

A Scalable Parallel Genetic Algorithm for X-ray Spectroscopic Analysis

Kai Xu
Dept. of Computer Science
and Engineering
University of Nevada, Reno
Reno, NV, 89557
xukai@cs.unr.edu

Sushil J. Louis
Dept. of Computer Science
and Engineering
University of Nevada, Reno
Reno, NV, 89557
sushil@cs.unr.edu

Roberto C. Mancini
Department of Physics
University of Nevada, Reno
Reno, NV, 89557
rcman@physics.unr.edu

ABSTRACT

We use a parallel multi-objective genetic algorithm to drive a search and reconstruction spectroscopic analysis of plasma gradients in inertial confinement fusion (ICF) implosion cores. In previous work, we had shown that our serial multi-objective Genetic Algorithm was a good method to solve two-criteria X-ray spectroscopy diagnostics problems. However, this serial version was slow and we therefore could not incorporate better physics and more criteria to solve larger problems and handle larger data sets. In this paper, we develop and use a parallel multi-objective genetic algorithm based on a master-slave model to solve three criteria spectroscopic analysis problems. The algorithm works well in reconciling experimental observations with theoretical physics model parameters. In addition, theoretical analysis and experimental results on the parallelized version show good scalability with up to 150 processors. This reduces the time for running the GA from 9.6 hours to 5.9 minutes.

Categories and Subject Descriptors

G.1.6 [Optimization]: Global Optimization; I.6.8 [Types of Simulation]: Parallel; J.2 [Physical Sciences and Engineering]: Physics

General Terms

Algorithms, Design, Performance

Keywords

Parallel Genetic Algorithms, Speedup, X-ray Spectroscopy

1. INTRODUCTION

X-ray spectroscopic analysis has proved to be a useful technique to determine temperature and density of astrophysical as well as laboratory plasmas [9]. Analysis based

on a uniform-model approximation yields conditions in the plasma source which can be interpreted as space averages. From a computational point of view these cases represent optimization problems which can be efficiently handled by a single objective genetic algorithm [7]. However, spectroscopic analysis of the spatial structure of plasma sources requires self-consistent and simultaneous consideration of several pieces of data. Then, the problem becomes a typical multiple objective optimization problem, which can be solved by a multi-objective genetic algorithm [8][6]. In earlier work, Louis [7] used such a multi-objective genetic algorithm to attack the problem of finding temperature and density gradients of ICF implosion cores through multi-objective optimization of an error function that minimizes the error between observed values and the values produced by our computational model of the X-ray line emission from the tracer argon gas in the core. In this paper, we use an improved Master-Slave parallel Multi-Objective Genetic Algorithm (MOGA) which allows us to handle more complex physics in our model, more criteria, and still runs in a reasonable amount of time on problem sizes of interest. Results indicate that the algorithm works well and both theoretical analysis and experimental data show that our parallelized multi-criteria genetic algorithm has a good scalability with increasing numbers of nodes. The organization of this paper is as follows. First, we introduce the physics problem being attacked, and then provide a summary of common approaches to parallelizing genetic algorithms. Section 2 presents our implementation of parallel multi-criteria genetic algorithm. Subsequently we develop a new framework that models our parallel genetic algorithms speedup behavior. Finally, the last two sections give experimental results and discuss future work.

1.1 X-ray Spectroscopic Analysis

X-ray spectroscopic analysis of ICF implosion cores uses x-ray line emission from a tracer gas added to the deuterium-filled spherical target to diagnose plasma conditions achieved in the implosion core [9]. Given the experimental data and our physics model of the X-ray line emission in the plasma the goal is to find temperature and density gradients that simultaneously yield good fits to 1) time-resolved spatially integrated X-ray line spectra and 2) X-ray monochromatic emissivity profiles for $He\text{-}\beta$ and $Ly\text{-}\beta$ line transition. Spatial emissivity profiles can be extracted from Abel inversion of X-ray monochromatic images provided that the plasma is

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO'05, June 25–29, 2005, Washington, DC, USA.
Copyright 2005 ACM 1-59593-010-8/05/0006 ...\$5.00.

