Online Supplement for

Parallel Stochastic Global Optimization Using Radial Basis Functions

INFORMS Journal on Computing

Rommel G. Regis
Mathematics and Computer Science Department, Saint Joseph's University,
Philadelphia, Pennsylvania 19131, USA, rregis@sju.edu

Christine A. Shoemaker
School of Civil & Environmental Engineering and School of Operations Research & Information Engineering, Cornell University, Ithaca, New York 14853, USA, cas12@cornell.edu

Appendix

A. Alternative Parallel Global Optimization Methods

We compare our proposed parallel RBF algorithm with a variety of alternative parallel global optimization methods for expensive functions. These methods are discussed below. The choice of alternative methods in this study is constrained by the availability of suitable software. Commercial software that runs on a cluster is expensive and developing software that accurately implements an optimization method is usually time consuming. Hence, we used publicly available software as much as possible. However, the simulation codes for our groundwater bioremediation application problems currently run only on Windows. Hence, we needed optimization software that satisfies three criteria: publicly available, runs in parallel, and runs on Windows. We found out that there are only few optimization software (e.g. APPS by Hough et al. (2001)) that satisfy these criteria. We also did not find any publicly available code that implements a parallel surrogate model-based global optimization algorithm that runs on Windows. Hence, the only alternative parallel surrogate model-based global optimization method in this study is ParCGRBF-R, which is our previously developed parallel RBF algorithm. There are several publicly available optimization software that run on Windows but do not run in parallel. This led us to develop parallel implementations of existing serial optimization codes such as BLMVM by Benson and Moré (2001) and PGSL by Raphael and Smith (2003). Finally, we also developed our own parallel implementation of a well-known evolutionary algorithm since this was fairly straightforward.

A.1. Multistart Parallel Finite-Difference Quasi-Newton Methods

In general, derivative-based methods are efficient at finding the local minima of smooth functions. Hence, a sensible alternative for the global optimization of expensive functions is to use a multistart quasi-Newton method where the derivatives are approximated by finite differencing. In a quasi-Newton method, the iterates are obtained in a manner similar to that in the line search Newton method except that we use an approximation to the inverse Hessian in place of the true inverse Hessian.

Local optimization algorithms (including quasi-Newton methods) require only one starting point. However, in practice, it makes sense to sample the domain first in order to find a good starting point before running a local optimization algorithm. Here, we will use a simple multistart procedure where the first starting point is chosen to be the best point from a space-filling experimental design. Subsequent starting points will be obtained by generating points that are as far away as possible from the trajectories of the previous local minimization runs. When allowed to run indefinitely, this multistart approach converges to the global minimum in the sense that the collection of all points from all local minimization trajectories generated by the procedure will be *dense* in the domain of the problem. This follows from a theorem in Regis and Shoemaker (2005), which was provided in connection with a response surface-based global optimization method. However, the theorem also applies to algorithms that do not use response surface models.

Now, assume that P processors are available. We parallelize a quasi-Newton method by performing *speculative gradient evaluations* (Byrd et al. 1988). That is, if $x_n + \lambda_n d_n$ is the current point that needs to be evaluated for the quasi-Newton method, then we simultaneously evaluate $f(x_n + \lambda_n d_n)$ and $\min(P-1, d)$ components of the gradient of f at $x_n + \lambda_n d_n$ by forward differences. If $x_n + \lambda_n d_n$ is accepted as the next iterate, which happens most of the time, then only $\max(d + 1 - P, 0)$ function evaluations are needed to complete the finite difference gradient that will be used by the quasi-Newton method. If $x_n + \lambda_n d_n$ is not accepted, then this gradient information is not used by the algorithm.

In the numerical experiments we use two quasi-Newton methods for bound constrained optimization: the *nag_opt_bounds_deriv* routine from the NAG C library (The Numerical Algorithms Group Ltd. 2005) and the BLMVM algorithm by Benson and Moré (2001). BLMVM determines its step direction by using projected gradients to construct a limited memory BFGS matrix. We refer to the resulting multistart parallel finite-difference quasi-Newton methods as *MultParNAGQN* and *MultParBLMVM*.

