Inducing a Cognitive Model from Examples Provided by an Optimal Algorithm

Bradley J. Best
Micro Analysis & Design
4949 Pearl East Circle
Boulder, CO 80304

Marsha Lovett
Cyert Hall, Suite 120
5000 Forbes Avenue
Pittsburgh PA 15213

Abstract

We have developed a methodology for using a normatively correct task model as the basis for a training methodology and for the provision of performance feedback within a national missile defense (NMD) simulated task environment (STE). This methodology has allowed us to explore: 1) the relative impact of expert versus optimal feedback, 2) the differences in task performance among an optimal model, a cognitively plausible rule-based model, and an instance-based model of the NMD task, and 3) the effectiveness of methodologies for constructing cognitive models directly from expert performance data..

Introduction

Successful training in complex environments is normally accomplished through the interaction of a trainee and a skilled expert, but experts are an expensive commodity. Using an optimal model of task performance subject to human constraints may be a more efficient way to develop models of skilled human performance for use in training, especially since optimal models are often simpler to validate, test, and debug than corresponding expert models. In addition, optimal models constrained by human plausibility can be constructed in domains where no experts are available or even exist. Cognitive architectures, such as ACT-R (Anderson & Lebiere, 1998), encapsulate human processing constraints and can be used as a framework for constructing *constrained optimal models*. Combining such a model with a simulated task environment (STE) permits close model-trainee interaction because the model can interact with the same interface as the trainee, allowing a model-based tutor to closely observe and guide trainee performance.

We aim to use an optimal model to achieve dual goals of guiding learning in a study of human participants and providing examples from which a cognitive model learns (i.e., rather than construct a model through programming, we will *induce* a model through learning mechanisms).

Our efforts to induce cognitive models from optimal models have shown that there *are* important fundamental and theoretical differences between optimal and constrained optimal models, and that the different feedback they provide can be expected to similarly impact both human learning and the induction of a cognitive model from that feedback.

This report presents an overview of some of the interesting aspects of inducing cognitive models from and being trained by optimal algorithms we have identified so far, together with an illustrative case study and an initial evaluation of the methodology.

Optimal Algorithms for Defining Task Performance

A key concept in computer science is the notion of optimal algorithmic approaches to particular problems. This notion has been used to characterize the relative performance of various algorithms, each with its own strengths and weaknesses. Algorithms may be optimal, but impractical, or as is more often the case, they may be sub-optimal, yet preferable for practical reasons.

The field of cognitive modeling has employed similar constructs, leading to research into the optimality of human behavior (e.g., Newell & Simon, 1972), and, in particular, the optimality of human behavior given human cognitive processing constraints (e.g., Kahneman, Slovic, & Tversky, 1982; Best & Simon, 2000; Best, 2004).

Research Objectives

The broad goals for our research are to understand the relationship between optimal task performance and human performance. Specifically, the research objectives for the current work are as follows:

- Investigate the different consequences of an optimal vs. an expert model in learning and performance (e.g., an expert model might not perform the task as well as an optimal system, and an optimal system might not direct a training session as well as an expert model).
- Develop a process model of (relatively) expert human performance based on normative performance and validated by empirical data.
- Determine if there is a continuum from normative to expert to novice performance or whether there are qualitative differences between the categories and what the empirical effect on training is.
- Investigate the cognitive limitations of human performance and the associated individual difference variables. Human performance may be effectively optimal given the constraints imposed by the perceptual system, memory system, etc. By carefully comparing

human performance with the various models proposed above, it should be possible to isolate the factors that influence human performance on the task (Lovett, Reder, & Lebiere, 1999).

In more general terms, we seek to use optimal task performance to frame human performance, especially when taken in the context of human processing constraints. Further, optimal performance may provide a basis for constructing suitable learning environments. Finally, optimal algorithms may also provide an automatic method for constructing a cognitive model: If an optimal task algorithm can be used to drive a learning cognitive model, it may be possible to induce a cognitive model of task performance with little development effort. The model simply does its best imitation of an optimal algorithm performing the task. This last research objective will be the focus of the remainder of this paper.

Instance-based Learning within ACT-R

The work described here depends heavily on the capability of the ACT-R cognitive architecture (Anderson & Lebiere, 1998) to learn correct behavior from examples. This learning capability, in this case, is specifically the declarative memory learning mechanism of ACT-R.

