Using HPC and PGAs to Optimize Noisy Computational Models of Cognition

Sue E. Kase[1] (skase@ist.psu.edu), Frank E. Ritter[1], Michael Schoelles[2]
[1]College of Information Sciences and Technology, The Pennsylvania State University
[2]Cognitive Science Department, Rensselaer Polytechnic Institute

***Abstract*-Cognitive modeling on high performance computing platforms is an emerging field. A preliminary analysis is presented on the use of parallel processing and genetic algorithms for optimizing the fit of non-linear, multivariable symbolic models of human cognition to experimental data. The effectiveness of this experimental optimization methodology is illustrated with a prototype model of a serial arithmetic task built in the ACT-R cognitive architecture. The results confirm that HPC-based optimization techniques could replace the manual optimization techniques used by cognitive modelers up until the present.**

I. Introduction

The number of available parameters for manipulating a cognitive model while running under the constraints of a cognitive architecture often outnumbers the experimental data points especially as the complexity of the task being modeled increases. For example, modifying model parameters to represent the effect of a threatening task appraisal in serial mental arithmetic performance, or the effects of 400 mg of caffeine on working memory capacity. The size of the search space grows combinatorially with the number of parameters used in the cognitive model.

Our research considers the role of genetic algorithms (GAs) in overcoming the combinative search spaces associated with cognitive models. GAs are a type of random search algorithm inspired by genetics and natural selection. They allow exploration of the space of potential cognitive theories, without preconceived notions of what the best parameters may be. GAs have disadvantages of being demanding in terms of computational load and memory. However, because the GA is an inherently parallel algorithm, parallel implementations of GAs (parallel genetic algorithms or PGAs) can provide considerable gains in terms of performance and scalability when studied and used on parallel machines.

The high-performance computing (HPC) platform utilized for the PGA component of the project is a Xeon Linux cluster with 1,450 dual-processor Dell PowerEdge 1750 servers located at the National Center for Supercomputing Applications.

The PGA runs the ACT-R cognitive architecture and the cognitive model. ACT-R is a theory for simulating and understanding human cognition. Researchers working with ACT-R are interested in understanding how individuals organize knowledge and produce intelligent behavior [1]. ACT-R provides a rigorous framework for cognitive modeling as well as an extensive set of parameters (over 100) and constraints on cognition to facilitate a priori predictions about a behavior of interest and psychologically plausible models in general.

In this case the behavior of interest is a serial mental subtraction task. Serial subtraction, repeatedly subtracting a 1- or 2- digit number from a 4-digit number is part of the Trier Social Stress Test used extensively to examine the physiological effects of stress in a laboratory setting [2]. Human performance data for the serial subtraction task was collected as part of a series of experiments investigating the effects of stress and caffeine on cognitive performance.

The human performance data is used to validate the cognitive model. A close correlation between the model's behavior and the human performance data is the goal. This fitting process is a key component in the Cognitive Science field, and in the end, determines success or failure of the research project.

Integrating HPC platforms, parallel processing, and optimization algorithms such as GAs into the modeling process points the way towards a more efficient and accurate model-to-data fitting process for the computational modeling community.

The paper briefly describes parallel implementations of GAs and the type of PGA used to optimize the prototype model. Cognitive models built in the ACT-R architecture are discussed next, as well as, the cognitive task being modeled, and the experimental data set used in the fitting process. Section IV compares the manual optimization used to date in the field to the parallel optimization methodology. The paper concludes with a discussion of the results from two PGAs.

II. Parallel Genetic Algorithms

Based on principles of natural selection and genetics, genetic algorithms (GAs) have been applied successfully to numerous problems in business, engineering, and science [3]. GAs are randomized, parallel search algorithms that search from a population of points [4]. The points (often referred to as genotypes) represent individuals in a population. The genotypes are evaluated for fitness, then propagated to later generations by means of probabilistic selection, crossover, and mutation. In the problem context of the project the genotypes are sets of ACT-R parameters applied to the cognitive task model. The population evolves to find better 'solutions' by selecting the most fit parameter sets (those that give the best match to the human data), and propagating these solutions to the next generation.

