Staged Hybrid Genetic Search for Seismic Data Imaging

Keith E. Mathias,† L. Darrell Whitley,† Christof Stork†† and Tony Kusuma††

Abstract — **Seismic data interpretation problems are typically solved using computationally intensive local search methods which often result in inferior solutions. Here, a traditional hybrid genetic algorithm is compared with different staged hybrid genetic algorithms on the geophysical imaging static corrections problem. The traditional hybrid genetic algorithm used here applied local search to every offspring produced by genetic search. The staged hybrid genetic algorithms were designed to temporally separate the local and genetic search components into distinct phases so as to minimize interference between the two search methods. The results show that some staged hybrid genetic algorithms produce higher quality solutions while using significantly less computational time for this problem.**

I. Introduction

Genetic algorithms are a population based method for globally sampling a search space and are capable of finding highly competitive solutions for optimization problems. They are tolerant of noise, work well in problem spaces where there are a large number of local optima and where gradient information cannot be calculated or is difficult to obtain [9]. These characteristics suggest that genetic algorithms might be useful in solving complex problems like the geophysical imaging static corrections problem.

The static corrections problem belongs to a class of complex geological problems. Geological exploration employs seismic reflection surveys to obtain subsurface images of geologic beds. These images can be distorted by ambient noise and heterogeneities in subsurface materials. Temporal adjustments, known as static corrections, can be applied to the signals to correct the images. Determining the correct static shifts is very difficult. This is due to the highly non-linear characteristics of the problem, large numbers of local optima and signal noise. Additionally, static corrections problems are quite large, with typical search space sizes on the order of 2^{6000} points. Geophysical static corrections are typically found using computationally intensive local search methods but they often produce unsatisfactory results. Therefore, the application of global search methods that are not sensitive to noise have been explored.

Early attempts to use simple genetic algorithms for solving the static corrections problem have proven to be computationally expensive [6, 11]. Nevertheless, some improvements in performance have been obtained by combining local search methods with genetic algorithms [7]. The coupling of genetic and local search methods to form *hybrid genetic algorithms* can be advantageous. Genetic algorithms converge on globally competitive solutions irrespective of local optima in the search space. Local search can potentially improve on the solutions discovered by genetic algorithms by climbing to the optima of their corresponding attraction basins.

Traditional hybrid genetic algorithms apply local search to every solution generated by the genetic search. This can significantly increase the time of the search as well as potentially inhibit the global sampling mechanisms of genetic search. Applying local search to *convergence* to every solution generated by genetic search is not only computationally expensive, it also biases the population such that it only contains samples representing local optima in the search space. Therefore, the two search methods should be coupled so that they search autonomously and interface only periodically.

The performances of four types of genetic algorithms were compared on the static corrections problem. All of the algorithms employed steepest ascent to form the initial genetic population and they also applied steepest ascent to some portion of the final genetic population. The first genetic algorithm compared was a basic GENITOR [8] algorithm that did not employ any local search other than as described above. The second algorithm used was a *traditional* hybrid simple genetic algorithm (SGA) [3]. This model used a generational style genetic algorithm and applied steepest ascent to the entire population after every generation. The other two algorithms tested were staged hybrid genetic algorithms.

There are several new and effective genetic algorithm strategies which have been introduced in the last few years that exploit some form of "staged" search [2, 4, 10]. These algorithms apply the genetic algorithm in the usual way for a specific period of time, constituting one stage of the search. Then the algorithm is stopped and the best strings in the population can be used to reinitialize the population so that search can be continued. Following each stage of genetic search, local methods can be applied to selected strings in the population. The resulting solutions can then be used to re-initialize the population for the next stage of genetic search.

One of the staged hybrid algorithms compared here employed an SGA for genetic search and applied steepest ascent to the entire population every 10^{th} generation. The other staged hybrid genetic algorithm tested used GEN-

†Department of Computer Science, Colorado State University; Fort Collins, CO 80523. (mathiask, whitley@cs.colostate.edu)

††Advance Geophysical Corporation, 7409 S Alton Ct, Englewood; CO 80112. (stof, tony@packhorse.advance.com)

This research was supported in part by a grant from the Colorado Advanced Software Institute.

