Preprint of the paper submitted to CMWR XVI 2006, Copenhagen, Denmark, 19-22 June, 2006

DERIVATIVE-FREE OPTIMIZATION METHODS FOR HANDLING FIXED COSTS IN OPTIMAL GROUNDWATER REMEDIATION DESIGN

T. HEMKER[1], K.R. FOWLER[2], O. VON STRYK[1]

[1]Simulation and Systems Optimization, Department of Computer Science, Technische Universität (TU) Darmstadt, Darmstadt, Germany
[2] Clarkson Center for the Environment, Department of Mathematics, Clarkson University, Potsdam, NY, USA

Abstract

We consider a hydraulic capture application for water resources management that includes a fixed installation cost in addition to operating costs. The result is a simulation-based, nonlinear, mixed-integer optimization problem. The motivation is that our preliminary studies have shown that convergence to an unsatisfactory, local minimum with many wells operating at low pumping rates is common when the fixed cost is ignored. Such optimization tasks are not unique to subsurface management, rather efficient simulation-based methods are needed in the whole field of computational engineering.

All the approaches used below do not need the gradient of the objective function, only function values for minimization. In one approach, we bypass including the number of wells as a decision variable by defining an inactive-well threshold. In another approach, we use penalty coefficients proposed in the literature to transform the discontinuous problem into a continuous one. For the two above formulations, we use the implicit filtering algorithm. In the third approach, we introduce a mixed-integer problem formulation and use an iterative stochastic modeling technique to build surrogate functions that approximate the objective function. With this new procedure the use of a branch-and-bound technique becomes possible to solve the mixed-integer problem in contrast to methods working directly on the simulation results, which impedes relaxation of integer variables. We present promising numerical results on the benchmarking problem and point the way towards improvement and future work.

1. INTRODUCTION AND MOTIVATION

A hydraulic capture (HC) problem involves the placement of wells to alter the direction of groundwater flow and halt the migration of the contaminant plume [2]. Subsurface simulation is needed to understand the response of the aquifer and predict the fate of the plume. Optimization techniques work in conjunction with the simulators to determine the optimal well-field that meets plume containment constraints at a minimal cost. Typically these numerical simulation codes have been developed for many years and have usually not been designed to meet the specific needs of optimization methods as, e.g., providing gradient information. Decision variables can be real-valued, in the case of pumping rates and well locations, or integer-valued in the case of the number of wells in the system design.

The starting point for such a problem is to develop an objective function that measures the costs to design and operate the well-field. The formulation of the cost objective and the constraints, which include the remediation design aspects, typically dictates which optimization approach can be used. Unfortunately, formulation simplifications are often made due to the availability or understanding of optimization software. It is becoming more accepted that fixed installations costs included with operating costs are needed when remediation time horizons are short, say five or ten years [18, 21]. Using the cost data in [20], it costs roughly \$20,000 to install an extraction well and \$1,000 to operate the well for a year. Ignoring installation costs, and starting with a large set of “candidate wells”, can result in a suboptimal final system design, with many wells operating at low pumping rates. Typically, low-pumping wells are then consolidated and optimization is re-run with the smaller set of candidate wells [1]. An approach that selects the appropriate number of wells in the course of the optimization is much more attractive.

In this work we focus on formulations that include fixed installation costs as well as operating costs, resulting in a simulation-based nonlinear mixed-integer optimization problem. The challenge in this formulation is the integer variable for the number of wells in the system design. Removing a well from the design leads to a large decrease in cost, meaning optimizers must be equipped to either handle a mixed-integer, approximate mixed-integer, or a black-box problem with discontinuities in the objective function. Moreover since evaluation of the objective function requires numerical results from a simulation which may add noise, derivative information is unavailable. Gradient based optimization methods are not appropriate for these applications, hence methods that rely only on function values are more appealing. We compare three derivative-free optimization approaches on an HC application proposed in the literature specifically for benchmarking [20]. This HC problem has been the focus of studies comparing constraint formulations, subsurface simulators, and optimization approaches [16, 7].

In this study we compare three approaches to handle the installation cost. In the first, the objective function remains discontinuous and we use an inactive-well threshold to remove wells from a system design. In the second, we reformulate the problem to preserve continuity by multiplying the installation component by a penalty term. For the above two approaches, we use an implementation of the implicit filtering algorithm for minimization. In the third, we use an iterative stochastic modeling technique to build surrogate functions that approximate the original objective function. With this procedure a branch-and-bound technique combined with sequential quadratic programming is applicable to solve the mixed-integer problem in contrast to methods working directly on the simulation results, which impedes relaxation of integer variables.