Instance-based models of human performance have been constructed for many individual decision processes. For example, Lebiere, Wallach, and West (2000) showed how the fundamental memory processes encoded in the ACT-R cognitive architecture can account for human behavior in games such as the Prisoner's Dilemma, a game in which a payoff matrix specifies positive and negative payoffs based on not just the player's move, but the opponent's move as well (games like this are of particular interest to economists since many economic systems can be analyzed in game-theoretic terms). In this case, the human performance was captured through two simple uniform architectural processes – power law learning (and decay), and stochasticity. These were incorporated into a general strategy based on determining the most likely outcome given each of the player's potential moves, and choosing the move with the best of the likely outcomes (stochasticity is essential to prevent the opponent from making easy predictions of the player's strategy). The ACT-R equation that produced both adaptation and stochasticity for this model is given below:

$$A = \ln \sum_{j=1}^{n} t_j^{\ -d} + N(0, \frac{\pi \bullet s}{\sqrt{3}}) \cong \ln \frac{n \bullet L^{-d}}{1-d} + N(0, \frac{\pi \bullet s}{\sqrt{3}})$$

The first term of this sum represents the strengthening in memory of a particular piece of information (a chunk) each time it is either retrieved from memory or (re)created. In this term, t_j represents the time since the j^{th} reference, while n is the number of references to the chunk, and d is the decay rate. The implications of this formula are that activation increases with use and decreases with time with a functional form that produces both the power law of learning and the power law of decay. Assuming even distribution of the chunk references over time allows for approximating activation with the shown function of decay, number of chunks, and total life of the chunk, L. Stochasticity is provided by the second term of the sum, normally distributed noise with a mean of 0 and a standard deviation determined by the activation noise parameter, s. These equations provide the basis for instance-based learning and decision-making within the ACT-R architecture. In general, instance-based techniques such as the method described here naturally generalize to situations which are similar to those that have been seen before, but are not exact matches to previous situations or behavior.

Task Environments for Studying Learning

The selection of task environments in which to pursue our research objectives has been driven by the need for an environment which is dynamic, fast-paced, time-pressured, and non-obvious, yet still amenable to the calculation of optimal performance. Initially, we have identified a National Missile Defense task, a task in which participants attempt to save lives by placing appropriate defensive units on cities of various populations in the face of an imminent attack. This task has the following beneficial properties:

- A straightforward optimal algorithm that is unlikely to be discovered by a participant.
- Time pressure.
- Repeated similar but non-identical trials that provide the opportunity both to learn and to measure learning.
- An interface operable by cognitive models and humans.

Using this task environment, we have embedded an optimal algorithm for use as a training aid. We have both tested human performance that leverages this training aid and constructed a cognitive model capable of learning from it.

The National Missile Defense Task

The NMD task is a simulated task environment that presents a scenario in which a subject must allocate a limited number of reserve Ground Based Interceptor (GBIs) missiles across individual cities under a ballistic missile attack. The interface in this task displays information about each city under attack, including the city population, the number of GBIs currently allocated to the city and the probability of intercepting the incoming missile.

A scenario consists of as many decisions as there are GBIs to be allocated. (The number of GBIs available depends on the number of cities.) The task participant is presented with a set of initial conditions for one of the scenarios—i.e., size of cities, probability of their targeting by enemy missiles and initial GBI allocation—and is then asked to make their allocations. The appropriateness of their decisions is based on the probabilistic expectation of lives saved by the allocation, where that expectation for a particular city is a function of the number of GBIs allocated as well as the population of the city:

$$\textit{Expected lives saved} = \textit{population} * (1 - 0.30^{\textit{Number of GBIs}})$$

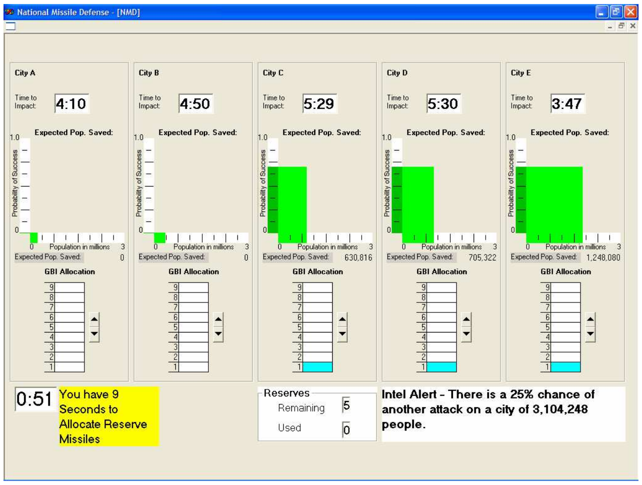

Figure 1: National Missile Defense Task interface

Figure 1 shows the primary NMD task display, consisting of a panel containing a set of cities, each with a population and some assignment of ground based interceptor (GBI) missiles to defend it from possible attack. The center bottom of the display shows the number of GBIs used and remaining in reserves while the lower left portion of the screen is focused on a countdown. The number of GBIs used is represented using a color bar and a slider control. The population and percentage of savings of life can be read off of the area for each city, most easily by paying attention to the area produced by combining these measures. The green areas above, then, represent the lives that will be saved by allocating the missiles as chosen above.

