Advances in analytics: Integrating dynamic data mining with simulation optimization

M. Better
F. Glover
M. Laguna

We introduce a simulation optimization approach that is effective in guiding the search for optimal values of input parameters to a simulation model. Our proposed approach, which includes enhanced data mining methodology and state-of-the-art optimization technology, is applicable to settings in which a large amount of data must be analyzed in order to discover relevant relationships. Our approach makes use of optimization technology not only for data mining but also for optimizing the underlying simulation model itself. A market research application embodying agent-based simulation is used to illustrate our proposed approach.

Introduction

In industry, managers are constantly faced with situations which require making decisions that can have important consequences for profitability, market share, customer satisfaction, or other key factors affecting company success. Often these situations are highly complex and exhibit considerable uncertainty. This makes it practically impossible for managers to predict accurately the effect of a particular decision on system performance. In such cases, it is desirable to have a simplified but realistic model to test real-world decision-making scenarios and to evaluate the outcomes of alternative decisions that affect the company's success. Computer simulation has become a methodology of choice in these situations by giving the user an ability to build models for testing numerous configurations of complex systems in a relatively quick and inexpensive manner.

A primary function of computer simulations is to evaluate the implications of operational or policy changes. For example, changes to the number of tellers at a bank, the number of machines in a job shop, or the number of lanes in a highway can be tested in order to determine their effect on the performance of a system as evaluated by one or multiple measures, such as cost, expected profit, risk, waiting time, or resource utilization. However, in cases with a high degree of complexity and uncertainty, it is not always obvious which variables to focus on in order to improve performance, nor is it evident to what extent these variables should be changed. Furthermore, the number of possible different system configurations, even in relatively simple models, is so large that it is impossible from a practical standpoint to enumerate them all to find the best configuration.

We propose an approach that incorporates an advanced data mining module to identify relevant system inputs and to analyze the way these inputs interact within the system. This data mining model uses traditional methods to select relevant features, cluster, and classify data. Its principal contribution makes use of an innovative dynamic data mining procedure that is activated at certain intervals during the optimization process in order to make use of information obtained during that process, with the goal of speeding the search for optimal solutions. In addition, we incorporate state-of-the-art optimization technology, the OptQuest** optimization engine, in order to guide the search for optimal policies or scenarios for a given system, based on user-defined performance measures. Our approach describes work in progress. It has elements in common with, and may be applied in, areas such as bio-informatics, which make use of small simulations in order to create fitness functions for an objective performance measure. However, the approach has not yet been applied by others to the areas of market research, finance, and other business settings, which we plan to explore in this paper.

0018-8646/07/$5.00 © 2007 IBM

The paper is organized as follows. In the next section, we discuss our proposed data mining methodology. The two sections that follow describe our approach to the optimization of simulation models, and the manner in which the dynamic aspects of data mining interact with the optimization process. The section on an integrated approach to simulation optimization illustrates our proposed approach in detail, focusing on an example using agent-based simulation in a market research application. The final section presents our concluding remarks and avenues for future research.

Data mining

The field of data mining is concerned with the efficient storage, access, modeling, and, ultimately, understanding of large data sets. A detailed discussion of these various aspects of data mining, both from a theoretical and from an implementation viewpoint, can be found in [1]. From the viewpoint of our approach, data mining can be specifically defined as the analysis of data in order to identify patterns or discover relationships among the various elements of a data set. From such a viewpoint, data mining—including classification, clustering, pattern analysis, discrimination, feature selection, and a broad array of related domains—looms as a critically important practice in business, science, and government.

The catalog of optimization methods used for data mining applications encompasses neural networks, support vector machines, Bayesian analysis, and Markov blankets, among others (see [2–5] for useful descriptions and tutorials). These methods are continuing to evolve and improve, as demonstrated by recent innovations in *conditional separation methods* based on new mixed-integer programming, quadratic optimization, and metaheuristic models, which have provided advances in applications such as evaluating the quality of banks, diagnosing cancer, and creating systems for scoring credit applications (see, e.g., [6–8]). The following are three prominent data mining problems that are relevant in our proposed approach.

Classification and discrimination

Let a_{ij} denote the value of a specific characteristic or attribute exhibited by elements in a data set, where each element i ($i = 1, \cdots, m$) is described by n different attributes indexed by j ($j = 1, \cdots, n$). We seek a decision rule to classify these elements in order to identify correctly whether a given vector $\mathbf{A}_i = (a_{i1}, \cdots, a_{in})$ should belong among the elements of Group 1 or instead among those of Group 2 (denoted G_1 and G_2, respectively). (The basic model can easily be extended to three or more groups.) For instance, the elements $\mathbf{A}_i$ may refer to a patient's physical characteristics, symptoms, and laboratory test results. We seek to classify patients according to whether they have a particular disease ($i \in G_1$) or not ($i \in G_2$), and the first component a_{i1} of $\mathbf{A}_i$ may refer to the applicant's age, the second component a_{i2} may refer to the patient's gender, and so forth.

Given the knowledge of the $\mathbf{A}_i$ vectors and their group membership, our goal is to provide a decision rule that not only performs well in discriminating whether one particular vector belongs to Group 1 or Group 2, but also whether a new vector $\mathbf{A}$, not among the original known vectors, should be classified as belonging to one group or the other.

Feature selection

From among the entire set of attributes (i.e., *features*) that are included in the data defining a classification problem, we seek to identify a subset consisting of those that are relevant to the target (i.e., the class variable). In general, the goal is to minimize the number of features considered in order to make a correct classification. This is desirable in order to reduce the computational cost of determining correct classifications, but it can also lead to improved classification accuracy by reducing the risk of "overfitting," a term that refers, in this instance, to a model that performs poorly when classifying new observations because it has been trained using a data set with a high ratio of attributes to known observations.