We proceed by describing the HC problem, the techniques for handling the installation cost term, and the applied optimization algorithms. We present numerical results in section 5 and conclude by pointing the way towards improvements and future work.

2. HYDRAULIC CAPTURE BENCHMARKING PROBLEM

In [20], an HC problem is posed for the optimization and environmental engineering communities to use for benchmarking purposes. We give a brief overview of the problem here.

The decision variables are the number of wells, $n \leq N$, the pumping rates $\{Q_i\}_{i=1}^n [m^3/s]$, and the $\{(x_i, y_i)\}_{i=1}^n$ locations. The objective function is the sum of the installation (capital) cost J^c and the operational cost J^o given by [20, 19]:

$$\begin{aligned} J \; &= \; \underbrace{\sum_{i=1}^{n} c_0 d_i^{b_0} + \sum_{Q_i<0.0} c_1 |Q_i^m|^{b_1} (z_{gs} - h^{min})^{b_2}}_{J^c} \\ &+ \; \underbrace{\int_0^{t_f} \left(\sum_{i,Q_i<0.0} c_2 Q_i (h_i - z_{gs}) + \sum_{i,Q_i>0.0} c_3 Q_i \right) dt}_{J^o} . \end{aligned} \tag{1}$$

In J^c, the first term accounts for drilling and installing all the wells and the second term represents the additional cost for pumps for extraction wells. Note that $Q_i < 0$ for extraction wells and $Q_i > 0$ for injection well. In J^o, the term pertaining the extraction wells includes the lift cost associated with raising the water to the surface. The cost coefficients and exponents, c_j and b_j are specified in [20]. In (1), $d_i = z_{gs}$ is the depth of well i, Q_i^m is the design pumping rate, and h^{min} is the minimum allowable head. We use the values $h^{min} = 10[m], Q_i^m = \pm 0.0064[m^3/s]$, and $d_i = 30[m]$ for each pump i. The simulation time is $t_f = 5$ years.

The hydraulic heads, $h_i[m]$ for well i, also vary with the decision variables and obtaining their values at each iteration requires a solution to the partial differential equation that models saturated flow. We use Modflow96 [17] for the saturated flow simulation.

We impose the following constraints;

$$-0.0064 \leq Q_i \leq 0.0064[m^3/s], i = 1, ..., n \tag{2}$$

$$30 \geq h_i \geq 10[m], i = 1, ..., n \tag{3}$$

$$Q_T = \sum_{i=1}^{n} Q_i \geq -0.032[m^3/s]. \tag{4}$$

To contain the plume, we implement a head gradient constraint at specified locations around the perimeter of the plume. Consider

$$h_k^{(1)} - h_k^{(2)} \geq d[m], k = 1, ..., M, \tag{5}$$

where M is the number of head gradient constraints imposed around the boundary, $h^{(1)}, h^{(2)}$ are hydraulic head values at specified adjacent nodes for each constraint k, and d is the bound on the difference. Calibration and postprocessing is needed to determine the values of k and d that ensure the plume is captured [1]. For this work we use $M = 5$ and $d = 10^{-4}$. Note that (3) and (5) require the output from a numerical flow simulation.

3. HANDLING THE INSTALLATION COSTS

3.1. **INACTIVE-WELL THRESHOLD.** The reference approach from [7] to determine the number of wells in the design is to set an inactive-well threshold, so that if a well rate becomes low enough, the well is removed from the design space. This eliminates the integer decision variable entirely. For this work, if

$$|Q_i| < 10^{-6}[m^3/s], \tag{6}$$

then the well rate is set to zero and well i is not included in the installation cost. Incorporating (6) leads to large discontinuities in the minimization landscape and a drastic decrease in cost once the well rate falls into this region of the design space.

3.2. **PENALTY COEFFICIENTS.** To reformulate the problem as a continuous one, the authors of [18] propose an approximate mixed-integer approach using a polynomial penalty coefficient method. Here a penalty coefficient is given by

$$\beta_i = Q_i/(Q_i + m), \tag{7}$$

where $0 << m < 1$ is a small number. This penalty term is then multiplied by the fixed cost for each well in J^c as part of (1). Note that in (7), if $Q_i = 0$, then $\beta_i = 0$ and the fixed cost for well i does not contribute to the objective function. For this work we used $m = 10^{-6}$.