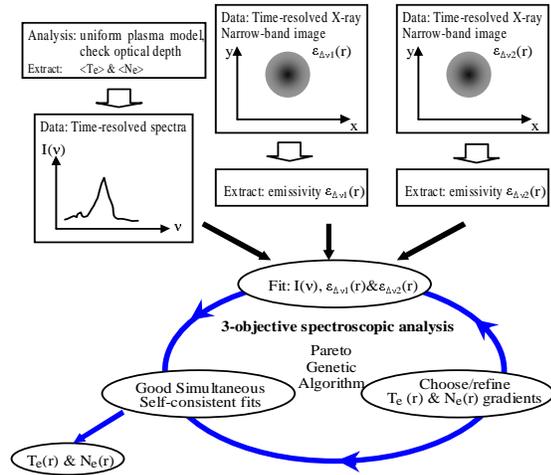


Figure 1: Fitting Plasma Temperature and Density Gradients

optically thin and spherically symmetric [6]. Figure 1 shows the role of the genetic algorithm in fitting theoretical model parameters to experimentally obtained data.

Temperature and density spatial gradients as well as other properties of plasmas can be computed using hydrodynamic modeling. Hydro simulations are model calculations that include hydrodynamics, thermal transport, atomic, radiation physics, etc. The models are very complex and it is important to be able to compare simulation results with independent information obtained from the analysis of experimental data. Our work is, to a large degree, independent of hydrodynamic simulations and hence can provide the data for these comparisons. Estimating plasma temperature and density gradients based on the analysis of experimental data is a complex inverse problem. This work builds upon our earlier work on this problem and extends our methodology to handle larger, more complex problems where the physics spectral model computed to evaluate the fitness function of the genetic algorithm becomes more computationally intensive. The results of the analysis of experimental data can be used to improve characterization of core plasma dynamics and to provide new data for detailed benchmarks of hydrodynamic codes.

1.2 Multi-objective Genetic algorithm

Genetic Algorithms (GAs) are search and optimization algorithms based on the mechanics of natural selection. They are capable of finding solutions in poorly understood search spaces while exploring only a small fraction of the space, and can robustly deal with complex non-linear problems. In this paper, our problem has a large, multi-dimensional search space. Thus, an efficient and robust algorithm is required to effectively implement the spectroscopic analysis.

One approach to solving this problem is to combine the multiple (in our case, three) criteria into a single scalar fitness function. Unfortunately, this simple method did not work well on our problem. Instead we turned to multi-objective optimization and used a multi-objective genetic algorithm [4]. Our strategy is to use the principle of pareto optimality in designing a multi-objective genetic algorithm. At each generation, there is a set of non-dominated solutions

in fitness space that form a surface known as the pareto optimal front (or the Pareto front). The goal of a multi-objective genetic algorithm is to find and maintain a representative sampling of the solutions on the pareto front. If the criteria are not self-contradictory, then there should be a point on the final convex front that satisfies all criteria well. We used the Elitist Non-Dominated Sorting Genetic Algorithm (NSGA-II) from Deb's book as our implementation of the pareto GA [4]. For each generation, we expand the current population size from n to $\lambda*n$, where λ is an expansion parameter. Then, after evaluation, we use the non-dominated sort algorithm to sort the new candidate population and cut the population size back to n by retaining only the first n individuals in the sorted population. The detail of this algorithm can be found in [4]. This paper develops a parallel version of NSGA-II for X-ray spectroscopic analysis.