A new approach that is proving very successful in feature selection and classification is based on the concept of Markov blankets (see [5] for a discussion of successful implementations in various application areas). A Markov blanket is a special case of a Bayesian network (BN). A BN is a directed acyclical graph $\mathbf{G}$, consisting of a set of nodes or vertices $\mathbf{V}$ that represent the features (i.e., random variables) in the data set, and directed edges, $\mathbf{E}$, between the nodes, representing conditional independence relationships among the variables. Given a graph $\mathbf{G} = \{\mathbf{V}, \mathbf{E}\}$, we say that a node $X \in \mathbf{V}$ is a *cause* of node $Y \in \mathbf{V}$ if there is a directed edge from X to Y. Equivalently, we say that X is a *parent* of Y, and Y is a *child* of $\mathbf{X}$. The *Markov condition* imposed on a BN renders a variable $\mathbf{X}_i \in \mathbf{V}$ independent of any non-descendant or non-parent, conditional on its parents. The Markov blanket of X_i includes its parents, its children, and the parents of its children. The Markov blanket, therefore, constitutes the set of variables that are not independent of the target variable, conditional on all the other variables in the set. In other words, any variable not in the Markov blanket of the dependent (target) variable is considered redundant for the purpose of predicting the value of the dependent variable. Thus, by finding the Markov blanket of the variable of interest, we can discard all of the other, irrelevant variables in the domain.

Clustering

Clustering involves the grouping of data into cohesive clusters according to context-dependent criteria. The vector $\mathbf{A}_i$ of the attributes of an individual consumer i can be considered a point in space. A very simple clustering algorithm would create clusters made up of those points closest to the center of a cluster, up to a certain distance threshold. If that threshold is exceeded, a new cluster is formed. The process continues until all points have been assigned to a cluster. Nearly as simple as this process, the classical k-means approach begins by randomly selecting k points as centers and then assigns each data point to the center closest to it. New centers are identified for the resulting clusters, and the process repeats until the clusters no longer change their composition. A variety of more complex clustering approaches are described in [9–12].

Optimizing computer simulation models

The issue of identifying best values for a set of decision variables falls within the realm of optimization. Until quite recently, however, the methods available for determining optimal decisions have been unable to cope with the complexities and uncertainties posed by many real-world problems of the form treated by simulation.

The need for optimization of simulation models arises when the systems analyst wants to select a set of model specifications (i.e., input parameters and/or structural assumptions) that leads to optimal performance. On one hand, the range of parameter values and the number of parameter combinations are too large for analysts to simulate all possible scenarios, so they need a method to intelligently guide the search for good solutions. On the other hand, without simulation, many real-world problems are too complex to be modeled by mathematical formulations that are at the core of pure optimization methods. This creates a conundrum; pure optimization models alone are incapable of capturing all of the complexities and dynamics of the system, so one must resort to simulation, which in turn cannot easily find the best solutions. Simulation optimization resolves this conundrum by combining both methods.

The merging of optimization and simulation technologies has seen remarkable growth in recent years. Today, most Monte Carlo and discrete event simulation software packages include an optimization tool as part of their product. Until relatively recently, however, the simulation community was often reluctant to use optimization tools. Optimization models were thought to oversimplify the real problem, and it was not always clear why a certain solution was the best [13]. However, recent developments are changing this picture.

Advances in the field of metaheuristics—the domain of optimization that augments traditional mathematics with artificial intelligence and methods based on analogs to physical, biological, or evolutionary processes—have led to the creation of optimization engines that successfully guide a series of complex evaluations with the goal of finding optimal values for the decision variables, as in [14–20]. One of those engines is the search algorithm embedded in the OptQuest optimization system [21]. OptQuest is designed to search for optimal solutions to the following class of optimization problems:

$$
\begin{array}{lll}
\text{Maximize or minimize} & F(x) & \text{(objective)}, \\
\text{subject to} & \mathbf{A}x \le b & \text{(constraints)}, \\
 & g_l \le G(x) \le g_{\mathrm{u}} & \text{(requirements)}, \\
 & l \le x \le u & \text{(bounds)},
\end{array}
$$

where x is a set of variables that can be continuous or discrete, with an arbitrary step size. Matrix $\mathbf{A}$ is described shortly.

A typical example might be to maximize the throughput of a factory by judiciously increasing machine capacities, subject to a budget restriction and a limit on the maximum work in process (WIP). In this case, x represents the specific capacity increases, and $F(x)$ is the expected throughput at capacity x. The budget restriction is modeled as $\mathbf{A}x \le b$, where $\mathbf{A}$ could represent a matrix of operating cost of capacity x, and b is the available operating budget. The limit on WIP is achieved by a requirement modeled as $G(x) \le g_{\mathrm{u}}$, where $G(x)$ represents the average WIP given capacity x, and g_{u} is an upper bound on the desired WIP. The distinction between constraints and requirements is subtle: The former involves only model inputs, whose values are known prior to running the simulation, while the latter is a requirement on one (or more) model outputs. Each evaluation of $F(x)$ and $G(x)$ requires a discrete simulation of the factory. By combining simulation and optimization, a powerful design tool is produced.

The optimization procedure uses the outputs from the system evaluator (i.e., the simulation), which measures the merit of the inputs that have been communicated to the model. On the basis of both current and past evaluations, the optimization procedure decides upon a new set of input values (see **Figure 1**).

