

A FRAMEWORK FOR THE OPTIMIZATION AND ANALYSIS OF AGENT-BASED MODELS

Benjamin Thengvall
Fred Glover

OptTek Systems, Inc
1919 Seventh Street
Boulder, CO, 80302, USA

ABSTRACT

OptAgent is a new framework which enhances the practical applications of agent-based models by giving them an optimization and analysis capability. Embedding such an ability within agent-based systems greatly expands their usefulness and applicability. Developed through an Army Research Office SBIR Phase II grant, OptAgent is built on our existing OptQuest engine to create a system that can be used alike with agent-based and other complex simulation models. Important components of the OptAgent system include new predictors based on Markov blankets and satisfiability data mining that speed the search for high quality solutions. OptAgent is structured to assist with the verification, validation, and accreditation process for a new model and can also be used to enhance the analysis accomplished with a more mature model.

1 INTRODUCTION

With increases in computing power in the last decade, agent-based models have become increasingly common and useful. Currently these models are being used to analyze a wide variety of military, government, and commercial problems. Agent-based simulation, a critical function of these models, provides a powerful tool to evaluate the behavior of systems that are too complex to be given tractable mathematical formulations. Recent applications of agent-based models in the commercial realm include modeling of financial markets and actors, supply chain and logistics; marketing and consumer behavior, social networks and information flow, distributed computing, diversity and manpower planning, and traffic management.

For the military, agent-based modeling represents the latest in analytic simulation technology and provides a simulation environment in which small- to large-scale joint war fighting scenarios are being constructed and explored. Agent based modeling tools are also being used to represent military operations other than war and irregular warfare. To see these models in the commercial or the defense arena reach their full potential, an integrated software environment that intelligently couples optimization and analysis with these agent-based models is needed.

Despite their widespread and growing use, agent-based simulations face the criticism that they have too many parameters, and that the impact of varying these parameters is not well understood. The complex interacting effects among these parameters renders customary analysis incapable of determining whether the parameters values selected are best. There is no guarantee that other parameter choices will not yield better performance.

Another commonly noted weakness in the use of agent-based simulations is a lack of understanding of the scope and sensitivities of model outputs for varying parameter choices. One would like to know the best- and worst-case behavior of the model, given different parameter choices. Similarly, it would be valuable to know if there are ranges of parameter values for which model outcomes remain above or below certain thresholds.

A user wrestling with such issues is compelled to rely on repeated what-if analysis or perhaps "expert judgment." A complex simulation of the type required to model real-world applications, however, will typically have myriad potential combinations of agent and environment parameters. The probability that a user iteratively manipulating the simulation will find optimal inputs is effectively zero. Large agent-based simulations are generally computationally intensive. Thus, enumerating all combinations of agents and environment parameters is not a practical strategy for such models.

A preferable alternative is to wrap an optimization algorithm around an agent-based model to search for high-quality parameter settings, offering the potential to obtain such parameter settings with a limited number of runs. Classical mathematical programming techniques such as linear and integer programming cannot be applied to optimize the settings for agent-

based models, which are highly nonlinear and stochastic. Instead, an intelligent search metaheuristic, to iteratively adjust input parameters and evaluate model responses, has been shown to be an effective approach to optimizing simulation models.

Integrating a generalized framework for optimization and robust analysis with agent-based simulations has the potential to offer far greater than the current mode of operation where a user manually manipulates the agent-based simulation. Metaheuristic-based stochastic optimization libraries can overcome this limitation of current systems by answering the question of whether the best combination of parameters was chosen to meet certain goals, and more particularly by identifying such a combination as a foundation for better decisions. The inclusion of an intelligent analysis component has the additional ability to provide statistics describing model behavior and to give decision makers a previously unmatched capacity for sensitivity analysis.

