GRID-ENABLED SIMULATION-OPTIMIZATION APPROACH FOR SOLVING GROUNDWATER CHARACTRIAZATION PROBLEMS

Baha Mirghani[1], Michael Tryby[1], Ranji Ranjithan[2], and Kumar Mahinthakumar[3]

ABSTRACT

Groundwater environmental characterization involves the resolution of unknown system characteristics from observation data, and is classified as an inverse problem. Inverse problems are relatively challenging to solve due to natural ill-posedness and computational intractability. Here we adopt the use of a simulation-optimization approach that couples a numerical pollutant-transport simulation model with evolution strategies for solution of the inverse problem. In this approach, the numerical transport model is solved iteratively during the evolutionary search, which in general can be computationally intensive since several hundreds to thousands of forward model evaluations are typically required for solution. Given the potential computational intractability of such a simulation-optimization approach, grid computing is explored as a possibility to ease and enable the solution of such problems. In this paper, the solution of two groundwater inverse problems will be explored. The computational experiments were performed on the National Scientific Foundation TeraGrid. The results demonstrate the performance of the grid-enabled simulation-optimization approach in terms of solution quality and computational performance.

KEY WORDS

Simulation-Optimization, Groundwater characterization, Inverse problem, Grid computing.

1. INTRODUCTION

Source identification and release history reconstruction problems are important in environmental forensics and characterization of contamination for the purposes of regulatory enforcement and assessing liability. In this problem context, source locations and/or historical contaminant release schedules are unknown model inputs, which are resolved from the spatially and temporally distributed observational data collected at monitoring wells. Problems such as these, where system characteristics are resolved from sparse observational data are classified as inverse problems. A forward model, usually a system of partial

[1] Graduate Research Assistant, Department of Civil Engineering, North Carolina State University, Raleigh, NC 27695, Phone: +19195154342, Fax: +19195157908, Email: {bmirgha, metryby}@ncsu.edu

[2] Associate Professor, Department of Civil, Construction & Environmental Engineering, North Carolina State University, Raleigh, NC 27695, Phone: +19195156979, Fax: +19195157908, Email: ranji@ncsu.edu

[3] Assistant Professor, Department of Civil, Construction & Environmental Engineering, North Carolina State University, Raleigh, NC 27695, Phone: +19195157696, Fax: +19195157908, Email: gmkumar@ncsu.edu

differential equations (PDEs), describes the transport processes of the groundwater system and defines the relationship between system inputs and outputs.

The solution complexity of such problems is proportional to the number of system inputs that must be determined. Inverse problems are inherently difficult to solve due to ill-posedness, and are several orders of magnitude more computationally challenging than solution of the corresponding forward model since several hundred to thousands of forward model evaluations are typically required for solution. Here a simulation-optimization solution approach is studied where the inverse problem is formulated and solved as an optimization model. The optimization model representation of the inverse problem is then solved by coupling the groundwater forward model with numerical search algorithms and/or evolutionary search heuristics.

Recent investments in national high speed network infrastructure have allowed the aggregation of geographically distributed high-performance computing resources into computational grids. In part, because computational grids promote reliable and economical access to, and sharing of, high-end computing resources, they have emerged as a new paradigm in scientific and engineering computation. Given the computational resource demands of the simulation-optimization solution approach, computational grids have the potential to enable the solution of environmental inverse problems that previously would not have been possible.

This paper discusses the solution to two groundwater inverse problems: (1) the contaminant source identification problem; and (2) the source history reconstruction problem. The computational experiments are performed on the National Scientific Foundation TeraGrid. The results demonstrate the efficacy of the grid-enabled simulation-optimization approach in terms of identifying the unknown system inputs and raw computational performance.

2. PROBLEM DESCRIPTION

In the real world, full spatial distributions of significant groundwater properties such as contaminant concentrations and locations are generally unknown. As alternative sparse measurements of hydraulic heads and concentrations at number of observation wells are available. Finding the source location and the time history of the solute in groundwater can be categorized as a problem of time inversion. This means that we have to solve the governing equations backward in time “inverse modeling”, (Atmadja & Bagtzoglou, 2001). Modeling contaminant transport using reverse time is an ill-posed problem since the dispersive process is irreversible. Thus, the solutions have discontinuous dependence on data and are sensitive to errors in the data. A problem is categorized as an ill-posed problem if: (1) the solution does not exist; (2) the solution is nonunique; and (3) the solution is unstable. The inverse problems occurring in groundwater modeling includes but not limited to contaminant source history reconstruction, source identification, and hydraulic conductivity estimation. Inverse problem are computationally challenging because they are generally ill-posed and they required extensive computational resources. Generally there are many approaches, such as simulation optimization, probabilistic, analytical, and direct approaches, to solve inverse problems (Sun, 1994).