In addition to this screen, every trial gives one of three forms of feedback: 1) a summary of what the student chose to do, 2) an after-action review of the outcome of the actions taken by the student and a comparison to optimal allocations, and 3) stepwise feedback during problem solution in the form of beeps when the student strays off of the optimal solution path.

Human Performance on the NMD Task

Space precludes fully describing human performance on the NMD task. We will instead focus on a key aspect of human performance on the task to explore the effectiveness of the modeling efforts. Figure 2 presents the learning exhibited by human participants across trials that we hope to capture with a cognitive model. This graph presents the percentage of decisions by trial for human participants that are, in fact, the optimal decision. Initially, participants make approximately 50% of their moves consistently with an optimal model, while by the 70th trial they are exceeding 80% decision optimality by trial. This measure represents optimality by move rather than optimality by problem solution and therefore tracks how well participants stayed in lock-step with an optimal model (the same steps taken in different order for a particular problem would produce a lower percentage of optimal decisions, but the same outcome for the trial, so this measure is significantly more sensitive to the process used by participants than overall outcome).

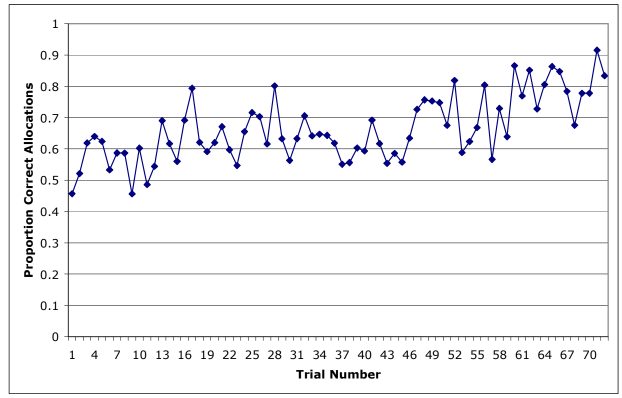

Figure 2: Percentage of Optimal Moves by Trial

Modeling the NMD Task

Potential alternatives for constructing a model of this task include the use of a pure optimal model (i.e., algorithm), a rule-based cognitive model that encodes the basic functionality of the optimal model, and an instance-based model that encodes the individual decisions but does not represent the optimal decision process in its entirety. Each of these will be discussed in turn below.

Pure Optimal Model

The first computational model developed as part of this project is a pure optimal computational model. The algorithm is based on a simple hill-climbing principle that can be applied to monotonic spaces like the NMD problem. Monotonic here means the individual moves do not interact, and will always have the same outcome, regardless of permutations. For example, a missile allocated in the NMD task and then de-allocated would leave the trial in exactly the same state it was in prior to the moves. Similarly, adding a missile to a city will always increase the protection afforded to that city by the same amount regardless of what has been done elsewhere within the trial.

Given this problem space, the construction of the initial algorithm was straightforward. The algorithm is an iterative algorithm that always seeks to change missile allocations in the direction of the greatest available gain. Since the excess missiles are initially in reserve, this means that each allocation must save more lives than the decrease in reserves costs, and that no allocation could save more than the chosen move. The algorithm terminates on the first iteration that no new assignments are made at which point the (optimal) solution is returned.

Rule-based Cognitive Model

Although the pure optimal model is conceptually straightforward, it was not obvious how to leverage the algorithm in a teaching situation. To investigate the utility of the algorithm for learning and teaching, we implemented the algorithm within the ACT-R cognitive modeling framework. The intention of this effort was to cast the optimal algorithm into cognitive operations, thereby providing both a task analysis and a potential mediating algorithm simultaneously. Those cognitive operations can be characterized within the ACT-R framework as the operation of production rules. An example of a production rule involved in the task is presented here in pseudocode:

```
If adding a missile to City B will save
more lives than adding a missile to City A
then City B is currently the best candidate
location for a missile.
```