The optimization procedure is designed to perform a special "non-monotonic search," in which the successively generated inputs produce varying evaluations, not all of them improving, but which over time provide a highly efficient trajectory to the best solutions. The process continues until an appropriate termination criterion, which is usually based on the user's preference for the amount of time to be devoted to the search, is satisfied. For a step-by-step description of a practical application of simulation optimization, see [22].

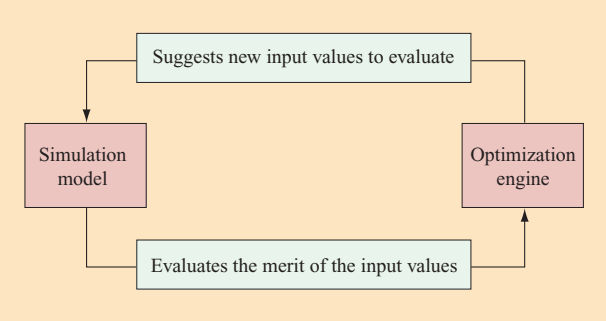

Figure 1

One cycle illustrating the coordination between the optimization engine and the simulation model. The optimization engine suggests new input variables for the simulation model, and the simulation model evaluates the merit of the input values.

The advent of simulation optimization heralds a new dimension for data mining, expanding the range of problems that can be effectively addressed by this area. We call this expanded realm *dynamic data mining*, and discuss its key features and the opportunities facilitated by its emergence.

Dynamic data mining

Classical data mining, as previously noted, presupposes the existence of a collection of data in some repository, such as a computer database, whose elements are often referred to as points (i.e., vectors) in some sort of space. (The space need not be numeric; for example, some vector components may represent qualitative rather than quantitative elements.) The data may be generated by historical records or by various types of deterministic or stochastic processes, which may in some cases be viewed as a dynamic source for the data. However, once generated, the data elements constitute a static collection that is analyzed by taking its elements as fixed. In other words, the data is not modified or updated, which is in contrast to the dynamic case, in which we may update the data by incorporating new features or factors as a result of information gained during the optimization process. To accommodate data that changes over time, successive snapshots or samples are taken using updated forecasts or other information, and the analysis proceeds in the form of a moving data window design.

However, a more responsive and effective approach involves conducting data mining for dynamic processes by analyzing the processes *in action,* as they are unfolding. This is the sense of the term *dynamic data mining* as we intend it to be interpreted here. The gains of being able to handle data mining considerations in this dynamic fashion are considerable. For example, the classical and customary approach of generating a fixed set of data to analyze often misses the opportunity to handle uncertainty and to respond to the range of problem characteristics that can be captured through the use of simulation. When we speak of data analysis, we note that the outcome of any finite simulation can be represented as a fixed set of data. However, by not interacting with and analyzing the data as it is being generated, one foregoes the opportunity to manipulate the generation of the data stream in many traditional approaches. Moreover, the methods designed for analyzing fixed data sets have a significantly different character than those designed for simulation optimization, with consequences illustrated by the advances that simulation optimization has brought about in many other problem areas.

Dynamic data mining based on simulation optimization can be illustrated by a problem of determining appropriate classifications of investments. The investments need not be financial, but can be related to a set of projects that a company is planning to undertake, or a set of departmental budget allocations that the company wants to make, such as allocations involving research and development, marketing and advertising, or sales. We do not merely want to divide the investments into categories such as "good" and "bad," but additionally to generate categories representing investment quality by the use of multiple criteria that include factors of risk and return in different combinations. Moreover, suppose that we are interested in classifying not just a single investment in isolation from others, but in classifying collections of investments (i.e., investment packages) that are not specified *a priori* but are to be determined dynamically as part of the overall classification process. Such a determination is highly relevant in investment contexts, because factors such as risk depend not only on a single investment but on the composition of a complete package.

Finally, we address a complication not considered in most classification efforts, by requiring the collections of assembled investments to satisfy a budget constraint so that the total investment cost of their members does not exceed a given limit. This increases the scope of the dynamic part of the classification process, since it is very challenging to enumerate fixed collections of points that meet the requirements of such a constraint, and we want to handle such a task without the need to explicitly itemize all of the points that qualify for consideration. In the same vein, at an even more challenging level, we want to be able to uncover the good *collections* of investments without having to itemize those collections, which are exponentially more numerous than the possibilities consisting of single investments by themselves.

Dynamic data mining using simulation optimization can accommodate all of these goals in an entirely

natural way. Formulating the problem in the simulation optimization setting is no different than formulating any other problem in this setting. The OptQuest optimization engine keeps a record of solutions (e.g., corresponding to investment packages) generated throughout the search. Such a record provides the source of the instances that meet the classification standards.

Most traditional forms of classification provide a set of rules, or functional specifications, as a way to classify elements. In simulation optimization, it is instead the simulation optimization process itself that provides the means to produce the classification. A simulation optimization process may be viewed as a set of rules, but the rules are more complex than customary rules, and they have an iterative character.

How does this approach facilitate the classification of a new element, not in a current data set, once it is encountered? If the element is already a collection of investments to be treated as a single package, the package must simply be subjected to the simulation process to evaluate it and to see where it belongs in relation to items previously classified. If the element, instead, is a group of securities that enrich the investment pool, the simulation optimization process is rerun using the modified pool, with the ability to include constraints such as those stipulating a minimum number of the new securities to be members of candidate packages produced. This results in generating new packages of investments that meet the desired classification standards. This approach is applicable beyond the financial setting as well. For example, the role taken by new securities and investment opportunities can be given to marketing initiatives not previously considered, new technology, or an acquisition target that has only recently been identified, such as a firm that is targeted for potential merger, acquisition, or strategic alliance.