The stochastic search properties of genetic algorithms provide an efficient tool for solving problems with large, poorly understood search spaces, thus, allowing for exploration of the space of potential cognitive theories in which to apply to the problem. The search space can also be seeded (constrained) as knowledge about the context of the problem space becomes known.

In many practical applications, GAs find good solutions in reasonable amounts of time. However, in some cases, GAs can require hundreds of thousands of expensive function evaluations, and depending on the cost of each evaluation, the GA may take days, months, or even years to find an acceptable solution [5]. In this project, the function evaluation consists of running the model in the cognitive architecture, analyzing the model's performance output, and calculating a fitness value for the model's predictions. When considered over a generation of 200 genotypes, for example, the computational resources required on a single processor would be significant.

The parallel nature of genetic algorithms has been recognized for a long time, and many researchers have successfully used parallel GAs to reduce the time required to reach acceptable solutions to complex problems. GAs, working with a population of independent solutions, can easily distribute the computational load among multiple processors.

There are several classes of PGAs distinguished by their level of parallelization. This project utilizes a master-slave global parallelization PGA. This type of PGA is characterized by a high computation to communication ratio. In a master-slave PGA, one master-processing node (with rank 0) executes the GA-related functions (selection, crossover, mutation), while the fitness function evaluation is distributed among several slave processors. The slaves evaluate the fitness of the genotypes in the population that they receive from the master, and then return the fitness results back to the master node. Figure 1 is pseudo code for optimizing a cognitive model using a master-slave PGA with a message-passing interface (MPI).

In the project, the slaves each receive a different set of cognitive architecture parameters from the master, run the cognitive model in the architecture, collect the model output, and calculate the associated statistics and fitness value from the model's performance. Each slave then sends its fitness value from the model run back to the master.

```
MPI_Init . . .
if (rank is 0)  // master
    Initialize population
. . . . .
for (each generation)
{
    if (rank is 0)  // master
    {
        Selection
        Crossover
        Mutation
    }

    // find fitness of genotypes in population
    // master and slaves
    MPI_Scatter individuals out to processors
    Run cognitive model
    Calculate fitness of model predictions
    MPI_Gather up resulting fitness values

    if (rank is 0)  // master
        Print out generational statistics
}
Test best solutions found  // master and slaves
MPI_Finalize . . .
```

Fig. 1. Pseudo code for master-slave GA using MPI

III. COGNITIVE MODELS

A symbolic approach to cognitive science holds that cognition can be explained using operations on symbols, by means of explicit computational theories and models of mental (but not brain) processes analogous to the working of a computer.

A cognitive model, in the form of a working computer program, is intended to be an explanation of how some aspect of cognition is accomplished by a set of primitive computational processes. A cognitive model performs a specific cognitive task or class of tasks and produces behavior that constitutes a set of predictions that can be compared to data from human performance. *A cognitive model produces both a theory of human behavior on a task and a computational artifact that performs the task*.

To represent the intended level of abstraction, many programming languages designed for cognitive modeling are production systems. Production systems are used as a flexible model of the control structure of human cognition. The flow of processing is controlled by a set of production rules (condition-action pairs) that can be selected to fire when their conditions are satisfied. Therefore, the flow of control is at run time, and is a function of dynamically evolving memory contents triggering the productions. A cognitive model written in a production system makes theoretical commitments at the level of the production rules, and when built within a cognitive architecture, defines a computationally complete system. In this cognitive architecture approach to modeling, the model is a byproduct of three components: cognitive constraints offered by the architecture; background knowledge residing in memory; and the task to be performed.

A. ACT-R

Many instances of cognitive architectures exist, for example: ACT-R [6], Soar [7], and Epic [8]. ACT-R is the product of a community of researchers led by John Anderson at Carnegie Mellon University. ACT-R is a two-layer modular cognitive architecture on a production system framework. One layer contains symbolic representations and has a serial flow in that only one production can fire at a time. The second layer is a sub-symbolic layer whose representations are numeric quantities that are the result of computations preformed as if they were executed in parallel. In ACT-R cognition emerges through the interaction of a number of independent modules. Each of these modules is associated with specific brain regions and theories about the internal processes of these modules [9].

The modularity of ACT-R permits the parallel execution of the verbal system with the control and memory systems (specifically involved in the serial subtraction task). ACT-R has been used in models of working memory tasks and arithmetic processing tasks by other researchers.

