UCRL-JC-119732
PREPRINT

Poor Man's Parallelism in Environmental Management

Virginia M. Johnson
Leah L. Rogers

This was prepared for submittal to the
Second Congress on Computing in Civil Engineering
Atlanta, Georgia
June 5-7, 1995

February 1995

Lawrence Livermore National Laboratory

Poor Man's Parallelism in Environmental Management

Virginia M. Johnson and Leah L. Rogers
Environmental Restoration and Earth Sciences Divisions
L-206, PO Box 808
Lawrence Livermore National Laboratory
Livermore, CA 94551

Introduction

Poor man's parallelism is a term to describe the harnessing of commonly available computational approaches containing a high degree of implicit or explicit parallelism with distributed computer resources to produce a large gain in processing time. The distinguishing features of poor man's techniques are their accessibility and relatively low cost. In some circumstances, the clever exploitation of existing hardware and software may achieve as much improvement in the timely completion of tasks as do high-end, state-of-the-art parallel technologies. The ANN-GA approach to the optimization of environmental remediation strategies is an example of poor man's parallelism: it integrates two well-known computational technologies, artificial neural networks (ANNs) and the genetic algorithm (GA), with a simple scheme for exploiting a network of Unix workstations to solve a nonlinear combinatorial optimization problem. Although this work has been motivated by the need to tame a computational tiger rather than to experiment with different flavors of parallelism, the approach has reached a level of maturity where it is instructive to examine how parallelism is embodied in its various components. It also stands as a demonstration of how even resource-lean organizations can take advantage of parallelism to solve problems.

The Optimization Problem

Figure 1 illustrates the kind of design problem for which the ANN-GA approach was developed. It shows a contour map of groundwater contaminated by volatile organic compounds at an industrial site, overlaid by a pattern of extraction and injection wells intended to clean up the contamination by means of the pump-and-treat strategy. The goal of optimization for this type of design problem is often to find one or more combinations of extraction and injection well locations that will capture or clean up the contamination at minimum cost or time. Although the number of well combinations is potentially infinite, it has been customary in groundwater optimization work to prespecify a grid of potentially good locations and then formulate the search to locate the most cost-effective subsets of those locations which meet remediation goals.

Early optimization work at an illustrative Superfund site used 20 preselected extraction locations with fixed pumping rates and searched for the subset producing the smallest volume of treated water (a convenient surrogate for cost) which contained the contamination over a 40-year planning period (Rogers and Dowla, 1994). Later work focused on 28 fixed-rate extraction and injection locations in a multiple-objective search which balanced cost-efficiency with mass-extraction performance, while meeting a containment constraint over a 50-year planning period (Rogers, Dowla, and Johnson, 1995). Current versions of the problem involve 225 potential extraction and 43 potential injection locations with variable pumping-rates and additional constraints regarding the dewatering of the aquifer.

Regardless of the problem formulation or the type of search technique employed, the fitness or cost function associated with a particular well pattern is largely supplied by a contaminant transport model which assesses the impact of the well pattern on the distribution of the contamination over some period of time. Even in 2-D, numerical models of this sort can take hours to evaluate a single pattern on a conventional Unix workstation. As the resolution, dimensionality, and complexity of the models increase, the time required for this evaluation can extend to days. Since even the

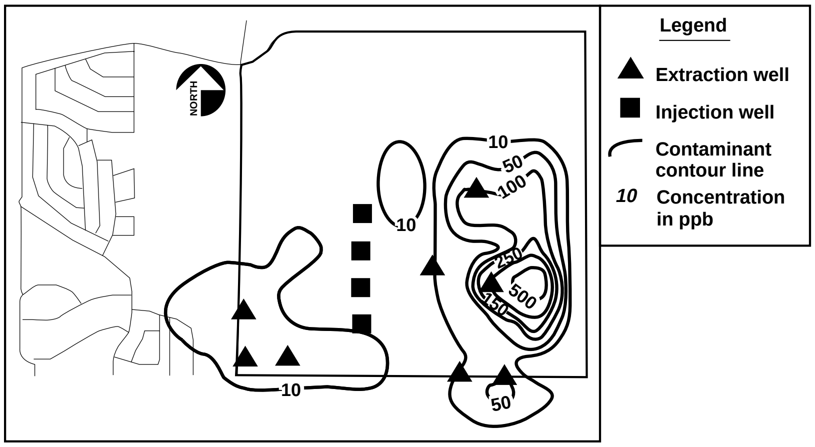

Figure 1. Example of a well pattern intended to remediate groundwater contamination at an industrial site.

most directed search techniques must evaluate hundreds of patterns, the modeling step becomes the biggest single impediment to the optimization of field-scale problems.