The resulting set of good solutions (i.e., investment packages) can then be ranked according to statistical criteria or a mix of quantitative and qualitative attributes that are weighted to produce an overall score.

An integrated approach to simulation optimization

We propose a practical approach for the optimization of simulation models, which is enhanced by a dynamic data mining module that provides input to the simulation, and state-of-the-art optimization technology in order to find optimal configurations or operating policies for a particular system. A high-level graphical description of the proposed approach is shown in **Figure 2**.

As shown in the figure, we begin with a repository of data concerning the domain of interest. This approach adds two critical features to traditional applications of computer simulation in industry: the data mining module and the optimization engine.

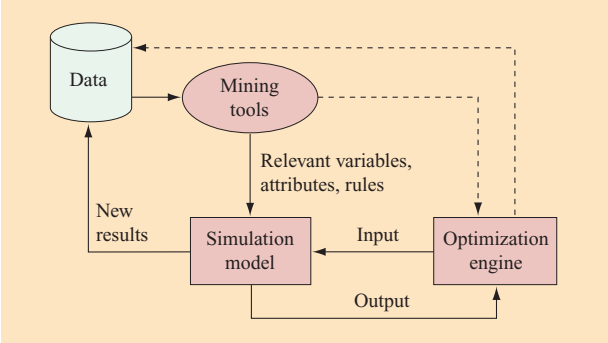

Figure 2

High-level view of the integrated approach to simulation optimization.

Through the use of dynamic data mining tools, we identify the relevant variables, attributes, and rules that govern the computer simulation model. In addition, we can dynamically classify scenarios and incorporate new elements of the system, as described in the section on data mining. The simulation model itself then provides an *evaluator* of the performance of different scenarios we wish to test, interacting with the optimization engine as described in the section on optimizing computer simulation models. In addition to guiding the search for optimal scenarios for the simulation model, the optimization engine also provides a critical component of the data mining tools we use for clustering, feature selection, and classification (indicated by the inner dashed arrow in Figure 2).

To better illustrate the proposed approach and describe each of the relevant modules, we focus on an agent-based simulation approach to a market research application. As a foundation, we first describe the agent-based simulation methodology.

Agent-based simulation

Within the computer simulation arena, Monte Carlo simulation, continuous flow simulation, and discrete event simulation are well-known tools that are widely used in industry. Recently, agent-based simulation has been gaining notoriety in a variety of application areas, including human resources management [23], market research [24], supply chain and logistics management [25], and materials science [26], to name only a few.

Also known as "Artificial Life," owing to the work of Langton [27], agent-based computer simulation makes use of artificial "agents" that represent the entities or participants in the system. An agent, for example, can be

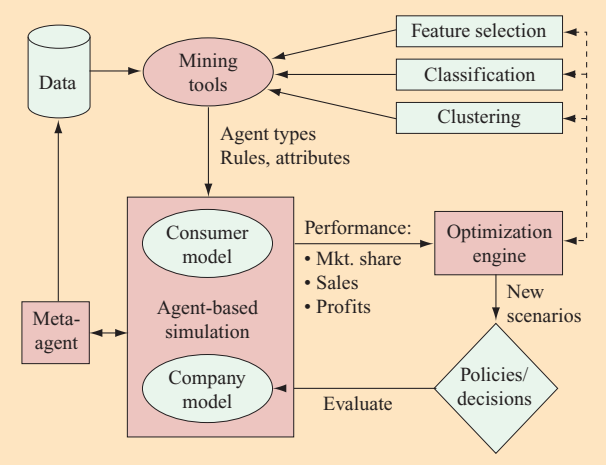

Figure 3

The optimization approach in the context of an agent-based simulation example for market research. Performance is measured by such factors as market share, sales, and profits, and the simulated performance data is transmitted from the agent-based simulation to the optimization engine.

a computer representation of an organism, a person, or an organization that interacts with the environment and with the other entities in the system. As explained in [28], even very complex phenomena can be modeled by a system of relatively simple, autonomous agents that follow simple rules of interaction. Each agent's activities are programmed as a set of rules, and agents can be identical or can belong to different types, or "breeds." These rules can also be designed to give agents behavior-learning and decision-making capabilities. This provides great flexibility in terms of the desired level of detail.

In the case of a market research application, for example, one type of artificial agent can represent typical consumers in a particular market segment, another can represent a consumer in a different market segment, and yet another type of agent can be programmed to represent firms that operate in the market and compete for a share of those consumers.

In some cases the definition of the artificial agent or agents is obvious, especially if the participants or entities can all be considered identical. However, in the vast majority of applications, the participants are not homogeneous. For instance, purchasing behavior may vary greatly among potential customers as a function of age, gender, income level, or geographic location, while investment strategies among firms may differ greatly on the basis of sales volume, brand equity, or financial strength. We consider these complications in the example presented in the following section.

An illustrative market research example

Consider a start-up firm that is seeking venture capital financing. The company founders need to prepare a detailed business plan that will impress potential investors. Particularly important is the strategy of the firm to gain a share of a market that is currently dominated by a few well-known competitors. Aside from market share, the investors are interested in the forecasted profitability of the company, as well as its expected market liquidity and the specific uses to which their investment will be applied. The founders have collected a large amount of data about the market, including demographic, economic, and purchasing behavior data related to consumers, as well as financial data with respect to their competitors and the quality, functionality, and consumer perception of the competition's products.

The founders have decided to develop a decision support system that relies on agent-based computer simulation (see **Figure 3** for a graphical representation). First, they seek to develop a *consumer model*, with artificial agents representing consumers in different market segments. Next, they wish to develop a *company model* that will interact with the consumer model. The company model uses additional agents programmed to represent the other firms in the market, as well as their own firm.