2 OPTAGENT OVERVIEW

OptAgent is a framework that wraps around agent-based models to provide support for optimization, analysis, and distributed computing. OptAgent is comprised of a series of additions and enhancements to the existing OptQuest simulation optimization software developed by OptTek Systems. Figure 1 depicts the three-stage process that is used to optimize and analyze agent-based models.

The OptQuest engine combines state-of-the-art metaheuristic procedures including Tabu Search, Neural Networks, and Scatter Search with classical mathematical programming techniques such as Linear and Integer Programming. This composite methodology chooses a set of values for the model input parameters (i.e., factors or decision variables) and uses the responses generated by the simulation model to make decisions regarding the selection of the next trial solution.

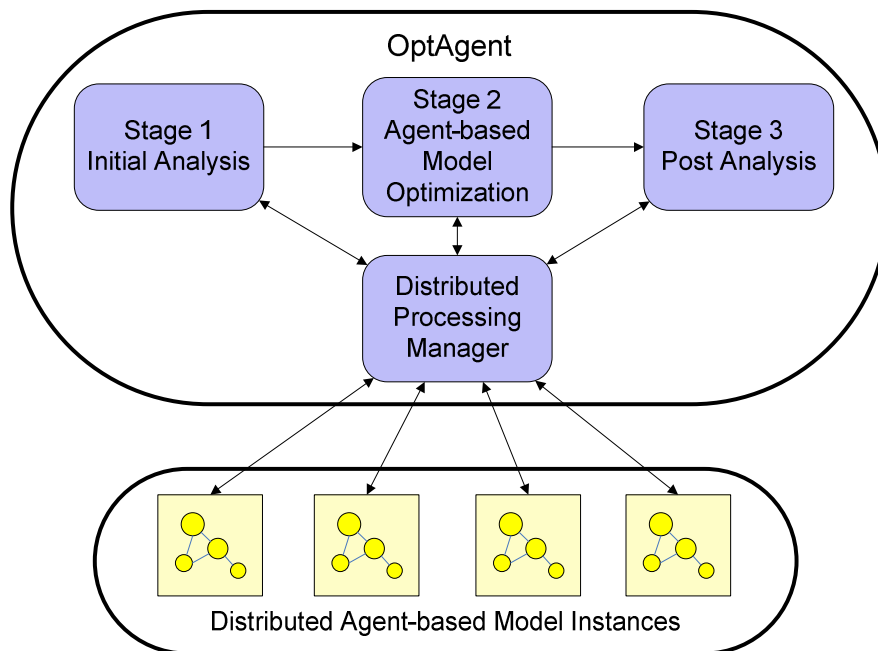


Figure 1. Conceptual View of OptAgent and a Distributed Set of Agent-based Model Instances.

Prior to Stage 1, the user identifies the agent and environment attributes from the agent-based model whose values the user wishes OptAgent to determine. These attributes are the decision variables over which OptAgent optimizes and performs model analysis. Decision variables may be continuous, discrete, integer, binary, or enumerations. The user also identifies outputs of the agent-based model to collect following each model run.

The user specifies an objective in terms of decision variables and outputs of the agent-based model. The objective may be a linear or nonlinear combination of decision variables and model outputs. The objective can also be a user-defined evaluation function of arbitrary form that takes the decision variables and model outputs and returns an objective value.

The user optionally sets constraints involving decision variables and model outputs. The constraints may be a linear or nonlinear combination of decision variables and model outputs or contain user-defined functions of arbitrary form. If a constraint is defined only in terms of the decision variables, variable values sent to the simulation will adhere to the constraint.

For constraints defined in terms of model outputs, the constraint will be tested for feasibility once the model outputs are returned.

For distributed optimization, the required software is set up on each node participating with one node selected as the optimization manager and the others as agent-based model simulators. Each simulator node registers with the optimization manager. If distributed computing is selected, each agent-based model run in each stage will be sent out as a job through the optimization manager to one of the available simulators.

2.1 Stage 1: Initial Analysis

OptAgent runs the agent-based model many times initially creating a diverse initial set of solutions. Statistics are collected on this initial solution set and displayed to the user, such as the range of objective values found in the initial set of solutions.

