

## MULTI-RESPONSE SIMULATION OPTIMIZATION USING STOCHASTIC GENETIC SEARCH WITHIN A GOAL PROGRAMMING FRAMEWORK

Felipe F. Baesler  
José A. Sepúlveda

Industrial Engineering and Management Systems  
University of Central Florida  
P.O. Box 162450  
Orlando, FL 32816-2450, U.S.A.

### ABSTRACT

This study presents a new approach to solve multi-response simulation optimization problems. This approach integrates a simulation model with a genetic algorithm heuristic and a goal programming model. The genetic algorithm technique offers a very flexible and reliable tool able to search for a solution within a global context. This method was modified to perform the search considering the mean and the variance of the responses. In this way, the search is performed stochastically, and not deterministically like most of the approaches reported in the literature. The goal programming model integrated with the genetic algorithm and the stochastic search present a new approach able to lead a search towards a multi-objective solution.

### 1 INTRODUCTION

Most of the mathematical programming applications found in the literature have been focused on single objective problems. Many simulation models and real world problems involve analysis of more than one objective. The most common approach to deal with this kind of situation is to simplify the problem selecting the most important objective and solving the resulting single objective model. The other objectives are just ignored or simply transformed into model constraints. This approach would generally lead to unrealistic solutions, especially when conflicting objectives are present. For example, consider an investment problem with two objectives, profit maximization and risk minimization. It usually happens that the higher the profit the bigger the risk. For this reason, treating this problem using a single objective will lead to a poor solution. A better approach to this problem would be a multi-objective model where both objectives, profit maximization and risk minimization, are considered as objective functions of the model.

In simulation optimization the situation is not different, generally just one measure of effectiveness is

selected as the response to be optimized. Even though most simulation packages are able to generate information regarded to several measures of effectiveness, such as, resource utilization, time in system, time in queues, etc., just one of them is used as the problem's objective function. A multi-response approach in simulation optimization represents a difficult task due to the stochastic nature of simulation. The optimization problem can not be represented and solved using a deterministic mathematical model.

Most of the applications of simulation optimization have been single objective problems. In the literature there are few attempts to solve multi-response simulation optimization problems. The majority of them are focused on response surface methodology, utility theory and interactive procedures where the decision-maker interacts with the model and leads the search. The major drawbacks of these approaches are local optimality and in most cases the lack of automated direct search. In addition to this, these methods generally analyze the problem in a deterministic way, in other words the randomness associated to simulation is not considered.

This study will focus on these three main issues. The objective is to propose a general approach able to find stochastically a global optimum, at least in theory, for a multi-response simulation optimization problem. This approach should be able to perform as intended in any type of problem regardless of the size and nature of the problem. Also it is expected to find the solution in a direct way, in other words no external intervention will lead the search to the optimal solution. The decision-maker will be included at the beginning of the problem where initial conditions are needed in order to start the optimization process.

### 2 BACKGROUND

The literature reports few attempts for solving multiple objective simulation problems in comparison to single

objective problems. Montgomery and Bettencourt (1977) applied a method based on the Geoffrion-Dyer interactive vector maximal algorithm. This approach make use of the response surface methodology to estimate the objective functions equations, and then solving a set of sub problems, the search direction is determined. Clayton, Weber and Taylor (1982) present a direct search approach for multi-response simulation optimization based on modified pattern search and goal programming with preemptive priorities.

Rees, Clayton and Taylor (1985) proposed a procedure for obtaining satisfactory solutions to multiple response simulation models using modified response surface methodology within a lexicographic goal programming framework. The most preferred goal is optimized first using the response surface approach. Then, an attempt to achieve the next highest ranked goal is made without violating the result obtained for the highest ranked goal. In other words, the achievement of the next goal cannot be made at the expense of the higher goal. The same procedure is repeated for each one of the goals.

Mollaghasemi, Evans and Biles (1991) present an aggregation approach for multi-response simulation optimization. The method uses a multi-attribute value function representing the decision-maker preferences. Then, a gradient search technique is used to find the optimum of the assessed function. Mollaghasemi and Evans (1994) proposed a modification of the multi-criteria mathematical programming technique called STEP method. This technique works in interaction with the decision-maker who is asked, in each iteration, to identified the least satisfactory performance measure, which is then improved at the expense of other responses using a gradient search method.

