We use a Pareto optimal genetic algorithm to determine plasma temperature and density through X-ray spectroscopic diagnostics. Pareto selection provides better solutions compared with linear combinations of the two optimization criteria. Parallelizing the genetic algorithm achieves linear speed-up and allows larger populations resulting in improved reliability in reasonable running time.

I. INTRODUCTION

X-ray spectroscopic analysis has proved to be a useful technique to determine temperature and density of astrophysical as well as laboratory plasmas [1]. Routinely analysis is performed manually, that is, using an interactive graphical user interface to compare experimental and theoretical spectra calculated using a particular set of plasma model parameters. When a good fit to experimental data is achieved, the parameters used to calculate the synthetic spectrum are considered to be representative of the state of the plasma during the formation of the spectrum. A criterion for measuring good fits is the distance between synthetic and experimental spectra defined in a given metric as well as visual similarity. This procedure is simple and convenient as long as the spectral model is not very complex, does not depend on too many parameters, and is relatively inexpensive (can be performed in real time). For many other cases the availability of a computer-driven automated analysis procedure is important. In this paper we will use a genetic algorithm to estimate plasma temperature and density gradients for high energy density plasmas.

Traditional (as described above) spectroscopic analysis has been used to determine averaged or effective temperatures \( T_e \) and densities \( N_e \). However, the spectroscopic analysis of plasma temperature and density gradients for Inertial Confinement Fusion (ICF) experiments. The goal here is to find temperature and density gradients that simultaneously produce good fits to 1) time-resolved spatially integrated X-ray line spectra and 2) X-ray monochromatic emissivity profiles. Spatial emissivity profiles can be extracted from Abel inversion of X-ray monochromatic images provided that the plasma is optically thin and spherically symmetric (see Figure 1).

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 is one of the first attempts in this direction. 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.

Since we need to simultaneously fit both spectra and emissivity the problem involves multi-objective optimization and makes it especially difficult to fit the data by hand. Spectroscopic analysis for non-linear models is also difficult for conventional minimization schemes and exhaustive
searches can be prohibitively expensive. We have reported previously [2] that Genetic Algorithms (GAs) perform very well when applied to spectroscopic diagnostics. In this paper we show that a Pareto optimal Genetic Algorithm is a robust and efficient tool to analyze experimental spectra and monochromatic images. Note that in contrast to many multi-criteria optimization problems we are less interested in the Pareto front and more interested in a single point that optimizes both our criteria. Since Pareto optimality makes no assumptions about the relationship between the two criteria that we are trying to simultaneously optimize, we find it suitable for our needs.

Genetic Algorithms are search algorithms based on the mechanics of natural selection [3]-[4]. They are capable of finding a solution in a poorly understood search space while exploring only a small fraction of the space and can robustly deal with complex non-linear spaces. A GA’s reliability, robustness, ease of use, and speed were our primary motivations to apply it for our purposes. In addition, spectral analysis has a threshold of sensitivity determined by the quality of the experimental data and by the model’s sensitivity with respect to parameter changes. Since GA’s are especially proficient at finding promising areas (corresponding to solutions within our threshold) of the search space, they are well suited for our problem.

We use the principles of Pareto Optimality in designing a Pareto optimal genetic algorithm [5]. This algorithm is well known and widely used. Moreover, our existing code could be readily modified to be used as a basis for the new niched Pareto genetic algorithm. Although there are several other multi-objective optimization schemes that can be applied to spectroscopic analysis, studying the performance of these algorithms can be a subject of further investigation.

At each generation, there is a set of non-dominated solutions that form a surface known as the Pareto optimal front (or Pareto front). The goal of a Pareto optimal GA is to find and maintain a representative sampling of the solutions on the Pareto front. If the criteria are not self-contradictory, there should be a point on the final convex front that satisfies all criteria well. In our case, this point will be considered as the solution to the spectral analysis problem (Figure 2).

