Tepper School of Business

11-22-2008

Optimization for Simulation: LAD Accelerator

Miguel A. Lejeune
George Washington University

François Margot
Carnegie Mellon University, fmargot@andrew.cmu.edu

Optimization for Simulation: LAD Accelerator

Miguel A. Lejeune[1], François Margot[2] *
[1] George Washington University, Washington, DC, `mlejeune@gwu.edu`
[2] Carnegie Mellon University, Pittsburgh, PA, `fmargot@andrew.cmu.edu`

November 22, 2008

Abstract

The goal of this paper is to address the problem of evaluating the performance of a system running under unknown values for its stochastic parameters. A new approach called LAD for Simulation, based on simulation and classification software, is presented. It uses a number of simulations with very few replications and records the mean value of directly measurable quantities called observables. These observables are used as input to a classification model that produces a prediction for the performance of the system. Application to an assemble-to-order system from the literature is described and detailed results illustrate the strength of the method.

Keywords: Simulation-Optimization, Logical Analysis of Data, Stochastic Models

1 Introduction

This paper focuses on the optimal design of production, distribution, or queuing systems subject to stochastic events. These models are configurable by specifying values for stochastic parameters and values for decision variables and their performance can be described by a single number. More precisely, we define a *parameter* to be a stochastic parameter of the model; we assume that the value of any parameter can not be controlled. An example is the arrival rate of clients in a queuing system. We define a *manipulable* to be a decision variable that can be set within given bounds. An example is the buffer length of a queue in the system. The performance of a system with specified values $\bar{p} = (p_1, \ldots, p_q)$ for its q parameters and $\bar{m} = (m_1, \ldots, m_r)$ for its r manipulables may be any function $\mathcal{P}(\bar{p}, \bar{m})$ or any other value that can be estimated at the end of a simulation. The names *Optimization for Simulation* [20] and *Simulation-Optimization* [7, 10, 22, 23] have been coined for the problem of finding values of the manipulables that optimize the performance of a system with given parameter values. We refer the reader to the special issue of the *INFORMS Journal of Computing*, Summer 2002, Volume 14 (3) for a thorough overview of the discipline.

Several approaches using discrete event simulation software have been introduced to find good values for the manipulables of a system with given parameter values [20, 36, 38]. A common feature of these approaches is a local search in the space of the feasible manipulable values, running simulations to evaluate the performance. Following the terminology of [10, 20], a *simulation* is a collection of runs, called *replications*, of the software for fixed values of the parameters. The sample mean performance

---

over replications is used as the expected performance of the system under the given settings of the parameters. While these approaches have relevance for solving real world problems, they do not address directly the main problem that a manager of a production system faces, namely how to assess and improve the performance of the system under current conditions. In many situations, the stochastic parameters are not known and can vary with time (hourly, daily, or seasonally depending on the application). This makes it difficult to set up a realistic simulation of the system.

The goal of this paper is to address, at least partially, the problem of evaluating the performance of a system, under the assumption that the values of the stochastic parameters may vary within some given bounds. We build a classification model to estimate how well the system is run under current conditions. This model takes as inputs a collection of values (called *observables*) that can be measured from the running production system and it outputs an indicator value for the performance. Examples of observables for a production line are observed minimum, observed maximum and observed average of service times, inventory levels, profit, production level, or production quality. We stress that stochastic parameter values are *not* part of the observables, but manipulable values can be. We validate the approach on an "assemble-to-order" system from the literature [36] and show that the classification obtained by the model matches closely the actual performance of the system. The classification model used in the application is based on the *logical analysis of data (LAD)* proposed by Hammer [26] and developed by Boros et al. [11] and the approach is called *LAD for Simulation*.

One of the crucial challenges in Simulation-Optimization resides in the allocation of the computational resources between the search for a better solution and the evaluation of the current candidate solution [22]. When a classification model with high accuracy can be built for a particular application, it can be used not only for evaluating performances of the current system, but also to enhance the local search heuristics for Simulation-Optimization mentioned above: it is possible to reduce the search space for the heuristic by rejecting quickly all settings not classified as "good". When the time for computing the classification of a setting is a fraction of the time to get the estimated performance of the setting, precious computing time can be saved during optimization. Our proposal is to execute a number of simulations with very few replications and record the mean value of the observables. These values are then used by the classification model and its output is used as the performance of the system under this particular setting of the manipulables.

Note that the design of the classification model for the problem at hand might be time consuming. However, in a situation where the same stochastic model has to be optimized periodically, investing time to build the classification model beforehand might be worthwhile. It also allows for optimization "on-the-fly" that is order of magnitude faster than with simulations with large number of replications. The approach developed here is thus also interesting when short reoptimization time is critical.

The paper is organized as follows. In Section 2, the proposed approach is described in more details and a review of related approaches is given. Section 3 presents briefly the LAD methodology. It will be clear that several LAD classification models can be constructed and the evaluation of their respective performances is necessary. Several performance measures are described in Section 4. In Section 5, an "assemble-to-order" application is presented. Several alternatives to construct the classification model are explored and a model with high classification performance is constructed. Finally, Section 6 provides concluding remarks.

2 LAD for Simulation and Related Approaches

Most approaches for Simulation-Optimization in the literature try to find good manipulable values for given values of the stochastic parameters. In this paper, we are interested in the problem of estimating the

performance of a system running under unknown values for the stochastic parameters. We want to build a classification model using directly measurable quantities (called observables) as inputs and whose output is an estimation of the system's performance. Assuming that such a reliable classification model can be built, it can be used to speed up heuristics for Simulation-Optimization by using the model evaluation as a guide. The idea is to perform a simulation with few replications, record the values of the observables, and feed them to the classification system to get the estimation of the performance.

We first review briefly some of the classical approaches for Optimization-Simulation. A typical approach for finding good values of the manipulables is sampling: In its simplest form (known as a *factorial design experiment* or *factorial sampling*), a discretization of the domain of each manipulable is chosen and all possible choices of values from their discretized domains are evaluated. This is the methodology used in [25] for optimizing manipulables for a trust-region solver. This method is of course limited to cases where the number of manipulables is small and where the number of discretized values for each manipulable domain is small. When factorial sampling is impractical, it is possible to sample only a subset of all possible choices of manipulable values from their discretized domains while covering well, in a statistical sense, the discretized manipulable space. The choice of points to evaluate is based on *orthogonal array designs*, a well-known tool in statistics [16, 32, 33] or on the construction of optimal design of experiments [45]. The number of evaluated points drops considerably compared to the number used in a factorial sampling, but the approach is limited by the fact that introducing many discretization points in the domain of the manipulables is still not practical. To overcome this difficulty, it is possible to iteratively refine the discretization around points of interests. Examples are discussed in [9, 41]. A more sophisticated approach coupling experimental designs with local search is developed by Adenso-Diaz and Laguna [1].