Two groundwater inverse problems were considered in this study. The forward model employed here is a parallel groundwater transport and remediation simulator (PGREM3D) (Mahinthakumar, 1999). The governing equations describing the groundwater transport is fully explained by (Bear, 1972).

2.1 SOURCE HISTORY RECONSTRUCTION PROBLEM (SHR)

One of the inverse problem applications considered in this study is an instance of a groundwater release history reconstruction problem (SHR). In this problem, unknown historical contaminant release schedules at given source locations are resolved from spatially and temporally distributed concentration data observed at a set of monitoring wells. The problem assumes that the contaminant source locations are known, but that the contaminant release histories at those sources are unknown. Concentration observations at the 18 monitoring wells are collected periodically (see the vertical cross-sections of the domain in Figure 1) to generate a concentration time series at each monitoring location. We parameterize a general concentration time series as follows;

$$\mathbf{C} = [c(0), c(\Delta t), c(2\Delta t), ...] \tag{1}$$

where c *(M/L3)* is a concentration and Δt is a periodic time interval. We wish to reconstruct the source release history over a finite time horizon tr extending from the time (ts_0) in the past when monitoring activities started and towards the present; this time span tr is referred to as the release history reconstruction period, with a periodic interval Δtr. Monitoring activities are conducted at n_m monitoring wells. The number of samples n_s taken at a monitoring well over the release history reconstruction period is equal to $(tr/\Delta tm)n_m$, where Δtm is the periodic monitoring interval. In the general case, t_0 and ts_0 do not necessarily correspond; however, for the problem being studied here we assume that sampling activities started at t_0.

We attempt to model the observed concentrations using the PGREM3D simulation, which provides the relationship $\mathbf{C}m = f(\mathbf{C}r)$, where $\mathbf{C}r$ is the source release concentration time series and $\mathbf{C}m$ is the time series of modeled monitoring concentrations. We wish to identify $\mathbf{C}r$ given $\mathbf{C}m$, thus the solution of inverse problem is necessary. The inverse problem is posed as an optimization model where the objective is the minimization of the root square error (RSE) between the observed and calculated concentrations;

$$\min_{\mathbf{C}r} \sqrt{\sum_{i=1}^{n} (\mathbf{C}o_i - \mathbf{C}m_i)^2} \tag{2}$$

where the concentration time series $\mathbf{C}o_i$, $\mathbf{C}m_i$ are observed and modeled concentrations, respectively, and $i = 1 \ldots n_m$ is the index of monitoring well locations. The time series of released concentrations $\mathbf{C}r$ are the set of decision variables over which the problem is solved. The following constraints are added to enforce non-negativity for the decision variables.

subject to:

$$\mathbf{C}_{\max} \geq \mathbf{C}r \geq 0, \tag{3}$$

where $\mathbf{C}_{max}$ is the maximum expected concentration. Depending on the number of contaminant sources, the number of decision variables is equal to the product of the number of contaminant sources times the number of time durations.

2.2 Contaminant Source Identification problem (CSI)

The second inverse problem application considered in this study is the groundwater contaminant source identification problem (CSI). In this problem, an unknown contaminant release at a single source location is resolved from spatially and temporally distributed concentration observations collected at monitoring wells. The problem assumes that the contaminant source location and the contaminant release at the sources are unknown. Similar to the previous application, concentration observations at the 18 monitoring wells are collected to generate a concentration time series at each monitoring location. The problem assumes that the source location and contamination release at the source is unknown. Furthermore, the signature of the source embedded in the monitoring data is a function of the source characteristics. Again, we attempt to model the observed concentration using PGREM3D, which describes the relation $\boldsymbol{Cm} = f(x_j, y_j, z_j, \boldsymbol{Cr})$, where x_j, y_j *and* z_j are the coordinates of the expected source location, $j = 1, 2$ denotes the vertices at opposite corners of the extent of the source, $\boldsymbol{Cr}$ is the source concentration, and $\boldsymbol{Cm}$ is the time series of modeled monitoring concentrations. The inverse problem is posed as an optimization model where the RSE between the observed and calculated concentrations (Equation. 4) is minimized. The following constraints are included to enforce decision variable bounds and feasibility.