In this capacity, the ACT-R language served primarily as a programming language, but one that introduced cognitive constraints. This (unusual) use of the architecture proved extremely useful from a task analysis perspective. In particular, the ACT-R architecture provides high fidelity estimates for the speed of cognitive operations. The optimal algorithm, as expressed in ACT-R, took more than a minute to complete a trial in a pilot version of the NMD task (using any reasonable estimates of the speed of the constituent operations). However, the original NMD task was temporally structured in a way that prevented students from interacting with the task for all but the last few seconds of a trial. The model clearly predicted the mental arithmetic could not be completed in such a short time frame. Attempts to apply the optimal algorithm to solve problems confirmed this prediction: the task as structured could not easily or practically be solved using the optimal algorithm with human constraints.

The performance of the algorithm, because it is not stochastic, and because time was not a serious consideration, is completely unremarkable: it produces the optimal answer exactly each and every time (and thus produces trivially perfect graphs and figures as well).

Instance-based Near Optimal Model

One of the primary technical objectives of this project is to investigate the utility of instance-based modeling techniques based on a cognitive architecture to quickly develop high-quality cognitive models. To this end, we constructed a performance model using the ACT-R cognitive architecture. This model, which is based on the rule-based ACT-R model discussed above, will now be described in detail.

Initially, the model starts with a particular scenario consisting of five cities each having a population, an initial allocation of missiles, and a follow-on city having a given probability of a follow-on attack and a number of reserve missiles that protect it. These facts are perceived and encoded into the declarative memory of ACT-R. For example, the first city in scenario 1 is represented as follows:

```
City1   99.326
    isa PERCEPT
    city 1
    pop 1839449
    miss 0
```

The model makes allocations by inspecting the cities in the scenario in a left-to-right fashion (echoing the solution method used by a human task expert), and identifying the city that would maximize the number of lives saved by increasing its allocation of missiles. For each city, the model attempts to retrieve a previous example of a similar evaluation (it has a bias to retrieve rather than calculate). If the model has never performed a similar calculation prior to this, or it cannot retrieve the result of the previous calculation, it instead calculates it. The potential gain of adding a missile is compared with the best possibility so far during the trial for each city and the follow-on city (thus limiting comparisons to the best previous city and the current city). Based on these calculations, the city that presents the opportunity for the greatest gain is selected to receive a defensive missile.

As the trial continues, the model either continues allocating missiles or decides to leave the rest of the missiles in reserve for a follow-on attack, thereby completing the trial. The model then receives feedback indicating the chosen and desired (optimal) allocation of missiles to the cities. Example feedback is show below:

```
City 5 population 1728296
missiles allocated 3: missiles desired 2
Follow-on city 6 population 2368000 probability
0.25 missiles left 0: missiles desired 1
```

The model processes the rows of the feedback one at a time, inspecting the chosen and desired coverage for that city. In this example, the model has allocated 3 missiles to city 5 (instead of the optimal 2) while not leaving any missiles in reserve to defend from a follow-on attack (instead of the optimal 1). Upon receiving this feedback, the model notes that an incorrect allocation was made, and that more missiles should have been kept in reserve.

The effect of this "noticing" is that the model stores an instance in declarative memory that corresponds to the action that should have been taken (the optimal choice). This instance, a chunk of information now stored in declarative memory, will have the opportunity to be retrieved the next time the model attempts to evaluate the alternative allocations that can be made during a trial. The assumption that is latent in this process is that the learner is consciously choosing to attend to the feedback screen, and that the same process would not necessarily be engaged by a more passive or less problem focused viewing of the information. This assumption is borne out by the differences in human performance observed in the three feedback conditions – the summary screen detailing the difference between actual performance and ideal performance appears to be a key driver of successful learning in this task.

Relating Model Performance to Human Performance

The preceding section describes the qualitative aspects of the instance-based model – the actions taken and their sequencing. This provides the first level of model performance description by demonstrating that the model performs the task and does so in a way that does not overtly violate the constraints of human performance. (This prerequisite step in model validation is often skipped before commencing with a quantitative analysis of second-order model effects. Finer-grained distinctions are certainly important, but the finer details of model correspondence are irrelevant if the broader details are wrong.)

Quantitative issues of interest include describing and explaining the proportion of time the model stays on the optimal path, the parameters that impact that performance, and issues surrounding learning. We chose to investigate the use of two parameters in tuning model performance: the accuracy of feedback, and the accuracy of calculations.

