MECHANISTIC-BASED GENETIC ALGORITHM SEARCH ON A BEOWULF CLUSTER OF LINUX PCS

Jin-Ping Gwo[+], Forrest M. Hoffman[++], and William W. Hargrove[+++]
[+]Computer Science and Mathematics Division and
[++]Environmental Sciences Division
[+++]Computational Physics and Engineering Division
Oak Ridge National Laboratory[1]
P.O. Box 2008, MS6203
Oak Ridge, TN 37831-6203
email: gwojp@ornl.gov

KEY WORDS

Genetic algorithm, Inverse problem, Fracture network, Solute transport, Parallel virtual machine (PVM), Beowulf Linux cluster.

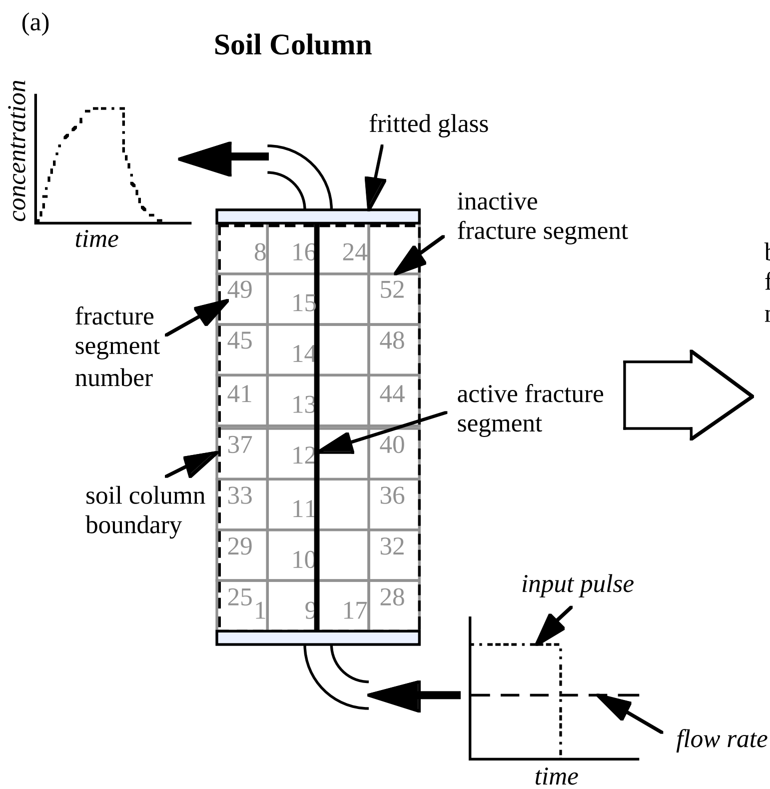
ABSTRACT

A simple genetic algorithm (SGA) was implemented on a cluster of Linux PCs to search for the most likely fracture networks in a soil column. The objective is to evaluate the performance of SGAs in a distributed computing environment that is widely and inexpensively available to environmental researchers and engineers. The Beowulf computer was built out of surplus personal computers at Oak Ridge National Laboratory by scientists in the Environmental Sciences Division (http://www.esd.ornl.gov). The communication on the Beowulf is via ordinary Ethernet connection private among the processors, with a peak bandwidth of 10 Mbit/s. The CPUs are mostly Intel 486DX-2/66 and Pentiums, with 16 - 32 MB of memory. Most of the software on the Beowulf is from the public domain. Using the PVM message passing library and a manager-worker paradigm, we seek to maximize the loads on CPUs of dissimilar speed and memory size. SGA is an inductive search algorithm that bases upon a few simple operators such as reproduction, crossover, and mutation. The underlying mechanisms of flow and transport phenomena in structured soils with discrete fractures are simulated by the computer code FRACTRAN. In a generation of SGA, hundreds of FRACTRAN simulations are required, which consume the majority of the CPU time needed by the SGA search process. For an entire SGA search, tens of millions of such simulations, often referred to as function evaluation in genetic algorithms literature, are performed. The minimal communication between the manager and workers, passing fracture networks represented in bit strings to the workers and bit string fitness back to the manager, suggests that small communication bandwidth is adequate to achieve high performance. The manager-worker paradigm is also highly effective in achieving load balance on heterogeneous, networked computers such as the Beowulf. In addition to reporting the performance of the implementation, we also explore the aspect of SGA related to information constraints. SGA may be trapped in local optima and genetic drifting may ensue. With additional information the SGA may be steered away from local optima and the uncertainty of the identified fracture networks may be reduced. Because multiple runs of the SGA search algorithm are necessary to determine the least uncertain fracture networks, a distributed computing environment proves to be highly effective.