Teleb and Azadivar (1994) proposed an algorithm based on the constrained scalar simplex search method. The method works by calculating the objective function value in a set of vertices of a complex. The method moves towards the optimum by eliminating the worst solution and replacing it with a new and better solution obtained by connecting the old point to the centroid of the remaining vertices. The process is repeated until a convergence criterion is met.

Mollaghasemi (1994) presents an interactive approach for optimizing multi-response simulation models based on the Geoffrion-Dyer-Feinberg (GDF) vector maximal algorithm. In this approach the decision-maker is asked to determine the tradeoff ratios between a reference criterion and the remaining responses. This information in addition to the gradient estimate of each response is used to formulate a directional sub problem that after solving it will lead to the determination of the optimum direction.

The process is repeated until the decision-maker is satisfied with the solution.

Boyle (1996) presents a method called Pairwise Comparison Stochastic Cutting Plane (PCSCP). This method combines features from interactive multi-objective mathematical programming and response surface methodology. The method works by finding the center of the feasible region in the decision space and performing a design of experiments centered at that point. Interaction with the decision-maker and cutting plane based techniques are used to determine the most preferred experimental point. Finally, formulating a new constraint based on the estimated gradient reduces the feasible region in the decision space. The process is repeated until the best compromise solution is found or terminating criteria are met.

### 3 METHODOLOGY

This section presents in detail a methodology developed to solve multi-response simulation optimization problems. This methodology integrates simulation, goal programming, and genetic algorithms. The simulation model serves as a black box representing the objective functions of the problem, which generates output responses for all the objectives involved in the analysis. These outputs are transformed using a goal programming framework enabling us to consider all the objectives during the optimization process. The genetic algorithm is responsible for performing the search for improved solutions. The selection process of the GA is performed using a multiple comparisons statistical technique. In this way, the stochastic nature of simulation is considered when a selection among different scenarios takes place.

The general structure of the methodology can be divided in nine major sections, each one containing subsections. A description of the methodology is presented next.

#### 3.1 Initialization

The first section or step of this methodology is related to the definition of the initial conditions of the problem. This information can be divided in two groups; information related to the problem it self, and information required by the genetic algorithm. This is explained in table 1.

#### 3.2 Goal Program

In this step, part of the information gathered during the initialization process is used to construct a goal programming model. The steps required to do this are explained next.

Table 1: Initial Conditions

Problem General Information	Description
Variables	Factors of interest for the analysis. These factors are the simulation input parameters.
Objectives	Measures of effectiveness of interest for the problem.
Weights	Relative importance of each objective. AHP suggested for weights estimations
Goals	Aspiration levels for each objective. These values are generally set using experts' opinions. Goal values are realistic but extremely good solutions.
Hard Constraints	Other problem constraints, resources capacities, system limitations, etc.
Genetic Algorithm	Description
Population Size	Number of chromosomes (scenarios) that will be kept from generation to generation.
Crossover Rate	Probability of performing crossover.
Mutation Rate	Probability of performing mutation

*Goal Constraints:* These constraints will contain the simulation output for each objective and the goal associated to it. The structure of the goal constraints is presented in equation 1.

$$E_i(x) + n_i - p_i = b_i, \quad i = 1 \dots k \quad (1)$$

Where  $E_i(x)$  is the expected value of the simulation output representing the measure of effectiveness for objective  $i$ ,  $b_i$  is the predefined goal value for objective  $i$ ,  $n_i$  and  $p_i$  are the negative and positive deviations of objective  $i$  with respect to the goal. One goal constraint is required for each objective considered in the analysis.

*Hard Constraints:* These constraints represent the system's limitations, e.g. resources capacity and availability, demand, etc.

*Objective Function:* The objective function for the goal programming model is presented as follows:

$$Min \quad z = \sum_{i=1}^k (w_i n_i + w_i p_i) \quad (2)$$

Where  $w_i$  is the weight that represents the DM preferences for objective  $i$ , and  $n_i$  and  $p_i$  are the negative and positive deviations of objective  $i$  with respect to the goal.