Instead of applying a heuristic optimization technique over the entire search space, we propose to derive an LAD classification model that discriminates the good settings from the average and bad ones. The construction of the LAD classification model is as follows. Given the stochastic model under consideration and domains for its stochastic parameters, a collection of simulations are performed with various values $p^j = (p_1^j, \ldots, p_q^j)$ for its q parameters and values $m^j = (m_1^j, \ldots, m_r^j)$ for its r manipulables for $j = 1, \ldots, N$. These simulations are done with sufficiently many replications so that the estimation $\hat{\mathcal{P}}(p^j, m^j)$ of the performance $\mathcal{P}(p^j, m^j)$ can be trusted. (This might require a large number of replications [14] as the rate of convergence of the mean of a sample of n replications is typically $O(\sqrt{n})$.) That estimation and the values of the s observables $o^j = (o_1^j, \ldots, o_s^j)$ are recorded. They form a collection of N data points $(\hat{\mathcal{P}}(p^j, m^j), o^j)$ called *experiment set*. The LAD classification model is then designed so that its evaluation function $f(o^j)$ behaves similarly to $\hat{\mathcal{P}}(p^j, m^j)$ for $j = 1, \ldots, N$. While the construction of the model itself is intricate, software implementing this step is available. Section 3 gives more information about the methodology and software.

Once the LAD classification model is built, it can be used jointly with most heuristics: For a given setting of the values of the manipulables, we execute a number of simulations with very few replications and record the mean value of the observables. The LAD classification model then is used as a predictor for the performance of the system under this particular setting of the manipulables. If the LAD classification model is accurate enough to discriminate good and bad settings, precious time can be saved by discarding the latter and focusing more on the former. Note also that the correct answer is obtained even if the prediction of the LAD classification model is not numerically accurate: it is enough for the LAD classification system to rank the choices of manipulable settings in the same order as given by their true performance.

This method is related to the concept of ordinal optimization first proposed by [35] (see also [17, 34]). Ordinal optimization does not aim at identifying the best setting, but concentrates on finding settings that can be shown to be good with a high probability, and reduces the required simulation time dramatically

[34]. It was shown [14, 17] that less simulation effort is needed to develop a good ordering of solutions than it is required to estimate their actual performance [10]. For a large class of systems, Dai [17] showed that the probability of correctly selecting the best system's setting using ordinal optimization converges at an exponential rate.

Building upon this, *selection*, *screening* or *indifference-zone ranking* techniques have been proposed (see [13] for a review). In [15], the best simulated system is identified through a Bayesian procedure. The "optimal computing budget allocation" method [14] defines, for a limited computational budget, the number of replications to be allocated to each simulation in order to maximize the probability of ranking the settings and selecting the good ones correctly (see also [24]). Two-step methods, applicable to simulation problems in which the number of settings is large but finite and when a number of replications have been conducted for each setting, are proposed in [10] and [43]. They first isolate subsets of good system settings from the very bad ones, before attempting to determine the best setting using a ranking procedure. While this is in the same spirit as the approach presented in this paper, a major difference is that the latter does not require to run a sample of replications for each possible setting, as the LAD classification model is (hopefully) able to extrapolate to the full search space the information given by performance values for a collection of settings. In addition, our approach is not restricted to discrete-event simulation models and it can handle directly manipulables with non discrete domain.

3   Logical Analysis of Data

The logical analysis of data (LAD) is a combinatorial logic-based optimization methodology that was initially created [26] and used for the analysis and classification of binary data [3]. Its application scope was later extended [11] to data sets containing numerical variables. This section presents an overview of the method as applied to the case studied in this paper, namely the construction of a system hopefully replicating the behavior of the function with value $\mathcal{P}(p^j, m^j)$ when evaluated at the point o^j for $j = 1, \ldots, N$. LAD models can be constructed using the `Datascope` software [2] which is available for academic research. We refer the reader to [4, 5] for a description of the most distinguishing features of the implementation of the LAD method in `Datascope`.

LAD is usually used as a classification mechanism: Given a collection of points labeled *positive* or *negative*, the goal is to derive a small set of patterns that returns, for each point in the collection, a classification that matches its label as accurately as possible. We describe in section 3.1 the LAD methodology for this case. Extension to the case where the label of an experiment is a continuous or categorical indicator instead of a binary one was studied by Hammer et al. [29, 30] and Kogan and Lejeune [39] for reverse-engineering the Standard&Poor's evaluation of entities' creditworthiness, which takes the form of an alphanumerical (i.e., AAA, AA+, etc.) credit risk rating system.

3.1   LAD: Positive/Negative Case

Assume that the experiment set is a collection of N points (z^j, o^j) where z^j has value 1 for a positive experiment and 0 for a negative one and o^j is an s-vector recording the values of the observables.

Table 1 illustrates an experiment set containing 5 experiments recording the values of three observables. Each component o_i^j of the $[3 \times 3]$-matrix in Table 1 gives the value taken by observable i in experiment j. The column labeled z returns the outcome (positive if $z^j = 1$, negative if $z^j = 0$) of the experiment.

The LAD method generates and analyzes exhaustively a major subset of combinations of possible values for the observables which can describe the positive or negative nature of an experiment. It uses optimization techniques to extract models taking the form of a limited number of significant *logical*

Table 1: Experimental Set

Experiment	Observables			Outcome
j	o_1	o_2	o_3	z
1	3.5	3.8	2.8	1
2	2.6	3.8	5.0	1
3	1.0	1.6	3.7	1
4	3.5	2.2	3.9	0
5	2.3	1.4	1.0	0

patterns. LAD has been very successfully applied to multiple medical classification problems (see [28] for a review) and credit risk rating and data mining problems [29, 30, 31, 39].

The purpose of LAD is to discover a binary-valued function f which is constructed as a weighted sum of logical functions of binary variables related to the observables, and provides an accurate discrimination between positive and negative experiments. To construct the function f, the experiment set is first transformed into a *binarized data set* in which the components can only take the values 0 and 1. Each original numerical observable is replaced by several binary ones. This is achieved by defining, for each observable o_i, a set of $K(i)$ values $\{c_{i,k} \mid k = 1, \dots, K(i)\}$ called *cut points* and associated binary variables $\{y_{i,k} \mid k = 1, \dots, K(i)\}$. The value of these binary variables for data point (z^j, o^j) is then defined as:

$$y_{i,k}^j = \begin{cases} 1 & \text{if } o_i^j \geq c_{i,k} \\ 0 & \text{otherwise.} \end{cases}$$

The choice of the values of the various cut points is based on a statistical analysis of the experiment set.

Table 2 provides the binarized experimental data set corresponding to the data above, and reports the values $c_{i,k}$ of the cut points k for each observable o_i, and those of the binary variables $y_{i,k}^j$ associated with any cut point k of observable i in experiment j. For example, $y_{1,1}^1 = 1$ since $o_1^1 = 3.5$ is larger than $c_{1,1} = 3.0$.