1.3 Parallel Genetic Algorithms

There are three basic types of parallel Genetic algorithms: 1) Master-slave: Here, a single processor performs the genetic operations and uses other processors only for evaluation of individuals. This model is useful when dealing with a small number of processors or when dealing with computationally intensive evaluations 2) Island model: In this model, every processor runs an independent Evolutionary Algorithm (EA), using a separate sub-population. The processors cooperate by regularly exchanging migrants (good individuals). The island model is particularly suitable for computer clusters, as communication is limited. 3) Diffusion model: Here, the individuals are spatially arranged, and mate with other individuals from the local neighborhood. When parallelized, there is a lot of inter-processor communication (as every individual has to communicate with its neighbors in every iteration), but the communication is only local. Thus this paradigm is particularly suitable for massively parallel computers with a fast local intercommunication network. For more detail about parallel genetic algorithms, please refer to [1]. In this paper, we use the Master-Slave model, since evaluations take quite long compared to communication time. The advantages of this model are ease of implementation and the property of producing the same result as the corresponding serial genetic algorithm.

1.4 Parallelization method

Each evaluation requires execution of the spectral model, which is somewhat expensive in time. In order to reduce computing time we parallelized our pareto GA using a Master-Slave model so that evaluation is done in parallel (Figure 2). In this implementation the behavior of the algorithm is the same as that of the sequential GA. It is possible to parallelize the other genetic operators as well, but it may not necessarily lead to any significant improvement. Crossover and mutation operators are very simple and any performance gain due to their parallelization will be diminished by communication costs. In any case, the master-slave kind of parallelization is quite straightforward and relatively easy to implement and should result in significant speed-up as long as the cost to evaluate an individual dominates communication cost [2].

For our application our results actually show better speedup with respect to Cantú-Paz's theoretical model of parallel GA performance [3]. Cantú-Paz developed a master-slave parallel genetic algorithm model where speedup was shown to

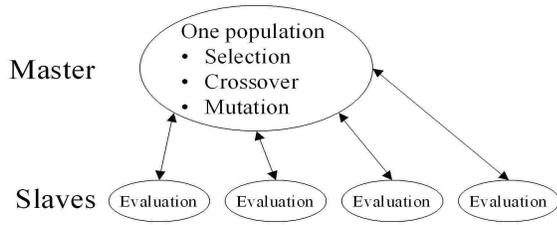


Figure 2: A Schematic View of a Master-Slave Parallel Genetic Algorithm

first rise and then fall as we increased the number of nodes in the cluster. To better fit experimental results, this paper develops a new theoretical model for parallel GAs. It is interesting that if we try to send multiple individuals in a single communications event (as in the Cantú-Paz model) we can have a decrease in performance with an increase in number of nodes thus giving an upper limit on the number of nodes or population size in evaluating the utility of a parallel GA. Our model does not ever lead to a decrease in performance with an increase in the number of nodes. We also consider other factors like potential variance in individual evaluation times caused by evaluation parameters and shared system scheduling. Thus, in our shared cluster, because other jobs may be running, we cannot assume that all our nodes take equal time in evaluating and communicating the fitness of an individual chromosome. In terms of implementation, we address this issue by using a producer-consumer strategy in chromosome distribution. Each slave node only gets one individual to evaluate; when done evaluating, the node returns the fitness to the master node, which will then assign it a new individual to be evaluated. Depending on the relationship between population size and number of nodes, this scheme may increase or decrease communication time. But overall, the communication time will not change significantly compared to the long evaluation time on each node. The next two subsections describe our implementation and develop a theoretical model of our parallel scheme.

1.5 Algorithm

We provide the pseudo-code for our parallel multi-criteria genetic algorithm below:

Slave Algorithm (Evaluation Node)

```

Initialize()
While (true)
  Receive instruction I from master node
  If (I is EVALUATION_COMMAND)
    do evaluation and return the fitness
  else if (I is FINISH_COMMAND)
    process exit
end while

```

Master Algorithm

```

Randomly initialize the population
Evaluate the whole population
While (not meet stop criteria)
  Pareto_Selection()
  Crossover_And_Mutation()
  Evaluate_Population()

```

End while

Evaluate_Population:

```

While (not all individuals evaluated)
  Wait (blocking) until a slave node  $N_i$  is available
  Send next individual to  $N_i$ 
  wait (non-blocking) for the fitness return.
End while