*Normalization:* Since different objectives are represented using different measure units, e.g. one objective could be presented in terms of millions dollars and other in terms of percentage (resource utilization), it is necessary to transform each unit to a common one. This could be achieved dividing the expression in brackets in equation 2 by its corresponding goal value. In this way, the new objective function is shown next:

$$Min \quad z = \sum_{i=1}^k \left( \frac{w_i n_i + w_i p_i}{b_i} \right) \quad (3)$$

It is important to mention that when an objective is to be minimized, then the  $n_i$  term should be eliminated from the objective function. When the objective is to be maximized,  $p_i$  has to be removed. In this way we are not penalizing the over achievement of an objective. If the objective is to get as close as possible to the goal, then both terms  $n_i$  and  $p_i$  should be included in the objective function.

### 3.3 Objective Function Variance Estimation

The objective function presented above represents the deterministic part of the problem. Since simulation is a stochastic technique, it is necessary to capture the randomness associated to it and include it in the objective function. A simulation model is able to generate the mean and the variance for each one of the objectives associated with the problem. The objective function's mean can be calculated using equation 3 and the variance has to be estimated considering all the variances associated to each response. If the objectives considered in the study were all independent of each other, the resulting variance would be the summation of all the independent variances. Unfortunately, this is not the most common case. A multi-objective analysis generally considers conflicting objectives, this means there exists correlation between them, and no independence could be assumed. For this reason, mean and variance estimation for the summation of non-independent random variables can be performed using the following theorem:

If  $y_1, y_2, \dots, y_n$  are random variables and  $l = a_1 y_1 + a_2 y_2 + \dots + a_n y_n$ , where  $a_i$  are constant values, then

$$E(l) = a_1 \mu_1 + a_2 \mu_2 \dots + a_n \mu_n \quad (4)$$

and

$$\begin{aligned} \sigma_f^2 = & a_1^2 \sigma_1^2 + a_2^2 \sigma_2^2 + \dots + a_n^2 \sigma_n^2 + 2a_1 a_2 \text{cov}(y_1, y_2) + \\ & 2a_1 a_3 \text{cov}(y_1, y_3) + \dots + 2a_1 a_n \text{cov}(y_1, y_n) + \\ & 2a_2 a_3 \text{cov}(y_2, y_3) + \dots + 2a_2 a_n \text{cov}(y_2, y_n) + \\ & \dots + 2a_{n-1} a_n \text{cov}(y_{n-1}, y_n) \end{aligned} \quad (5)$$

This theorem can be used to estimate the objective function's mean and variance as follows:

$$\mu_f = \left(\frac{w_1}{b_1}\right) d_1 + \left(\frac{w_2}{b_2}\right) d_2 + \dots + \left(\frac{w_n}{b_n}\right) d_n \quad (6)$$

For notation simplification and because we are not penalizing the over achievement of any objective, the  $d_i$  was used to represent  $n_i$  (maximization) or  $p_i$  (minimization) depending on each objective.

$$\begin{aligned} \sigma_f^2 = & \left(\frac{w_1}{b_1}\right)^2 \sigma_1^2 + \left(\frac{w_2}{b_2}\right)^2 \sigma_2^2 + \dots + \left(\frac{w_n}{b_n}\right)^2 \sigma_n^2 + \\ & 2\left(\frac{w_1}{b_1}\right)\left(\frac{w_2}{b_2}\right) \text{cov}(y_1, y_2) + 2\left(\frac{w_1}{b_1}\right)\left(\frac{w_3}{b_3}\right) \text{cov}(y_1, y_3) + \\ & \dots + 2\left(\frac{w_1}{b_1}\right)\left(\frac{w_n}{b_n}\right) \text{cov}(y_1, y_n) + 2\left(\frac{w_2}{b_2}\right)\left(\frac{w_3}{b_3}\right) \text{cov}(y_2, y_3) + \\ & \dots + 2\left(\frac{w_2}{b_2}\right)\left(\frac{w_n}{b_n}\right) \text{cov}(y_2, y_n) + \dots \\ & + 2\left(\frac{w_{n-1}}{b_{n-1}}\right)\left(\frac{w_n}{b_n}\right) \text{cov}(y_{n-1}, y_n) \end{aligned} \quad (7)$$

Since the covariance is not known, it is necessary to generate a set of randomly selected points of the response surface. This is done by running the simulation model for all the selected scenarios and collecting the mean of each one of the measures of effectiveness under study. These values can be used to calculate a covariance matrix. All the covariance values that appear to be significant will be included in the formula.

Each time a scenario is simulated, the mean and variance of all its measures of effectiveness have to be collected and used to evaluate equation 6 and 7. These two equations will represent the multi-objective aggregated MOE that will be used to search for better scenarios. The first step in the optimization process is presented in the next section.