Using the initial solution set, a collection of Markov blanket and Satisfiability Data Mining solution predictors are created. These predictors are used to evaluate the expected results of model runs based on the proposed variable values. As shown in Figure 2, by using the collection of predictors, the search can be strictly limit to runs with good expected results or the search can be skewed toward searching more frequently in areas with good expected results.

Between Stage 1 and Stage 2, the user is presented with the results of Stage 1 and will have the option to deselect variables from consideration or to cancel the optimization process altogether.

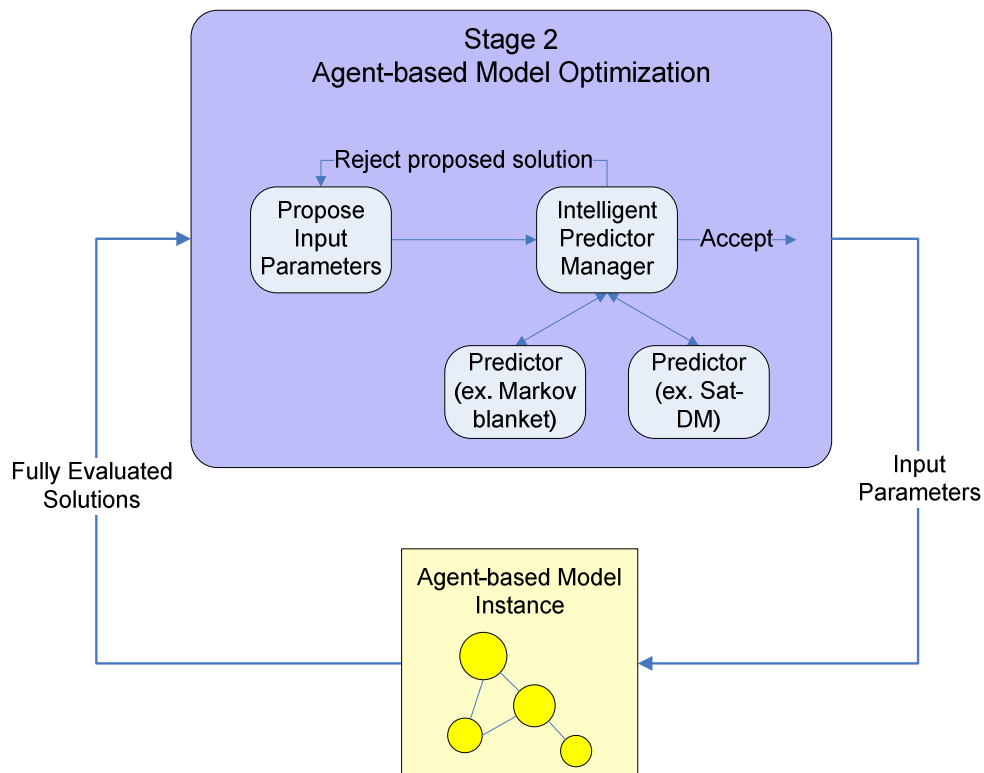


Figure 2. Stage 2 Interaction between OptAgent and the Agent-based Model.

2.2 Stage 2: Agent-based Model Optimization

OptAgent uses the existing OptQuest software to perform the core agent-based model optimization search process. The OptQuest software is also used for solution management. Figure 2 shows the interaction between the OptQuest metaheuristic search engine and an agent-based model. The metaheuristic optimizer chooses a set of values for the input parameters (i.e., factors or decision variables) and uses the responses generated by the simulation model to make decisions regarding the selection of the next trial solution.