Table 2: Binarized Data Set

Observables		o_1			o_2		o_3			
Cut Points		$c_{1,1}$	$c_{1,2}$	$c_{1,3}$	$c_{2,1}$	$c_{2,2}$	$c_{3,1}$	$c_{3,2}$	$c_{3,3}$	
		3.0	2.4	1.5	3.0	2.0	4.0	3.0	2.0	
	j	$y_{1,1}$	$y_{1,2}$	$y_{1,3}$	$y_{2,1}$	$y_{2,2}$	$y_{3,1}$	$y_{3,2}$	$y_{3,3}$	z
	1	1	1	1	1	1	0	0	1	1
Binary	2	0	1	1	0	0	1	1	1	1
Variables	3	0	0	0	0	1	0	1	1	1
	4	1	1	1	0	0	0	1	1	0
	5	0	0	1	0	1	0	0	0	0

A *discretized data set* can then be created [27]; it has the same dimension as the original experiment set and whose components x_i^j, thereafter called *variables*, are mapped to the observables. The discretization of the binarized data set is carried out as follows:

$$x_i^j = \sum_{k=1}^{K(i)} y_{i,k}^j, \quad \text{for } i = 1, \dots, s, j = 1, \dots, N.$$

5

Table 3 below displays the discretized data set.

Table 3: Discretized Data Set

Experiment	Variables			Outcome
j	x_1	x_2	x_3	z
1	3	2	1	1
2	2	0	3	1
3	0	1	2	1
4	3	0	2	0
5	1	1	0	0

Positive (resp., negative) *patterns* impose upper and lower bounds on the values of a subset of the variables, such that a high proportion of the positive (resp., negative) experiments in the discretized data set satisfy the conditions imposed by the pattern, and a high proportion of the negative (resp., positive) experiments violate at least one of the conditions of the pattern. An experiment in the data set is *covered* by a pattern if it satisfies the conditions of the pattern. The selection of the patterns is achieved by solving a set covering problem (see [11] for details). The *degree* of a pattern is the number of variables whose values are bounded by the pattern. The *prevalence* of a positive (resp., negative) pattern is the proportion of positive (resp., negative) experiments covered by it. The *homogeneity* of a positive (resp., negative) pattern is the proportion of positive (resp., negative) experiments among those covered by it.

Considering the data above,

$$x_1 \leq 2 \quad \text{and} \quad x_2 \leq 2$$

is a positive pattern of degree 2 covering two positive observations and one negative observation. Therefore, its prevalence and homogeneity are both equal to 2/3.

The first step in applying LAD to a data set is to generate all patterns for the data set, a collection called *pandect* in the LAD terminology. The number of patterns contained in the pandect can be extremely large. The substantial redundancy among the patterns of the pandect makes necessary the extraction of (relatively small) subsets of patterns, sufficient for differentiating positive and negative experiments in the data set. Such collections of positive and negative patterns are called LAD *models*. Limitations on the size of the model, i.e. the number of included patterns, are usually imposed by restricting the degree, the prevalence, and the homogeneity of the considered patterns. Models incorporating patterns of low degree, high prevalence and high homogeneity have been shown to be the most effective in LAD applications [12]. A model is supposed to include sufficiently many positive (resp., negative) patterns to guarantee that each of the positive (resp., negative) experiments in the data set is covered by at least one of the positive (resp., negative) patterns in the model. Furthermore, good models tend to minimize the number of experiments in the discretized data set covered simultaneously by both positive and negative patterns in the model.

An LAD model can be used for classification in the following way. An experiment which satisfies the conditions of some of the positive (resp., negative) patterns in the model, but which does not satisfy the conditions of any of the negative (resp., positive) patterns in the model, is classified as positive (resp., negative). An experiment satisfying both positive and negative patterns in the model is classified with the help of a *discriminant* that assigns specific weights to the patterns in the model [12]. More precisely, let n_p and n_n represent the number of positive and negative patterns in the model, and let $c_p(j)$ (resp., $c_n(j)$) represent the numbers of positive (resp., negative) patterns which cover a new experiment j. The value of the discriminant is defined as

$$\Delta(j) = \frac{c_p(j)}{n_p} - \frac{c_n(j)}{n_n}$$

and the classification by the model is determined by the sign of $\Delta(j)$. An experiment for which $\Delta(j) = 0$ is left unclassified. Note that in the above formula, the discriminant of an experiment j is computed by giving the same weight $w_r^+, r = 1, \ldots, n_p$ (resp., $w_s^-, s = 1, \ldots, n_n$) to all positive (resp., negative) patterns with $w_r^+ = w_r^-$ if and only if the number of positive patterns is equal to that of the negative ones. Alternative weighting schemes in which the weight of a pattern is a function of its homogeneity and/or prevalence, its degree, or is determined in such a way that the classification accuracy of the discriminant is maximized are detailed in [12].

4 Evaluation Methodology

In order to evaluate the accuracy of a classification model, several measures are traditionally used. This section reviews those that are used in the application. Subsection 4.1 covers the *quality* of a classification. Subsection 4.2 describes the *cumulative accuracy profile* (CAP), also called *Lorenz curve* or *power test*.

4.1 Classification Quality

Results for the classification of the experiments in a data set are displayed in the form of a classification matrix (Table 4).

Table 4: Classification Table

Experiment	Classification of Experiments		
Classes	Positive	Negative	Unclassified
Positive	a	c	e
Negative	b	d	f

The value a (resp., d) represents the percentage of positive (resp., negative) experiments that are correctly classified. The value c (resp., b) is the percentage of positive (resp., negative) experiments that are misclassified. The value e (resp., f) represents the percentage of positive (resp., negative) experiments that remain unclassified. Clearly, $a + c + e = 100\%$ and $b + d + f = 100\%$. The quality of the classification is defined [6] by

$$\mathcal{Q} = \frac{a+d}{2} + \frac{e+f}{4} \ . \tag{1}$$

4.2 Cumulative Accuracy Profile

A key measure of the quality of a classification model is the *cumulative accuracy profile* (CAP), also called *Lorenz curve* or *power test*. It has been widely used in medicine [40] and credit risk [18, 19, 47] to estimate the quality of machine learning, support vector machine and data mining approaches. Consider a classification model which returns a discriminant value for each experiment. An experiment with large discriminant value is classified as positive. Assume that the N experiments $(z^1, o^1), \ldots, (z^N, o^N)$ are ordered in non-increasing of their discriminant values. For $i = 0, \ldots, N$, let $q(i)$ be the number of positive experiments (i.e., experiments j with $z^j = 1$) in $\{(z^1, o^1), \ldots, (z^i, o^i)\}$, let q be the total number of positive experiments and let $PR = \frac{q}{N}$ be the fraction of positive experiments. The point $(\frac{i}{n}, \frac{q(i)}{q})$ is then a point of the CAP curve of the model. That curve is drawn in the 2-dimensional unit square with x-axis corresponding to percent of experiments and y-axis to percent of positive experiments.

A perfect classification function would have experiments $\{(z^1, o^1), \ldots, (z^q, o^q)\}$ as the q positive experiments and its CAP curve would be a line from $(0,0)$ to $(PR, 1)$ followed by a straight horizontal line joining $(PR, 1)$ to $(1, 1)$. Conversely, the expected CAP curve of a model without any discriminative power (i.e., random classification) would be a straight line from $(0,0)$ to $(1, 1)$. In reality, the CAP curve of classification models run between these two extremes.