ITOR as the genetic search component. In this staged hybrid algorithm steepest ascent was only applied to the best population members every 10,000 trials. These improved strings were then used to probabilistically re-seed the population for each genetic search interval.

The staged hybrid GENITOR algorithm resulted in better quality solutions than those found by the other algorithms tested here and required less computational time. The solutions generated by the staged hybrid GENITOR algorithm were approximately 7% better than those yielded by the staged hybrid SGA and took approximately $\frac{1}{2}$ as much CPU time to generate. Additionally, the staged hybrid GENITOR algorithm generated solutions that were approximately 5% better than the traditional hybrid SGA and basic GENITOR algorithm.

II. The Geophysical Static Corrections Problem

There are several large optimization problems in geophysics for which locating good solutions is both difficult and computationally time consuming. One such problem in seismic imaging is known as the "static corrections" problem [7]. The geological industry uses seismic reflection surveys to construct subsurface images of geologic beds. These images can be distorted by the change in surface materials and ambient noise. These images can be corrected by application of static corrections or shifts. However, it is very difficult to determine the correct static shifts because of the highly non-linear characteristics of the problem and the existence of many local optima [7].

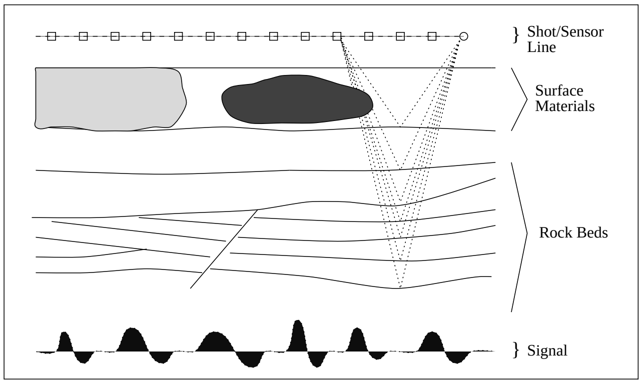

Figure 1: Sensor Shot Signals.

Seismic reflection surveys are performed by setting off a detonation (shot) and then recording the resulting signals at a series of sensor sites. Figure 1 illustrates the reflection signals that a single sensor (□) would collect for a single shot (∘) as well as the waveform corresponding to the signal. Each of the sensors along the sensor-line would collect a similar set of reflection signals for this shot. A number of shots would be detonated over time, moving the shot site to the adjacent sensor site, for each new signal collection (top of Figure 2).

The signals collected at each sensor can be filtered and re-combined so that all reflections occurring at common midpoints are grouped into corresponding synthesized traces. Figure 2 illustrates the reflection signals that would be grouped into a single common midpoint gather. These synthesized traces are then stacked and rotated to form a distorted image of the subsurface strata. The collection of all of these common midpoint gathers is known as a *stack*. Applying the true temporal corrections to each signal results in an accurate image of the subsurface strata.

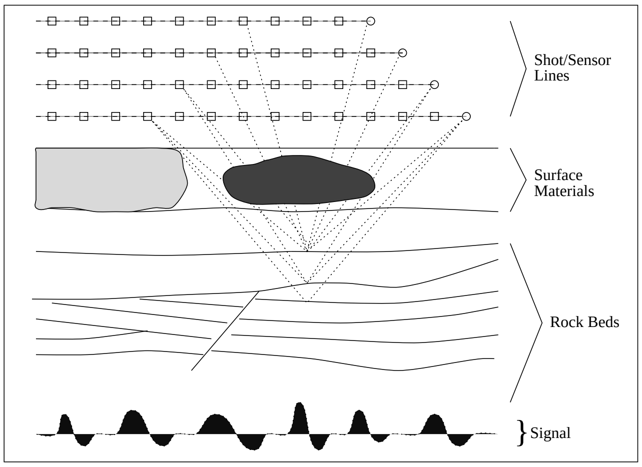

Figure 2: Common Mid Point Gather.

There are two types of temporal phenomena affecting the traces in a common midpoint gather. The first type of temporal phenomenon results from the difference in materials at the earth surface. For example, the signal leaving the shot site may be delayed when it passes through very loosely packed material such as sand. However, the material directly under the sensor may be dense, such as clay, which would increase the velocity of the signal. Thus, each signal in the common midpoint gather has two independent "static" correction values. The static corrections are the parameters which we wish to optimize.