B. Serial Subtraction Task

The cognitive model for this project simulates a human subject performing the serial subtraction task. Serial subtraction is the mental arithmetic stressor portion of the Trier Social Stressor Test (TSST). The TSST has been used to provide an acute physiological stress response in human subjects since the 1960's. The serial subtraction task consists of four 4-minute blocks of mentally subtracting by 7's and 13's from 4-digit starting numbers.

C. Experimental Data

The cognitive model of the serial subtraction task is validated with human subject data collected from a larger project to study the effects of stress, task appraisal, and caffeine on biomarkers of cardiovascular health [10].

In the serial subtraction task, subjects' answers were scored against a list of correct answers from the starting number. Task performance was voice recorded on a digital camera and laptop computer. For each subject the number of subtraction problem attempts were recorded and a percent correct score was calculated by dividing the total number of correct attempts by the total number of attempts for each block of the subtractions. The audio recordings were transcribed to obtain subtraction pace and details about error types.

Table I shows the subtraction rates for the subjects' performance on two 4-minute blocks of subtracting by 7s. There is a wide range of performance on this task suggesting a high degree of individual differences within the subject pool.

TABLE I
HUMAN SUBJECT (N=15) MEAN PERFORMANCE AND STANDARD DEVIATION FOR SERIAL SUBTRACTION ON 4-MINUTE BLOCKS OF SUBTRACTING BY 7S

	7s – 1st block	7s – 2nd block
Number of Attempts	47.3 (15.2)	47.8 (19.2)
Percent Correct	82.0 (10.0)	88.8 (7.0)

Subtraction performance was also analyzed by task appraisal. During the experiment, pre- and post-task appraisals were assessed immediately before and at the end of the serial subtraction stressor session. Based on the appraisal responses, subjects were categorized into one of two appraisal groups: challenge or threat. A challenge condition equates to a subject's perceived stress being less than or equal to their perceived ability to cope with the task. In a threat condition, the subject's perceived stress is greater than their perceived ability to cope with the task. Table II shows the subtraction rates for the subjects grouped by post-task appraisal condition. For the project the cognitive model was fit to the mean of each appraisal group.

TABLE II
MEAN PERFORMANCE AND STANDARD DEVIATION BY POST-TASK APPRAISAL GROUP

7s – 1st block	Threat (N=8)	Challenge (N=7)
Number of Attempts	40.3 (10.1)	55.3 (16.7)
Percent Correct	78.1 (8.2)	85.4 (10.8)
7s – 2nd block		
Number of Attempts	44.8 (10.2)	70.7 (23.7)
Percent Correct	84.2 (4.6)	92.5 (6.2)

IV. OPTIMIZATION PROCESS

A. Manual Optimization

Traditionally, cognitive modeling researchers use a manual optimization process to fit the model to the human data. This time consuming iterative process involves selecting a set of parameters, assigning a numeric value to each parameter, running the model in the cognitive architecture, and evaluating the resulting output against the human data. If the fit is unsatisfactory, the process is repeated.

The optimization process can be complicated by the stochasticity built into the cognitive architecture. With a static set of parameters and values, the combination of the model and architecture yield a distribution of performance scores, not a single value. When models include stochastic effects, the model may require 10, 20, or 100s of runs in order to compute stable predictions. Table III compares five example sets of ACT-R parameters used in the serial subtraction model. For each parameter set, the model was run 10 times and then 100 times. The number of attempts and percent correct are averaged over the number of model runs.

TABLE III

MEAN PERFORMANCE AND STANDARD DEVIATION ON SERIAL SUBTRACTION BY 7S FOR ONE 4-MINUTE BLOCK BY POST-TASK APPRAISAL GROUP

ACT-R Parameters			Mean Across 10 Model Runs		Mean Across 100 Model Runs	
ANS	BLC	SYL	Number of Attempts	Percent Correct	Number of Attempts	Percent Correct
0.463	1.693	0.526	42.30	82.56	42.88	85.00
0.399	1.839	0.555	40.30	78.05	40.03	89.74
0.251	1.588	0.529	42.50	80.49	41.57	82.93
0.766	2.619	0.531	40.90	90.00	41.11	78.57
0.654	2.078	0.588	39.60	78.95	38.77	89.47

When comparing model performance between 10 and 100 runs, percent correct shows more variance than number of attempts. This makes for difficult optimization especially if both performance statistics are used simultaneously in the fitting process. Previous attempts at fitting the serial subtraction model to data from other human subject experiments using manual optimization techniques have been unsuccessful [11].