In order to identify the different market segments and the behavioral rules that govern the artificial consumers (agents) within each segment, a clustering methodology is applied to the original data, mentioned previously. Then, once the segments have been identified, a feature selection method identifies the attributes that are relevant in modeling the consumers' purchasing behavior. In other words, the founders want to use the smallest possible set of rules to model the consumer agents, and they are aware that some of the data collected on real-life consumers may not be important in terms of modeling the computerized agents' purchasing decisions.

Finally, the goal is to obtain from the computer model an optimal solution in terms of the allocation of funds to the various business functions within the firm—such as marketing, research and development, and supply chain management—that will produce the maximum market share while meeting prespecified levels of profitability, liquidity, sales volume, and other important factors. This solution can then be translated into real-world actions and policy decisions that the founders can implement. In order to achieve this, an optimization engine interacts with the agent-based computer simulation to guide the search for optimal fund allocation scenarios.

DDM module

In agent-based simulation applications, the process by which the different types of agents are identified—and

their corresponding behavioral rules defined—is often not thoroughly examined. In other words, agents are merely considered to be inputs into the simulation model, and there is little attempt to assess their validity. To correct this oversight, we incorporate a dynamic data mining (DDM) module at the forefront of our approach. The DDM function is represented by the mining tools block and the meta-agent block in Figure 3.

The DDM module first makes use of various data mining techniques, aided by the optimization engine, to identify different classes or types of agents. In the case of the market research application, the DDM module is used to identify agents that represent the types of consumers in different market segments, and a set of parameters that represent the agents' rules of behavior and interaction.

To begin with, we use a Markov blanket methodology for feature selection. Once the relevant features have been identified, we apply a clustering algorithm that consists of a metaheuristic solution method to unconstrained quadratic optimization. The procedure clusters consumers into different market segments according to the attributes chosen during the feature selection stage. These clusters represent the different market segments for the agent-based consumer model.

Once the market segments are defined, we again use a Markov blanket-based procedure in order to define behavioral rules for the consumer. The structure of the Markov blanket permits us, for example, to calculate the probability of a purchase given the state of the attributes of the consumer. It also helps identify threshold values for certain attributes that will result in a change of purchase behavior. For example, if the annual income of a certain consumer increases, the probability of a purchase by that consumer also increases, with a corresponding decrease associated with a decrease in the consumer's income.

A "master agent," or "meta-agent," is also created (Figure 3) as part of the computer simulation model. This meta-agent interacts with the agent-based simulation model by obtaining information about the progress of the model in terms of the search for an optimal solution. At this point, the enhanced dynamic nature of data mining is applied. As the meta-agent obtains information about the different scenarios evaluated by the agent-based model, it updates the data. When the repository is updated in real time, the data can be analyzed by traditional data mining techniques to modify the rules that govern the behavior of consumer agents, in order to speed up the search for a better solution. The specific activities of the meta-agent are described in greater detail later in the example.

Agent-based simulation module

The simulation module of the application consists of an agent-based simulation model with two types of agents, consumers and companies, which are modeled as follows.

Consumer model

A group of consumers belonging to a particular segment of the market is modeled as an agent. As mentioned above, market segmentation is the result of the application of the data mining tools. Each type of agent is modeled according to four basic "rules" of purchasing behavior, closely following the approach suggested by Piana in [24]. We note that while other consumer behavior models exist that may be more sophisticated than Piana's, as in [29], it is not our intention to endorse one particular model over another, but merely to illustrate how our approach might aid the search for better solutions given one particular model. Wherever Piana's approach considers deterministic parameter values, we introduce a certain degree of uncertainty about the parameters. Our four basic rules of purchasing behavior are the following:

1. *Purchase decision rule:* We define a "reservation price," R_i, for each type of agent i. (Piana calls this reservation price the "maximum acceptable price.") The reservation price is the maximum price at which the consumer in a particular market segment will still consider buying a certain product or service. Whereas in Piana's approach the decision to buy is a variable with a value of 1 below the reserve price and a value of 0 above it, we introduce a stochastic variable y for which the probability decreases as the actual price C approaches the reservation price. Therefore, if we have an agent that represents consumers in market segment 1, we can denote p_1 as the purchasing decision for agent 1, and we can then express the probability that the agent will purchase the product as

   $$y = P(p_1 = 1) = \alpha(C - R_1),$$

   where α is a normalizing constant. Thus, we model the behavior of an agent as a binomial probability distribution with parameter y.
2. *Brand selection rule:* This rule relates to product differentiation. We define a *value proposition score* (VPS) that is a weighted average of various product (or service) attributes that consumers consider important in terms of value. Price is always included as one of the attributes. Depending on the market segment, some attributes are weighted more heavily and others less. For instance, for a high-tech product, a segment of "early adopters" might value quality and function more heavily than price, while a more conservative segment might weigh price and quality more heavily than function. We add a subscript k to our p variables defined in rule 1 in order to track the

sales volume for each brand. Thus, p_{ik} is a binary variable denoting whether agent i will purchase a product of brand k. As in rule 1, we use a probability of purchase related to the degree to which the attributes of the brand k product satisfy the agent in segment i.

3. *Divisibility of goods rule:* This rule relates to the nature of market demand. We assume that goods and services can be purchased only in discrete quantities. We then define a reservation price for each successive unit of the good, or commodity, so that the first unit purchased will have a certain reservation price, the second unit another, and so on. The reservation price may be the same for all units, or it may be different. Of course, the reservation price for the second unit is not available to the consumer unless a first purchase is made. In this way, special offers with discounts based on volume may be attractive for consumers whose second-unit reservation price is lower than the first-unit price, as long as the product can be stored for the appropriate length of time.
4. *Periodicity of demand rule:* This rule has to do with various consumer attributes that affect how often an agent is expected to buy a commodity. Brand loyalty, shopping patterns, and the nature of the commodity itself all affect this factor (see [24] for a more detailed discussion). However, we again make this rule more flexible by fitting a probability distribution to each type of consumer on the basis of the customer segment's purchase frequency average and standard deviation.