2.3 Stage 3: Post Analysis

During the post analysis stage, statistics are collected on the key performance measures in the agent-based model across the various solutions found. Statistics of interest, such as maxima, minima, means, quartiles, and standard deviations are displayed. These statistics provide the user insight into the behavior of the model across many possible combinations of inputs. The user is able to tailor reported statistics to provide specific information on questions of interest. For example, what are the values of the statistics when a parameter has a certain value or has a certain range of values. If an insufficient number of solutions have been collected, more solutions may be analyzed during this stage to provide the requested statistics.

Sensitivity analysis is also performed during Stage 3, providing ranges of parameter values that provide solutions with objective values in a certain range, e.g., ranges of parameter values that all provide solutions with an objective value within 10% of the best objective found.

3 A MISSILE DEFENSE EXAMPLE

As a test case we constructed an agent-based model representing a missile defense scenario. This model was chosen for its relevance to the Army and for its ability to explore many agent and environment parameters that are representative of those commonly found in many agent-based models. The model included a two-dimensional physical landscape on which the agents interact. The agent types represented in the model are blue leadership, blue airborne lasers (ABLs), blue ground-based interceptors (GBIs), blue GBI stations, blue Patriots, blue cities, and blue bases, red leadership, red transport-erect-launchers (TELS), and red missiles. We used the Repast repast.sourceforge.net agent-based modeling toolkit for building the missile defense example.

For our primary tests we fixed the battle space size and the initial location and number of red TELS and red missiles. We created variables representing the number of blue ABLs at each blue base, the number of GBIs at each GBI station, and the number of Patriots at each blue city. Constraints were included that limit the total number of ABLs, the total number of GBIs, and the total number of Patriots. The overall model objective was to maximize the final population in the blue cities.

During the execution of the agent-based model, the red leadership sends launch commands to the red TELS. Each red missile launched is targeted at a blue city or base. Blue ABLs move from their home base to patrol and back. As red missiles pass within range, intercept attempts are made by ABLs, GBIs, and Patriots. Missiles that are not intercepted and strike a blue base degrade the base's ability to service ABLs and thus degrade the ABLs' effectiveness. Missiles that are not intercepted and strike a blue city result in population loss which is then reflected in the model objective function.

This model demonstrates the technical feasibility of optimizing the input parameters for an agent-based model. By adjusting the battle space size and the initial location and number of red TELS and red missiles, adjusting the numbers of ABLs, GBIs, and Patriots at each location, and adjusting the constraint limits on total numbers of ABLs, GBIs, and Patriots, we showed our optimization process for agent-based models to be effective for the model across a variety of model settings.

Our base model had 20 input variables and three constraints on subsets of those variables. We ran the model with up to 100 agents of 9 base agent types operating for up to 100 time units. The number of agents and the length of the simulation could have both been increased significantly. We intentionally limited the number of agents and the number of time steps to keep model run times short and allow for a large number of trials during testing.

Wrapping optimization around this missile defense agent-based model using OptAgent allowed us to answer questions of the following forms:

- What is the best mix of assets to acquire under various budget, production, or theater constraints?
- How can assets be best positioned to most effectively address the red missile threat?
- What are the best tactics to employ during the fight?

Red war gaming could also be done to determine blue vulnerabilities answering questions such as:

- Given a blue defense, what timing of missile launches produces the greatest blue loss?
- Given a blue defense, what assignment of missiles to cities and bases inflicts the greatest loss?

While this model does not include all of the agents and agent interactions that may potentially be embraced in a missile defense model, it captures the essentials of a missile defense fight in a realistic fashion. This agent-based model can easily be extended to include a greater number of agents, a greater variety of agents, more complex agent interactions, and far more time units. Our results imply that the optimization techniques used in our research would apply well to extensions of this model. Our experience in other practical settings reinforces the conclusion that these outcomes apply generally to many other agent-based models as well as other forms of simulation modeling.

4 SOLUTION PREDICTORS

Through the use of predictors the search space can be effectively narrowed by limiting full agent-based model runs to proposed input parameters with good predicted results. In OptAgent, we have extended the core OptQuest engine to utilize both Markov blanket (MB) and Satisfiability Data Mining (Sat-DM) solution predictors during Stage 2 of the optimization process to speed the search and improve the results.