The second type of temporal phenomenon affecting the traces in a common midpoint gather is due to the difference in distance that the signals must travel. Since the signals come from various sensor/shot pairs that are not equidistant from the reflection point, the signals reflecting from a common depth point will experience different delays. The temporal correction for this effect is known as the *normal move out* (NMO). The NMO for a trace can be approximated since the temporal differences between signals at a common depth point are directly related to the distance traveled. It is important to note, however, that there are minor adjustments in the NMO that depend on the specific set of "static" corrections that are used.

Conventional methods for finding static corrections include gradient methods that locate the nearest local optima. To be useful, these results must be "close" to the global optimum. However, these methods are unacceptable when severe surface heterogeneities exist or when high levels of ambient noise are present.

Two common static correction methods are cross-correlation picking and maximum stack power optimization. Cross-correlation picking determines the set of static shifts by aligning peaks of the cross-correlation signals with a master trace in the common midpoint gather. Maximizing the power of the "stack" involves calculating the cross-correlation between all pairs of reconstructed signals corresponding to the common midpoint gathers.

There have been several attempts to use basic genetic algorithms to solve the static corrections problem but they have proven to be computationally ineffective [6, 11]. Implementations by Stork and Kusuma [7] combining gradient ascent methods with genetic algorithms have yielded significantly improved performance. However, the method is still computationally expensive. Any algorithmic improvement resulting in decreased computational time or higher quality solutions for the static corrections problem would be beneficial to the geological industry.

III. Algorithm Descriptions

A. Genetic Algorithms and GENITOR

Genetic algorithms are a class of computational search algorithms based on processes observed in organic evolution such as natural selection, genetic recombination, and mutation. Using a genetic algorithm for parameter optimization requires that the variables of the function be encoded onto a string. The parameters can be encoded using a binary alphabet where a one-to-one mapping between each binary pattern and a parameter value is established, thus discritizing the solution space [9]. The search process begins by randomly generating a population of $\mathcal{N}$ strings. Each string is then decoded and evaluated by an *objective function* [9].

The GENITOR algorithm uses a population where the strings in the population are kept in ranked order from the most highly fit string in position 1 to the worst in position $\mathcal{N}$ (where $\mathcal{N}$ is the population size). After the population is initialized and sorted, strings are paired for recombination using a linear bias selection mechanism. Crossover is applied to the strings selected for reproduction. One of the offspring is chosen for evaluation via the objective function and inserted into the population, ranked with respect to it's fitness. The worst individual in the population is discarded. This guarantees that the $\mathcal{N} - 1$ best solutions found during the search are always in the population. This mating process is called a *trial*.

The basic GENITOR style algorithm tested here employed a population of binary strings seeded from a small number of candidate solutions (e.g. 5). The candidate solutions are developed from *a-priori* information about signal correlations. Then waveform steepest ascent is iteratively applied to these candidate solutions. The improved solutions are converted to binary representations which serve as templates for initializing the genetic population. Each original template is used to seed an equal portion of the population. This is done by repeatedly mutating some percentage (e.g. 15%) of the bits in the template strings and placing the resulting strings in the population. Seeding the population by mutating templates in this fashion is known as cataclysmic mutation [2]. None of the original templates are included in the population.

The NMOs for each of the candidate solutions are averaged to form a single NMO that serves as the NMO for each member of the population during evaluation. The evaluation function used to evaluate each candidate solution returns the cross-correlation power of the stack. Genetic search, using the GENITOR algorithm, is then allowed to proceed until the population has converged. No local search is applied to the offspring during the genetic search. After the genetic search has been terminated the best solutions are converted to floating point model representations and waveform steepest ascent is iteratively applied as a final step.

B. Hybrid Genetic Algorithms

Hybrid genetic algorithms are formed by coupling genetic search with a local search method. These two search methods are complementary in that genetic algorithms have been shown to converge on globally competitive solutions, but they typically are not aggressive at exploiting local improvements. Local search methods, on the other hand, are designed to climb from a given starting point in the search space and locate the local optimum of the corresponding attraction basin. Therefore, applying local search methods to globally competitive solutions discovered by genetic search has the potential for discovering even better solutions by climbing to a local optimum.