subject to:

$$\min_{x_j, y_j, z_j, Cr} \sqrt{\sum_{i=1}^{n} (\mathbf{C}o_i - \mathbf{C}m_i)^2} \tag{4}$$

$$\mathbf{C}_{max} \geq \mathbf{C}_0 \geq 0, \tag{5}$$

$$x_{max} \geq x_j \geq 0; \quad y_{max} \geq y_j \geq 0; \quad z_{max} \geq z_j \geq 0; \quad j=1,2 \tag{6}$$

$$x_2 \geq x_1; \quad y_2 \geq y_1; \quad z_2 \geq z_1, \tag{7}$$

Depending on the number of contaminant sources, the number of decision variables is equal to the product of the number of contaminant sources times the number of unknowns used to describe each source. Thus, for this case we have 7 unknown decisions variables.

Table1: Hypothetical Domain Parameters

Parameter	Values
Problem size	51x31x11 grids
Number of time steps	100
Time step size (dt)	0.15day
Dispersion parameters	$\alpha L = 0.1m$, $\alpha T = 0.1m$, $Dm = 0.001 m^2/d$
Flow filed, velocity	Homogenous, 1 m/day
True source location	$x_1 = 2$, $y_1 = 15$, $z_1 = 6$, $x_2 = 5$, $y_2 = 17$, $z_2 = 8$

2.3 DESCRIPTION OF THE TEST PROBLEM DOMAIN

A hypothetical 3-dimensional field of 51x31x11 grids was employed in this study. Detailed geometrical and hydraulic parameters are shown in Figure 1 and Table 1.

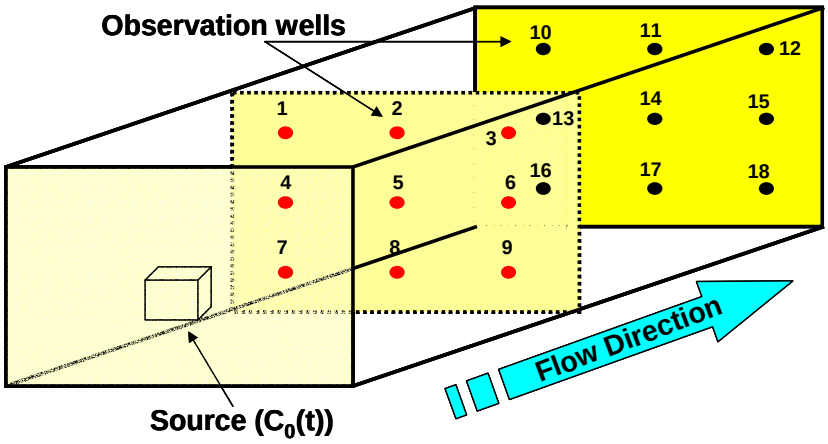

Figure 1: Hypothetical three dimension domain

3. FRAMEWORK ARCHITECTURE

The LAarge Scale Simulation-Optimization framework (LASSO) (illustrated in Figure 2) is used in this study (Tryby et al., 2005). LASSO consists of a centralized optimization application that utilizes a master-worker task distribution strategy. The optimization, master, and worker processes are executed on grid-based computational resources. The worker processes interface with instances of the forward model for distributed task execution. Results are returned to the master for processing by the centralized optimization application. In the following sections of the paper, key components of the application architecture are described in greater detail.

Several different search procedures have been implemented in Java, making up the centralized optimization application. Here, the optimization model representation of the inverse problem is solved using evolution strategies (ES) − a stochastic search heuristic conceptually similar to natural evolution (Back, 1997). An ES-based procedure encodes within an individual the decision variables that describe a potential solution to the problem. The ES search starts with a collection of individuals, referred to as a population. The objective function of the optimization model is used to quantify a fitness value indicating how well an individual solves the inverse problem. Fitness values are calculated using the results of forward model evaluations. During the search process, the population is iteratively subjected to stochastic selection and mutation search operators. Each iteration of the algorithm constitutes a generation. This search process continues until a predefined convergence criterion is satisfied. The application of ESs to inverse problems is advantageous because of their robustness and global search characteristics. Some drawbacks, however, include the computational intensity of a typical ES search and slow final convergence prior to termination.