B. *Parallel Optimization*

The ACT-R architecture and cognitive model are written in the Lisp. Generally, message-passing interfaces available on cluster computing resources are called from C or Fortran programs. To utilize parallel processing in the cognitive model optimization process, ACT-R and the cognitive model are packaged into an executable Lisp image or core file. This image file can be run by a system call from a C program on each processor in parallel while utilizing MPI to communicate genotypes and fitness values among the processors.

The population of genotypes (ACT-R parameter sets), in the form of a matrix, are ‘scattered’ row-wise to the processors. Each processor executes the Lisp image file that runs the model within the ACT-R architecture. Each processor then calculates a fitness based on the model’s performance predictions and the human data statistics. In this case, sum of the squared error is calculated on both number of attempts and the percent correct from a block of subtracting by 7s. The fitness values calculated by the processors are ‘gathered’ up by the master process, which then applies genetic functions to the population based on the fitness of the genotypes (refer to Figure 1). This is repeated through any number of generations with the effect of evolving a set of candidate solutions.

C. *Serial Subtraction Optimization*

Two PGAs were set up to run 50 generations of 200 binary-encoded genotypes. One PGA optimized the model to the challenge appraisal group means (55.3 attempts, 85.4% correct), and the other to the threat appraisal group means (40.3 attempts, 78.1% correct).

A genotype consisting of one 36-bit chromosome is divided into three 12-bit substrings each representing the value of an ACT-R parameter. We investigated: activation noise representing variance in applying procedural knowledge (ANS), the base level constant affecting declarative memory retrieval (BLC), and syllable rate, seconds per syllable (SYL)—because the model verbalizes the answers as the human subjects do. One processor was allocated for each genotype.

The selection probability (selection of the fittest) was set to 0.5 meaning half the population is replaced each generation by offspring of the fittest genotypes. Random mutations alter a certain percentage of the bits in the list of chromosomes. This operation introduces traits in the original population and keeps the GA from converging too quickly before sampling the entire search space. The mutation rate was set at 0.15. The terminating condition was a specified number of generations (50), instead of proximity to the appraisal data means. The fitness function compared the sum of the squared error for the model’s predicted number of attempts and percent correct to the corresponding human data.

V. RESULTS

Typically, GAs generate new points in the search space by applying operators to current points and statistically moving toward more optimal positions in that search space. In this optimization problem, the fitness is in terms of error (or cost) and is the discrepancy between the model’s predictions and the actual human performance on the cognitive task. The PGA in this case is seeking a global minima in the ACT-R parameter space.

Figures 2 and 3 plot the progress of the PGA as it seeks a global minima across the 50 generations. Figure 2 shows the minimum and average fitness when optimizing to the challenge appraisal group means. Figure 3 is optimizing to the threat appraisal group means.

Normally, what would be expected for this type of plot is a smooth, maybe slightly bumpy, curvilinear downward sloping line as the GA converges on a solution. Figure 2, and especially Figure 3, show the PGA ‘bouncing’ around the search space; finding a fit solution in one generation, and then tossing it out in the next. Additionally, it appears that this pattern would continue for infinitely many generations.

Because of the previously discussed stochastic effects embedded in the model and architecture these results are not that surprising. Several modifications were built into the PGA to compensate for a single genotype returning a distribution of performance predictions instead of an exact value.