Typical hybrid genetic algorithm implementations embed the local search component in the reproduction process, locally improving every offspring generated by the genetic search component [1, 5]. However, this can potentially inhibit the global sampling mechanisms of the genetic algorithm. Additionally, applying local search algorithms to each offspring generated can be computationally expensive.

If strings are improved by local search methods and then placed back into the population, then the genetic algorithm in essence is exploiting a form of Lamarckian evolution. One criticism of Lamarckian evolution is that changing information in the offspring results in a loss of the statistical information about hyperplane partitions that is implicitly contained in the population [9]. The resulting search could be less global, since it is hyperplane sampling that is used to explain the global sampling behavior of genetic search.

Despite the theoretical concerns, hybrid genetic algorithms typically do well at optimization tasks. Nevertheless, from a theoretical viewpoint it would be better to combine local search with the genetic algorithm so that the local search does not interfere with the hyperplane sampling of the genetic algorithm. Therefore, we compared a traditional hybrid genetic algorithm and two different implementations of staged hybrid genetic algorithms that were designed to temporally separate the local and genetic search components in a simple and straightforward manner. The local search component in all of these algorithms is a form of waveform steepest ascent. Steepest ascent is applied for a fixed number of iterations, and thus does not necessarily converge to a local optimum [7].

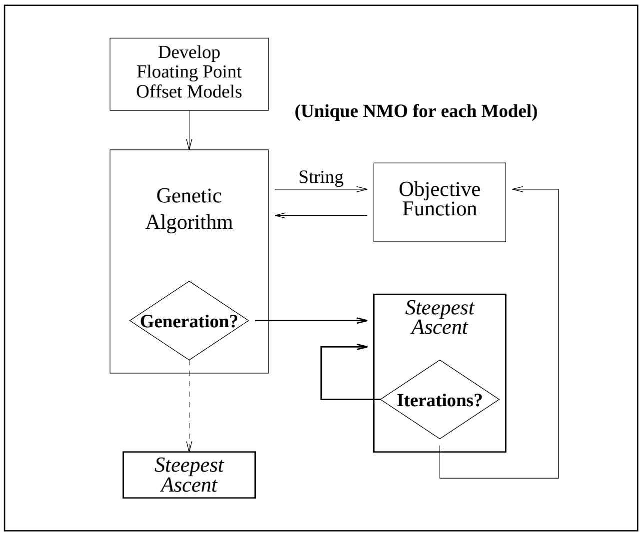

Figure 3: Generic Hybrid SGA Model.

The Traditional Hybrid SGA Model

The traditional hybrid SGA is an elitist SGA (Figure 3). Waveform steepest ascent is **repeatedly** applied to *every member of the population at each generation*. The elitist SGA always copies the best individual encountered up to that point in the search into the population of the next generation.

The traditional hybrid model utilizes a population of floating point strings which correspond to the static corrections for each of the traces in the stack. The entire population is initialized using the same *a-priori* information described earlier for the GENITOR algorithm, however each member of the population has a unique NMO. Genetic crossover is applied only to the static shift corrections in the model. However, the steepest ascent algorithm modifies both the static shift corrections and the NMO for each individual. The search is concluded by iteratively applying waveform steepest ascent to the final genetic population for 7 iterations.

The Staged Hybrid SGA Model

The design of the *staged* hybrid SGA differs from the traditional hybrid SGA in that after the initial population has been improved via waveform steepest ascent, genetic search is allowed to continue uninterrupted for 10 generations. At the end of this *genetic search stage*, a single iteration of waveform steepest ascent is applied to each individual in the population. This temporally separates the two search methods so as to limit their interference and decrease the computational time related to the steepest ascent component of the search. This allows the genetic algorithm to perform more work in the search process. This cycle is repeated for some number of generations and concludes by applying waveform steepest ascent to the entire population for 7 iterations. This process is also illustrated in Figure 3 where steepest ascent *iterations* equals 1 and *generations* equals 10.