INTRODUCTION

The advent of Beowulf-style computers has brought cluster computing within the reach of many environmental scientists [e.g., Hargrove and Hoffman, 1999]. Beowulf personal computer (PC) clusters were first devised by scientists at NASA Goddard Space Flight Center (see http://www.beowulf.org) to enable earth and space science applications on low cost, off-the-shelf computer components. The NASA Beowulf project was also facilitated by the freely available Linux operating system. The open source nature of Linux allows programmers to enhance the operating system to meet the requirements of cluster computing (see http://www.opensource.org). As a result, a collection of tools are freely available on the Internet to assist in the building of Beowulf-style PC clusters (http://www.beowulf.gov). The Beowulf cluster in the Environmental Sciences Division at Oak Ridge National Laboratory (ORNL) was built out of surplus personal computers. The PCs are mostly Intel 486DX-2/66 and Pentiums, with 16 - 32 MB of memory, connected by

[1] managed by Lockheed Martin Energy Research, Corp. for the U.S. Department of Energy under contract DE-AC05-96OR22464



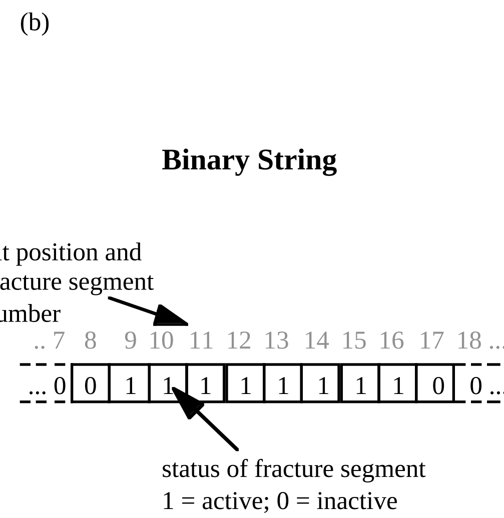


Fig. 1. (a) Schematic of laboratory soil columns, with inflow at the bottom and outflow at the top; (b) GA binary string illustrating mapping from fracture segments in (a) to positions in the string in which the status of the fracture segments are coded into the binary 1 (active) and 0 (inactive).

ordinary Ethernet with a peak bandwidth of 10 Mbit/s (Hoffman and Hargrove, 1999).

Various science applications have been successfully demonstrated on the ORNL Beowulf cluster [e.g., Hargrove and Hoffman, 1999; http://stonesoup.esd.ornl.gov]. This paper describes a mechanistic-based genetic algorithm that is used to search for near-optimal fracture networks in laboratory soil columns. Genetic algorithms (GAs) are inductive search algorithms that explore and exploit the similarity among individuals within a population [Goldberg, 1989]. Often times the most expensive kernel of a GA application is the evaluation of the fitness functions. In our case, the calculation of fitness involves solving a system of linearized equations derived from the conservation laws of mass and momentum. Fortunately the evaluations are independent of each other and can be carried out in parallel. This class of GA search problems is therefore highly amenable to the low bandwidth, high processor power nature of Beowulf clusters.

The objective of this research is to evaluate the performance of SGA in the distributed computing environment of Beowulf-style PC clusters. Using a simple genetic algorithm (SGA), a fracture flow and transport code for structured soils, and the PVM message passing library and a manager-worker paradigm, we seek to maximize the loads on CPUs of dissimilar speed and memory size. The timely evaluation of the fracture networks within SGA populations enables us to identify the near-optimal fracture networks and the appropriate constraints for the SGA search algorithm. Performance of the SGA search algorithm on the Beowulf cluster is reported. Implication of the SGA search constraints on the characterization of the uncertainties associated with the near-optimal fracture networks is discussed.