The LAD model assigns a discriminant value $\Delta(j)$ to each experiment j, used to draw its CAP curve. A high discriminant value indicates that the considered experiment is positive and should result in a high value (i.e., within 10% of the best known value) of the metrics of interest, thereafter referred to as *response*.

The classification measure derived from the CAP curve is called *accuracy rate AR*

$$AR = \frac{AC}{AC + AP} \tag{2}$$

and is defined as the ratio of the area AC (shaded area in Figure 1) between the Lorenz curve (dotted line in Figure 1) of the classification model and the line joining $(0, 0)$ to $(1, 1)$ to the area AP between the line representing the perfect classification (bold line in Figure 1) and the line joining $(0, 0)$ to $(1, 1)$ representing the random classification.
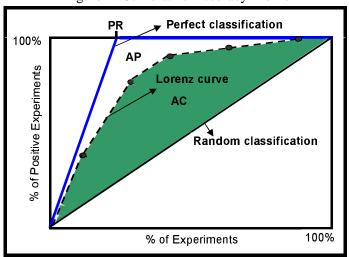
Figure 1: Cumulative Accuracy Profile

The AR measure is related to the classical parametric Wilcoxon rank sum and Mann-Whitney U tests [42] that are typically used to verify whether two distributions are identical. Denoting by $\Delta(j_1)$ (resp., $\Delta(j_2)$) the discriminant value of an experiment j_1 (resp., j_2) classified as positive (resp., negative) by the classification model, the Mann-Whitney U test counts the number of pairs (j_1, j_2) for which the inequality $\Delta(j_2) < \Delta(j_1)$ holds. If the classification function were perfect, the number of correctly classified pairs would be equal to $|N_p| \cdot |N_n|$ with $|N_p|$ (resp., $|N_n|$) referring to the number of positive (resp., negative) experiments. If the classification were random, $\Delta(j_2) < \Delta(j_1)$ would happen with probability 0.5, and thus $0.5 \cdot |N_p| \cdot |N_n|$ experiments would be correctly classified. Therefore, in this paper, we calculate the value of the accuracy ratio AR by means of the unbiased Mann-Whitney estimator $\hat{U}$

$$\hat{U} = \frac{\sum_{(j_1, j_2)} \alpha_{j_1, j_2}}{|N_p| \, |N_n|} \tag{3}$$

8

with $\alpha_{j_1, j_2} = 1$ if $\Delta(j_2) < \Delta(j_1)$ and equal to 0 otherwise.

4.3 Informational Content of Classification Metrics

The accuracy rate is the ratio of the performance improvement of the model being evaluated over the naive model to that of the perfect model over the naive model. It reflects the ability of the model to correctly rank the experiments. The accuracy is at its highest level (100%) if the ranking/ordering of the experiments in non increasing order of their discriminant values is exactly the same as the one obtained by ranking the experiments in non increasing order of the metrics with respect to which one seeks to classify them.

On the other hand, the classification quality reflects the ability of the model to discriminate good experiments from bad ones but does not account for the ranking of the experiments according to a given numerical or categorical criterion. It takes into account the percentage of true positives ($a/2$), the percentage of true negatives ($d/2$), and the percentage of unclassified experiments ($(e + f)/4$).

We illustrate the complementarity of the two classification metrics with the following example. Consider a data set composed of 50 positive and 50 negative experiments and assume that the LAD model generates the classification results displayed in Table 5.

Table 5: Relationship between Classification Metrics: Example 1

Experiment	Classification of Experiments		
Classes	Positive	Negative	Unclassified
Positive	90%	10%	0
Negative	0	100%	0

The classification quality is equal to 95%. On the other hand, the accuracy rate significantly varies depending on the ranking of the five misclassified experiments. All we can say is that the accuracy rate ranges between 80% and 100%. Recall that the accuracy rate orders the experiments by decreasing value of the discriminant value. If, in this ordering, the five positive experiments classified as negative by the LAD model occupy positions

- 46 to 50, the accuracy rate is equal to 100%;

- 71 to 75, the accuracy rate is equal to 90%;

- 96 to 100, the accuracy rate is equal to 80%;

Proposition 1 *A classification quality of 100% implies that the model has a 100% accuracy rate.*

The proof is straightforward. The converse is not necessarily true, as the order of the experiments by non increasing discriminant value can start with all the positive experiments followed by all the negative ones, but with all experiments classified as negative. While the accuracy rate of the model is 100%, its classification quality is 50% if there are as many positive experiments as negative ones in the experiment set.

The above discussion attests the relevance and the complementarity of the information provided by the two classification metrics to evaluate the discrimination power of a model.

5 Application Example

In this section, we evaluate the added value of our approach using a computational study. We were not successful in obtaining a variety of complex problems from the literature. (We are supportive of Pasupathy and Henderson's initiative [44] to develop a taxonomy and a publicly available testbed of Simulation-Optimization problems.)

Therefore, we evaluate our optimization approach on several variants of an assemble-to-order problem used in [36], in which items are made to stock to supply the demands for finished products, and various finished products are assembled to order from the items. The system operates using a continuous-review base-stock policy: each demand for a unit of an item triggers a replenishment order for that item. Items are produced one at a time on specific facilities, and production intervals are usually stochastic.

5.1 Problem Description

The specific assemble-to-order system we study is well known in the simulation-based optimization literature [36]. It has eight items $v_i, i = 1, \ldots, 8$, and five types of customers $c_j, j = 1, \ldots, 5$. Different types of customers come into the system as Poisson arrival processes with different rates $\lambda_j, j = 1, \ldots, 5$, and each of them requires a set of key items and a set of secondary items. If any of the key items are out of stock, the customer leaves. If all key items are in stock, the customer buys the product assembled from all the key items and the available secondary items. A sold item generates a unit profit $p_i, i = 1, \ldots, 8$, and each item in inventory has a holding cost per period $h_i, i = 1, \ldots, 8$. There are inventory capacities $C_i, i = 1, \ldots, 8$ for each item. The item production time is normally distributed with mean $\mu_i, i = 1, \ldots, 8$ and standard deviation $\sigma_i, i = 1, \ldots, 8$. The objective is to find the the optimal inventory levels $m_i, i = 1, \ldots, 8$ for each item to maximize the expected total profit w defined as the response.

We thus have a model with:

- 21 stochastic parameters which are the customers' arrival rates $\lambda_j, j = 1, \ldots, 5$ (Table 6), the mean production times $\mu_i, i = 1, \ldots, 8$ and the standard deviation of the production times $\sigma_i, i = 1, \ldots, 8$ (Table 7);

Table 6: Arrival Rate of Customers

j	1	2	3	4	5
λ_j	3.6	3	2.4	1.8	1.2

- 8 manipulables which are the optimal inventory levels $m_i, i = 1, \ldots, 8$.