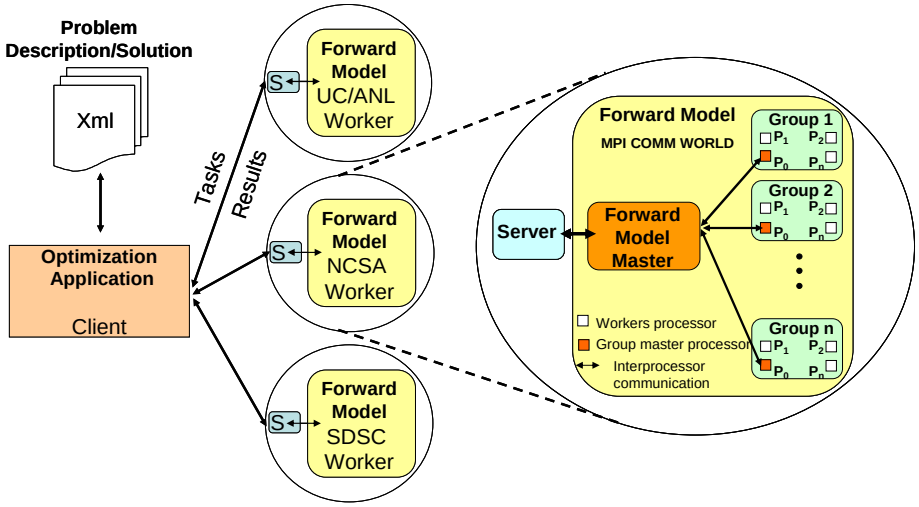

Figure 2: A schematic of the framework architecture

4. FRAMEWORK PARALLELISM

The optimization application is coupled with the forward model (PGREM3D) for fitness evaluation. The simulator is based on a finite element method (FEM) and is written in Fortran. Each FEM evaluation is handled by a group of moderate number of processors (typically 1-8) called "group size" or number of processors per group. The MPI message passing interface is used to group processors, associate processors to computational domains, and for fine grained message passing within each of these groups (Mahinthakumar, 2005). The solution procedure adopted here involves two levels of parallel granularity exhibited by the search procedure and the forward model. Each iteration of the search procedure exhibits a coarse grained parallel structure that requires an uncoupled forward model evaluation for each individual in the population. The optimization application acts as the master process in the master-worker task distribution strategy. The master, worker, and task pool used in the framework were designed and implemented as part of Vitri (Baugh, 2003). The master maintains a pool of remote tasks − a bundle of individuals requiring evaluation. Aggregating individuals in this manner reduces communications overhead. Worker processes running on distributed grid resources, having established a TCP-IP socket connection with the master, signal their readiness and draw tasks from the task pool. The worker process transfers the remote task to the MPI zeroth processor. From this point forward, standard MPI group communications are utilized. The forward model manages multiple MPI groups, and each group evaluates an individual in the task bundle. The results of these simulations are then aggregated into a result bundle and returned to the optimization application for processing by the search algorithm. Finally, the next generation of the search is initiated.

The computational experiments presented here were performed on the National Science Foundation (NSF) TeraGrid, NCSA. The TeraGrid is a heterogeneous agglomeration of computational resources distributed across the United States and connected through a specialized interconnection network designed for high-band width data transfer (Catlett, 2002).

5. APPLICATION EXPERIMENT DESIGN AND SETTINGS

Two sets of runs were conducted on the TeraGrid NCSA site to address inverse problems and to measure the performance of the current framework. A set of runs were devoted to attempt solving the application problems by changing the ES parameters: number of generations, population size and sigma value. Another set of runs were conducted to measure the performance of the framework in terms of fine and coarse grained parallelism. The following section defines the forward model and the search parameter setting used in this study.