FLOW AND TRANSPORT IN FRACTURED POROUS MEDIA

The governing equations for the movement of fluids and solutes in fractured porous media are:

$$\frac{\partial}{\partial x_i} K_{ij} \frac{\partial h}{\partial x_j} = 0, \qquad i, j = 1, 2 \tag{1}$$

$$\theta \frac{\partial c}{\partial t} + q_i \frac{\partial c}{\partial x_i} - \frac{\partial}{\partial x_i} \theta D_{ij} \frac{\partial c}{\partial x_j} = 0, \qquad i, j = 1, 2 \tag{2}$$

$$(2b)K_f \frac{d^2 h_f}{dl^2} - q_{n^-} + q_{n^+} = 0 \tag{3}$$

$$2b\left[\frac{\partial c_f}{\partial t} + q_f \frac{\partial c_f}{\partial l} - \frac{\partial}{\partial l} D_f \frac{\partial c_f}{\partial l}\right] - \Gamma_{n^-} + \Gamma_{n^+} = 0 \tag{4}$$

where, for the matrix domain, K_{ij} is the hydraulic conductivity, x_1 and x_2 are the spatial dimensions, h is the hydraulic head, c is the solute concentration, t is time, θ is water content, q is specific discharge, D_{ij} is the hydrodynamic dispersion coefficient as given in Bear [1972]; for the fracture domain, $2b$ is the fracture aperture, K_f is the hydraulic conductivity as defined in Sudicky and McLaren [1992], h_f is the hydraulic head, l is the spatial dimension along the fracture, c_f is the solute concentration in the fracture, and D_f is the hydrodynamic dispersion coefficient as given in Tang et al. [1981]. The last two terms in eq. (3) account for the fluid mass loss and gain, respectively, due to mass transfer with the matrix domain. Similarly, the last two terms in eq. (4) are the solute loss and gain, respectively. Equations (1) - (4) with their boundary conditions are implemented in the computer code FRACTRAN [Sudicky and McLaren, 1992] which is our computational kernel in the SGA search algorithm.

To enable the simulation of flow and transport in a fractured porous medium, one needs to specify the parameters identified above. For this application, the hydraulic conductivity, dispersivity, and water content of the matrix and fracture domains were measured directly from laboratory experiments or calculated theoretically using fracture aperture sizes [Gwo et al., 1998]. Additionally, one would need to implement a finite-difference grid to identify the individual matrix blocks and fracture segments, the initial conditions of solute concentrations in these structures, and the boundary conditions that are the driving force for the movement of the fluids and solutes. A two-dimensional soil column with a 4 (horizontal) by 8 (vertical) grid is used for our simulations (Fig. 1a). We therefore have 32 matrix blocks and 52 candidate fracture segments. These segments are subject to the manipulation by the SGA to select the near-optimal fracture networks. The hydraulic condition of the soil column is assumed to have reached steady state, and the matrix and fracture domains are in hydraulic equilibrium. The soil column is assumed initially depleted of the solute and a pulse of the solute is injected at the bottom of the soil column, followed by another pulse of the carrying fluid without the solute (Fig. 1a).

A PARALLEL SGA SEARCH ALGORITHM

The finite difference grid depicted in Fig. 1a is encoded into a bit string of length 52. The numbers of the fracture segments correspond to the positions of the binary bits (Fig. 1b). The binary bits encode the status of a fracture segment, either active (on as 1's) or inactive (off as 0's). The binary string in Fig. 1b therefore represents a vertical fracture centered in the soil column from bottom to top (Fig. 1). Three GA operators, reproduction, crossover, and mutation [e.g., Goldberg, 1989], are used to manipulate a population of 128 individuals. These individuals are generated randomly for the first generation and subsequently selected for the GA operators according to the following fitness function:

$$F = \frac{1}{N}\sum_{j=1}^{N}\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(c_i - c_i^*\right)^2} - \frac{\alpha}{A}\sigma_{RMSE} \qquad (5)$$

where N is the number of solute breakthrough curves (BTCs) presented to the SGA, n is the number of measurements on a BTC, c_i and c_i* are the calculated and measured solute concentration, respectively, α is the number of matching surface features, e.g., exposed flowing fracture segments, among a total of A, and σ_{RMSE} is the standard deviation of the root mean squared error (RMSE). The SGA terminates its search when a stopping criterion is met or the number of allowable generations, 128 here, is exhausted. Various methods of selecting winning individuals for reproduction are available for SGA search [e.g., Goldberg, 1989]. Among the methods tested, i.e., roulette wheel selection and tournament selection with or without replacement, tournament selection without replacement produces the best outcome. We also tested single and multiple point mutation and uniform crossover [Goldberg, 1989] and it was found that uniform crossover performs better for the encoding scheme described above. We hereby restrict our discussion in this paper to tournament selection with uniform crossover.

Parallel implementation of the SGA search algorithm utilizes the parallel virtual machine (PVM) library [Geist et al., 1994] and a manager-worker paradigm [Mahinthakumar and Gwo, 1999]. The manager consists of an SGA Fortran code (written by Ulrich Hermes of the University of Dortmund, Germany) within a driver routine that doles out FRACTRAN simulations to the workers that report finishing a previously assigned job. The workers are individual FRACTRAN processes on the Beowulf compute nodes that receive the encoded binary string and carry out the flow and transport simulations.

PARALLEL PERFORMANCE OF PVM VIRTUAL MACHINES

To test the performance of the parallel SGA search algorithm on the ORNL Beowulf cluster, we ran a series of five-SGA-generation simulations, using combinations of Pentiums and 486DX-2/66's. The communication and child process spawning times account for a small fraction of the total execution time (Fig. 2, all CPUs are Pentium 80-200 MHz). Nonlinear scaling behavior is expected for processors of dissimilar CPU speeds. However the performance of the virtual machine appears to encounter a threshold with more than 10 Pentiums. The performance threshold is caused by the faster CPUs waiting on the slower CPUs during the last few FRACTRAN simulations of each SGA generation. The manager routine does not discern a fast CPU from a slow one and thereby try to optimize the virtual machine. Because the SGA population is of fixed size, the optimization should be straightforward. Alternatively, one may use a subset of the 22 Pentiums, e.g., 10, and the speed up is close to 10 fold. For this particular application, this option is equally attractive.

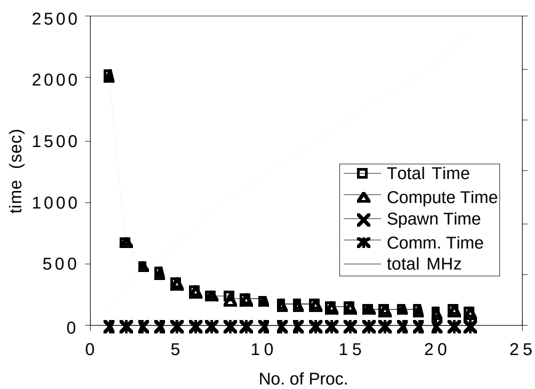


Fig.2 Parallel performance of all Pentium PVM virtual machines up to 22 CPUs.

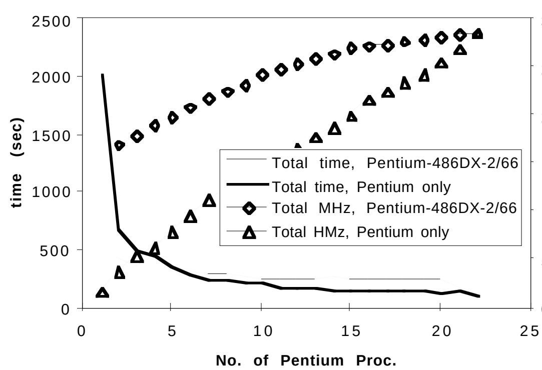


Fig. 3. Parallel performance of all Pentium and Pentium-486DX-2/66 PVM machines.