### 3.4 Optimization Process

The first stage to start the optimization process is to generate an initial population of solutions (scenarios). The number of point estimations required to start is equal to the population size parameter defined in the initialization step of the methodology. The selection of these scenarios is carried out randomly. The reason for this is to cover a wide range of the surface response. Generally, DMs have a good idea where good solutions could be found. Some of the DMs suggestions could be considered in the initial population.

After running all the scenarios, the mean and variance for each MOE considered in the analysis have to be collected and saved in a database. Then, calculate  $d_i$  using equation 1. Remember that  $d_i$  represents either  $n_i$  or  $p_i$  depending on that particular objective (maximization, minimization). Finally using equations 6 and 7, calculate mean and variance of the GP function. Even though the covariance is a random variable, and its value may change depending on the sample used for its estimation, we will assume that it is constant from iteration to iteration, it does not have to be estimated again.

### 3.5 Selection

The next step in the methodology and following the genetic algorithm paradigm is to select the chromosomes (scenarios) that will be considered for crossover. The selection will be carried out in terms of each chromosome's fitness, which it is represented by the GP function mean and variance.

One of most common selection method is called roulette wheel selection. The idea is to determine selection probabilities for each chromosome proportional to the fitness value. If  $f_i$  is the fitness value for chromosome  $i$ , then its probability of selection  $p_i$ , is equal to:

$$p_i = \frac{f_i}{\sum_{i=1}^{pop. size} f_i} \quad (8)$$

Then, a wheel according to these probabilities can be created and each time the wheel is spun a chromosome is selected. We need to select half of the population using this method, the other half will be generated by mating the selected chromosomes. Because our problem is not deterministic, it is necessary to apply some modifications to the selection procedure. Because  $f_i$  is a random variable represented by a mean and a variance, statistical techniques have to be applied in order to differentiate among the chromosomes. In other words, it is not possible to say that a chromosome is better than another because its  $f_i$  mean is better. The randomness associated to these values has to be considered. There are many statistical techniques that can

be applied to solve this problem, they are called multiple comparison methods. These methods are able to form sets of treatments considering the randomness associated to them. These methods work by forming groups of treatments. It is considered that there is statistical difference among the groups but not within them. In this way, we can form groups instead of individual chromosomes that can be ranked. Using this approach we can compute the fitness for each group and use this value to calculate the probability of selection of that group proportional to the population of groups. Every time the wheel is spun and a group selected, a chromosome from that group is randomly chosen and passed to the next generation. Boesel et al. (1999) applied this approach to a single objective simulation optimization problem. This method make use of the single response variance estimation generated by the simulation model to create groups using a statistical grouping procedure based on Calinski and Corsten (1985). The selection procedure proposed in our study is based on Tukey’s multiple comparison procedure. In addition to this, since this is a multi-objective problem, our method makes use of the variance of each one of the responses, as well as the covariance of all the pairs of objectives.

Tukey’s method selects a critical distance,  $w$ , so that the probability of making a type I error is  $\alpha$ . Any pair of sample means (GP means) that differ by less than  $w$  can be considered equal. In this way, groups of chromosomes can be created considering that their fitness values do not differ by more than  $w$ . The estimation of  $w$  is shown in equation 9.

$$w = q_{\alpha}(p, v) \frac{s}{\sqrt{n}} \tag{9}$$

Where:

- $P$  = Number of sample means (population size)
- $s$  = Square root of MSE. For the case where the number of observations is equal for all the samples,  $s$ = Square root of the average GP variance.
- $v$  = Number of degrees of freedom associated to MSE.
- $n$  = Number of observations in each of the  $p$  samples.
- $q_{\alpha}(p, v)$  = Critical value of the Studentized range.

After calculating  $w$ , all the chromosomes are ranked in terms of its fitness value and grouped considering chromosomes that differ by less than  $w$ . In this way, we have created groups of chromosomes that are not statistically different.

After the groups have been created it is necessary to assign the probability of selection for each group. This is done by adding up all the fitness values  $f_i$  within each

group calculated in equation 8. Equation 10 shows the formula for assigning the probability of selection for each one of the groups.

$$p_k = \frac{\sum f_i}{\sum f_i} \text{ (for every } i \text{ in group } k) \tag{10}$$

Having the probability of selection for each group ( $p_k$ ), the roulette wheel can be constructed. The groups containing better chromosomes would have a higher chance to be selected. In each spin, a group is selected and a chromosome within the group is chosen randomly. The chosen chromosome will be passed to the next generation and utilized as a parent in the mating process. The half of the new generation will be created using this method, and the other half will be the offspring generated from the mating process.