If there is no such point (concave front), an expert decision has to be made about which (if any) solution on the concave Pareto optimal front represents the most acceptable, physically sound solution. Analysis in this case may not be reliable because a self-consistent high quality spectral model should be able to describe all radiative properties of the plasma. Tracing the Pareto optimal front also helps address the issue of solution uniqueness.

In this paper, we show that the method we developed is capable of determining correct gradients. To this end, we performed numerous calculations based on the analysis of synthetic experimental data. Running our spectral model using known gradients as input produces synthetic data. We add noise to the generated spectra and emissivity profiles to represent experimental uncertainties.

Successful analysis should be able to recover the gradients we used to construct the synthetic data, by fitting emissivities and spectra. We will show, however, that under certain circumstances GA can find alternative gradients that produce good fits to both emissivities and spectra. It is therefore very important to understand the limitations of the method before applying it to the analysis of real experimental data with unknown gradients.

As an application, we consider analysis of Ar Heβ and associated Li-like satellites spectral feature observed during the collapse of Ar-doped laser-driven ICF implosions. This feature is widely used for diagnostic purposes. Based on the spectra of this feature effective emissivity-averaged plasma density and temperature can be determined. Combined with the information that can be obtained from monochromatic X-ray images of the plasma (photon energy at the center of Ar Heβ line) the spectra can be used to bracket temperature and density spatial gradients. Our spectral model is a collisional-radiative atomic kinetics steady state model that includes the effects of Stark broadening, line overlapping, and opacity [6].

Each evaluation requires execution of the spectral model, which is somewhat expensive. In order to reduce computing time we parallelized our GA so that evaluation is done in parallel (Figure 3). All the genetic operators, most importantly selection, may act upon every individual, and therefore the behavior of the algorithm is the same as that of sequential GA. This kind of parallelization is quite straightforward and relatively easy to implement. It should result in significant speed-up as long as the cost to evaluate an individual dominates communication cost [7].

![Figure 3. A Schematic View of a Global Parallel GA](image)

Our results indicate that spectroscopic analysis driven by the Pareto optimal Genetic Algorithm is capable of determining plasma temperature and density gradients. The method is generalizable, and with minor modifications can be used to drive searches in other plasma spectroscopy applications. We achieved almost perfect linear speed-up for our parallel code. Performance of the algorithm improves when using larger population sizes.

The next section describes our implementation of the algorithm that results in good performance. Then we will
discuss the results of the computations. Finally, we will present our conclusions and some guidelines for future work.

II. IMPLEMENTATION

The 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 gradient choices to 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 encoding of a parabolic gradient.

\[
\begin{align*}
\text{Temperature} & : T(r) = a_1 r^2 + c_1 \\
\text{Density} & : N(r) = a_2 r^2 + c_2
\end{align*}
\]

| 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 |

\( T(0) \quad T(R) \quad N(0) \quad N(R) \)

[260 eV - 1500 eV] \quad [1.00 \times 10^{23} \text{ cm}^{-3} \quad 1.65 \times 10^{24} \text{ cm}^{-3}]

Figure 4. Encoding of Temperature and Density

The idea of the analysis is to minimize the difference \( \chi^2 \) between the experimental and synthetic data; therefore we measure the fitness or performance of each candidate as \( 1/\chi^2 \) (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:

\[
\sum_{i=exp}^{n} \omega_i (I_i^{exp} - I_i^{theor})^2 ,
\]

where \( I_i^{exp} \) and \( I_i^{theor} \) are intensities (emissivities) of experimental and theoretical data respectively and \( \omega_i \) 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 [8]. Since our primary goal was to study the performance of the GA, we have chosen \( \omega_i = 1 \) for the spectra and \( \omega_i = (1/I_i^{exp})^2 \) 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.

One approach to solve this problem is to combine the multiple 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 the Niched Pareto Genetic Algorithm [5].