One set of solutions to the modeling bottleneck involves reducing the execution time required by the model through parallel algorithms and computer architectures (e.g. Dougherty, 1991). This represents a "rich man's" approach because of the costs normally associated with gaining access to computer resources of this kind. The ANN-GA approach, in contrast, addresses the problem by training neural networks to predict selected model results. The trained networks, rather than the original model, are then used by a simple genetic algorithm (Goldberg, 1989) to obtain fitness information in fractions of a second.

The network architecture used for this prediction task is a multilayer perceptron with the conjugate gradient variation (Johansson, Dowla, and Goodman, 1992) on the standard backpropagation learning algorithm. Networks are trained from examples associating well pattern variations to selected outcomes such as the amount of contamination that has been removed, the highest remaining concentrations after treatment is complete, and whether or not contamination has spread beyond certain boundaries. The examples are drawn from a knowledge base initially created by running the contaminant transport model on a representative sample of well patterns. Since there are no dependencies among the model runs, they can be distributed over a network of processors using only the basic remote file system and execution facilities that are now a standard part of most network environments. Components of the approach are shown in Figure 2.

Parallelism in the ANN-GA Approach

Apart from theoretical treatments of the intrinsic parallelism of schema processing, discussions of parallelism in the genetic algorithm have focused mainly on parallelization of population processing (Gordon, Whitley, and Bohm, 1992). At the global end, a single population is maintained in shared memory and accessed by different processors assigned to different reproductive subtasks. In the island approach, the population is divided into subpopulations, each handled by a different processor. Periodic migration of individuals among subpopulations aids the location of global optima. When a large number of processors is available, as in the case of massively parallel architectures, the number of individuals per subpopulation can go as low as one.

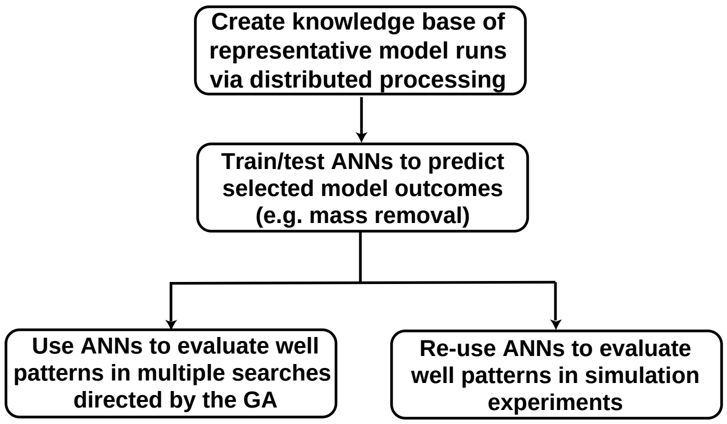
Create knowledge base of representative model runs via distributed processing
Train/test ANNs to predict selected model outcomes (e.g. mass removal)
Use ANNs to evaluate well patterns in multiple searches directed by the GA
Re-use ANNs to evaluate well patterns in simulation experiments

Figure 2. Basic steps in the ANN-GA approach to optimization.

Possibly because it is so problem dependent, parallelization of the evaluation of fitness has received less attention in the literature (Reeves, 1993). However, this is, in essence, how the ANN-GA approach parallelizes the optimization task. On the surface, the simple GA that is being used appears to process the fitness evaluations in the customary serial manner. However, it is doing so by calling upon networks to supply the fitness information. And those networks were trained from data obtained by modeling runs accomplished by distributed processing over a network of workstations. So far as the computational time required for modeling is concerned, the simple GA's directed-random, serial search has been converted to a random, parallel search.

The key to parallelism in the ANN-GA approach lies in how the initial knowledge base of model runs is created. Example well patterns are generated by random sampling over the space of well patterns, supplemented by expert judgment and analyses highlighting particularly promising combinations. The required number of examples is determined by the networks' predictive accuracy on test examples. This, in turn, is roughly related to the number of variables (number and type of locations, pump rates, etc.) in the problem formulation and the complexity of the underlying relationships. An initial set of patterns is run through the model, results are analyzed, and a second refined set is generated and run. Within a set, the examples are independent of each other; so they can be farmed out to different machines either manually or by an automated scheduler. Once trained, the networks interpolate over the space of well combinations to generate the fitness information required by the GA. So, an initial investment in model runs on the order of hundreds can serve to evaluate thousands or even millions of well patterns. The cost-efficiency of the approach depends on the degree to which the networks are used and re-used in various ways.

Some Illustrative Numbers

Performance comparisons conducted so far have focused on the total number of calls to the model. An early study (Rogers and Dowla, 1994) of the 20-well problem compared the ANN-GA approach with NPSOL, a nonlinear optimization procedure well-known in groundwater management. Both procedures used the 2-D contaminant transport model SUTRA (Voss, 1984) to evaluate well patterns. 245 runs, each requiring an average of 2.5 hours to complete, were needed to construct the ANN-GA knowledge base. The NPSOL procedure converged (with results similar to those obtained by the ANN-GA approach) after 206 calls to the model, which would appear to make it the winner. However, NPSOL requires that the calls be made serially, as it searches through the space of well patterns. Runs for the knowledge base, in contrast, were obtained in parallel from a mix of machines. Furthermore, the trained networks are re-usable in subsequent searches and for different analytic purposes.