### 3.6 Crossover (Mating Process)

The half of the new generation has been created using the selection process described above. Now, in order to fill the remaining half it is necessary to mate pairs of chromosomes to create offspring. This process is known as crossover. This is performed selecting a pair of chromosomes from the new generation randomly. Then, the crossover point has to be chosen. This point represents the position where the chromosome is cut to exchange genetic information within the pair of parents. There are many different types of crossover, the most common one is called single point crossover. Other types are two points, multiple points and uniform crossover. All these types of crossover work very good when the chromosome is represented using binary code. Since we will use real number codification, a different kind of crossover will be used. The method that will be used in this study is a variation of a group of techniques called blending crossover. Some examples of blending crossover can be found in Haupt and Haupt (1998). The method starts selecting a pair of parent chromosomes. Then, one gene is randomly selected as the crossover gene. The new chromosomes will be created using a combination of the two parents. This is shown in next.

Crossover  
Gene

Chrom. A	$[a_1, a_2, a_3, a_4, \dots, a_n]$
Chrom. B	$[b_1, b_2, b_3, b_4, \dots, b_n]$
Offspring C	$[a_1, a_2, X, b_4, \dots, b_n]$
Offspring D	$[b_1, b_2, Y, a_4, \dots, a_n]$

$$X = a_3 - \beta(a_3 - b_3)$$

$$Y = b_3 + \beta(a_3 - b_3)$$

$\beta$ : Random number between 0 and 2

If the last gene is selected, then all the genes to the left are swapped. This method of crossover will allow adding new information to the new chromosomes. After the crossover process has been completed for all pairs of parent chromosomes, the mutation operator has to be applied. This process is explained in the next section.

### 3.7 Mutation

Mutation is the second way a genetic algorithm explores a surface. The basic idea about mutation is to keep the GA from fast convergence to local optima. The simplest type of mutation is called uniform mutation. In this type of mutation a gene is replaced with a randomly selected number within the boundaries of the parameter. The probability of mutation is specified in the initialization step of the methodology and represents the chance that a gene would mutate in each generation. If a mutation rate is 5%, the population size is 10, and the number of genes (parameters) in each chromosome is 4, then,  $(10 \times 4 \times 0.05) = 2$  genes should mutate in each generation. There is a concept called *elitism* that leaves the best chromosomes out of the mutation candidates. The reason for this is to avoid the change of a very good solution. A percentage of the population could be selected as elite solutions.

### 3.8 Evaluation

The new generation has been created using the crossover and mutation genetic operators. The problem is that for some of these chromosomes its fitness value is unknown. In order to estimate the fitness of a chromosome it is necessary to run the simulation model for each one of the scenarios represented by the chromosomes. Before running the simulation the database containing all the simulation results should be checked to see if that particular scenario was run already. This is important since simulation time is costly. If that parameter configuration was contained in the database, then its output can be used for estimating the chromosome fitness. If it is not found then the simulation model has to be used to estimate the output. These values should be stored in the database and the fitness value computed using equations 6 and 7.

### 3.9 Stopping Criteria

The last step in the methodology is to check if the GA found a solution that is good enough to meet the DM expectations. The first possibility is that the GA has reached a point of total convergence. This happens when all the chromosomes in a generation are exactly the same. In this case the process should be stopped since no further improvement could be reached. This is not the most common case in real world problems, so other stopping criteria are necessary. The simplest one is related to the amount of time or number of

simulated scenarios that the DM is willing to spend in the analysis. If this is the case the process should be stopped when this constraint has been met and the best solution reached so far can be selected as the final solution. Other types of stopping criteria are related to the improvement observed among generations. For example, if no significant improvement is observed in a number of consecutive generations the GA should be stopped.

## 4 FUTURE RESEARCH

Since this is an ongoing research, there are several aspects that still have to be investigated. The natural next step would be to apply the methodology to test bed cases. This would offer important results that would help for testing the performance of the methodology. These results should be compared to the results obtained using other techniques. This will offer an empirical validation and justification for the application of this technique. Some possible alternatives to compare the methodology with could be some of the methods presented in the literature review. The opinion of experts in the field of the particular test bed utilized to test the methodology could offer important information for the analysis of its performance. The utilization of meta-models such as response surface methodology would offer important insights to analyze the problem's surface and evaluate the results obtained using the proposed approach.