A.2. Multistart Asynchronous Parallel Pattern Search

Pattern search (Torczon 1997) is a derivative-free local optimization method that has been used for expensive objective functions. Each iteration consists of searching for a point whose objective function value is better than that of the current iterate from a finite collection of points around that iterate. This collection of points is independent of the objective function and it is formed by adding the current iterate to a scalar multiple of a fixed collection of vectors called a *pattern*. Asynchronous parallel pattern search (APPS) (Hough et al. 2001, Kolda and Torczon 2004) is a parallel implementation of pattern search that effectively balances the computational load among all available processors, and hence, it works best when there is substantial variation in the function evaluation times. However, the function evaluation times for our groundwater bioremediation models are roughly constant so the performance of APPS is expected to be similar to a synchronous parallel pattern search algorithm. We implement APPS using the same multistart approach described in the previous section and we refer to the resulting algorithm as *MultAPPS*.

A.3. Parallel Evolution Strategy

Evolutionary algorithms have an inherently parallel structure, and so, they are trivially parallelizable. Here, we use a (μ, λ)-Evolution Strategy (or simply (μ, λ)-ES) (Bäck, 1996; Schwefel, 1995), which is an evolutionary algorithm where each generation consists of μ parents that produce λ offspring via crossover and mutation. It differs from a standard genetic algorithm (GA) in that it allows self-adaptation of the parameters that control the algorithm. Typical parameters for an ES are the standard deviations of the normal random perturbations (i.e. mutation rates) that will be applied to the components of an offspring solution vector from a crossover operation. Self-adaptation of the parameters means that the algorithm not only evolves a good set of solutions, but it also evolves a good set of parameters that control the algorithm. The rate of convergence of a (μ, λ)-ES can be very slow. Hence, in any given trial, we keep track of the number of consecutive generations that did not improve the best function value. If the number of such generations exceeds a specified threshold, we restart the parallel (μ, λ)-ES from scratch using a new space-filling experimental design as the initial population. We refer to the resulting parallel algorithm as the *Parallel Evolution Strategy with Restart (ParES-R)*.

A.4. Asynchronous Parallel Probabilistic Global Search Lausanne

Probabilistic Global Search Lausanne (PGSL) is a derivative-free, stochastic global search method developed by Raphael and Smith (2003). The main idea in PGSL is to sample with higher probability in regions containing good solutions. Raphael and Smith (2003) reported that PGSL performed better than a genetic algorithm and an adaptive simulated annealing algorithm in 19 out of 23 test problems in their study.

In this investigation, we implement an asynchronous parallel PGSL by running PGSL simultaneously (without communication) on multiple processors. We refer to this algorithm as *Parallel PGSL (ParPGSL)*. For a given set of processors, we use different random seeds to get different results from the simultaneous PGSL runs. However, since ParPGSL will be run for multiple trials on each test problem, we made the random seed dependent on both the trial number and the processor number. Moreover, we use the parameters for PGSL that are consistent with the ones used by Huyer (2004) when he compared algorithms for bound constrained global optimization, including PGSL.

A.5. Parallel Controlled Gutmann RBF with Restart

Parallel Controlled Gutmann RBF with Restart (ParCGRBF-R) was developed by Regis and Shoemaker (2007c) and it is a parallel implementation of Controlled Gutmann RBF with Restart (CGRBF-R) (Regis and Shoemaker 2007b), which is a modified version of the RBF method by Gutmann (2001). The basic idea in the RBF method by Gutmann is as follows. Let $x_1, \ldots, x_n$ be the previously evaluated points in $\mathcal{D}$ and assume that we have a guess f_n^* of the global minimum value of the expensive function f. Furthermore, for each $y \notin \{x_1, \ldots, x_n\}$, assume that there is a unique RBF s_n^y that interpolates the data points $(x_1, f(x_1)), \ldots, (x_n, f(x_n))$ and the additional data point (y, f_n^*). The next evaluation point x_{n+1} is chosen to be the point $y \in \mathcal{D}$ such that some measure of "bumpiness" of s_n^y is minimized. Regis and Shoemaker (2007b) observed some convergence problems with the RBF method by Gutmann and developed CGRBF-R to address these issues. In particular, CGRBF-R is very similar to the RBF method by Gutmann (2001) except that it has better balance between local and global search and it performs a restart whenever the algorithm fails to make substantial improvements after some threshold number of consecutive function evaluations. Finally, the parallelization of CGRBF-R is achieved by considering multiple guesses of the global minimum value of the expensive function f in each iteration in order to generate multiple points for simultaneous function evaluation in parallel.