The Staged Hybrid GENITOR Algorithm

The *staged* hybrid GENITOR algorithm utilizes a population of binary strings. The strings in the population and the NMO used for string evaluation are developed using the same initialization procedures described for the basic GENITOR algorithm. However, genetic search for the staged hybrid GENITOR algorithm is only allowed to proceed for some predetermined number of trials. No local search is applied during the genetic search stage.

After the current "stage" of genetic search is terminated, the best solution in the population is converted to a floating point model and inserted into the original set of improved models used to seed the population. A single iteration of waveform steepest ascent is applied to this set of models. The models are then converted back to binary strings and serve as templates to re-initialize the new population.

The NMOs associated with the set of improved models are again averaged, producing a single NMO to be used by all strings for evaluation. Genetic search is then resumed. This staged process isolates genetic and local search in distinct and separate phases for several iterations. Upon the completion of the final iteration, the best solutions are converted to floating point model representations and waveform steepest ascent is iteratively applied 7 times as a final step. This algorithm is outlined in Figure 5.

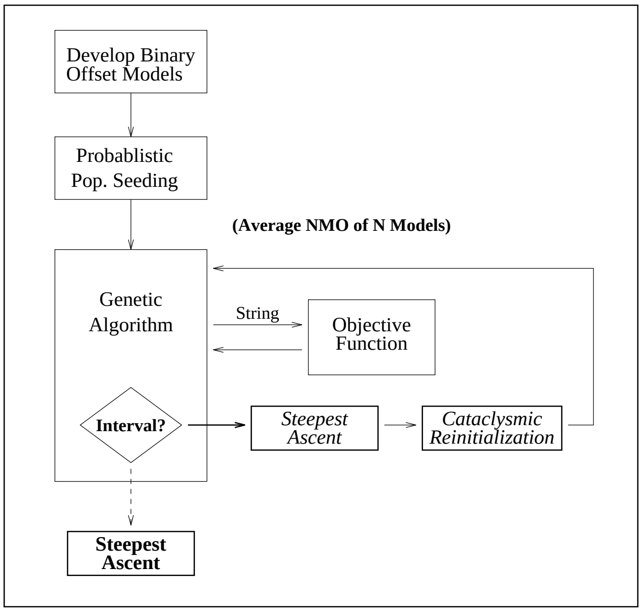

Figure 5: Staged Hybrid GENITOR Algorithm.

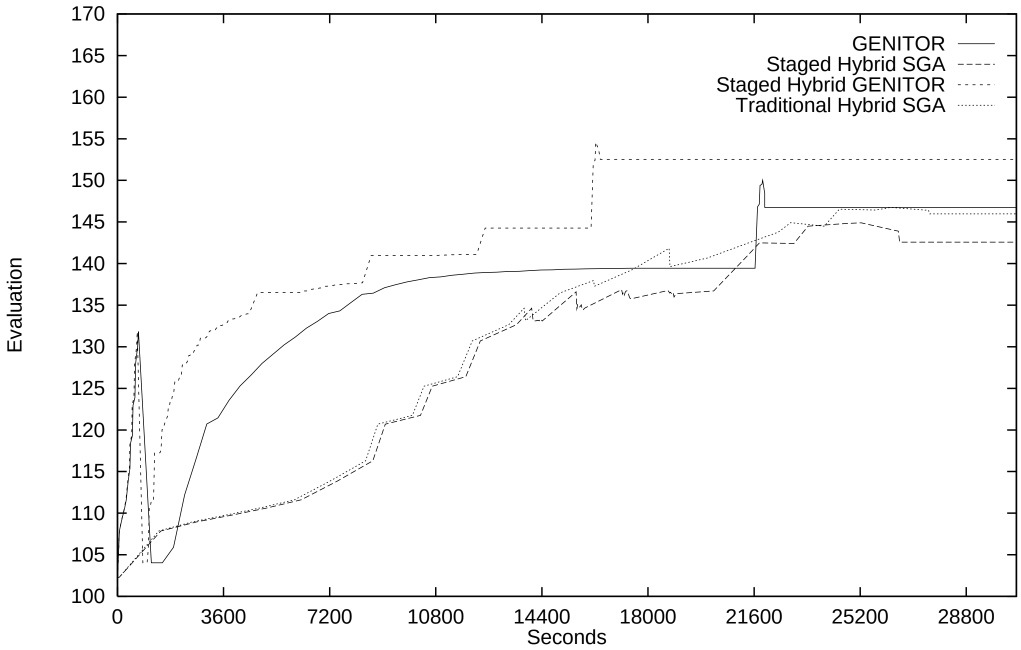

Figure 4: Timing Performance Comparisons. Note that 3,600 seconds represents 1 hour of computation.