5.2 Procedures for Model Construction and Validation

The methodology used to generate an LAD model in the cases listed below is as follows: Observables are selected (Section 5.3.1 describes the different set of observables considered in our tests), and a set of experiments is created, each experiment being specified by a choice of values for all the manipulables and for the stochastic parameters. We generate 1000 experiments:

- 500 experiments have fixed values for the stochastic parameters (using values in Tables 6 and 7). We use a 2-folding approach and assign 250 of them to the training set with fixed stochastic parameters (TRFSP) and we use them used to derive LAD models; the other 250 are assigned to

Table 7: Item Production Time: Average and Standard Deviation

i	μ_i	σ_i
1	0.15	0.0225
2	0.40	0.0600
3	0.25	0.0375
4	0.15	0.0225
5	0.25	0.0375
6	0.08	0.0120
7	0.13	0.0195
8	0.40	0.0600

the testing set with fixed stochastic parameters (TEFSP) and are used to validate the LAD models derived using TRFSP;

- 500 experiments have variable values for the stochastic parameters which can take any value (with uniform distribution) within $\pm 20\%$ of their mean values reported in Tables 6 and 7; 250 of them are assigned to the training set with variable stochastic parameters (TRVSP) and are used to derive LAD models and to validate the LAD models derived using TRFSP; the other 250 are assigned to the testing set with variable stochastic parameters (TEVSP) and are used to validate the LAD models derived using TRFSP and TRVSP.

For each experiment j, a simulation with a given number s of replications is run with a warm-up process of 20 periods and the average profit w^j over the next 50 periods is computed. In the remainder, a *long simulation* (resp., *short simulation*) is a simulation with $s = 50$ (resp., $s = 5$).

Short simulations are run for 215 (of the 250 experiments in the training set) to build the LAD classification model without using too much time. Long simulations are used to classify the experiments. Experiments are then ordered in decreasing order of their long-simulation average profit w^j. Denoting by $w^* = \max\limits_{j} w^j$, we associate to each experiment j an outcome z^j taking value

- 1 if $w^j \in [0.9 \cdot w^*, \ w^*]$

- 0 if $w^j < 0.85 \cdot w^*$

- -1 if $w^j \in [0.85 \cdot w^*, \ 0.9 \cdot w^*[$

Experiments in the training sets TRFSP and TRVSP with $z^j = 1$ or $z^j = 0$ are used to construct patterns and derive the LAD models whose accuracy is evaluated with respect to the measures presented in Section 4. We set the threshold values 0.9 and 0.85 defining z_j based on the results of numerous prior tests. (As Table 13 shows, the results are robust with respect to these threshold values.)

As explained in Section 3, the LAD method depends on a few control parameters (degree, prevalence, etc.). In this paper, all the LAD models are constructed by using a standard/default setting for the LAD control parameters. More precisely, the selected patterns have degree 3, 100% homogeneity and prevalence at least equal to 10% and 5 cut points are generated for each observable. Two main reasons motivate our choice of not customizing the setting of the LAD control parameters to the studied problem. First, the very conclusive results (see next section) obtained with standard settings show that the proposed approach can be used by non-LAD experts and does not require spending excessive time on the understanding of the arcane of the LAD method. Second, the reliance upon standard setting is a way

11

to hedge against the risk of developing an overfitted model. Nevertheless, as standard LAD settings does not prevent overfitting, we resort to a 2-folding cross-validation procedure to obtain a more definite statement regarding the overfitting issue. We derive the classification model with a subset of the experiments (the training set) and evaluate the quality of the model with respect to the classification it generates on the experiments not used for its construction (the testing set).

5.3 Constrained Simulation-Optimization Problem with Integer-Ordered Variables

In this section, we consider the original definition of the problem [36] in which all the decision variables, i.e. the manipulables, are discrete. Based on Pasupathy and Henderson's taxonomy [44], we call the associated problem a *constrained* Simulation-Optimization problem with *integer-ordered variables*.

We study two variants of the problem: The first one has fixed values for the stochastic parameters and the second one let them vary within 20% of their expected values. The results show that, in the first variant, a model based on the manipulables and the expected profit gives good results. However, in the second variant, having a broader set of observables is essential.

5.3.1 Fixed Value of Stochastic Parameters

In this section, we use the training set TRFSP (fixed stochastic parameters) to derive and analyze three LAD models:

- MOD I uses as observables the 8 manipulables o_i, $i = 1, \ldots, 8$;

- MOD II uses as observables the 8 manipulables and the short simulation average profit o_9;

- MOD III uses as observables the 8 manipulables, the short simulation average profit o_9 and the short simulation averages of:

    - item inventory levels o_i, $i = 10, \ldots, 17$,
    - lead time o_{18},
    - ratio of lead time to total time o_{19},
    - number of production lots in a work cell (work-in-progress) o_{20},
    - revenue o_{21},
    - total number of items produced o_{22},
    - number of stockouts o_{23}.

Note that when the stochastic parameters have fixed values, the expected profit is a function of the 8 manipulables. MOD I is trying to find an LAD model working with this set of variables. MOD II adds one (important) observable, the observed average profit. MOD III adds 14 observables, whose utility will be demonstrated in the next section when stochastic parameters are not fixed.

Table 8 displays the classification of the experiments in the training set TRFSP with the three models. The three models have a classification quality superior to 90% on the training set. We note that MOD I does not generate incorrect classification, but leaves about 20% (resp., 8%) of the positive (resp., negative) experiments unclassified and that MOD II and MOD III are the two top performers on the training set. The classification accuracy AR of MOD I (resp., MOD II, MOD III) is equal to 97.62% (resp., 98.74%, 98.83%). We note that a better classification might be obtained by fine-tuning the LAD control parameters for the problem at hand or by using another margin classifier.

12

Table 8: Classification Quality on Training Set TRFSP.

| Experiment Classes | Classification of Experiments | | | | | | | | |
| | MOD I | | | MOD II | | | MOD III | | |
	Positive	Negative	Unclassified	Positive	Negative	Unclassified	Positive	Negative	Unclassified
Positive	80.49%	0%	19.51%	95.12%	0%	4.88%	100%	0%	0%
Negative	0%	92.31%	7.69%	0%	99.41%	0.59%	0%	100%	0%
$\mathcal{Q}$	93.20%			98.63%			100%		

We now validate the three derived models and check whether the high classification accuracy subsists when the models are applied to experiments not used in their derivation. Indeed, overfitting, i.e., the phenomenon of building a model that is in close concordance with the observed data but has no predictive ability, may occur. In case of overfitting, the model has a high classification power on the training data, but performs poorly on new observations.

The classification quality of the three models on the testing set TEFSP is given in Table 9.

Table 9: Classification Quality on Testing Set TEFSP

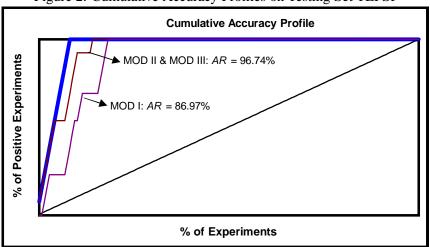| Experiment Classes | Classification of Experiments | | | | | | | | |
| | MOD I | | | MOD II | | | MOD III | | |
	Positive	Negative	Unclassified	Positive	Negative	Unclassified	Positive	Negative	Unclassified
Positive	69.23%	0%	30.77%	100%	0%	0%	100%	0%	0%
Negative	8.77%	83.33%	7.89%	0.88%	98.25%	0.88%	0.88%	98.25%	0.88%
$\mathcal{Q}$	85.95%			99.34%			99.34%		

Figure 2 shows that MOD II and MOD III have the same classification quality, almost 10% higher than the one of MOD I. We recall that the bold (resp., diagonal) line in Figure 2 represents the perfect (resp., random) classification.