```

1.6 Parallelization Analysis

Assume that n is the given population size and P , the number of processors. In every generation, all the slave nodes perform fitness evaluation with average time \overline{T}_E and the average communication time between two nodes is \overline{T}_C . Thus communication in each generation takes time $(n-1) * \overline{T}_C$. Genetic algorithm processing (selection, mutation, and crossover) time can be described by $T_B(n)$, which in our implementation is only related to population size, not P . So given P processors, the total time in one generation has the following form:

$$\begin{aligned} T_p &= T_B + T_C + T_E \\ &= T_B(n) + (n-1) * \overline{T}_C + \frac{n}{P} \overline{T}_E \end{aligned} \quad (1)$$

The speed-up for k processors can be calculated by:

$$\begin{aligned} Speedup_k &= \frac{T_1}{T_k} \\ &= \frac{T_B(n) + (n-1) * \overline{T}_C + \frac{n}{k} * \overline{T}_E}{T_B(n) + (n-1) * \overline{T}_C + \frac{n}{k} * \overline{T}_E} \\ &= \frac{T(n) + n \overline{T}_E}{T(n) + \frac{n}{k} \overline{T}_E} \\ &= \frac{1 + \frac{n \overline{T}_E}{T(n)}}{1 + \frac{n \overline{T}_E}{k T(n)}} \end{aligned} \quad (2)$$

As is show in the derivation, we combined the first two terms $T_B(n) + (n-1) * \overline{T}_C$ into one function of n , $T(n)$, and the expected speedup can be known if we know what $T(n)$ looks like. More specifically, the actual speedup is determined by the ratio $\frac{\overline{T}_E}{T(n)}$, the ratio of evaluation time to all other operations $T(n)$ (communication and GA processing on master node). In the ideal case, $\frac{\overline{T}_E}{T(n)} \gg k/n$, and the speedup will be almost linear (thick line in Figure 3); but in more realistic cases, speedup will not be a linear function. The equation simplifies when $T(n)$ is linear, i.e., $T(n) = C * n$, or $n / T(n) = s$, where s is a constant value. In this case, the speedup becomes:

$$\begin{aligned} Speedup_k &= \frac{T_1}{T_k} \\ &= \frac{1 + \frac{\overline{T}_E}{T(n)}}{1 + \frac{n * \overline{T}_E}{k * T(n)}} \\ &= \frac{T_B(n) + (n-1) * \overline{T}_C + \frac{n}{k} * \overline{T}_E}{T_B(n) + (n-1) * \overline{T}_C + \frac{n}{k} * \overline{T}_E} \\ &= \frac{1+s}{1+\frac{s}{k}} \end{aligned} \quad (3)$$

Figure 3 plots speedup versus number of processors for the linear case. The dashed line corresponds to $n \overline{T}_E / T(n) = 1$, thin line to 5, the intermediate line to 50, and the thick line to 500. The dotted line is the ideal (linear) speedup.