The MB solution predictor is based on the research of Better (2007). The creation of the MB takes place in three phases: discovery, construction, and improvement. In the discovery phase, all variables are tested for independence and conditional independence to a specified depth. This creates the set of variables from the model that are the primary influencers of the objective value. These variables of primary influence become the candidate nodes from which the MB is formed. Candidate variables are then used in the construction phase and are assembled into a directed acyclic graph including all of the variables as either parents of the objective node, children of the objective node, or parents of children of the objective node. This directed acyclic graph is an MB in which each node represents a variable (including a variable representing the objective) and each directed edge expresses a relationship between the variables. In the improvement phase, node/edge positions in the directed acyclic graph are varied in an attempt to create an MB with greater prediction accuracy. The prediction accuracy of the MB is evaluated by testing it against the solutions contained in the initial sample set of solutions.

An MB is a special case of a Bayesian network (BN). A BN is a directed acyclic graph G , consisting of a set of nodes or vertices V that represent the features (i.e. decision variables) in the data set, and directed edges, E , between the nodes, representing conditional independence relationships between the variables. Given a graph $G = \{V, E\}$, a node $X \in V$ is a *cause* of node $Y \in V$ if there is a directed edge from X to Y . Equivalently, X is a *parent* of Y , and Y is a *child* of X . The *Markov condition* imposed on a BN renders a variable $X_i \in V$ independent of any non-descendant or non-parent, conditional on its parents. The MB of X_i includes its parents, its children and the parents of its children. The MB, 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 MB of the dependent (target) variable is considered redundant for the purpose of predicting the value of the dependent variable. Thus, by finding the MB of the variable of interest, we can discard all the other, irrelevant variables in the domain.

Data binarization is not required for the construction of MBs, but it is often desirable. One primary reason is that the compression – both in the number of attributes and in the amount of information – that results from data binarization can greatly reduce the computational cost of creating MBs. Recent research indicates that binarizing data can result in comparable and often better classification accuracy than using the original raw data. We perform a binarization process against the agent-based model inputs we are considering. A modified version of the IDEAL algorithm developed by Moreira et al., (1999) is used called Controlled Discriminant Selection.

The MB provides the user of the agent-based model insight into which variables are the primary drivers of the model. This may lead to more care in modeling the agents related to these variables. This may also lead the agent-based model creator to remove or rethink some of the variables that are not of primary influence to the model outcome. The MB can be used to predict whether potential solutions will fall above or below a desired objective value before they are run through the agent-based model. Potential solutions that are predicted to yield poor solutions can be discarded or placed in a list of predicted bad solutions that will only be evaluated if there is time after evaluating all possible predicted good solutions.

We also implemented a Satisfiability Data Mining (Sat-DM) solution predictor based on the research of Glover (2007). Sat-DM is a method for binary data classification in which a collection of logical clauses is generated, or equivalently a collection of inequalities in zero-one variables, for each group of points representing a given classification. A point with unknown membership is classified as belonging to a particular group based on comparing the number or proportion of the inequalities it satisfies for that group versus the number or proportion it satisfies for other groups. Sat-DM makes use of a fundamental observation which states that inequalities are satisfied by a subset of elements of a particular group (and correspondingly violated by a subset of elements from a complementary group) if and only if these inequalities correspond to feasible solutions to a special variant of a satisfiability problem. Based on this, the method generates membership-defining systems of inequalities that provide a filter for segregating points lying in different groups.

Satisfiability data mining may be viewed as a procedure for generating multiple hyperplanes that segregate points of different groups by isolating their logical properties. The inequalities produced capture classification regions in feature space that are more varied and complex than those derived from hyperplane separating procedures such as those used in support vector machines and related procedures based on linear programming and convex analysis. A particularly useful feature is the ability to generate the collections of segregating inequalities (complementary half-spaces) in a highly efficient manner, allowing the approach to handle large data sets without difficulty.