The very high value of the two classification metrics for the testing sets strongly support the claim that no overfitting occurs. Indeed, the very high classification quality ($> 98.6\%$) and accuracy ($> 96.7\%$) of MOD II and MOD III do not decrease significantly when applied to the experiments in the testing set. The same comment does not extend to MOD I.

The above results show that, when values of the stochastic parameters are fixed, MOD II and MOD III are able to identify good decision settings based on the short-simulation average value of their observables. Clearly, the LAD models MOD II and MOD III predict with high accuracy when the performance of the system is within a predefined percentage of its optimal performance value. This is very useful and is described as "an ideal performance guarantee" in [8]. Moreover, the fact the LAD model provides this by relying on the observables of short simulations matters very much. Indeed, as noted by [21, 37], the determination of a high quality decision "in the fewest number of evaluations is the core problem". We also note that MOD II is more parsimonious in the sense that it uses less observables than MOD III to reach similar classification quality and accuracy.

The objective is now to verify whether the above conclusion can be extended when the defining values of the stochastic parameters vary within an interval. More precisely, the second validation phase pertains to the application of the three models to experiments not used in their derivation and in which the

Figure 2: Cumulative Accuracy Profiles on Testing Set TEFSP

stochastic parameters are not fixed but can take any value within $\pm 20\%$ of their mean value. The three LAD models are used to classify the experiments in the sets TRVSP and TEVSP, with results displayed in Table 10.

Table 10: Classification Quality on Experiments in Sets TRVSP and TEVSP.

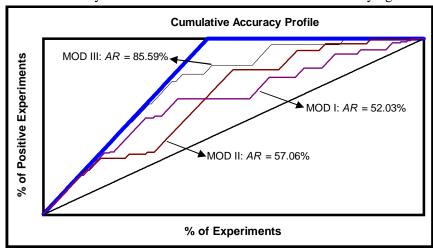| Experiment Classes	Classification of Experiments								
	MOD I			MOD II			MOD III		
	Positive	Negative	Unclassified	Positive	Negative	Unclassified	Positive	Negative	Unclassified
Positive	39.31%	45.09%	15.61%	74.57%	15.03%	10.40%	78.61%	17.92%	3.47%
Negative	4.07%	93.02%	2.91%	11.05%	86.63%	2.33%	8.72%	83.72%	7.56%
$\mathcal{Q}$	70.79%			83.78%			83.92%		

For each model, the classification quality on the sets containing experiments with varying stochastic parameters (Table 10) is significantly lower than that on the training (TRFSP, Table 8) and testing (TEFSP, Table 9) sets containing experiments with fixed values of the stochastic parameters. This seriously challenges the applicability of the LAD models derived with experiments having fixed stochastic parameters for the differentiation of experiments having varying stochastic parameters. Moreover, the drop in classification quality is accompanied by a decrease in the classification accuracy (Figure 3) for each model. The reduction in classification accuracy is particularly significant for MOD I and MOD II.

It appears clearly that none of three models built with respect to experiments with fixed values of the stochastic parameters enables the determination of good decision settings when the values of the stochastic parameters vary.

5.3.2 Varying Values of Stochastic Parameters

In this section, we derive three new LAD models, MOD IV, MOD V and MOD VI, on the basis of the experiments in the training set TRVSP. The patterns characterizing MOD IV (resp., MOD V, MOD VI) are defined with respect to the exact same observables used by MOD I (resp. MOD II, MOD III). A key

14

Figure 3: Cumulative Accuracy Profiles on Sets TRVSP and TEVSP with Varying Stochastic Parameters

difference is that MOD IV, MOD V and MOD VI are inferred from a set of experiments (TRVSP) having varying values of the stochastic parameters, instead of, as it is the case for MOD I, MOD II and MOD III, being derived from a set of experiments (TRFSP) having fixed values of the stochastic parameters.

The first objective of this section is to construct models enabling the accurate differentiation of good and bad decision settings based on short simulation of experiments characterized by varying values of the stochastic parameters. The second objective is to check the robustness of the models and to validate them using the 2-folding technique described in Section 5.2.

Table 11 gives the details of the classification obtained with MOD IV, MOD V and MOD VI on the training set TRVSP.

Table 11: Classification Quality on Training Set TRVSP

Experiment Classes	Classification of Experiments								
	MOD IV			MOD V			MOD VI		
	Positive	Negative	Unclassified	Positive	Negative	Unclassified	Positive	Negative	Unclassified
Positive	67.27%	0%	32.73%	88.18%	0%	11.82%	100%	0%	0%
Negative	0%	43.64%	56.36%	0%	91.82%	8.18%	0%	100%	0%
$\mathcal{Q}$	77.73%			95.00%			100%		

The commonalities between the three models are that they have about the same classification accuracy (Figure 4) and that none of them wrongly classified any of the observations in the training set. The models however differ in terms of their discrimination power: MOD VI classifies perfectly all experiments while MOD V (resp., MOD IV) leaves about 9% (resp., 50%) of the experiments unclassified. Clearly, MOD IV which solely relies on the manipulables does not have the classification ability required.

We now proceed to the validation of the models, and we check in particular whether the high classification quality of MOD V and MOD VI remains when they are applied to the experiments (not used for their derivation) of the testing set TEVSP.

The classification quality of MOD V and MOD VI on the testing set decreases (compared to Table

Figure 4: Cumulative Accuracy Profiles on Training Set TRVSP

Table 12: Classification Quality on Testing Set TEVSP

| Experiment | Classification of Experiments | | | | | | | | |
| | MOD IV | | | MOD V | | | MOD VI | | |
Classes	Positive	Negative	Unclassified	Positive	Negative	Unclassified	Positive	Negative	Unclassified
Positive	69.84%	6.35%	23.81%	85.71%	3.17%	11.11%	100%	0%	0%
Negative	11.29%	43.55%	45.16%	9.68%	88.71%	1.61%	9.68%	90.32%	0%
$\mathcal{Q}$	73.94%			90.39%			95.16%		

11) but remains very high ($> 90\%$). The following observations highlight the superior performance of MOD VI. First, the classification quality and the accuracy rate of MOD VI are higher than those of MOD V for both the testing and training sets. Second, the classification accuracy of MOD VI is invariant ($> 95.25\%$) regardless of whether it is used on the testing or training set. This contrasts with the classification accuracy of MOD V which drops from 92.64% to 84.09% when used to classify the experiments in respectively the training and testing sets. We now provide the results of six additional tests in which the binary outcome z^j (i.e. the long simulation expected profit) of each experiment in the testing set TEVSP is successively defined as

$$z^j = 1 \quad \text{if and only if} \quad w^j > \alpha \cdot w^*$$

for $\alpha = 85\%, 87.5\%, 90\%, 92.5\%, 95\%, 97.5\%$.