The aggregation of the agents' individual behavior according to these four basic rules can be translated directly into the performance of the various firms in the simulation model, as described in the following subsection.

Company model

We first model each participating firm in the market according to its product offering. We include the following product attributes: price, including discounts and special offers; quality; time; functionality; service; customer relationship management; and brand.

We also include an instance of our firm whose attributes will be the input variables used by the optimization engine in its search for an optimal allocation of capital. In other words, we assume that funds from investors will be allocated to activities that directly affect one or more of the product attributes mentioned above. For example, an investment in advertising potentially creates brand loyalty, while an investment in research and development may increase product functionality, and an investment in manufacturing and supply chain management may allow the firm to set a better (i.e., lower) price for the product. Given these characteristics of the product, the consumer agents operate according to their specific rules during the simulation by creating a certain demand for the product over a specified time horizon. By tracking the behavior of individual consumer agents, we are able to calculate aggregate performance measures for the firm. Thus, the consumer's purchase decision rule translates into a decision on whether or not to buy the product, and an expected market penetration at the aggregate level of the firm. For example, by aggregating the results of the individual agents, by summing the individual purchases of product a and product b, we can calculate the market penetration achieved by the firm that sells product a and the firm that sells product b, respectively.

Similarly, the brand selection rule at the consumer level translates into a measure of market share at the level of the firm. Finally, the combination of the divisibility of goods and the periodicity of demand rules at the individual consumer level translates into a measure of overall market demand for the product over the planning horizon.

Optimization module

The optimization engine interacts with the agent-based simulation module in the manner described in the section on optimizing computer simulation models. We define x_i as the portion of capital investment allocated to attribute i. As a reminder, these attributes usually relate to a product—for example, a cost, quality, or after-sales support level of a product. We also impose a lower and upper bound on each allocation, which results in constraints of the form

$$l_i \leq x_i \leq u_i,$$

for each attribute $i = 1, \cdots, n$. In addition, we add an overall budget constraint so that all funds allocated do not exceed the available capital investment amount b, as follows:

$$\sum_{i=1}^{n} x_i \leq b.$$

The objective of the optimization is to maximize the performance of the firm according to some prespecified performance measures. Examples of performance measures are market share, profit, sales revenue, net cash flow, and ratio of debt to equity. The objective can be expressed as a single performance measure, as a weighted combination of more than one measure, or as both.

For example, let us assume for now that the investors are interested primarily in maximizing profit; however, they also want to capture a certain percentage of market share, and they seek to minimize the total amount of

borrowed funds. For this case, we can formulate the optimization problem as follows:

Denote $E[P]$ as the *expected value* (*mean*) *of* P, and $E[M]$ as the *expected value* (*mean*) *of* M, where P is profit and M is market share. Also, denote d as the total amount of borrowed funds.

Objective:
Maximize $E[P] - d$
subject to the following constraints:

$$E[M] > m \qquad \text{(market share)};$$

$$\sum_{i=1}^{n} x_i \leq b + d \qquad \text{(available budget)};$$

$$l_i \leq x_i \leq u_i \quad \text{for } i = 1, \cdots, n \qquad \text{(bounds on } x_i\text{)};$$

$$d \geq 0 \qquad \text{(lower bound on } d\text{)}.$$

The objective in this example is to maximize expected profits while minimizing the amount of borrowed funds. This is a complex objective function because it implies a tradeoff between the first term ($E[P]$) and the second term (d), and thus we use a negative sign in front of d. We desire to maximize $E[P]$ and minimize d. Note that it is not necessary to include a constraint that imposes an upper bound on d, because net profit P is indirectly proportional to d owing to the interest expense generated by debt. Note, however, that we can compose a more complex objective that is a weighted sum of various performance measures of interest, as in a balanced scorecard (BSC). For simplicity, we continue the discussion by focusing on the objective function discussed in the mathematical formulation above.

In order to evaluate $E[P]$ and $E[M]$ for a given allocation x_i of funds, we need to conduct many simulation trials. Then, the evaluation is transmitted to the optimization engine which, based on its search algorithm, in turn suggests another "solution" (i.e., allocation of funds) which is sent back to the simulation module for evaluation. The following is a sequence of steps to evaluate the merit of a solution:

1. Determine the allocation of investment amounts in each attribute x_i and re-express this amount as a budget allocation in each division or activity, such as ordering, production, sales, research and development, logistics, services, market research, and advertising.
2. Determine sales as a function of market demand, taking into consideration volume discounts and other special promotions planned for the product.
3. Determine product costs (cost of goods sold) in terms of such considerations as logistics, materials, and sales commissions.
4. Determine projected cash flow entries, such as interest expense, depreciation and amortization, and overhead expenses.
5. Given the allocation of funds from step 1, conduct enough trials of the simulation for a planning horizon of t periods to obtain a probability distribution of market penetration, demand, and expected sales.
6. From step 5, obtain the expected value of various performance measures, such as net profit, market share, net cash flow, debt, and return on invested capital.
7. Construct the corresponding financial statements for each period in the planning horizon, using the expected values or using other measures such as percentiles or confidence intervals as desired.
8. Construct a balanced score card (BSC) with the above measures to gauge overall company performance.