The crucial difference between a canonical GA and the Niched Pareto GA is in the implementation of selection. We implemented Pareto domination tournament selection where two candidates are picked at random from the population. A comparison set of individuals is also picked randomly from the population. Each of the candidates is then compared against each individual in the comparison set. If one candidate is dominated by the comparison set, and the other is not, the latter is selected for reproduction. If neither or both are dominated by the comparison set, then we must use sharing to choose a winner. Equivalence class sharing implemented in our model defines the winner as an individual that has the smallest number of the other individuals inside its niche. This technique helps to maintain diversity along the Pareto front. Niche size gets adjusted automatically for each generation based on the average area of the front. We also normalize the objective function for each generation so that the objective function for each criterion ranges from 0 to 1.

In order to increase selection pressure we use an elitist scheme where: 1) members of the current generation and offsprings are combined in a common pool in each generation; 2) the solutions along the Pareto front are selected for the next generation and removed from the pool; 3) the procedure is repeated until the next generation is filled. We have found empirically that elitism combined with uniform crossover provides reliable and rapid convergence. The size of the comparison set controls selection pressure. However, when using our elitist scheme, the algorithm is not very sensitive to size. In our implementation we compare each candidate against 5 individuals.

Probability of crossover and mutation is 0.95 and 0.05 respectively. We performed a number of runs varying GA parameters to ensure reliability and optimize the performance of the algorithm. The implementation we discussed above is well suited for our purposes.

We developed and tested our parallel code on an 8-node Beowulf machine (PII - 400, 100Mbs) using PVM 3.4 for distributing the GA’s evaluations. As noted before, evaluation of the individuals is performed in parallel and is done by assigning a certain fraction of the population to each processor. Evaluation of each individual is completely independent of the other individuals in the population. Communication occurs only when each slave receives its fraction of the population to evaluate and when it returns fitness values to the master. As long as evaluation takes a long time compared to communication, this scheme is very efficient. In our case each evaluation takes approximately 1.6 seconds, which is much longer than communication time (milliseconds). Therefore we expect very good speed-up.

Although it is possible to parallelize the genetic operators as well, 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. Selection in our case is relatively complex, but parallelization will require constant communication between the processors. Our results support this analysis.

We can control the size of the sub-population (packet) that is being sent to the slaves for evaluation. In order to have good load balance, population size should be divisible by the number of slaves times packet size. We did not observe any noticeable change in performance for different packet sizes as long as this condition is satisfied. For all the results presented in this paper the packet size is equal to 5.
III. RESULTS

Level of noise in real experimental data can vary from a few percent (high quality data) to 20 percent (poor data). We performed a series of calculations with the following noise levels: 1, 5, 10, 15, and 20%. For each case we performed 5 runs starting with different initial random seeds. We used simple parabolic gradients to generate the data in all cases and parabolic functions to encode solutions. Two parameters are required to define a parabola for both temperature and density. We used 5 bit encoding for each parameter, which results in a 20-bit chromosome. With 100 individuals in each generation we achieve rather stable performance.

We now discuss the results of the GA’s runs for 10% noise in more detail. The propagation of the Pareto front in the objective space shows the dynamics of the run. Figure 5 displays propagation of the front for a typical run.

![Figure 5. Propagation of Pareto Front](image)

The behavior of the front indicates that our GA successfully drives the search towards better solutions. The upper right point represents the solution that has good spectral and emissivity fits (Figures 6 and 7).

![Figure 6. Good Spectrum Fit (10% Noise)](image)

![Figure 7. Good Emissivity Fit (10% Noise)](image)

It is important to study the points in the vicinity of the right solution. If the gradients for these points are similar to the one for the correct solution then the analysis is successful. The family of gradients then is indicative of the uncertainty intervals of the analysis. If, on the other hand, there are some points nearby that satisfy all criteria well but nevertheless have different gradients, then we have alternative solutions and the analysis is ambiguous.