Table 13 provides the classification quality $\mathcal{Q}$ and accuracy AR (see also Figure 6) of the LAD model MOD VI when applied to the experiments of the set TEVSP whose outcome is defined as described above.

The very convincing results displayed above provide a further validation of MOD VI and show its applicability and high accuracy to classify experiments with varying values of the stochastic parameters and that were not used to construct the model. The very high values of the classification quality and accuracy rates for various definitions of the outcome of the experiments in the testing set is a very strong indicator of the stability of the model and the absence of overfitting. This result is very important in view

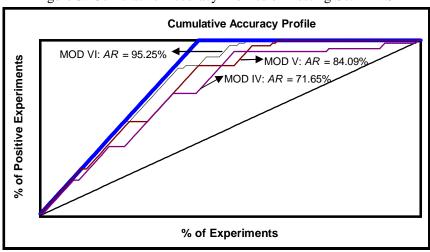Figure 5: Cumulative Accuracy Profiles on Testing Set TEVSP

Table 13: Classification Quality and Accuracy Rate

α	$\mathcal{Q}$	AR
85%	95.16%	92.20%
87.5%	95.16%	95.99%
90%	95.16%	94.83%
92.5%	95.16%	95.25%
95%	92.00%	92.90%
97.5%	90.62%	93.18%

of the occasional reports in the literature that the high fit of machine learning methods such as support vector machine is achieved at the cost of overfitting (see, for example, [46]).

5.4 Constrained Simulation-Optimization Problem with Continuous and Integer Variables

We have also considered a variant of the assemble-to-order problem where half of the manipulables are defined as continuous variables. The inventory levels m_i of items $i = 1, \ldots, 4$ are defined as continuous variables, taking any value in $[1, 12]$, while inventory levels of items 5 to 8 are defined as integer variables, taking any integer value in $[1, 12]$. The associated problem is called a *constrained* Simulation-Optimization problem with *continuous variables*.

The results are in perfect agreement with those obtained for the problem described in the preceding section and illustrate the applicability of the proposed method to both Simulation-Optimization problems with integer-ordered variables and Simulation-Optimization problems with continuous variables. For sake of brevity, we do not include the details of the computational tests.

17

6 Conclusion

In this paper, we develop an approach to identify good settings of decision variables of a stochastic system. A distinct feature of this work is the assumption that the values of the stochastic parameters of the system are unknown, but whose effect on the system can be captured by recording the values of quantities readily available at the end of a simulation. The LAD model provides an "an ideal performance guarantee" [8], since it predicts with striking accuracy when the performance of the system is within a predefined percentage of a targeted value or of its optimal performance value. An important feature [21, 37] is that the very accurate classification is obtained by using as inputs the short-simulation (i.e., limited number of replications) expected value of the observables.

Obviously, the construction of the LAD model requires time. However, this must be put in perspective with the following observations. First, the LAD model is very economical. For the problem studied, the derivation of the LAD model only requires the running of a limited number (250) of simulations with only a sample of them being used in the construction of the model. Moreover, in order to accurately separate bad from good experiments, the LAD model uses as only inputs (i.e., observables) the manipulables and other observables whose values are obtained by running short-simulations. This means that the LAD model gives its verdict (good or bad) about an experiment in very short-fashion (after only 5 replications). Second, the construction effort is largely offset by the gain in time when using the model as part of a local search heuristic. Moreover, in a case where the same stochastic model must be optimized periodically, the time invested to construct the classification model beforehand is worthwhile. What we can safely say at this point is that the running of 5 replications for an experiment takes less than 1 second on a standard PC and that the construction of the LAD model using the Datascope software takes a few seconds.

It is part of our future research plans to assess the time needed to construct the LAD model and, more importantly, to evaluate the overall computational savings that our approach will permit over the entire Simulation-Optimization process. Indeed, the LAD classification model is very accurate in distinguishing "good" from "bad" experiments and can thus improve the use of computing resources by allocating more time to promising experiments. To test the benefits of this key feature, we plan to use the LAD-based classification model in an iterative optimization procedure. An experiment classified "bad" by the LAD classification model would be immediately dropped from further consideration by the optimization-based simulation algorithm, while an experiment classified as "good" would receive more attention (i.e., more replications would be run for this experiment) in order to obtain a very accurate estimate of its results.

Finally, we also note that the proposed approach is not contingent on the running of a sample of replications for each possible setting and that empirical results show that the LAD for Simulation approach performs equally well for Simulation-Optimization problems with integer-ordered variables and with continuous and integer variables.

References

[1] Adenso-Diaz B., Laguna M. 2006. Fine-Tuning of Algorithms Using Fractional Experimental Designs and Local Search. *Operations Research* 54, 99-114.

[2] Alexe S. 2002. Datascope - A New Tool for Logical Analysis of Data. *DIMACS Mixer Series*, DIMACS, Rutgers University.

[3] Alexe G., Alexe S., Bonates T., Kogan A. 2007. Logical Analysis of Data The Vision of Peter L. Hammer. *Annals of Mathematics and Artificial Intelligence* 49, 265-312.

[4] Alexe G., Hammer P.L. 2006. Spanned Patterns for the Logical Analysis of Data. *Discrete Applied Mathematics* 154 (7), 1039-1049.

[5] Alexe S., Hammer P.L. 2007. Accelerated Algorithm for Pattern Detection in Logical Analysis of Data. *Discrete Applied Mathematics* 154 (7), 1050-1063.

[6] Alexe S., Hammer P.L. 2007. Pattern-Based Discriminants in the Logical Analysis of Data. In: *Data Mining in Biomedicine*. Eds: Pardalos P.M., Boginski V.L., Vazancopoulos A. Springer.

[7] Andradóttir S. 1998. Simulation Optimization. Chapter 9 in *Handbook of Simulation*. Ed: Banks J. John Wiley & Sons, New York, 307-333.

[8] Andradóttir S. 2002. Simulation Optimization: Integrating Research and Practice. *INFORMS Journal on Computing* 14 (3), 216-219.

[9] Barton R.R. 1987. Testing Strategies for Simulation Optimization. *Proceedings of the 1987 Winter Simulation Conference*. Eds: Thesen A., Grant H. IEEE Press, 391-401.

[10] Boesel J., Nelson, B.L., Kim S.-H. 2003. Using Ranking and Selection to "Clean Up" After Simulation Optimization. *Operations Research* 51 (5), 814-825.

[11] Boros E., Hammer P.L., Ibaraki T., Kogan A. 1997. Logical Analysis of Numerical Data. *Mathematical Programming* 79, 163-190.

[12] Boros E., Hammer P.L., Ibaraki T., Kogan A., Mayoraz E., Muchnik I. 2000. An Implementation of Logical Analysis of Data. *IEEE Transactions on Knowledge and Data Engineering* 12 (2), 292-306.

[13] Branke J., Chick S., Schmidt C. 2005. New Developments in Ranking and Selection: An Empirical Comparison of the Three Main Approaches. *Proceedings of the 2005 Winter Simulation Conference* 708-717.