Creating the Sat-DM predictors works as follows: let a_{ij} denote the value of a decision variable for the agent-based model, where each solution i ($i=1, \dots, m$) is described by n different decision variables indexed by j ($j=1, \dots, n$). We seek a rule to

classify these elements in a manner to identify correctly whether a given vector $A_i = (a_{i1}, \dots, a_{im})$ should belong among the elements of Group 1 or instead among those of Group 2 (denoted G_1 and G_2 , respectively). In our case, we will seek to classify possible combinations of decision variables according to whether they will yield a solution with an objective value above a certain threshold ($i \in G_1$) or not ($i \in G_2$).

From our initial diverse solution set generated in Stage 1 we have the A_i vectors and their group membership for any chosen objective threshold. The goal is to provide decision rules that perform well in discriminating whether a new vector A not among the original known vectors should be classified as belonging to G_1 or G_2 .

Test runs were executed with and without the new solution predictors. For each configuration runs were completed using four different initial sample sizes. For each configuration and sample size a number of runs were conducted varying the random seed for each run and average results were computed. Results for the test runs are shown in Table 1.

Each run consisted of 1500 iterations of the agent-based model and results of the test runs are summarized in Table 1 below. The summary base case results, along with results for runs using only MB solution predictors, runs using only Sat-DM solution predictors, and runs using both MB and Sat-DM solution predictors are shown in Table 1. The percentages displayed in the table represent the gap between the best known solution and the best solution obtained for each set of test runs. Four columns contain the results for each of four initial sample sizes, 25, 50, 100, and 150. The final column shows the average of the results across the four sample sizes.

Table 1. Summary Results Comparing Base Case Results with Results Obtained Using the Markov Blanket and Satisfiability Data Mining Solution Predictors.

	Initial Sample 25	Initial Sample 50	Initial Sample 100	Initial Sample 150	Overall % From Best Known
Base Case / No Predictors	3.3%	5.8%	3.3%	4.8%	4.3%
Markov blanket	3.0%	4.9%	2.8%	3.4%	3.5%
Sat-DM	3.2%	3.3%	3.2%	4.6%	3.6%
Markov blanket & Sat-DM	3.1%	3.4%	2.7%	3.0%	3.1%

Comparing the test runs in which MB solution predictors were used with the base case, we see that the addition of these predictors improved the average performance for each initial sample size. The overall average gap between the best solution found and the best known solution is reduced 0.8%. The tests in which Sat-DM solution predictors were used yielded solid improvements over the base case. The reduction in the overall average gap between the best solution found and the best known solution is 0.7%.

The results achieved when using both the MB and the Sat-DM solution predictors together yield the best results of any of our configurations. The average gap between the best solution found and the best known solution is better than the base case result for each sample size. A comparison of the base case row and the final row in Table 1 shows the average gap between the best solution found and the best known solution was reduced by 1.2%. For the base case, the average gap was 4.3%, and when using both types of solution predictors, it is 3.1%.

There is not a clear pattern when comparing the initial sample size used and the quality of the best solution found.

The results of the runs in which the solution predictors were used to discard predicted bad solutions demonstrate the value of these new solution predictors and their ability to produce higher quality solutions than are obtained without their use. In each test, the use of solution predictors improved overall performance and the greatest gain was seen when both the blanket and the Sat-DM solution predictors were used in combination.

5 USE DURING THE MODEL VERIFICATION, VALIDATION, AND ACCREDITATION PROCESS

The Verification, Validation and Accreditation process generally involves both data verification and validation and model verification and validation. The Department of Defense’s Modeling and Simulation Glossary offers the following definitions. Verification is the process of determining that a model implementation and its associated data accurately represent the developer’s conceptual description and specifications. Validation is the process of determining the degree to which a model and its associated data are an accurate representation of the real world from the perspective of the intended uses of the model. Using the OptAgent framework with an agent-based model helps with both the data and model verification and validation.