B. Problem Descriptions

B.1. Test Problems

We compare the performance of the parallel global optimization algorithms on twenty synthetic test functions with dimensions ranging from 2 to 6. Six of the test functions belong to the collection of nonconvex test functions proposed by Dixon and Szegö (1978). The GWSS and GWSC test problems are described in Schoen (1993) and they have the form:

$$\text{GWSS}(x) = \frac{\sum_{i=1}^{k} f_i \prod_{j \neq i} \|x - z_j\|^2}{\sum_{i=1}^{k} \prod_{j \neq i} \|x - z_j\|^2} \qquad \text{and} \qquad \text{GWSC}(x) = \frac{\sum_{i=1}^{k} f_i \prod_{j \neq i} \|x - z_j\|^3}{\sum_{i=1}^{k} \prod_{j \neq i} \|x - z_j\|^3}$$

where $k \geq 1$; $z_j \in [0,1]^d \ \forall j = 1, \ldots, k$; and $f_i \in R \ \forall i = 1, \ldots, k$. We also include the highly multimodal 3-dimensional Rastrigin3 and Ackley3 functions. Finally, we include eight of the Moré-Garbow-Hillstrom (MGH) test problems (Moré et al. 1981), which we modified by adding box constraints. We added 1 to each of the MGH test functions whose global minimum value is 0 since we will be measuring wall clock times until an algorithm gets a relative error that is less than some pre-specified value (see Section 7.2 of the paper). The relative error is only defined if the global minimum value is nonzero. Each of these MGH test problems has only one local minimum but it is typically hard to find. Table 1 summarizes the characteristics of the synthetic test functions used in this investigation.

Our test problems are not really expensive to evaluate. However, we can still conduct meaningful comparisons of parallel performance for the different algorithms by assuming that these functions are computationally expensive. The relative performance of parallel algorithms on these test problems are expected to be similar to the relative performance of these parallel algorithms on truly expensive functions that have the same general structure as our test problems.

B.2. Management of Groundwater Bioremediation

Parallel Local Metric Stochastic RBF with Restart (ParLMSRBF-R) and other alternative parallel global optimization algorithms will be applied to the problem of minimizing the cost associated with groundwater bioremediation (Minsker and Shoemaker 1998, Yoon and Shoemaker 1999). Groundwater bioremediation is the process of using the energy-producing and cell-synthesizing activities of soil bacteria to transform contaminants in the groundwater into harmless substances. It involves using injection wells that will supply electron acceptors

Table 1: Test Problems for the Computational Experiments. Each problem has exactly one global minimum.

Test Function	Dim	Domain	No. of local min	Global min value
Goldstein-Price	2	$[-2,2]^2$	4	3
GWSC2	2	$[0,1]^2$	≥ 13	-97.5831
Beale	2	$[1,4]\times[-1,2]$	1	0
Jennrich & Sampson	2	$[0,1]^2$	1	124.362
Hartman3	3	$[0,1]^3$	4	-3.86
GWSS3	3	$[0,1]^3$	≥ 15	-82.6229
Broyden Tridiagonal	3	$[-1,1]^3$	1	0
Rastrigin3	3	$[-1,3]^2$	≥ 49	-3
Ackley3	3	$[-1,3]^2$	≥ 30	-22.7183
Box 3-D	3	$[-20,20]^3$	1	0
Shekel5	4	$[0,10]^4$	5	-10.1532
Shekel7	4	$[0,10]^4$	7	-10.4029
Shekel10	4	$[0,10]^4$	10	-10.5364
Discrete Boundary Value	4	$[-3,3]^4$	1	0
Powell Singular	4	$[-1,3]^4$	1	0
GWSC5	5	$[0,1]^5$	≥ 10	-82.0265
Variably Dimensioned	5	$[-2,2]^5$	1	0
Hartman6	6	$[0,1]^6$	4	-3.32
GWSS6	6	$[0,1]^6$	≥ 17	-61.1606
Trigonometric	6	$[-1,3]^6$	1	0

(e.g. oxygen) or electron donors (e.g. hydrogen) into the groundwater to promote the growth of the soil bacteria that will detoxify the pollutants. Monitoring wells are also needed to measure the concentration of the contaminant at specific locations and ensure that it will be below some threshold level at specified time periods.

We consider a setup in which we need to pump oxygenated water into injection wells whose locations have been fixed. Our optimization formulation involves a 2-dimensional finite element simulation model called Bio2D (Taylor 1993) that describes groundwater flow and changes in the concentrations of the contaminant, oxygen and biomass. We evenly divide the entire planning horizon into management periods and the problem is to determine the pumping rates for each injection well at the beginning of each management period in order to minimize the total pumping cost subject to some nonlinear constraints on the contaminant concentration at the monitoring wells. Yoon and Shoemaker (1999) eliminated the constraints in the formulation by incorporating them into the total pumping cost objective by means of a penalty term. This results in a box-constrained global optimization problem

with a nonlinear objective function.