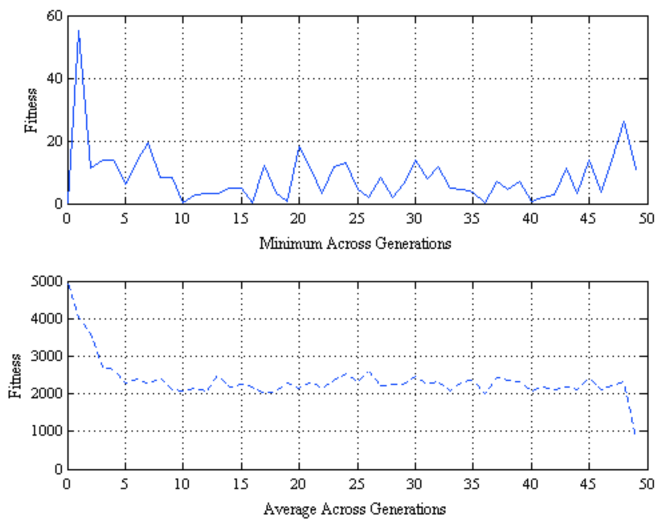

Fig. 2. PGA optimizing to challenge appraisal group

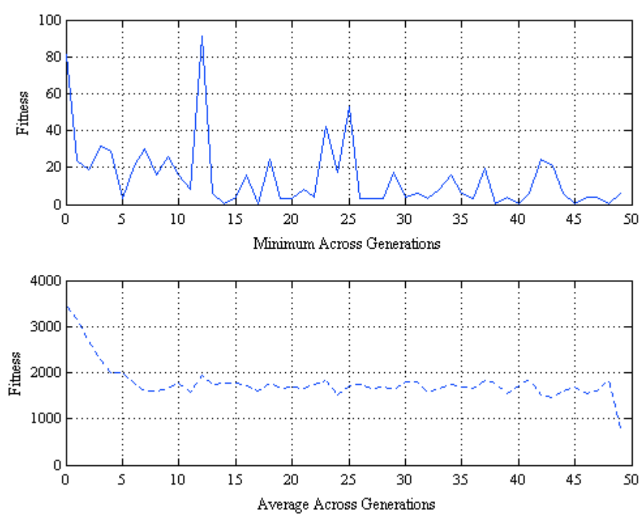

Fig. 3. PGA optimizing to threat appraisal group

During its generational journey, if the PGA finds a 'good enough' solution, as determined by a boundary fitness value, that particular genotype is remembered for a post-PGA testing phase. In essence, the PGA is gathering up good solutions across the generations, instead of converging on a so-called best set of solutions. Once the PGA terminates, each of the collected genotypes is run on all the processors with the fitness calculated from the mean number of attempts and percent correct across all runs (200 runs per genotype).

While optimizing to the challenge appraisal group the PGA collected 17 genotypes for additional testing. During threat appraisal group optimization, 9 genotypes were collected. After the final generation of the PGA, these genotypes were run in parallel on all the processors using a master-slave/MPI approach. Table IV lists the four best fitting genotypes (by post-PGA test) from each appraisal optimization collected by the PGA. The second column shows the genotypes' original fitness value as reported by the PGA compared to their fitness values from post-PGA testing in the third column.

TABLE IV
GENOTYPE FITNESS COMPARISON BETWEEN PGA AND POST-PGA TESTING

Genotypes	Fitness	
Challenge Optimization	PGA Reported	Post –PGA Testing
0.500, 2.083, 0.365	0.093	0.133
0.271, 1.558, 0.360	0.093	1.244
0.561, 2.279, 0.366	0.093	1.696
0.500, 2.083, 0.365	3.597	3.610
Threat Optimization	PGA Reported	Post-PGA Testing
0.727, 2.538, 0.535	6.191	0.986
0.729, 2.524, 0.593	6.008	7.072
0.693, 2.446, 0.586	3.075	7.896
0.713, 2.446, 0.593	3.614	8.325

In the threat appraisal optimization, the genotype producing the best fitness value reported in the PGA (3.075) does not correspond to the genotype producing the best fitness value from the post-PGA testing phase (0.986). In the challenge appraisal optimization, there were three genotypes with a fitness of 0.093. One of those genotypes produced the best post-PGA test fitness value (0.133).

As a validation effort, the best fitting set of ACT-R parameters from each appraisal optimization was tested with three additional sets of 200 runs each. Table V shows the number of attempts and percent correct averaged over each of the 200 runs, and a comparison of the model's mean performance to the subjects' mean performance by appraisal group.