IV. Results

The performance curves shown in Figure 4 indicate the progress, as measured by the best individual in the population, for each of the algorithms in finding static corrections to a synthetic stack over the course of time. These curves are single runs typical of the average performance for each of the algorithms. The "Evaluation" label on the Y axis is the cross-correlation power of the stack returned from the objective function. The time on the X axis is represented in CPU seconds as recorded on Sparc II machines.

All of the algorithms tested here applied waveform steepest ascent 14 times during the initialization process and 7 times at the completion of the search process. The basic GENITOR algorithm used a population of 750 strings and a selection bias of 1.25. The population was seeded as described previously using 5 strings. Genetic search was terminated when the population converged. This occurs after approximately 6 hours. The search makes rapid progress initially. However, minimal improvement is noted after approximately 3 hours of genetic search (approximately 54,000 trials).

In addition to the iterative application of steepest ascent during the initialization process, the traditional hybrid SGA applied 2 passes of the waveform steepest ascent algorithm to every string in the population after each generation. A population of 100 strings was used. The computational time necessary to initialize the population and iteratively apply steepest ascent prohibited running the algorithm for more than 3 generations. The staged hybrid SGA however, only applied a single iteration of the waveform steepest ascent to every member of the population every 10 generations. Thus, 40 generations were completed in approximately the same amount of time used by the traditional hybrid SGA. The staged hybrid SGA also used a population of 100 strings.

Both the traditional hybrid SGA and staged hybrid SGA applied the steepest ascent algorithm to the entire population of strings during initialization. Therefore, genetic search did not even begin until almost 4 hours elapsed. The search time for each algorithm is 7 hours. Less than $\frac{1}{3}$ of that time is used for genetic search in the staged hybrid SGA and even less time is dedicated to genetic search in the traditional hybrid SGA.

In the final stages of search the steepest ascent often results in decreases in the objective function value. This is partially due to the method used to match NMOs to offspring. It is difficult to know which parent's NMO should be applied to which child and exhaustively testing all possible combinations is very time consuming as the evaluation function is computationally expensive. The final climb and plateau seen in both the traditional hybrid SGA and staged hybrid SGA performances beginning just after the 6 hour point indicates the beginning of the final iterative application of the steepest ascent algorithm.

The staged hybrid GENITOR algorithm tested here used a population of 300 strings and a linear selection bias of 1.5. The cataclysmic mutation rate was set to 15% and a single iteration of steepest ascent was applied every 10,000 trials. The staged hybrid GENITOR algorithm search produces final models with superior objective function values and does so in less time than the other algorithms tested here.

The staged hybrid GENITOR search, like the basic GENITOR search, displays a characteristic peak followed by a decrease in the objective value of the best model in the population around the 20 minute point. The peak represents the termination of the initial iterative application of the waveform steepest ascent algorithm to the candidate models. The dramatic depression following the peak is the result of excluding the best candidate models from the initial genetic population. The depression occurs slightly later for the basic GENITOR algorithm as there are more than twice as many strings to seed in the population. Although excluding the seed strings from the population seems counterproductive, it has been empirically established that inclusion of the best model in the genetic population dramatically influences the search such that the end results are significantly inferior to those produced when the best solution is initially excluded from the population.

Additionally, the peak found at the termination of both the staged hybrid GENITOR and basic GENITOR searches corresponds to the final iterative application of steepest ascent. The decrease in fitness of the best member of the improved population for the staged hybrid GENITOR is similar to that noted in the hybrid SGA algorithms. However, the basic GENITOR algorithm appears to exhibit a more dramatic decrease in the fitness of the best individual at that point. This may indicate that the optimization performed by the steepest ascent algorithm on the NMO is significant and periodic application is beneficial in adjusting the NMO for the model.

