUENCE DIAGRAM** - Once the fundamental objectives have been identified, it is important to reflect on what aspects of the model and use cases have an impact on achieving this objective. An influence diagram [14] (Figure 2), depicts probabilistic dependencies between the design variables and the objective function. It is important to note that the influence diagram is not a flow chart, and thus cannot contain any loops or cycles. The arrows indicate relevance or temporal sequence, thus reflecting the flow of information. Chance events (e.g. weather, component reliability, user skill) also affect many parameters that impact the utility. Some of these should be included in the model or optimization while others can be ignored if they have little impact on the objective. Regardless, the influence diagram helps the decision maker to represent the problem pictorially and to make sure the model he is using encompasses all of the relevant information but omits extraneous details.

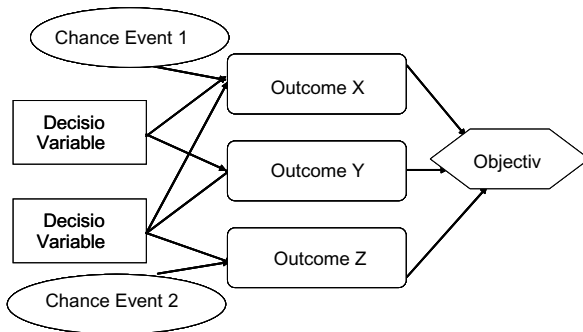


Figure 2: Generic Influence Diagram

**UTILITY FUNCTION ELICITATION** - The elicitation process starts with the fundamental objectives hierarchy and an influence diagram which aid in identifying the relevant attributes. A multi-attribute utility function must contain at least two attributes, otherwise tradeoffs are not encountered. In the influence diagram in Figure 2, the selected attributes for the utility function would be computation outcomes X, Y, and Z, the entities that feed directly into the objective block.

Before using a multiplicative utility function, mutual utility independence must be verified [9]. An attribute is utility independent of another attribute if the risk attitude for one attribute does not change as a function of the other attribute. That is, the certainty equivalent for a gamble on one attribute is the same for all values of the other attribute. It is important to note that utility independence is stronger than preferential independence [15]. For two attributes to be mutually preferentially independent, a preference for a particular outcome of one attribute should not be dependent on the value of the other attribute. For decisions under certainty, mutual preferential independence is a sufficient condition for the use of an additive utility function. However, for decisions under uncertainty, we must use the stronger utility independence criterion.

For mutually utility independent attributes, the multi-attribute utility function takes on a simple, multiplicative form, as shown for three variables in equation (2) [9, 16].

$$u(x, y, z) = k_x u(x) + k_y u(y) + k_z u(z) + k_{xy} u(x)u(y) + k_{xz} u(x)u(z) + k_{yz} u(y)u(z) + k_{xyz} u(x)u(y)u(z) \quad (2)$$

with  $k_x + k_y + k_z + k_{xy} + k_{xz} + k_{yz} + k_{xyz} = 1$

where  $k_x, k_y, k_z > 0$ .

In general, for  $n$  attributes, the multiplicative multi-attribute utility function is the following:

$$u(X) = \frac{1}{K} \left( \left\{ \prod_{i=1}^n [K k_i u_i(x_i) + 1] \right\} - 1 \right) \quad (3)$$

with  $1 + K = \prod_{i=1}^n (1 + K k_i)$

where  $u_i(x_i)$  is the single attribute utility function for attribute  $i$ , and  $u(x)$  is the overall utility of attribute set  $X$  for each alternative.

To begin the weight elicitation process, it is necessary to establish a scale for each attribute. During this step, we assign a utility between 0 (worst case) and 1 (best case) to a set of values for the attribute. It is best to be realistic during this step; if the range between the worst and best cases is chosen too broadly then it is difficult to elicit preferences precisely in the range of values of interest. If, on the other hand, the range is too narrow, one may not find the optimum or one may extrapolate beyond the range of elicitation.

Once a scale for each attribute is established, we elicit points of indifference. We first select a fixed reference point (using utilities and associated values for each attribute under consideration), and then select a point composed of different utilities/values where the decision maker's preference for either design scenario is equivalent. This process must be repeated multiple

times until there are more equations than unknown weight.