TABLE V
VALIDATION OF BEST FITTING PARAMETER SETS FROM APPRAISAL OPTIMIZATIONS

Challenge Appraisal Performance		
ACT-R Parameters	Number of Attempts	Percent Correct
0.500, 2.083, 0.365	55.0	83.5
	55.1	83.0
	55.0	84.7
Model Performance Means	55.0	83.7
Human Data Means	55.3	85.4

Threat Appraisal Performance		
ACT-R Parameters	Number of Attempts	Percent Correct
0.727, 2.538, 0.535	40.8	76.5
	40.9	78.5
	40.8	77.4
Model Performance Means	40.8	77.5
Human Data Means	40.3	78.1

For number of attempts the fit is nearly perfect; a difference of 0.3 for the challenged subjects, and 0.5 for the threatened subjects—half a subtraction problem or less. The fit is slightly less accurate for percent correct; a difference of 1.7 correct subtractions for challenged subjects, and half a correct subtraction (0.6) for threatened subjects.

In summary, this is a very good fit considering the complexity of model and the wide range of human performance on the serial subtraction task. The total run time on the cluster was minimal; 117 minutes for the two PGAs including the post-PGA testing phase, and 4 minutes for the additional best solution validation runs.

It would be important to consider past cognitive science research and potential theory development in the analysis and interpretation of the most promising of the PGA genotypes returned from the testing phase.

VI. Concluding Remarks

By integrating parallel processing on high-performance computing platforms with stochastic search algorithms, such as PGAs, cognitive models can be optimized to fit human subject data efficiently and more accurately then tradition manual optimization techniques.

The stochasticity built into the architecture requires cognitive models of tasks characterized by wide performance variance to be run 10, 20, or 100s of times to compute stable performance predictions. The serial subtraction task is one such task showing a wide range of human performance. Using manual optimization techniques to fit a wide distribution of performance is difficult.

Using 200 processors and approximately two hours of HPC run time the prototype model of the serial subtraction task produced over 25,000 predictions of human performance enabling the fitting of subject appraisal groups and, in the future, individual subjects. Additionally, the results from this exploratory optimization process introduce questions about the nature of the ACT-R parameter space and validity of the architecture in general. Visualization of the parameter space is needed to determine if rough terrain corresponding to noisy data or non-continuity exists.

Acknowledgments

This project is partially supported by ONR grant N000140310248. Computational resources were provided by TeraGrid DAC TG-IRI070000T. The authors would like to thank Dr. Laura Klein and her lab at the Department of Biobehavioral Health, Penn State University, for collection of the human performance data.

References

[1] J. R. Anderson, *The architecture of cognition*. MA: Harvard University Press, 1983.

[2] C. Kirschbaum, K. M. Pirke, and D. H. Hellhammer, “The Trier Social Stress Test—A tool for investigating psychobiological stress responses in a laboratory setting,” *Neuropsychobiology*, vol. 28, pp. 76-81, 1993.

[3] D. Goldberg, “Genetic and evolutionary algorithms come of age,” *Communications of the ACM*, vol. 37, no. 3, pp. 113-119, 1994.

[4] D. Goldberg, *Genetic algorithms in search, optimization and machine learning*. MA: Addison-Wesley, 1989.

[5] E. Cantu-Puz, *Efficient and accurate parallel genetic algorithms.* Kluwer Academic Publishers, 2001.

[6] J. R. Anderson, *The adaptive character of thought*. Hillsdale, NJ: Erlbaum, 1993.

[7] A. Newell and H. A. Simon, *Human problem solving*. Englewood Cliffs, NJ: Prentice-Hall, 1972.

[8] D. E. Meyer and D. E. Kieras, “A computational theory of executive cognitive processes and multiple-task performance: Part 1: basic mechanisms,” *Psychological Review*, vol. 104, no. 1, pp. 3-65, 1997.

[9] J. R. Anderson, *How can the human mind occur in the physical universe*? Oxford University Press, 2007.

[10] C. A. Whetzel, F. E. Ritter, and L. C. Klein, “DHEA-S and cortisol responses to stress and caffeine in healthy young men: Is DHEA-S a reliable marker for stress?” *Psychosomatic Medicine*, vol. 68, pp. A-77, 2006.

[11] F. E. Ritter, M. Schoelles, L. C. Klein, and S. E. Kase, “Modeling the range of performance on the serial subtraction task,” in *Proceedings of the 8th International Conference on Cognitive Modeling*, pp. 299-304, June 2007.