The first parameter, accuracy of feedback, is of interest for two reasons: 1) we determined that the existing version of the task had a bug in it which had resulted in occasionally incorrect (noisy) feedback in a prior study, and 2) expert feedback is likely to be at least slightly sub-optimal. To investigate this, we conducted a search across the parameter space of these variables to identify a portion of that space that corresponded to a range of human performance observed. The key measure is the percentage of moves made on the optimal path while varying the feedback accuracy and noise in mental calculations.

Figure 3 shows the percentage of moves made on the optimal path while varying the feedback accuracy and noise in mental calculations (note that "noisy" feedback was still close to optimal feedback). These parameters have roughly equal effects and combine approximately linearly.

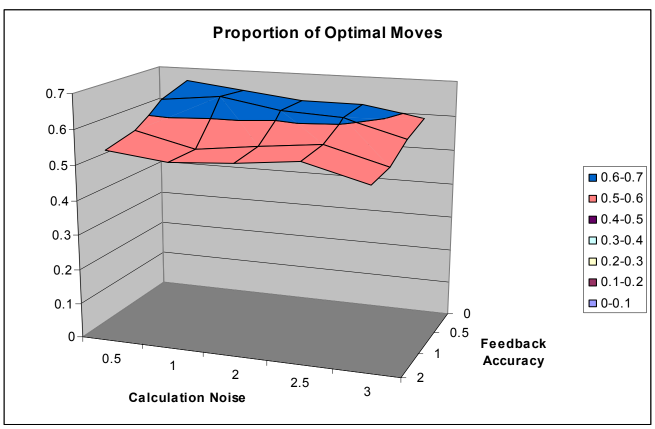

Figure 3: Effect of Feedback Accuracy and Mental Noise

Taken together, more accurate feedback and less calculation noise both result in better (learning) performance of the cognitive model, and are additive factors.

Learning in the Instance-Based NMD Model

The NMD model described above is a learning model that accumulates knowledge. As trials progress, the model depends more and more on recalling previous items rather than calculating the proper choice of action. This transition from calculation to memory retrieval might be labeled as "simply memorization", but in fact it does go beyond that.

Figure 4 shows the learning performance of the instance-based model of the NMD task compared to human performance (repeated from Figure 2), plotting trial number against percentage above optimal performance and presenting trendlines for both data sets with their equations:

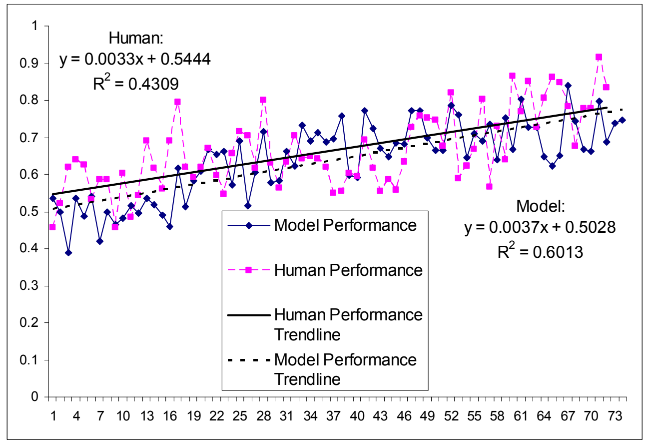

Figure 4: Human and Model Optimality by Trial

The near-identity of the best-fitting equations for human and model learning demonstrates that learning within the ACT-R framework, represented here as the transfer of operations from a procedural, algorithmic form into a memory-based pattern matching process, captures the key aspects of the learning demonstrated by the human participants. The lesser variability exhibited by the model is a product of the value of the ACT-R noise parameter and the number of individual model runs (in this case, 20). These values can be chosen to closely match human variability, or, alternatively, to expose the trend in the model learning. A more subtle point is that the learning demonstrated by the model is constrained by the representation chosen to learn within. These topics will now be addressed in greater detail.

The learning that takes place during an experimental session involves, qualitatively, the transfer of the seat of performance from a set of rules that algorithmically explain how to perform the calculations necessary for the task to the declarative memory of ACT-R. This has the flavor of what has been termed "Recognition Primed Decision Making", but rather than a vague notion of expertise having to do with recognition, it is made explicit here: expertise is the accumulation of task knowledge instances that allow the expert to make ever finer discriminations.