As mentioned above, this same process can be performed for various combinations of objectives in order to find a set of solutions that seem attractive. Then, on the basis of an overall BSC score, the solutions can be ranked and selected.

During the process described above, we take advantage of the information generated as more and more solutions are evaluated. Our dynamic meta-agent collects information about the characteristics shared by "good" solutions (i.e., solutions with a high objective value). Using these characteristics, the meta-agent updates the data repository. After a certain number of iterations, the DDM module is activated again, and the rules of behavior that govern the agents are changed. Also, in order to speed up the search, the meta-agent "filters out" solutions that have a high likelihood of being "bad," so that the filtered solutions are not subjected to evaluation by the simulation model. This is achieved by applying enhanced mixed-integer programming (MIP) classification methods to the different solutions generated, so that the set of separating hyperplanes that results from solving the MIP can be incorporated as additional constraints to the model during future iterations.

Conclusions and future research

In many computer simulation applications in industry, very little attention is given to the relationship among the input variables and their effect on the performance of the system that is being modeled. Limited effort is devoted

to assessing which inputs have the biggest effect on performance. The search for an optimal configuration of the system involves extensive trial and error, which is expensive and time-consuming. As an alternative, we propose a practical approach that includes a dynamic data mining module to identify the relevant inputs and discover the nature of their relationships to the performance of the system. The dynamic data mining model also makes use of information learned during the optimization process to separate good-quality solutions from bad, so that only promising solutions need to be evaluated during future iterations. Underlying and supporting this module is an optimization engine that makes use of state-of-the-art algorithms to aid in the data mining and to ultimately guide the search for optimal configurations for the simulation model.

A wide range of applications can benefit from the proposed approach, including business process management, portfolio management, project life-cycle management, health care, prevention and control of epidemics, bioterrorism detection and control, vaccination benefits assessment, clinical trial simulations, and numerous applications in the social sciences, physical sciences, and materials sciences.

Through a detailed market research example, we have shown how our approach can be used to find the best scenario with respect to a user's desired performance measures. A detailed illustration of how this is achieved is provided by a market research example utilizing agent-based simulation at both the consumer level and the company level. We envision that simulation optimization will find a growing and increasingly fertile field of application in data mining by means of this approach, drawing on the ability of our dynamic data mining approach to simulation optimization that is used to represent and respond to complex relationships in ways that cannot be achieved by alternative approaches.

**Trademark, service mark, or registered trademark of OptTek Systems, Inc. in the United States, other countries, or both.

References

1. D. J. Hand, H. Mannila, and P. Smyth, *Principles of Data Mining*, The MIT Press, Cambridge, MA, 2001.
2. M. Craven and J. Shavlik, "Using Neural Networks for Data Mining," *Future Generation Computer Syst*. (special issue on data mining) **13**, No. 2/3, 211–229 (1997).
3. N. Christiani and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*, Cambridge University Press, Cambridge, U.K., 2000.
4. D. Heckerman, "A Tutorial on Learning with Bayesian Networks," *Technical Report MSR TR-95–06*, Microsoft Research, 1995; see *http://research.microsoft.com/research/pubs/view.aspx?msr_tr_id=MSR-TR-95-06*.
5. X. Bai and R. Padman, "Tabu Search Enhanced Markov Blanket Classifier for High Dimensional Data Sets," *Proceedings of the 9th INFORMS Computing Society*, Springer, New York, 2004, pp. 338–354.
6. M. Better, F. Glover, and M. Samorani, "Multi-Hyperplane Formulations for Classification and Discrimination Analysis" (working paper), University of Colorado, Boulder, CO, 2006; see *http://www.opttek.com/white.html*.
7. F. Glover, "Improved Classification and Discrimination by Successive Hyperplane and Multi-Hyperplane Separation" (working paper), University of Colorado, Boulder, CO, 2006; see *http://www.opttek.com/white.html*.
8. M. Better, F. Glover, G. Kochenberger, and H. Wang, "A Novel Approach to Classification in Financial Applications," *Proceedings of the AI/DM Workshop of the INFORMS Annual Meeting*, 2006; see *http://ieweb.uta.edu/vchen/AIDM/AIDM-Better.pdf*.
9. L. J. Heyer, S. Kruglyak, and S. Yooseph, "Exploring Expression Data: Identification and Analysis of Co-Expressed Genes," *Genome Res*. **9**, No. 11, 1106–1115 (1999).
10. F. J. Rohlf, "Hierarchical Clustering Using the Minimum Spanning Tree," *Computer J*. **6**, No. 1, 93–95 (1973).
11. M. Ng, "A Parallel Tabu Search Heuristic for Clustering Data Sets," presented at the International Conference on Parallel Processing Workshops (ICPPW'03), Kaohsiung, Taiwan, 2003.
12. G. Kochenberger, F. Glover, B. Alidaee, and H. Wang, "Clustering of Microarray Data via Clique Partitioning," *J. Combinatorial Optimization* **10**, No. 1, 77–92 (2005).
13. M. Barnett, "Modeling and Simulation in Business Process Management," *BP Trends Newsletter, White Papers & Technical Briefs*, pp. 1–10; see *http://www.bptrends.com*.
14. V. Campos, F. Glover, M. Laguna, and R. Martí, "An Experimental Evaluation of a Scatter Search for the Linear Ordering Problem" (working paper), University of Colorado, Boulder, CO, 1999; available from authors.
15. V. Campos, M. Laguna, and R. Martí, "Scatter Search for the Linear Ordering Problem," *New Methods in Optimization*, D. Corne, M. Dorigo, and F. Glover, Editors, McGraw-Hill, New York, 1999, pp. 331–339.
16. F. Glover, "A Template for Scatter Search and Path Relinking," *Artificial Evolution, Lecture Notes in Computer Science* **1363**, J.-K. Hao, E. Lutton, E. Ronald, M. Schoenauer, and D. Snyers, Editors, Springer-Verlag, New York, 1998, pp. 13–54.
17. F. Glover and M. Laguna, *Tabu Search*, Kluwer Academic Publishers, New York, 1997.
18. F. Glover, M. Laguna, and R. Marti, "Fundamentals of Scatter Search and Path Relinking," *Control and Cybernet*. **29**, No. 3, 653–684 (2000).
19. F. Glover, M. Laguna, and R. Marti, *Scatter Search, Advances in Evolutionary Computing: Theory and Applications*, Springer-Verlag, New York, 2003, pp. 519–537.
20. M. Laguna, "Scatter Search," *Handbook of Applied Optimization*, P. M. Pardalos and M. G. C. Resende, Editors, Oxford University Press, New York, 2002.
21. OptTek Systems, Inc., *Optquest Engine Manual* (available online); see *http://www.OptTek.com*.
22. J. April, F. Glover, and J. P. Kelly, "OptFolio®—A Simulation Optimization System for Project Portfolio Planning," *Proceedings of the 2003 Winter Simulation Conference*, S. Chick, P. J. Sánchez, D. Ferrin, and D. J. Morrice, Editors, New Orleans, LA, 2003; see *http://www.opttek.com/publications/wsc03Final-kellyj85794i.pdf*.
23. J. April, "OptForce™: New Human Resource Optimization Methods," presented at the INFORMS Conference on O.R. Practice, Miami, FL, April 30–May 2, 2006.
24. V. Piana, "Consumer Decision Rules for Agent-Based Models," Economics Web Institute, 2004; see *www.economicswebinstitute.org/essays/consumers.htm*.
25. Y. Fu, R. Piplani, R. de Souza, and J. Wu, "Multi-Agent Enabled Modeling and Simulation Towards Collaborative Inventory Management in Supply Chains," *Proceedings of the 2000 Winter Simulation Conference*, J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick, Editors, 2000, pp. 1763–1771.