In this investigation, we consider two hypothetical contaminated aquifers whose characteristics are symmetric about a horizontal axis. Each aquifer is discretized using a two-dimensional finite element mesh with 18 nodes in the horizontal direction and 9 nodes in the vertical direction. The two aquifers are the same except that the second one has more contaminant than the first one. In each aquifer, there are 6 injection wells and 84 monitoring wells that are also symmetrically arranged so we only need to make pumping decisions for the 3 injection wells on one side of the axis of symmetry. There are 4 management periods, giving us 12 decision variables for our optimization problems. The maximum pumping rate was set to 1 so the search space is $[0, 1]^{12}$. We refer to the groundwater bioremediation problems corresponding to the two aquifers as GWB12A and GWB12B. Since there is more contaminant in GWB12B than in GWB12A, we expect that the global minimum value of the objective function for GWB12B to be larger than that for GWB12A.

This groundwater bioremediation model uses a relatively coarse grid so its simulation time is only about 0.25 second on a 3.6 GHz Xeon processor. However, it is representative of the type of function used in more complex groundwater bioremediation problems, whose simulation times can vary between minutes and hours depending on the complexity of the model and the size of the modeled region (Shoemaker et al. 2001).

B.3. Calibration of Groundwater Bioremediation Models

We also consider the problem of estimating the biokinetic parameters of a groundwater bioremediation model of chlorinated ethenes (Mugunthan et al. 2005). In situ bioremediation of chlorinated ethenes involves injecting organic compounds such as butyrate and lactate into the groundwater. The fermentation of these organic compounds produces hydrogen, which serves as the electron donor for biological dechlorination. Our groundwater bioremediation model is called DECHLOR (Willis and Shoemaker 2000), which is a multispecies reactive transport model which includes microbial competition for electron donor. DECHLOR uses MODFLOW (McDonald and Harbaugh 1988) and RT3D (Clement 1997). MODFLOW is a modular finite difference model for simulating flow while RT3D is a reactive transport code for modeling contaminant transport.

The optimization problem for calibration of our groundwater bioremediation model is to

minimize the total squared residual error given by

$$\mathrm{SSE}(x) = \sum_{t=1}^{T} \sum_{j=1}^{N} \sum_{i=1}^{S} w_{i,j,t} (C^s_{i,j,t}(x) - C^0_{i,j,t})^2 \tag{1}$$

where x is the vector of parameters, $C^s_{i,j,t}(x)$ is the output of DECHLOR that specifies the concentration of species i at monitoring location j and time period t, and $C^0_{i,j,t}$ is the measured concentration of species i at monitoring location j and time period t. There are five species of interest in the calibration problem (i.e. $S = 5$), namely perchloroethene (PCE), trichloroethene (TCE), cis-dichloroethene (DCE), vinyl chloride (VC) and ethene.

We consider a 3-dimensional hypothetical confined aquifer with an initial PCE plume. The aquifer was modeled using a finite difference grid with 3 layers, 14 rows and 26 columns, which gives a total of 1092 nodes. Injection and extraction were carried out from layer 2 and butyrate was used as the organic donor that ferments to hydrogen. Synthetic observations (i.e. concentrations) for the five chlorinated ethene species were generated by simulating DECHLOR over 120 days using a given set of initial and boundary conditions. Five biokinetic parameters and the initial concentration of the dechlorinating bacteria were considered as the unknown parameters to be estimated in the calibration problem. The five biokinetic parameters include half-velocity constants for successive dechlorination of PCE to ethene (one each for PCE, TCE, DCE and VC) and the half-velocity constant for the dechlorination with respect to hydrogen. Thus, we have a total of 6 parameters to calibrate. Using reasonable upper and lower bounds on the calibration parameters, we rescale these parameters so that their values are between 0 and 1 in the calibration problem. For this hypothetical aquifer, each simulation of DECHLOR (which produces all the simulated concentrations $C^s_{i,j,t}(x)$) takes about 220 seconds on a single 3.6 GHz Xeon processor. We shall refer to this optimization problem as DECHLOR6.

C. Relative Speedups

Tables 2 and 3 show the relative speedups of the different parallel algorithms. The discussion of these results can be found in the main paper.