Another important aspect to consider is related to the genetic algorithm parameters selected for implementation in the methodology. The selection of the population size, crossover rate, mutation rate, and other aspects as type of crossover and mutation could be analyzed in order to find the best GA parameter configuration for this particular methodology.

Finally, interesting analysis could be done in the field of statistical multiple comparison techniques. There is an important number of methods that could be used in the selection step of this methodology.

## 5 CONCLUSIONS

This paper has presented a new approach to solve multi-response simulation optimization problems. This approach integrates a simulation model with a stochastic genetic algorithm heuristic and a goal programming model.

A literature review was presented including the most relevant research performed in the field of multi-response simulation optimization.

Finally, a future research section summarizes some of the aspects that have to be considered to continue with this research. These include the application of the methodology to some test bed cases, the selection of the most appropriate GA conditions, and further study in the multiple comparison techniques area.

## REFERENCES

- Boesel, J., B. Nelson, and N. Ishii. 1999. A framework for simulation optimization software. Technical Report, Dept. of industrial Engineering and Management Sciences, Northwestern University.
- Boyle, C.R. 1996. An Interactive Multiple Response Simulation Optimization Method. *IIE Transactions*. 28: 453-463.
- Calinski, T., and C.A. Corsten. 1985. Clustering means in ANOVA by simultaneous testing. *Biometrics*. 41: 39-41.
- Clayton, E.R., W.E. Weber, and B.W. Taylor. 1982. A goal programming approach to the optimization of multiresponse simulation models. *IIE Transactions*. 14: 282-287.
- Haupt, R.L., and S.E. Haupt. 1998. *Practical Genetic Algorithms*. Wiley.
- Mollaghasemi, M., G.W. Evans, and W.E. Biles. 1991. An approach for optimizing multiple response simulation models. In *Proceedings of the 13<sup>th</sup> Annual Conference on Computers in Industrial Engineering*, 201-203.
- Mollaghasemi, M. 1994. Multiple response optimization of simulation models. *Transactions of the Society for Computer Simulation*. 11: 179-192.
- Mollaghasemi, M., and G.W. Evans. 1994. Multicriteria design of manufacturing systems through simulation optimization. *Transactions on Systems, Man, and Cybernetics*. 24: 1407-1411.
- Montgomery, D., and V. Bettencourt. 1977. Multiple response surface methods in computer simulation. *Simulation*. 29: 113-121.
- Rees, L.P., E.R. Clayton, and B.W. Taylor. 1985. Solving multiple response simulation optimization models using modified response surface methodology within a lexicographic goal programming framework. *IIE Transactions*. 17: 447-457.
- Teleb, R., and F. Azadivar. 1994. A methodology for solving multi-objective simulation-optimization problems. *European Journal of Operational Research*. 72: 135-145.

**JOSÉ A. SEPÚLVEDA** is Associate Professor in the Department of Industrial Engineering and Management Systems Department at the University of Central Florida. He received his Ingeniero Civil Químico degree from the Universidad Santa Maria, Valparaíso, Chile, in 1970. From the University of Pittsburgh he received a M. S. degree in Industrial Engineering in 1972, a Master in Public Health degree in 1974, and a Ph.D. degree in Industrial Engineering in 1981. Dr. Sepúlveda is a registered Professional Engineer in Florida (1983) and Chile (1975). He is a member of the Institute of Industrial Engineers, the Society for Computer Simulation, the Institute of Management Sciences, the American Public Health Association, and the Hospital Information Management Systems Society. E-mail address, <sepulved@mail.ucf.edu>.

## AUTHOR BIOGRAPHIES

**FELIPE F. BAESLER** is a doctoral candidate in the Department of Industrial Engineering and Management Systems, University of Central Florida, Orlando, Florida. He received his Ingeniero Civil Industrial degree from the Universidad del Bío-Bío, Concepción, Chile and a MS in IE from the University of Central Florida. During the last three years Mr. Baesler has worked as a research assistant at UCF. He also serves on the Industrial Engineering faculty of the Universidad del Bío-Bío, Chile.