26. P. Bilkstein and U. Wilenski, "MaterialSim: An Agent-Based Simulation Toolkit for Learning Materials Science," presented at the International Conference on Engineering Education, University of Florida, Gainesville, FL, 2004.
27. C. Langton, *Artificial Life: An Overview*, MIT Press, Cambridge, MA, 1995.
28. D. A. Samuelson and C. M. Macal, "Agent-Based Simulation Comes of Age," *ORMS Today* **33**, No. 4, 34–38 (2006).
29. K. J. Lancaster, "A New Approach to Consumer Theory," *J. Political Econ.* **74**, No. 2, 132–157 (1996).

Received September 12, 2006; accepted for publication October 14, 2006; Internet publication May 23, 2007

Marco Better *OptTek Systems, Inc., 1919 Seventh Street, Boulder, Colorado 80302 (better@OptTek.com)*. Mr. Better is a Research Associate of OptTek Systems, Inc. He holds a B.S. degree in industrial engineering from Pennsylvania State University and an M.B.A degree from the University of Colorado, and is a doctoral candidate at the Leeds School of Business of the University of Colorado at Boulder. Mr. Better has more than 12 years of professional work experience in the automobile, banking, and telecommunications industries.

Fred Glover *OptTek Systems, Inc., 1919 Seventh Street, Boulder, Colorado 80302 (glover@OptTek.com)*. Dr. Glover is the Chief Technology Officer of OptTek Systems, Inc. and is in charge of algorithmic design and strategic planning initiatives. He is a leading figure in the field of metaheuristics, a name he coined in the 1980s—an area that is now the subject of numerous books and international conferences, focusing on the development of models and methods enabling the solution of complex nonlinear and combinatorial problems that lie beyond the ability of classical optimization procedures. Dr. Glover also served as the MediaOne Chaired Professor in Systems Science at the University of Colorado, Boulder, where he holds the title of Distinguished Professor of the University of Colorado System. He has authored or co-authored more than three hundred fifty published articles and eight books in the fields of mathematical optimization, computer science, and artificial intelligence, with particular emphasis on practical applications in industry and government. Dr. Glover is the recipient of the distinguished von Neumann Theory Prize, is an elected member of the National Academy of Engineering, and has received numerous other awards and honorary fellowships, including those from the American Association for the Advancement of Science (AAAS), the NATO Division of Scientific Affairs, the Institute of Operations Research and Management Science (INFORMS), the Decision Sciences Institute (DSI), the U.S. Defense Communications Agency (DCA), the Energy Research Institute (ERI), the American Assembly of Collegiate Schools of Business (AACSB), Alpha Iota Delta, and the Miller Institute for Basic Research in Science.

Manuel Laguna *OptTek Systems, Inc., 1919 Seventh Street, Boulder, Colorado 80302 (laguna@OptTek.com)*. Dr. Laguna is the Vice President of Research of OptTek Systems, Inc. He is a Professor of Operations Management in the Leeds School of Business of the University of Colorado at Boulder. He received master's and doctoral degrees in operations research and industrial engineering from the University of Texas at Austin. Dr. Laguna has conducted extensive research on the interface between computer science, artificial intelligence, and operations research to develop solution methods for problems in areas such as logistics and supply chain, routing and network design in telecommunications, combinatorial optimization on graphs, and optimization of simulations. His research has appeared in numerous academic journal articles and books. He is the editor-in-chief of the *Journal of Heuristics* and a member of the international advisory board of the *Journal of the Operational Research Society*.