Table 2: Relative speedups of parallel global optimization methods on 12 test problems with multiple local minima when using $P = 4, 8$ processors assuming each function evaluation takes 1 hour.

Test Function	P	Par LMSRBF-R	Par CGRBF-R	ParES-R	ParPGSL	MultPar NAGQN	MultPar BLMVM	Mult APPS
Goldstein-	4	3.07	6.80	3.98	4.19	2.23	4.06	16.86
Price	8	3.96	10.28	7.91	6.29	2.01	7.65	2.21
GWSC2.10.1	4	3.48	2.33	3.90	4.73	2.55	3.51	1.93
	8	5.81	3.50	7.94	6.71	3.55	7.40	2.69
Hartman3	4	4.14	3.18	3.95	8.87	3.72	3.85	2.38
	8	7.00	4.33	7.76	11.85	4.26	7.53	3.75
GWSS3.10.8	4	3.30	3.86	4.00	2.03	3.74	3.34	2.24
	8	4.48	5.48	7.99	3.60	4.82	2.80	2.86
Rastrigin3	4	3.57	4.55	3.98	2.53	4.01	4.09	2.21
	8	6.13	3.77	7.92	4.89	5.04	4.50	2.44
Ackley3	4	3.89	3.50	3.97	3.74	2.92	5.26	2.62
	8	6.75	4.11	7.88	12.51	5.47	7.26	4.26
Shekel5	4	4.96	4.68	4.00	4.42	2.48	2.19	2.71
	8	3.61	6.06	7.99	5.30	5.99	4.17	4.34
Shekel7	4	2.43	4.05	4.00	3.37	2.58	2.27	1.68
	8	2.70	6.11	7.99	4.00	4.45	5.14	3.86
Shekel10	4	2.75	3.35	4.00	3.89	2.28	1.77	3.23
	8	3.49	4.54	7.98	4.69	5.34	3.66	6.22
GWSC5.8.4	4	3.43	3.36	3.99	4.98	2.56	2.74	2.51
	8	6.51	4.71	7.97	9.73	5.10	5.47	4.84
Hartman6	4	2.94	3.39	4.00	30.92	3.37	3.62	5.88
	8	5.13	3.71	7.98	34.80	6.68	7.19	12.00
GWSS6.8.3	4	3.15	3.65	4.00	3.69	2.90	3.23	1.86
	8	5.62	5.30	7.99	5.70	5.77	6.41	3.22

References

Bäck, T. 1996. *Evolutionary Algorithms in Theory and Practice*. Oxford Univ. Press, New York.

Benson, S.J., J. Moré. 2001. A limited-memory variable-metric algorithm for bound-constrained minimization. Technical Report ANL/MCS-P909-0901, Mathematics and Computer Science Division, Argonne National Laboratory.

Byrd, R.H., R.B. Schnabel, G.A. Shultz. 1988. Parallel quasi-Newton methods for unconstrained optimization. *Mathematical Programming* **42** 273-306.

Table 3: Relative speedups of parallel global optimization methods on 8 Moré-Garbow-Hillstrom test problems when using $P = 4, 8$ processors assuming each function evaluation takes 1 hour.

Test Function	P	ParLMSRBF-R	ParES-R	ParPGSL	MultParBLMVM	MultAPPS
Beale	4	3.85	3.97	2.05	3.17	2.69
	8	4.14	7.81	2.62	4.80	2.97
Jennrich	4	2.39	3.98	1.68	2.93	1.16
& Sampson	8	3.98	7.90	2.19	3.27	2.06
Box 3-D	4	3.11	3.96	NA	3.66	2.71
	8	5.66	7.80	NA	4.44	4.14
Broyden	4	3.35	4.00	4.39	4.03	2.52
Tridiagonal	8	4.71	7.99	6.27	2.10	2.76
Powell	4	4.39	3.99	4.64	2.70	2.37
Singular	8	13.48	7.96	6.58	5.06	3.37
Discrete	4	3.44	4.00	2.20	2.74	2.27
Boundary Value	8	7.01	7.98	3.38	5.07	3.38
Variably	4	4.64	4.00	1.20	3.00	2.56
Dimensioned	8	10.08	7.99	1.30	5.93	6.68
Trigonometric	4	3.39	3.99	1.43	3.52	2.81
	8	6.03	7.93	1.65	6.80	5.31

Clement, T.P. 1997. RT3D - a modular computer code for simulating reactive multispecies transport in 3-dimensional groundwater aquifers. Technical Report, Pacific Northwest Laboratory. Richland, WA.