The performance threshold of the all Pentium virtual machines is also complicated by the fact that the CPU speeds are in a relatively narrow range of 80 MHz to 200 MHz. Putting together a virtual machine of contrasting CPU speeds, however, defeats the purpose of Beowulf clusters, unless the particular application warrants such combination. Nevertheless, we are interested in the relative performance between an all Pentium virtual machine and a Pentium-486DX-2/66 mixed virtual machine. Shown in Fig. 3 is a comparison between Pentium-486DX-2/66 virtual machines of fixed size (22 CPUs) and all Pentium virtual machines of variable size (1 - 22 CPUs). The performance of a 22 CPU Pentium-486DX-2/66 virtual machine is similar to that of a Pentium virtual machine at approximately 4 to 6 CPUs. Replacing more 486DX-2/66's with Pentiums in the 22 CPU Pentium-486DX-2/66 virtual machines does not improve the virtual machine's performance, because of the synchronization required at the end of each SGA generation.

SGA SEARCH CONSTRAINTS AND FRACTURE NETWORK UNCERTAINTY

The SGA search algorithm, not unlike other search methods, may be trapped in local optima. This is likely to occur particularly when the search space is not appropriately constrained. For example, in typical laboratory tracer injection experiments, the soil column is assumed a one-dimensional flow and transport domain, and only one tracer breakthrough curve (BTC) is collected [Jardine et al., 1993]. Mixing or averaging is likely to smooth the signals of individual tracer parcels exiting at various locations along the exit cross section. Presenting one single BTC to the SGA may not be enough to appropriately constrain the search space for a structured soil in which the structure may very likely be multidimensional.

The one-dimension assumption, implying symmetry perpendicular to the bulk flow direction, is also inherent to a simple vertical fracture running centrally from bottom to top of the soil column (Fig. 1a and Fig. 4a). The SGA found the global optimum at generation 45, with the aide of one single BTC (data not shown). This symmetry assumption, however, results in the SGA being trapped in local optima for a nonsymmetric configuration (Fig. 4b). The global optimum emerges at generation 77 after two exposed flowing fracture segments (positions 1 and 8 in Fig. 1a) are also presented to the SGA and the individuals with the two segments in their binary strings are rewarded with the last term in eq. (5). This same combination of information, one single BTC and two exposed fracture segments, however, is not able to guide the SGA away from local optima for the tortuous fracture network in Fig. 4c. The situation is not remedied until three breakthrough curves along the exit cross section, in addition to the two exposed fracture segments, are presented to the SGA. The global optimum is identified at generation 48 (Fig. 4d).

Projecting the findings above to laboratory and field applications, one needs to characterize the uncertainty of the SGA-found fracture networks, given that true global optima are unlikely to be available. For laboratory soil column experiments, RMSE is frequently used to examine how a curve-fitting or parameterization procedure performs. Often the underlying mechanisms of flow and transport are soundly based upon first principles, but the RMSE is used