The tradeoffs established during the elicitation process are used to setup an over-determined system of equations. To obtain the weights for the multi-attribute utility function, we solve an over-determined system of equations using a least squares algorithm. For example, if we have  $n$  weights and  $m$  equations we arrive at the following set of over-determined equations:

$$[A]_{m \times n} [K]_{n \times 1} = [B]_{m \times 1} \quad (4)$$

By imposing the constraint that the coefficients must add up to one, we can reduce the size of  $K$  by one row and solve for  $K$  numerically using a least squares fit for the over-determined system of equations:

$$[K] = ([A]^T [A])^{-1} ([A]^T [B]) \quad (5)$$

Once the weights have been determined, we substitute them into our utility function. The final step prior to performing the optimization under uncertainty is to do a sensitivity analysis to determine the most significant uncertain parameters.

## SENSITIVITY ANALYSIS – METHOD OF MORRIS

System design optimization problems often involve a large number of design variables and uncertain parameters that affect the measurable attributes used to calculate the utility. Optimizing all of these variables at once is often computationally expensive and inefficient because often a number of these variables do not have a significant impact on the results. We use the Method of Morris screening test [17] to determine which factors have a significant impact on the utility. The optimization is then performed using only these significant variables.

The Method of Morris (MoM) is a one-factor-at-a-time (OAT) method that takes into account the entire range of a variable rather than a local neighborhood around a particular point. It also allows us to identify interaction effects between the variables. The MoM is more applicable to engineering design problems than many other screening methods [18]. It is more attractive than fractional factorial or central composite designs because the number of samples needed is only on the order of the number of input variables [17]. The MoM also requires fewer model evaluations than other global methods such as Sobol sensitivity matrices [19] or the Fourier Amplitude Sensitivity Test (FAST) [20, 21] while predicting with similar accuracy which factors are important and unimportant [22, 23]. However, unlike Sobol matrices or FAST, the MoM does not provide a decomposition of the output variance.

The MoM examines the changes in an output based on experimental plans composed of randomized designs for the input factors. Only one input factor is changed at a

time allowing change in output to be unambiguously attributed to change in that input. This is done by estimating the mean and variance of elementary effects attributed to input factors. One sample of the elementary effect for the  $i^{th}$  input factor is defined as:

$$d_i(x) = \frac{[y(x_1, \dots, x_{i-1}, x_i + \Delta, x_{i+1}, \dots, x_k) - y(x)]}{\Delta} \quad (6)$$

where  $x$  is a  $k$ -dimensional vector of model inputs, and  $\Delta$  is often chosen as:

$$\Delta = \frac{p}{(2(p-1))} \quad (7)$$

where  $p$  is the number of grid levels in the region of experimentation. The finite distribution of elementary effects associated with the  $i^{th}$  input factor, obtained by randomly sampling different  $x$ 's, is denoted by  $F_i$ . We take the mean,  $\mu$ , and the standard deviation,  $\sigma$ , of  $F_i$  to be informative sensitivity measures. Input factors with large  $\mu$  are likely to have an overall important influence on the output, while input factors with large  $\sigma$  may have interaction with other factors or may have non-linear effects.

To create a MoM design, sampling matrices denoted  $B^*$  are constructed using the following:

$$B^* = \left( J_{k+1,1} x^* + \frac{\Delta}{2} [(2B - J_{k+1,k}) D^* + J_{k+1,k}] \right) P^* \quad (8)$$

where  $J_{n,m}$  is a  $n$  by  $m$  matrix of ones,  $x^*$  is a randomly chosen base value of  $x$ ,  $D^*$  is a  $k$ -dimensional diagonal matrix in which each element is either 1 or -1 with equal probability, and  $P^*$  is a  $k$  by  $k$  random permutation matrix. One sampling matrix,  $B^*$ , is needed for every sample of a main effect. If one desires  $n$  main effect samples, then  $n \cdot (k+1)$  function evaluations are needed.

## OPTIMIZATION UNDER UNCERTAINTY

**OPTIMIZATION APPROACH** - When solving a decision problem using utility theory, one must maximize the expected value of the utility (from equation (1)). We use Latin hypercube sampling (LHS) to estimate the expected value of the utility. This process requires numerous evaluations of the deterministic model and is often computationally expensive. The computational expense is compounded by the fact that the optimization algorithm requires several evaluations of the objective function per iteration and many iterations to find the maximum. To reduce the optimization computation time, we use an adaptive kriging interpolation model to predict the results of the underlying deterministic model. If the adaptive kriging model is not sufficiently accurate, the algorithm evaluates the underlying model. A diagram representing this overall approach is shown in Figure 3.