Dixon, L.C.W., G. Szegö. 1978. The global optimization problem: an introduction. L.C.W. Dixon, G. Szegö, eds. *Towards Global Optimization 2*. North-Holland, Amsterdam, The Netherlands. 1–15.

Gutmann, H.-M. 2001. A radial basis function method for global optimization. *Journal of Global Optimization* **19(3)** 201–227.

Hough, P., T.G. Kolda, V.J. Torczon. 2001. Asynchronous parallel pattern search for nonlinear optimization. *SIAM Journal on Scientific Computing* **23(1)** 134–156.

Huyer, W. 2004. A comparison of some algorithms for bound constrained global optimization. *Unpublished manuscript*.

Kolda, T.G., V. Torczon. 2004. On the convergence of asynchronous parallel pattern search. *SIAM Journal on Optimization* **14(4)** 939–964.

McDonald, M.G., A.W. Harbaugh. 1988. A modular three-dimensional finite-difference

ground-water flow model. Technical Report, U.S. Geological Survey. Denver, Colorado.

Minsker, B.S., C.A. Shoemaker. 1998. Dynamic optimal control of in-situ bioremediation of groundwater. *Journal of Water Resources Planning and Management* **124(3)** 149–161.

Moré, J.J., B.S. Garbow, K.E. Hillstrom. 1981. Testing unconstrained optimization software. *ACM Transactions on Mathematical Software* **7(1)** 17–41.

Mugunthan, P., C.A. Shoemaker, R.G. Regis. 2005. Comparison of function approximation, heuristic and derivative-based methods for automatic calibration of computationally expensive groundwater bioremediation models. *Water Resources Research* **41**, W11427, doi:10.1029/2005WR004134.

Raphael, B., I.F.C. Smith. 2003. A direct stochastic algorithm for global search. *Applied Mathematics and Computation* **146** 729–758.

Regis, R.G., C.A. Shoemaker. 2005. Constrained global optimization using radial basis functions. *Journal of Global Optimization* **31** 153–171.

Regis, R.G., C.A. Shoemaker. 2007b. Improved strategies for radial basis function methods for global optimization. *Journal of Global Optimization* **37(1)** 113–135.

Regis, R.G., C.A. Shoemaker. 2007c. Parallel radial basis function methods for the global optimization of expensive functions. *European Journal of Operational Research* **182(2)** 514–535.

Schwefel, H.-P. 1995. *Evolution and Optimum Seeking*. Wiley, New York.

Schoen, F. 1993. A wide class of test functions for global optimization. *J. of Global Optimization* **3** 133–137.

Shoemaker, C.A., M. Willis, W. Zhang, J. Gossett. 2001. Model analysis of reductive dechlorination with data from Cape Canaveral field site. V. Magar, T. Vogel, C. Aelion, A. Leeson, eds. *Innovative Methods in Support of Bioremediation*. Battelle Press, Columbus, OH. 125–131.

Taylor, S. 1993. Modeling enhanced in-situ biodegradation in groundwater: model response to biological parameter uncertainty. *Proceedings of the 1993 Groundwater Modeling Conference*. International Groundwater Modeling Center, Golden, CO, (4-51)–(4-60).

Torczon, V. 1997. On the convergence of pattern search algorithms. *SIAM Journal on Optimization* **7** 1–25.

Willis, M.B., C.A. Shoemaker. 2000. Engineered PCE dechlorination incorporating competitive biokinetics: Optimization and transport modeling. G. Wickramanayake, A. Gavaskar, B. Alleman, V. Magar, eds. *Bioremediation and phytoremediation of chlorinated and recalcitrant compounds*. Battelle Press, Columbus, Ohio, 1311–1318.

Ye, K.Q., W. Li, A. Sudjianto. 2000. Algorithmic construction of optimal symmetric latin hypercube designs. *Journal of Statistical Planning and Inference* **90** 145–159.

Yoon, J.-H., C.A. Shoemaker. 1999. Comparison of optimization methods for ground-water bioremediation. *Journal of Water Resources Planning and Management* **125** 54–63.

The Numerical Algorithms Group Ltd. 2005. *NAG C Library Manual, Mark 8.* Oxford, UK.

The Mathworks, Inc. 2008. *Matlab Compiler: User's Guide, Version 4.* Natick, MA.