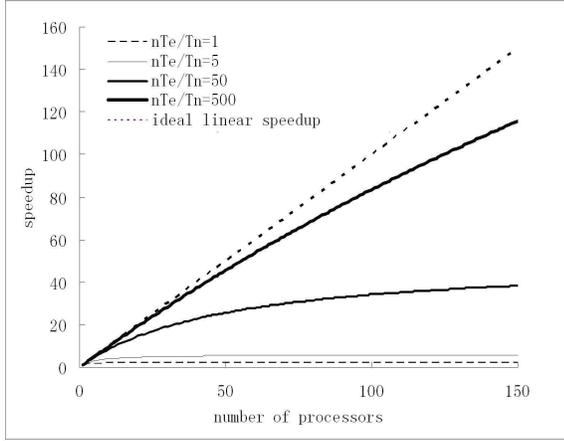


Figure 3: Model-predicted theoretical speedups

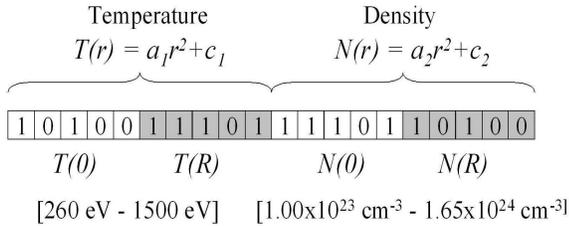


Figure 4: Encoding of Temperature and Density

2. ENCODING

The X-ray spectra are fully determined by the spatial distribution of plasma, electron number density and temperature. Both density and temperature should be smooth functions of position and all values must fall within a certain range in order to be physically meaningful. Moreover, the functions must be symmetric with respect to the center of the plasma and the first derivative of temperature should be equal to zero at the center. There are several plausible encoding choices for gradients that satisfy these properties. A parabolic function ($f(x) = ax^2 + b$) is the simplest choice, requires only two parameters to be fully defined, and can accurately describe many possible gradients. Figure 4 illustrates our encoding of a parabolic gradient. It is important to employ a flexible and robust encoding algorithm for characterizing the temperature and density gradient functions. Otherwise, the search and optimization might just fail because of the lack of ability of the gradient encoding algorithm to produce good approximations to the solution to the problem. We use six spatial zones; each one of them represents a small region of the plasma 10 microns in size; this choice is motivated by the spatial resolution of the X-ray imager instruments. Within each spatial zone, temperature and density can take any of 32 uniformly spaced values spread over a relevant range of values. This represents a 5 bit encoding case where the 32 integers from 0 to 31 are mapped into 32 uniformly spaced real numbers in a given range. To avoid sudden changes between adjacent zones, a bound is placed on the maximum relative change. The total number of possible pairs of temperature and density gradient functions that can be generated with this algorithm is 1.2×10^{18} . This

is the size of the parameter space. This large size renders an exhaustive search in parameter space impractical since the evaluation of the spectral physics model is computationally intensive. The idea behind evaluating an individual is to minimize the 2 difference between the experimental and synthetic data. Therefore we measure the fitness or performance of each candidate as $1/2$ (the higher the performance, the better the fit). Equation 1 defines a standard method of measuring the difference between the data and the fits for both emissivities and spectra:

$$\chi^2 = \sum_{i \in exp} \omega_i (I_i^{exp} - I_i^{theor})^2 \quad (4)$$

where I^{exp} and I^{theor} are intensities (emissivities) of experimental and theoretical data respectively and i is a weight factor. A particular choice of the weight factor may have an impact on the performance of the algorithm, it also may be important for estimation of uncertainty intervals (Coldwell, 1991). Since our primary goal was to study the performance of the GA, we have chosen $\omega_i = 1$ for the spectra and $\omega_i = (\frac{1}{I^{exp}})^2$ for the emissivities. This is done to compensate for possible substantial changes in the emissivity profile. The objectives are to find the best possible fits to spectra and emissivities simultaneously. Also, we normalize the objective function for each generation so that the objective function for each criterion ranges from 0 to 1.

3. RESULTS

The parallel algorithm was implemented in C++ with the MPICH library [5][10]. We used a cluster with 128 dual processor nodes connected by a myrinet switch.. Each node consisted of dual P4 Xeon 2.4GHz processors with 512K cache, and 1.5GB of RAM. Since our cluster was a shared cluster, other programs may have been running during our experimental runs. Each set of experimental data, i.e. X-ray line spectrum and spatial distribution of emissivities obtained via Abel inversion of $He-\beta$ and $Ly-\beta$ monochromatic images, was run at least ten times. As for genetic algorithm parameters, we set the mutation rate to 0.1 and used a simple two point crossover with probability of 0.95. The population size was 300, run for 450 generations. In each of these runs, the genetic algorithm driven analysis code was initialized with a different initial random seed; the other input parameters are the same. This is done in another effort to address the uniqueness of the solution issue for this data analysis problem.