V. Evaluation

The staged hybrid GENITOR algorithm produces static corrections models with objective function values more than 7% better than the staged hybrid SGA and does so approximately 3 hours faster. Additionally, the staged hybrid GENITOR model produces solutions to the static corrections problem that are approximately 5% better than the other two algorithms and does so in significantly less time. These time savings are primarily attributed to the way in which the waveform steepest ascent algorithm is applied. The staged hybrid GENITOR algorithm performs better and faster than the basic GENITOR search algorithm indicating that periodic local search is indeed beneficial. However, the long execution times and inferior results of the SGA algorithms indicate that application of local search to large numbers of strings is computationally expensive and may not result in solutions commensurate with the CPU time expended.

A careful study of the performance curves associated with the "staged" algorithm shows separate intervals of slow improvement separated by rapid improvement. The sharp inclines indicate improvement associated with the application of steepest ascent. The quick improvements noted during the local search stages might lead one to believe that it would be advantageous to skip the genetic stages of the search and iteratively apply only the steepest ascent. However, the genetic search stages actually support the local search by moving the search to new points in different hyperplane partitions that provide new points in the search space from which to climb. The best model from a previous iteration may represent a local optimum, in which case it could not be improved by repeated iteration of the waveform steepest ascent algorithm. In general, the repeated application of steepest ascent at a single point in the search space yields diminishing returns and inferior solutions.

In conclusion, we note that to the best of our knowledge the seismic data imaging problem presented here is one of the largest and most difficult parameter optimization problems that has been attempted using genetic algorithms to date. Additionally, this work has shown that the staged approach to combining local and genetic search, previously used for large traveling salesman problems [4], results in improved performance for large binary encodings as well.

Acknowledgements

Our thanks to Raja Das, John Dzubera and Scott Gordon for making suggestions along the way and reading earlier drafts of this work.

References

[1] Lawrence Davis. *Handbook of Genetic Algorithms.* Van Nostrand Reinhold, New York, 1991.

[2] Larry Eshelman. The CHC Adaptive Search Algorithm. How to Have Safe Search When Engaging in Nontraditional Genetic Recombination. In G. Rawlins, editor, *Foundations of Genetic Algorithms*, pages 265–283. Morgan Kaufmann, 1991.

[3] David Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley, Reading, MA, 1989.

[4] Keith Mathias and L. Darrell Whitley. Genetic Operators, the Fitness Landscape and the Traveling Salesman Problem. In R. Manner and B. Manderick, editors, *Parallel Problem Solving from Nature, 2*, pages 219–228. Elsevier Science Publishers, 1992.

[5] H. Muhlenbeinn. Evolution in Time and Space: The Parallel Genetic Algorithm. In G. Rawlins, editor, *Foundations of Genetic Algorithms*, pages 316–337. Morgan Kaufmann, 1991.

[6] J. Scales, M. Smith, and T. Fischer. Global Optimization Methods of Highly Nonlinear Inverse Problems. In *Proceedings of the International Conference on Numerical Aspects of Wave Properties and Phenomenon*, 1990. France.

[7] C. Stork and T. Kusuma. Hybrid Genetic Autostatics: New Approach for Large-Amplitude Statics with Noisy Data. In *Proceedings of SEG* 62^{nd} *Annual International Meeting*, pages 1127–1131. Society of Exploration Geophysicists, 1992.

[8] L. Darrell Whitley. The GENITOR Algorithm and Selective Pressure: Why Rank Based Allocation of Reproductive Trials is Best. In J. D. Schaffer, editor, *Proceedings of the Third International Conference on Genetic Algorithms*, pages 116–121. Morgan Kauffman, 1989.

[9] L. Darrell Whitley. A Genetic Algorithm Tutorial. Technical Report Nb. CS-93-103, Colorado State University, 1993.

[10] L. Darrell Whitley, Keith Mathias, and Patrick Fitzhorn. Delta Coding: An Iterative Strategy for Genetic Algorithms. In L. Booker and R. Belew, editors, *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 77–84. Morgan Kauffman, 1991.

[11] W. Wilson and K. Vasudevan. Application of the Genetic Algorithm to Residual Statics Estimation. *Geophysical Res. Let.*, 18(12):2181–2184, 1991.