For both applications, the grid resolution for the simulation model resulted in 17,391 finite element nodes within the groundwater domain, and the simulation duration was 100 time steps, using 18 monitoring wells. The SHR problem was formulated at a frequency (Δtm) of 10 times the simulation model time step. Thus, a total of 180 observations are used corresponding to 18 wells and 10 periodic samples. The release history reconstruction period, tr, was set equal to the simulation duration, and the reconstruction period frequency, Δtr, was set equal to 10 times the simulation model time step, thus the number of decision variables was 10. The population size for the LASSO evolution strategies was 300 and the algorithm was executed for 100 generations. The best solution was found at a sigma setting of 8.0. The CSI problem was formulated using 18 wells and 100 time steps. The boundary parameters used for this problem were $C_{max} = 100$, $x_{max} = 51$, $y_{max} = 31$ and $z_{max} = 11$. With 7 decisions variables in the optimization problem, the best solution was found at a setting of sigma equal to 3.0, population size equal to 300 and number of generations equal to 100.

6. RESULTS

Several trials were first conducted by tweaking the LASSO evolution strategies parameters, *population size, number of generations and sigma*. Figure 3 and 4 show the true and the calculated values for the RHC and the CSI problems, respectively. Noticeably, for the RHC problem the true and the estimated values are almost equal. However, for the CSI problem the true and the estimated values do not match as closely. In the RHC problem the unknowns are only concentration values over time period while for the CSI problem the unknown are the source location and concentrations. That makes the CSI problem more complex due to nonuniqness in the nature of the problem. From the groundwater governing equations, the concentration has a linear trend while the source location has nonlinear trend, which makes, in general, the estimation of source location more complex than concentration. In comparison to the RHC problem, tweaking the evolution strategies parameters for the CSI problem required several runs due to the significant effect of nonuniqueness. While the RMS is not fully minimized yet, improved sigma value could result in better solution performance since sigma is more sensitive than other parameters.

The number of forward model evaluations performed during a search is a function of ES population size times the number of generations. For instance, for the CSI problem we had a total of 30000 evaluations distributed among 32 parallel forward model groups with 2 processors per group for a total of 64 processors. On a normalized basis, an evaluation took around 0.0696 second, versus 1.6668 second on a single processor, i.e. each run took around 35 minutes instead of 14 hours.

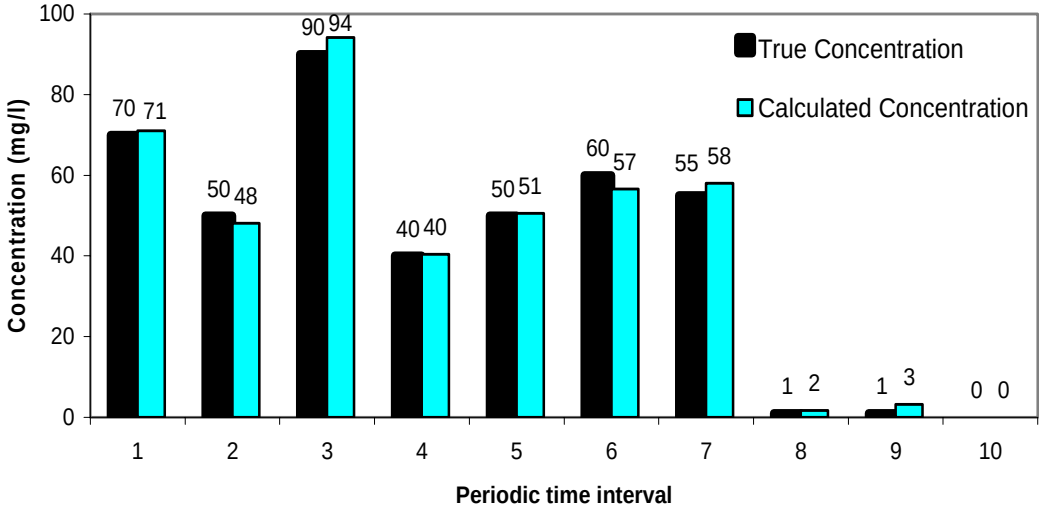

Figure 3: Calculated and observed concentrations for the RHC problem

To evaluate the performance of the simulation-optimization framework, three sets of runs were conducted. One set of runs investigated fine grained parallelism within the FEM simulator. In this study, we evaluated the wall time corresponding to an increasing number of processors per group, while number of groups and number of tasks remained constant. Figure 5 illustrates that the evaluation time decreases until 4 processors per group then increases thereafter. That is because the improvement in fine grained parallelism is associated with the problem size. As the problem size used in this study is relatively small, the improvement is limited.