3.1 Fitting the physics model

The pareto genetic algorithm successfully finds the temperature and density gradient functions that yield good quality fits to the three objectives using a population size of 300 for 450 generations. Hence, the total number of spectral physics model evaluations is 135,000. This represents a small fraction of the 1.2×10^{18} possible evaluations associated with an exhaustive search of parameter space. This clearly shows that the Pareto genetic algorithm plays a critical role in making this spectroscopic analysis method practical. We show results from one of three experiments with laboratory plasmas. The GA finds parameters for the physics model to fit the three observed profiles that form our criteria. These are the $Ly-\beta$ emissivity, $He-\beta$ emissivity, and spectral intensity. Figure 5 shows that the genetic algorithm is able to find

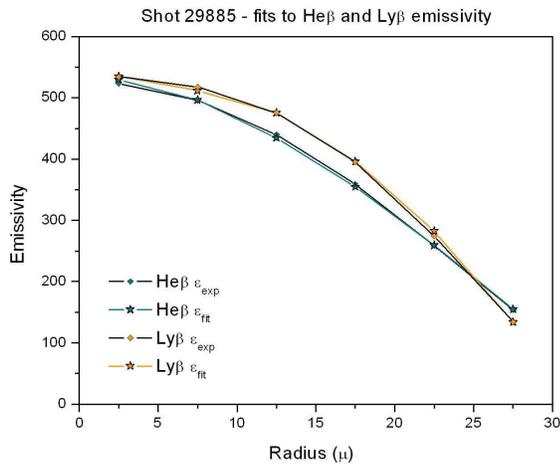


Figure 5: Observed vs. MOGA-fitted Emissivity

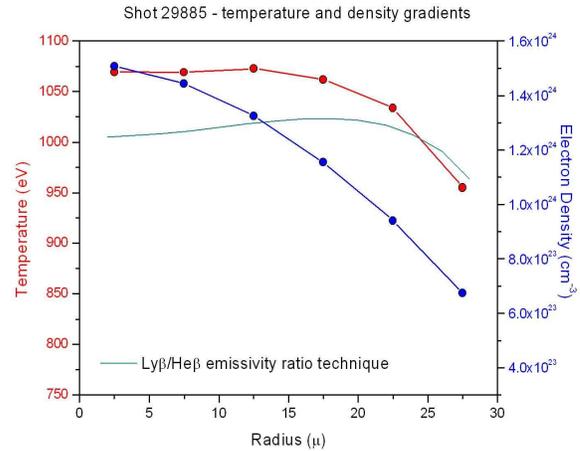


Figure 7: Observed vs. MOGA-fitted temperature and density

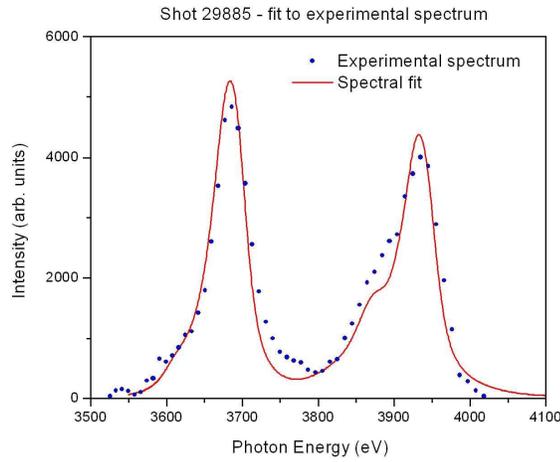


Figure 6: Observed vs. MOGA-fitted spectrum

temperature and density gradients that do a good job in fitting the $Ly\text{-}\beta$ and $He\text{-}\beta$ emissivity profiles. Figure 6 shows that the third criteria, spectral intensity, is simultaneously well fitted.

These results indicate that viability of multi-criteria techniques for our problem and show that the pareto genetic algorithm performs well. Figure 7 shows the fit obtained by a different physics model called the $Ly\text{-}\beta/He\text{-}\beta$ emissivity ratio technique; clearly the pareto genetic algorithm does significantly better in fitting observed experimental data.