Re-useability was demonstrated on a later 28-well multiple-objective problem (Rogers, Dowla, and Johnson, 1995). 325 SUTRA runs were conducted to create the initial knowledge base. Initially, the fitness function for the GA was the simple sum of three components: 1) whether or not a containment constraint was met, 2) the amount of contaminant mass removed, and 3) the cost of installing and operating the well pattern. To test the impact of each component on the search results, the search was repeated six times, systematically zeroing out or doubling the weight of each component. Each search was allowed to run for 12,000 generations of the GA and required 22 hours on a Sun Sparcstation II. In that space of time, only 8.8 calls to the model could have been made, which is not nearly sufficient for convergence by any nonlinear search technique. In addition, the same networks were re-used in Monte Carlo simulation experiments testing the contributions of particular well locations alone or in combination (Johnson and Rogers, 1995). Four experiments evaluated 112,834 well patterns, a task that would have required 225,668 hours if SUTRA had been used but which was completed by the networks in under an hour.

The current version of the problem uses 225 extraction and 43 injection locations, is based on a higher-resolution numerical grid, makes predictions over a 100-year planning period, and contains several additional constraints. As a result, execution time per SUTRA run now averages 8 hours and the minimum size of the knowledge base is in the thousands rather than hundreds. A simple automated scheduler was written to generate a well pattern, find a free machine from a list of 75 networked Sun workstations (ranging from Sparcstation 10s to IPCs), remotely initiate a model run for that pattern on that machine, and continue with the next pattern. All I/O is directed from/to a central exported filesystem. Because the models execute at lowest-possible priority, they exploit the resources of these largely idle computers without significantly interfering with their owners' tasks. Because no synchronization among runs is required, no additional message-passing software need be installed. Despite machine and network down-time and various public relations problems, the scheduler completed over 10,000 model runs in two months of operation. This may sound like a very long period of time. However, the engineering design time-frame for this type of optimization problem is normally measured in years.

Conclusions

The ANN-GA approach does much more than simply parallelize the evaluation of fitness in the genetic algorithm. Once embodied in the trained neural networks, the parallelization of the space of well combinations can be put to use, in a modular fashion, by other search techniques such as simulated annealing and other applications such as sensitivity analysis and inverse modeling. Although the ease with which the knowledge base is built is dependent on access to networks of computers, this represents a reasonable alternative for many organizations seeking to exploit resources they already have before they invest in new hardware and software.

References

Dougherty, D. E. (1991) "Hydrologic applications of the connection machine CM-2." *Water Resources Research*, 27, 3137-3147.

Goldberg, D. E. (1989) *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley, Reading, MS.

Gordon, V.S., Whitley, D., and Bohm, A. P. W. (1992) "Dataflow parallelism in genetic algorithms." *Parallel problem solving from nature*. R. Manner and B. Manderick, eds., North-Holland, New York, N. Y., 2, 553-542.

Johansson, E. M., Dowla, F. U., and Goodman, D. M. (1992) "Backpropagation learning for multi-layer feed-forward neural networks using the conjugate gradient method". *Int. J. of Neural Systems*, 2(4), 291-301.

Johnson, V. M. and Rogers, L. L. (1995) "Location analysis in ground water remediation using artificial neural networks." UCRL-JC-117289, Lawrence Livermore National Laboratory, Livermore, CA. See also *Ground Water* (in press).

Reeves, C. R. (1993) "Genetic algorithms." *Modern heuristic techniques for combinatorial problems.* C. R. Reeves, ed., Halstead Press, New York, N. Y., 151-196.

Rogers, L. L. and Dowla, F. U. (1994) "Optimization of groundwater remediation using artificial neural networks and parallel solute transport modeling." *Water Resources Research*, 30(2), 457-481.

Rogers, L. L., Dowla, F. U., and Johnson, V. M. (1995) "Optimal field-scale groundwater remediation using neural networks and the genetic algorithm." UCRL-JC-113773, Lawrence Livermore National Laboratory, Livermore, CA. See also *Environmental Science & Technology* (in press).

Voss, C. I. (1984) *A finite-element simulation model for saturated-unsaturated, fluid-density-dependent groundwater flow with energy transport or chemically-reactive single-species solute transport*. U. S. Geological Survey, Water Resources Investigations Report #84-4369.

Acknowledgements

This work was made possible by support from Lawrence Livermore National Laboratory's Environmental Restoration Division, under the auspices of the U. S. Department of Energy, contract W-7405-ENG-48. It has benefited from critical review by Kenneth Jackson.