The second set of runs was used to investigate coarse grained parallelism. We observed the evaluation time with increasing number of groups. Other parameters such as number of evaluations, number of processors per group and number of tasks per group were kept unchanged. Theoretically, the application is expected to scale linearly with the number of processors used. Scalability results shown in Figure 6 indicate that the framework scales almost linearly until 32 groups and scales slight sub-linearly when the number of group exceeds 32. This could be due to the effect of communication that plays an important role in increasing the wall time when more than 32 groups are used.

The third set of runs was utilized to investigate the effect of the chunk size (number of tasks) sent by the optimizer (the client). In those runs we observed the evaluation time with increasing chunk size. Other parameters, such as number of groups and number of processors per group, were kept constant. Figure 7 shows that the evaluation time for one generation increases until the chunk size is equal to 16 and then decreases until the chunk size is equal to 64. When the chunk size is greater than 64 the evaluation time increases again. A possible interpretation for that is for small chunk sizes (less than 16) the latency dominates the evaluation time, while for large chunk sizes (greater than 64) the bandwidth dominates the evaluation time. For this study the best performance occurs when the chunk size between 32 and 64.

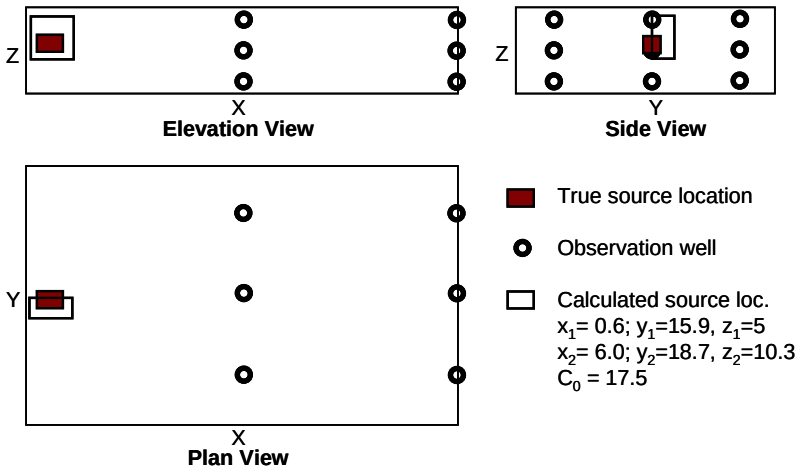

Figure 4: Estimated and true source location for the CSI problem

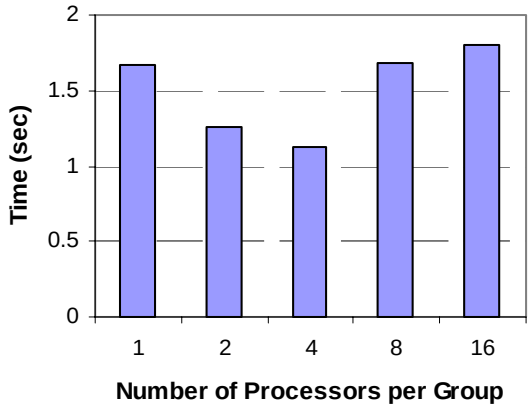

Figure 5: Fine grained parallelism, Number of Group = 1, Tasks/Group = 1:1

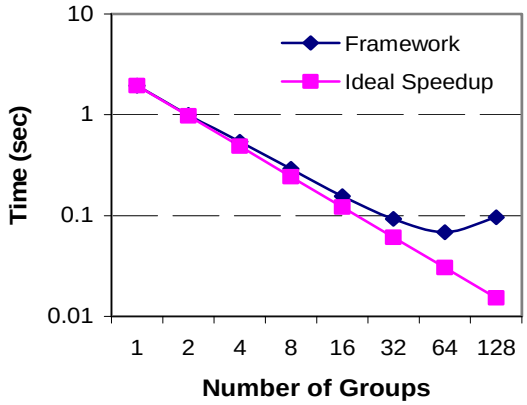

Figure 6:. Coarse grained parallelism, Procs/Group =1:1, Tasks/Group =1:1

7. OBSERVATIONS AND REMARKS

In this paper we have addressed two instances of inverse groundwater problems using a simulation-optimization framework implemented on the TeraGrid site at NCSA, where the solution procedure employs an ES-based search procedure. The results indicate that the solution for the SHR problem performs better than the CSI problem, because the effect of nonuniqueness is