[14] Chen H.-C., Chen C.-H., Yücesan E. 2000. Computing Efforts Allocation for Ordinal Optimization and Discrete Event Simulation. *IEEE Transactions on Automatic Control* 45 (5), 960-964.

[15] Chick S. E., Inoue K. 2001. New Two-Stage and Sequential Procedures for Selecting the Best Simulated System. *Operations Research* 49, 732743.

[16] Colbourn C.J., Dinitz J.H. 2007. *Handbook of Combinatorial Designs*, 2nd ed. Chapman/Hall/CRC.

[17] Dai L. 1996. Convergence Properties of Ordinal Optimization Comparison in Simulation of Discrete Event Dynamic Systems. *Journal of Optimization Theory and Applications* 91 (2), 363388.

[18] Engelmann B., Hayden E., Tasche, D. 2003. Testing Rating Accuracy. *Risk* 16 (1), 82-86.

[19] Engelmann B., Rauhmeier R. 2006. *The Basel II Risk Parameters - Estimation, Validation, and Stress Testing*. Springer, Berlin Heidelberg.

[20] Fu M.C. 2002. Optimization for Simulation: Theory vs. Practice. *INFORMS Journal on Computing* 14 (3), 192215.

[21] Fu M.C. 2002. Simulation Optimization in the Future: Evolution or Revolution? *INFORMS Journal on Computing* 14 (3), 226-227.

[22] Fu M.C. 2007. Are We There Yet? The Marriage Between Simulation & Optimization. *OR/MS Today* June 2007, 16-17.

[23] Fu M.C., Glover F.W., April J. 2005. Simulation Optimization: A Review, New Developments, and Applications. In *Proceedings of the 2005 Winter Simulation Conference*. Eds: Kuhl M.E., Steiger N.M., Armstrong F.B., Joines J.A.

[24] Glynn P.W., Juneja S. 2004. A Large Deviations Perspective on Ordinal Optimization. *Proceedings of the 2004 Winter Simulation Conference*, 577-585.

[25] Gould N.I.M., Orban D., Sartenaer A., Toint P.L. 2005. Sensitivity of Trust-Region Algorithms. *4OR* 3, 227-241.

[26] Hammer P.L. 1986. Partially Defined Boolean Functions and Cause-Effect Relationships. *International Conference on Multi-Attribute Decision Making Via OR-Based Expert Systems*. University of Passau, Passau, Germany.

[27] Hammer P.L., Bonates T.O. 2005. Linear and Nonlinear Set Covering Problems in the Logical Analysis of Data. *Workshop on Mathematical Programming for Data Mining and Machine Learning*. McMaster University, Hamilton, Ontario.

[28] Hammer P.L., Bonates T.O. 2006. Logical Analysis of Data: From Combinatorial Optimization to Medical Applications. *Annals of Operations Research* 148 (1), 203-225.

[29] Hammer P.L., Kogan A., Lejeune M.A. 2006. Modeling Country Risk Ratings Using Partial Orders. *European Journal of Operational Research* 175 (2), 836-859.

[30] Hammer P.L., Kogan A., Lejeune M.A. 2007. Reverse-Engineering Country Risk Ratings: Combinatorial Non-Recursive Model. *Working Paper* available at `http://www.optimization-online.org/DB_HTML/2007/03/1619.html`

[31] Hammer P.L., Kogan A., Lejeune M.A. 2007. Reverse-Engineering Banks' Financial Strength Ratings Using Logical Analysis of Data. *Working Paper* available at `http://www.optimization-online.org/DB_HTML/2007/02/1581.html`

[32] Hedayat A.S., Sloane N.J., Stufken J. 1999. *Orthogonal Arrays: Theory and Applications*. Springer.

[33] Hicks C.R., Turner K.V. 1999. *Fundamental Concepts in the Design of Experiments*. Oxford.

[34] Ho Y.C., Cassandras C.G., Chen C-H., Dai L. 2000. Ordinal Optimization and Simulation. *Journal of Operational Research Society* 51, 490-500.

[35] Ho Y.C., Sreenivas R., Vakili, P. 1992. Ordinal Optimization of Discrete Event Dynamic Systems. *Journal of Discrete Event Dynamic Systems* 2 (2), 61-88.

[36] Hong L.J., Nelson B.L. 2006. Discrete Optimization via Simulation Using COMPASS. *Operations Research* 54 (1), 115129.

[37] Kelly J.P. 2002. Simulation Optimization is Evolving. *INFORMS Journal on Computing* 14 (3), 223-225.

[38] Kleijnen J.P.C., Wan J. 2007. Optimization of Simulated Systems: OptQuest and Alternatives. *Simulation Modelling Practice and Theory* 15 (3), 354-362.

[39] Kogan A., Lejeune M.A. 2009. Combinatorial Methods for Constructing Credit Risk Ratings. In: *Handbook in Quantitative Finance*. Eds: Lee C.-F., Lee A.C. Springer. Forthcoming.

[40] Lee W. C. 1999. Probabilistic Analysis of Global Performances of Diagnostic Tests: Interpreting the Lorenz Curve-based Summary Measures. *Statistics in Medicine* 18, 455-471.

[41] Lin B.W., Rardin R.L. 1979. Controlled Experimental Design for Statistical Comparison of Integer Programming Algorithms. *Management Science* 25, 1258-1271.

[42] Mann H., Whitney D. 1947. On a Test of Whether One of Two Random Variables is Stochastically Larger than the Other. *Annals of Mathematical Statistics* 18, 50-60.

[43] Nelson B. L., Swann J., Goldsman D., Song W. 2001. Simple Procedures for Selecting the Best Simulated System When the Number of Alternatives is Large. *Operations Research* 49 950-963.

[44] Pasupathy R., Henderson S.G. 2006. A Testbed of Simulation-Optimization Problems. *Proceedings of the 2006 Winter Simulation Conference*.

[45] Pukelsheim F. 2006. *Optimal Design of Experiments*. SIAM (Reprint 1993).

[46] Rimer M., Martinez T. 2006. Classification-Based Objective Functions. *Machine Learning* 63 (2), 183-205.

[47] Sobehart J., Keenan S., Stein R. 2001. Benchmarking Quantitative Default Risk Models: A Validation Methodology. *Algorithmic Research Quarterly* 4 (1-2).

Figure 6: Cumulative Accuracy Profiles

Cumulative Accuracy Profile - $z^j = 1$ if $w^j > 0.85w^*$

% of Positive Experiments

% of Experiments

Cumulative Accuracy Profile - $z^j = 1$ if $w^j > 0.875w^*$

% of Positive Experiments

% of Experiments

Cumulative Accuracy Profile - $z^j = 1$ if $w^j > 0.9w^*$

% of Positive Experiments

% of Experiments

Cumulative Accuracy Profile - $z^j = 1$ if $w^j > 0.925w^*$

% of Positive Experiments

% of Experiments

Cumulative Accuracy Profile - $z^j = 1$ if $w^j > 0.95w^*$

% of Positive Experiments

% of Experiments

Cumulative Accuracy Profile - $z^j = 1$ if $w^j > 0.975w^*$

% of Positive Experiments

% of Experiments