During the course of an optimization run OptAgent generates a large number of trial solutions. In the post analysis phase a user can explore the range of solutions generated. While the search process focuses intensely on regions of local maxima the search algorithms also include diversification processes such that a wide variety of possible trial solutions are generated during a run. Subject matter experts (SME) can often provide reasonable ranges for model outputs. Examining solutions generated across the range of input variables and at solutions with very good and very bad objective values one can quickly identify areas in which the model does not perform in accordance with the expectations of the SMEs.

When a solution is found where model outputs do not fall within expected ranges the model performance can be examined to determine if the deviation is due to model assumptions or implementation or due to underlying data. The cases where deviations exist between model outputs and expected ranges, as disclosed by the analysis software, forms a basis for further discussions with the SMEs to determine next steps.

The ability to specify variables of interest, constraints, and an objective for an optimization run are also very useful during verification and validation. Controlling these inputs allows the user to test out different portions of a model's solution space with great specificity.

6 CONCLUSIONS AND FUTURE WORK

Our computational study demonstrates that OptAgent provides an optimization framework for agent-based models that yields an effective way to find high quality solutions and to carry out effective model analysis. Important additional applications of the OptAgent software stem from its ability to assist with the verification, validation, and accreditation process for a new agent-based model. Useful forms of analysis that can be done by adding optimization to an agent-based model are further illustrated by the inclusion of an agent-based missile defense model.

Our study also discloses the advantages obtained from Markov blanket and satisfiability data mining solution predictors to speed the search process and find better solutions in the search. We are currently investigating the best ways to measure and track the accuracy of the predictors, when to initially train and then retrain the predictors, and how best to use multiple predictors in combination. Research now underway focuses on creating a number of prototype instances where the OptAgent framework is applied both to existing agent-based and other complex simulation models. Discussions are proceeding with parties in the Department of Defense to use OptAgent with models for irregular military operations and for air defense simulation. An additional initiative consists of exploring integration opportunities to partner with makers of agent-based modeling toolkits. Such integration at the toolkit level will make the functionality embedded in OptAgent available to anyone building an agent-based model using one of these toolkits.

REFERENCES

- Better, M. (2007) "Data Mining Techniques for Prediction and Classification in Discrete Data Applications," *doctoral dissertation*, University of Colorado at Boulder.
- Moreira, M., A. Hertz and E. Mayoraz (1999) "Data Binarization by Discriminant Elimination," *IDIAP Research Report 99-04*, Dalle Molle Institute of Perceptual Artificial Intelligence, Lausanne, Switzerland.
- Glover, F. (2007) "Satisfiability Data Mining," (working paper), OptTek Systems, Inc., Boulder, CO.
- Glover, F. and M. Better (2007) "Improved Classification and Discrimination by Successive Hyperplane and Multi-Hyperplane Separation," (working paper), University of Colorado, Boulder.

AUTHOR BIOGRAPHIES

BENJAMIN THENGVALL is an Optimization Architect of OptTek Systems, Inc. He earned his M.S.E. and Ph.D. in Operations Research and Industrial Engineering from the University of Texas at Austin. Prior to joining OptTek, he was an Operations Research Scientist at Navitaire, Inc., a leading operations research technology firm serving the airline industry. His primary interests have been in mathematical modeling and scheduling. He manages the activities in OptTek's office in Omaha, Nebraska providing support for USSTRATCOM. His email is <thengvall@opttek.com>.

FRED GLOVER is the Chief Technology Officer in charge of algorithmic design and strategic planning initiatives for OptTek Systems, Inc. He also holds the title of Distinguished Professor of the University of Colorado System. Dr. Glover is a leading figure in the field of metaheuristics, whose 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

Thengvall and Glover

nonlinear and combinatorial problems that lie beyond the ability of classical optimization procedures. He has authored or co-authored more than three hundred seventy 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, 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. His email is glover@opttek.com.