more significant in the CSI than the SHR problem. Grid-enabled parallelized framework along two levels, coarse grained at the search algorithm level and fine grained at the forward model level, reduced the evaluation time drastically, to minutes instead of hours. For future work, different solution procedures would be attempted for better solution performance such as modeling to generate alternatives and hybrid algorithms. For

the CSI problem, the source dimensions could be constrained further to simplify the search and to reduce the non-uniqueness issue.

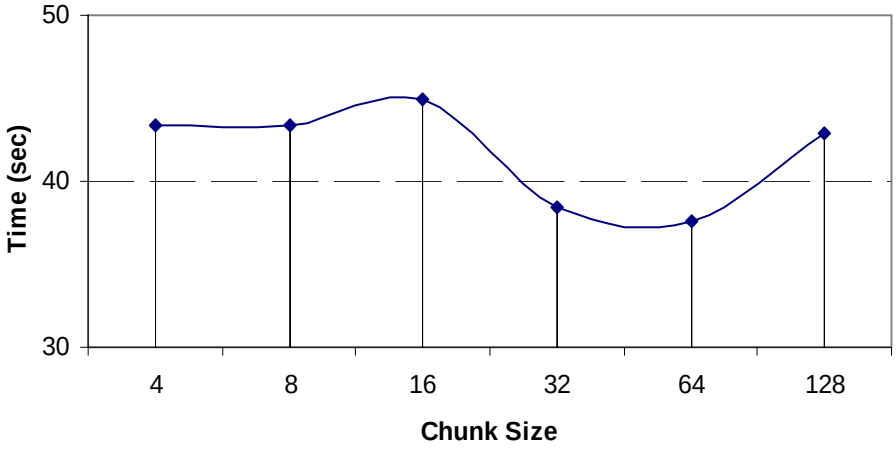

Figure 7: Chunk size versus Evaluation time
Number of groups = 4, population size = 128, Procs/Group =1:1

ACKNOLWDEGEMENTS

This work was supported by National Science Foundation (NSF) under Grant number BES-0312841 and BES-0238623. The authors are grateful for the TeraGrid supercomputer resources provided by NCSA.

BIBLIOGRAPHY

Atmadja, J. and Bagtzoglou, A.C. (2001), "State of Art Report on Mathematical Methods for Groundwater Pollution Source Identification", Environmental Forensics, 2(3) 205-214, 2001b.

Back, T. (1997), "Handbook of Evolutionary Computation", IOP Publishing Ltd. and Oxford University Press.

Baugh, J. W. (2003), "Vitri 2.0.", Available: http://www4.ncsu.edu/~jwb/vitri/vitri-2.0/

Bear, J. (1979), "Hydraulics of groundwater", New York: MCGraw-Hill.

Catlett, C. (2002), "The teragrid: A primer". Available: http://www.teragrid.org/

Mahinthakumar, G. (1999), "Pgrem3d: Parallel groundwater transport and remediation codes," users Guide. Available: http://www4.ncsu.edu/~gmkumar/pgrem3d.pdf

Mahinthakumar, G. and Sayeed, M. (2005), "Hybrid genetic algorithm local search methods for solving groundwater source identification inverse problems", Water Resource. Plng. and Mgmt., vol. 131, no. 1, pp. 45–57, January/February.

Sun, N. (1994), "Inverse problem in groundwater modeling, Theory and application of Transport in porous media", (Ed. Jacob Bear), vol. 6, Kluwer Academic Puplishers, 337p.

Tryby, M., Mirghani, B., Ranjithan, R., Baessler, D., Nicholas, K. and Mahinthakumar, M. (2005), "LASSO: A Grid-Enabled Simulation Optimization Framework", Poster presentation at Grid Computing Workshop, SC|05, Seattle WA.