By parallelizing the algorithm, we were able to go from about nine hours to under nine minutes for the time taken to fit our physics model to observed data, that makes evolutionary multi-objective optimization techniques a viable approach to attacking X-ray spectroscopic analysis problems. The next subsection addresses our parallel implementation of this pareto GA.

3.2 Speedup

We ran our algorithm implementation on a cluster using up to 150 processors with the same population size of 300 for 450 generations. Figure 8 shows the actual speedup in

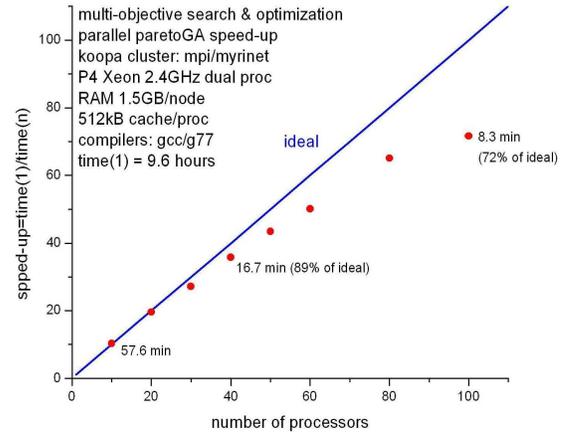


Figure 8: Theoretical vs. actual speedup

our experiment, the x-axis plots the number of nodes, while the y-axis plots the speedup value. As we expected, the speedup does not increase linearly and based on our prediction model, we will get different speedup with different value of the ratio $n\bar{T}_E/T(n)$. Using a ratio of $n\bar{T}_E/T(n)$ of around 300:1, our theoretical model of the parallel pareto GA fits the experimental speedup data quite well. Within the range of nodes tested, the model is very accurate, and we can predict that the speedup will increase if we can add more processors but the rate of increase will slow and finally stop.

We also computed the $n\bar{T}_E/T(n)$ ratio from data gathered during runs of our parallel pareto GA. It is not easy to find the ratio directly in our theoretical model, but if we convert the experimentally obtained speedup data into average load L on each processor and each evaluation, we will get a linear relation between L and P as shown in Figure 9. Then we can do a model fit using the method of least squares.