It is intuitively clear that the higher the level of noise, the larger the chances to find alternative solutions. For 1, 5, and 10% of noise the solution is unique. Starting at 15% noise we see alternative solutions. Analysis becomes ambiguous since any pair of the gradients produces acceptable fits to spectra and emissivities and no preference can be given to either of the solutions.

![Figure 8. Speed-up vs. Number of Cores](image)

Another interesting aspect is the ability of the genetic algorithm to satisfy all objectives. Table 1 summarizes statistics for different levels of noise.

<table>
<thead>
<tr>
<th>NOISE LEVEL</th>
<th>ALL OBJECTIVES SATISFIED</th>
</tr>
</thead>
<tbody>
<tr>
<td>1%</td>
<td>1 run out of 5</td>
</tr>
<tr>
<td>5%</td>
<td>3 runs out of 5</td>
</tr>
<tr>
<td>10%</td>
<td>5 runs out of 5</td>
</tr>
<tr>
<td>15%</td>
<td>5 runs out of 5</td>
</tr>
<tr>
<td>20%</td>
<td>5 runs out of 5</td>
</tr>
</tbody>
</table>

We performed several runs to study performance of the parallel code. Figure 8 shows speed-up vs. number of
processors for data with 10% noise and population size 100. The speed-up is almost linear for modest number of processors.

![Linear Regression: $Y = A + B \times X$](image)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.00501</td>
</tr>
<tr>
<td>B</td>
<td>0.97631</td>
</tr>
</tbody>
</table>

Figure 8. Speed-up vs. Number of Processors

As the number of processors grows, cost of communications increases accordingly. This may result in performance deterioration. However, some recent runs on a 20-node Beowulf machine indicate linear speed-up. Thus, evaluation time still dominates communication time. The significant reduction in computing time because of parallelization allowed us to increase population sizes so that we could increase the reliability with which good solutions were found. The serial code with a population size of 100 took about four (4) hours to run for 100 generations. The increased population size significantly improves the reliability of the algorithm (Table 2). The quality of the solution, however, does not improve noticeably.

<table>
<thead>
<tr>
<th>NOISE LEVEL</th>
<th>ALL OBJECTIVES SATISFIED</th>
</tr>
</thead>
<tbody>
<tr>
<td>1%</td>
<td>4 run out of 5</td>
</tr>
<tr>
<td>5%</td>
<td>5 runs out of 5</td>
</tr>
<tr>
<td>10%</td>
<td>5 runs out of 5</td>
</tr>
<tr>
<td>15%</td>
<td>5 runs out of 5</td>
</tr>
<tr>
<td>20%</td>
<td>5 runs out of 5</td>
</tr>
</tbody>
</table>

TABLE 2: NUMBER OF SUCCESSFUL RUNS FOR DIFFERENT NOISE LEVEL (POPULATION SIZE: 200)

IV. CONCLUSIONS

We studied the possibility of using Pareto optimal genetic algorithms for the estimation of plasma temperature and density gradients by performing simultaneous analysis of experimental X-ray spectra and monochromatic images. This information may be used to improve characterization of core plasma dynamics and to provide new data for detailed benchmarks of hydrodynamic codes.

The algorithm performs well when we analyze synthetic data produced using simple parabolic gradients. The Genetic Algorithm is capable of finding quality solutions while exploring only a tiny part of the search space. We have shown however that unambiguous determination of the gradients is sensitive to the quality of the experimental data.

We developed a parallel implementation of the code to be run on our in-house Beowulf machines. We achieved almost perfect linear speed-up using global parallelization scheme. This allows us to increase population sizes and improve reliability of the search for complex problems and thus establishes the utility of our approach for this problem.

Acknowledgments

This work was supported in part by DOE High Energy Density Science Grant No. DE-FG03-98DP00213 and in part by NSF Grant No. 9624130.

References