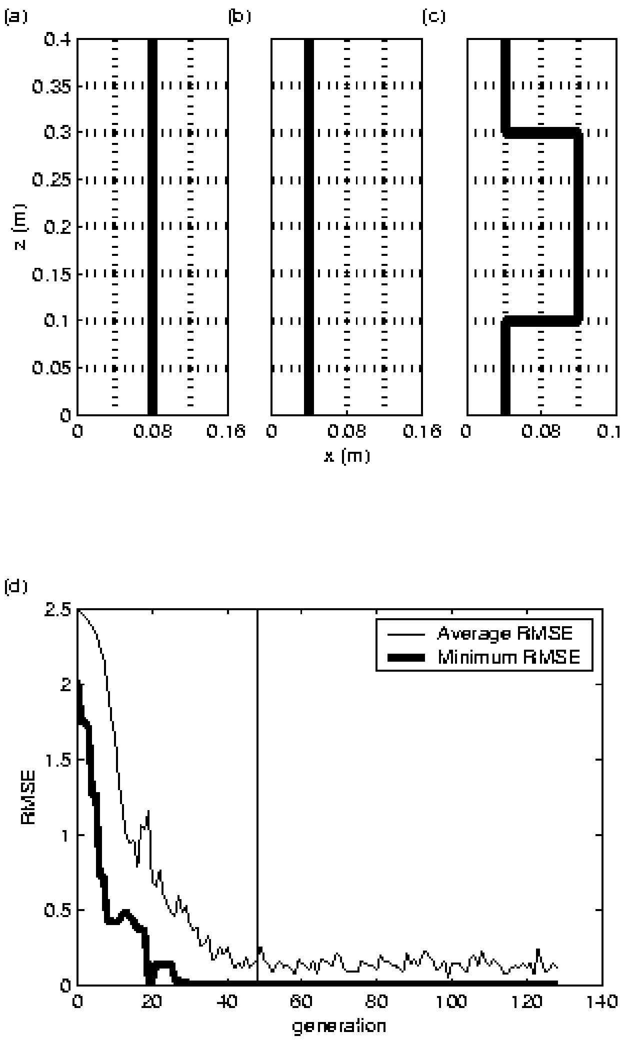


Fig. 4. Three fracture networks of increasing complexity (a) - (c) and (d) the SGA convergence history of (c).

without further examination of the model assumptions [e.g., Jardine et al., 1993]. Presented in Fig. 5a is the *true* BTC of the tortuous fracture network (Fig. 5d) and those of the other two near-optima found by the SGA (Figs. 5b and 5c). Visually the BTCs agree with the *true* BTC very well and the RMSEs are very small. However, the fracture networks are, in fact, local optima. They bear little resemblance with the global optimum. Their fracture-matrix contact areas are larger than that of the global optimum and may result in a much larger mass transfer rate estimation after being upscaled to field soils and geological formations. This result suggests that more rigorous SGA search end point measures must be devised and the uncertainty regarding SGA near-optima must be carefully examined.

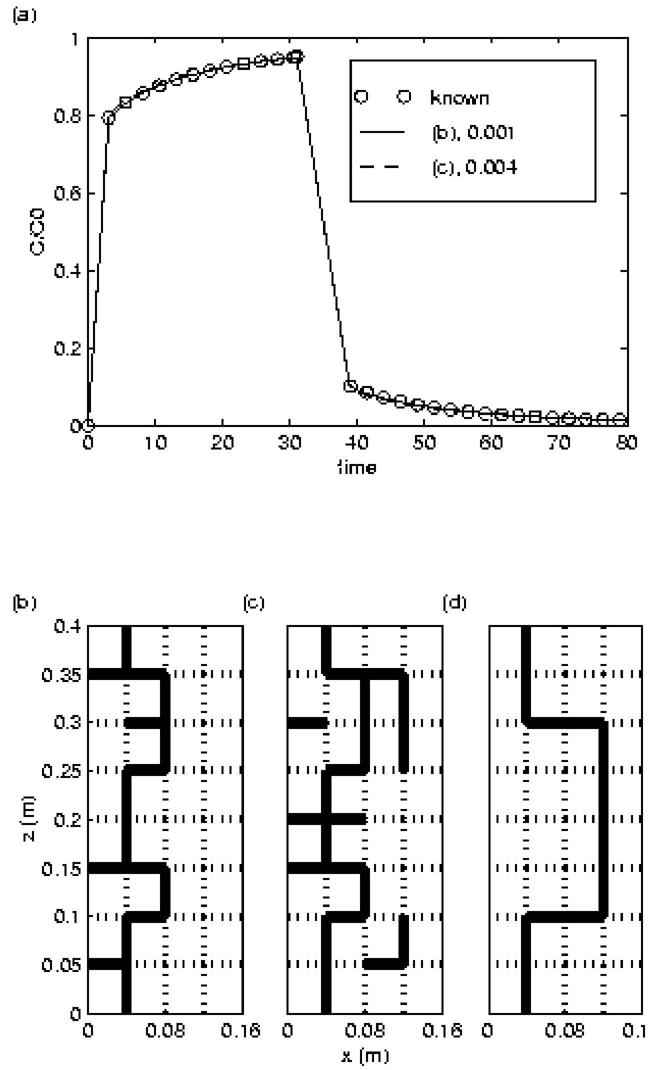


Fig. 5. Two SGA near-optima (b) and (c), the calculated BTCs (a), and the known solution (d).