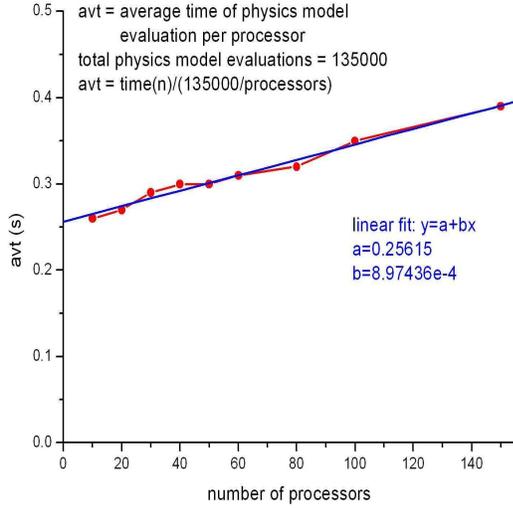


Figure 9: Averaged computation time per evaluation

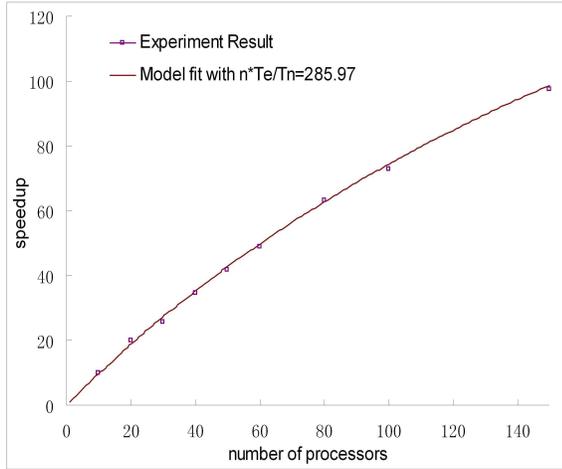


Figure 10: Observed vs. predicted speedup

$$\begin{aligned}
 T_k &= T(n) + \frac{n}{k} \bar{T}_E \\
 \Rightarrow \frac{T_k}{N/k} &= \frac{kT(n)}{N} + \frac{n}{N} \bar{T}_E \\
 \Rightarrow \begin{cases} \frac{n}{N} \bar{T}_E = 0.25615 \\ \frac{T(n)}{N} = 8.97436E^{-4} \end{cases} & \quad (5) \\
 \Rightarrow \frac{n \bar{T}_E}{T(n)} &= \frac{0.25615}{8.97436E^{-4}} = 285.97
 \end{aligned}$$

Now, we can fit the experimental data with the above ratio of 285.97. Figure 10 shows the resulting speedup curve along with the actual speedup obtained; our parallel speedup model fits the experimental speedup data well.

4. CONCLUSIONS AND FUTURE WORK

This paper presented a new parallelizing multi-objective genetic algorithm for application in X-ray spectroscopic analysis. We used a producer-consumer method in distributing

chromosomes within the cluster. This led to a simplified theoretical analysis of performance speedup. Experimental results showed that the new parallel algorithm is useful in solving time-consuming X-ray spectroscopic analysis problems for physicists. In addition, our theoretical speedup framework accurately predicts observed speedup and is a useful tool in analyzing parallel implementations. We are now investigating system scalability and system performance with increasingly complex physics.

5. ACKNOWLEDGMENTS

This material is based in part upon work supported by contract number N00014-03-1-0104 from the Office of Naval Research and by DOE NLUF Grant No. DE-FG03-03SF22696.

6. REFERENCES

- [1] E. Alba and M. Tomassini. Parallelism and evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 6(5):443–462, 2002.
- [2] E. Cantú-Paz. Designing efficient master-slave parallel genetic algorithms. Technical Report IlliGAL 97004, Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign, 1997.
- [3] E. Cantú-Paz and D. Goldberg. On the scalability of parallel genetic algorithms. *Evolutionary Computation*, 7(4), 1999.
- [4] K. Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, LTD, Baffins Lane, Chichester, West Sussex, PO19 1UD, England, 2001.
- [5] E. Gabriel, G. E. Fagg, G. Bosilca, T. Angskun, J. J. Dongarra, J. M. Squyres, V. Sahay, P. Kambadur, B. Barrett, A. Lumsdaine, R. H. Castain, D. J. Daniel, R. L. Graham, and T. S. Woodall. Open mpi: Goals, concept, and design of a next generation mpi implementation. In *Proceedings, 11th European PVM/MPI Users' Group Meeting*, Budapest, Hungary, September 2004.
- [6] I. Golovkin, R. Mancini, S. Louis, K. Fujita, H. Nishimura, H. Shirga, H. Azechi, R. Butzbach, I. Uschmann, J. Deletrez, J. Koch, R. W. Lee, and L. Klein. Spectroscopic determination of dynamic plasma gradients in implosion core. *Physical Review Letters*, 88(4), 2002.
- [7] I. Golovkin, R. Mancini, S. Louis, R. Lee, and L. Klein. Analysis of x-ray spectral data with genetic algorithms. *Journal of Quantitative Spectroscopy and Radiative Transfer*, pages 625–636, 2002.
- [8] I. E. Golovkin, S. J. Louis, and R. C. Mancini. Parallel implementation of niched pareto genetic algorithm code for x-ray plasma spectroscopy. In *Proceedings of the World Congress on Computational Intelligence*, pages 1820–1824, 2002.
- [9] Griem and H.R. Plasma spectroscopy in inertial confinement fusion and soft x-ray laser research. *Plasma Physics*, 15:2346–2361, 1992.
- [10] W. Gropp, E. Lusk, N. Doss, and A. Skjellum. A high-performance, portable implementation of the MPI message passing interface standard. *Parallel Computing*, 22(6):789–828, Sept. 1996.