In this case, the ACT-R model is learning how many missiles go with a city of a particular size, and that begs the question of what it should be learning. The answer to that question depends on whether the goal is to model human performance, or whether there is an absolute performance goal or criterion to reach. From the perspective of the optimal algorithm the number of missiles depends completely on the relative needs of the other cities presented in the scenario. A city of 100,000 people, for example, may receive two GBIs if the remainder of the other cities in the scenario includes less than 50,000 people, but that same city might receive no GBIs if the other threatened cities all had populations exceeding a million people. Thus, the current representation may impose a ceiling on performance.

Representational Choices

The model was developed with two free parameters: feedback accuracy and mental arithmetic accuracy. We initially expected that learning effects exhibited by human task participants could be captured, if somewhat unsatisfactorily, by a systematic variation of the mental arithmetic accuracy parameter. However, the learning exhibited by the model based purely on memory effects captures both the time scale and the qualitative shape of the learning demonstrated by students in the laboratory.

The representation itself, though appearing impoverished, is actually sufficient for performing the task at a high level of competence. The following chunk is an example of the actual content of the learning:

```
Increment417    1.417
    isa INCREMENT
    pop 1738842
    miss 2
    prob 0.75
    eval 0
```

This particular fact is used by the model to decide that, if confronted with a city of population 1,738,842 that already has two missiles allocated, and a 75% chance of a follow-on attack, it is best to leave it at that (this is how the model interprets the meaning of “eval 0” – the number represents an estimate of the number of lives that would be saved by increasing the allocation, or 0 if it would be expected to cost more lives than it saves). These incremental solution improvement chunks are created during the feedback session, and though they are initially difficult to retrieve and sparse (there are many situations there is no relevant memory for early on) they eventually cover the problem space and support good performance.

Optimal Feedback versus Expert Feedback

The simulation studies conducted examined learning with varying amounts of noise added to the conclusions reached by the automated feedback engine. The results from the simulation studies indicated that, while small differences in feedback quality (like those expected between optimal and expert performance) have small impact, the impact on performance tends to scale with the difference in quality between optimal and actual feedback.

The simulation results also support the use of actual expert performance to drive the development of a cognitive model. The initial actions taken by the current model, though based on an optimal algorithm, contribute little to the eventual learning. Rather, the initial algorithm provides a scaffolding that enables some task performance; the quality of that initial performance is largely irrelevant. The optimal rule-based model simply provides a framework to get the model into the task, where the learning is actually memory based and is produced by *making mistakes*, not by performing correctly. That is, the mistakes provide the opportunity for feedback. The difference between what was done and what should have been done is what forms the basis for all of the learning exhibited by the model.

Evaluation of Methodology and Conclusions

The simulation studies conducted examined learning with varying amounts of noise added to feedback. The results from the simulation studies indicated that, while small differences in feedback quality (like those expected between optimal and expert performance) have small impact, the impact on performance tends to scale with the difference in quality between optimal and actual feedback.

This paper demonstrates a method for quickly developing cognitive models that capture human behavior in the absence of both human experts, and human performance data to validate the model against. An optimal model provided guidance to a cognitive model that was initially simply a rough shell of the task structure. The feedback from the optimal model then provided the details to hang on this scaffolding. The resulting model, though simpler to construct than many cognitive models, captured not only the eventual performance of human task participants, but also mimicked their learning performance.

References

Anderson, J. R., & Lebiere, C. (1998). *The Atomic Components of Thought*. Mahwah, NJ: Lawrence Erlbaum Associates.

Best, B. J. (2004). *Modeling Human Performance on the Traveling Salesperson Problem: Empirical Studies and Computational Simulations*. Doctoral dissertation, Department of Psychology, Carnegie Mellon University, Pittsburgh, PA.

Best, B. J. & Simon, H. A. (2000). Simulating Human Performance on the Traveling Salesman Problem. In *Proceedings of the Third International Conference on Cognitive Modeling,* 42-49. Veenendaal, Netherlands: Universal Press.

Kahneman, D., Slovic, P., and Tversky, A. (eds.) (1982). Judgment under Uncertainty: Heuristics and Biases. Cambridge University Press.

Lebiere, C., Wallach, D., & West, R. L. (2000). A memory-based account of the prisoner's dilemma and other 2x2 games. In *Proceedings of the Third International Conference on Cognitive Modeling,* 185-193. NL: Universal Press.

Lovett, M. C., Reder, L. M., & Lebiere, C. (1999). Modeling working memory in a unified architecture: An ACT-R perspective. In A. Miyake & P. Shah (Eds.) *Models of Working Memory*. pp. 135-182. Cambridge, MA: Cambridge.

Newell, A., & Simon, H. A. (1972*). Human problem solving*. Englewood Cliffs, NJ: Prentice-Hall.