SUMMARY AND CONCLUSION

We presented a mechanistic-based, parallel SGA search algorithm to identify near-optimal fracture networks in structured porous media. Performance data of the search algorithm was collected on the ORNL Beowulf-style Linux cluster. The PVM virtual machine using up to 22 Pentiums (80 to 200 MHz) has an optimal performance near 10 CPUs. Above 10 Pentiums, the performance encountered a threshold which cannot be overcome without further optimization of the manager routines. The cause of the degradation is the time used to wait on the slower processors to finish the last few FRACTRAN simulations of an SGA generation. Simulations with larger SGA population size may be less vulnerable to this performance degradation. This problem associated with the manager-worker paradigm on heterogeneous clusters was further manifested by the 22 CPU Pentium-486DX-2/66 virtual machines. Performance of the SGA search algorithm on this latter class of virtual machines suggests that, against our intuition, replacing the slower 486DX-2/66's with Pentiums may not improve the performance of the virtual machine.

We also investigated the effect of information constraints on the SGA search algorithm. Without appropriate information to constrain the SGA search space, it is likely that the SGA may be trapped in local optima. This finding suggests that the near-optimal fracture networks identified by the SGA, especially those within laboratory and field soils and geological formations in which the *true* fracture networks are rarely known, may need to be examined carefully to determine their associated uncertainties. Without reducing these uncertainties, field mass transfer rates estimated using the upscaled, SGA-found fracture networks may be over-estimated.

ACKNOWLEDGMENT

This research is supported by the Natural and Accelerated Bioremediation (NABIR) Program of the Office of Biological and Environmental Research, U. S. Department of Energy. The authors would like to thank Dr. Frank Wobber, who is a contract officer for the DOE's NABIR program, for financially supporting this research. This work is also partially supported by the Mathematical, Information, and Computational Sciences Division of the U.S. Department of Energy.

REFERENCES

Bear, J., Dynamics of Fluids in Porous Media, Elsevier Science, New York, 1972.

Geist, A., A. Beguelin, J. Dongarra, W. Jiang, R. Manchek, and V. Sunderam, PVM: Parallel Virtual Machine, A Users' Guide and Tutorial for Networked Parallel Computing, Cambridge, Mass., MIT Press, 279p, 1994.

Goldberg, D. E., Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley Pub. Co., Reading, Mass., 412p, 1989.

Gwo J. P., R. O'Brien, P. M. Jardine, Mass transfer in structured porous media: embedding mesoscale structure and microscale hydrodynamics in a two-region model, J Hydrol 208(3-4): 204-222, 1998.

Hargrove, W. W., and F. M., Hoffman, Using multivariate clustering to characterize ecoregion borders, Computing in Science and Engineering 1(4): 18-25, 1999.

Hoffman, F. M., and W. W. Hargrove, Cluster computing: Linux taken to the extreme, Linux Mag. 1(1): 56-59, 1999.

Jardine, P. M., G. K. Jacobs, and G. V. Wilson, Unsaturated transport processes in undisturbed heterogeneous porous media: I. Inorganic contaminants, Soil Sci. Soc. Am. J., 57: 945-953, 1993.

Mahinthakumar, G. and J. P. Gwo, Task parallel and data parallel computing for subsurface inverse characterization problems. In A. Tentner (ed.), High Performance Computing 1999, Grand Challenges & Computer Simulation, p.217-223, 1999.

Sudicky, E. A., and R. G. McLaren, The Laplace transform Galerkin technique for large-scale simulation of mass transport in discretely fractured porous formations, Water Resour. Res. 28: 499-512, 1992.

Tang, D. H., E. O. Frind, and E. A. Sudicky, Contaminant transport in fractured porous media: Analytical solution for a single fracture, Water Resour. Res. 17(3):555-564,1981.