3.3. **MIXED-INTEGER FORMULATION.** If we take the system as it is given naturally, the management decisions are to install or de-install wells, to move the wells to better positions, and to set the pumping rates in order to minimize total costs subject to constraints (2) through (5).

Motivated by this thought, we introduce a binary vector $s \in \{0,1\}^n$, with s_i as a switch for installation or de-installation of the i-th well. This changes the original capital costs term J^c into

$$\tilde{J}^c = \sum_{i=1}^{n} s_i c_0 d_i^{b_0} + \sum_{Q_i<0.0} s_i c_1 |Q_i^m|^{b_1} (z_{gs} - h^{min})^{b_2}, \tag{8}$$

as well as the operating costs term J^o to

$$\tilde{J}^o = \int_0^{t_f} \left(\sum_{i,Q_i\leq 0}^{n} s_i c_2 Q_i (h_i - z_{gs}) + \sum_{i,Q_i \geq 0}^{n} s_i c_3 Q_i \right) dt. \tag{9}$$

The installation costs $\tilde{J}^c$ are not only simulation independent as in (1), continuous in Q_i for $s \in \{0,1\}^n$, but also continuous in s, if the integrality constraints are relaxed, $s \in [0,1]^n$. This is elementary for the application of a branch-and-bound approach to find a candidate (s^*, Q^*, x^*, y^*), that minimizes $\tilde{J} = \tilde{J}^c + \tilde{J}^o$.

4. OPTIMIZATION APPROACHES

4.1. **IMPLICIT FILTERING.** Implicit filtering (IF) is a projected quasi-Newton method that uses a sequence of finite difference gradients [11]. The difference increment is reduced as the optimization progresses to take advantage of the fast convergence of quasi-Newton methods near a local minimum. Because (IF) relies on finite difference gradients, only function values are needed to guide the minimization. For this work, we use a FORTRAN implementation called IFFCO, with the symmetric rank one quasi-Newton update [5]. We used the default optimization parameter settings. There are several convergence theorems for implicit filtering, which was particularly designed for the optimization of noisy functions, and indeed IFFCO has been successfully applied to other groundwater management problems [7, 8, 3].

4.2. **ITERATIVE UPDATED SURROGATE-FUNCTIONS.** Solving mixed-integer nonlinear problems is a challenging task, even when the objective is described analytically, because the process combines difficulties from both continuous and discrete optimization. An overview of optimization methods for these problems is given in [4]. In our approach we use a classical branch-and-bound (BB) method to guaranty the integrality constraints on the switching vector s for new candidate system designs (s, Q, x, y). The BB starts with relaxing the integrality constraints on s so only a standard nonlinear program (NLP) has to be solved. However, this normally doesn't carry out a feasible minimizer of $\tilde{J}$, with $s \in \{0,1\}^n$, so that the different s_i are set step by step to one and to zero, to generate a BB-tree. The leaves of the tree represent all possible zero-one combinations for s. All branches in the BB-tree with higher function values than prior found feasible solutions are discarded from further evaluations. See also Grossmann [12] who characterizes such techniques based on the decomposing the optimization problem, or the textbook of Floudas [6] for a broader introduction on these methods.

The problem considered here is challenging in that the operating cost depends on the results of the subsurface flow simulation, so that $\tilde{J}$ is not relaxable with respect to s and as described above, and gradient information is not available. To avoid these problems, we apply an extension of the classical design and analysis of computer experiments (DACE) approach [22] described in [13]. Here, the underlying process with real-valued and integer-valued variables is approximated to build an analytic surrogate function with real-valued variables only. Under the assumption of a real-valued $s \in [0,1]^n$, a DACE-model $\hat{J}^o$ for $\tilde{J}^o$, which satisfies the interpolation constraints

$$\hat{J}^o(s^{(j)}, Q^{(j)}, x^{(j)}, y^{(j)}) = \tilde{J}^o(s^{(j)}, Q^{(j)}, x^{(j)}, y^{(j)}), \text{ for } j = 1, ..., N,$$

for a set of system designs $(s^{(j)}, Q^{(j)}, x^{(j)}, y^{(j)})$, $j = 1, ..., N$, is given by

$$\hat{J}^o(s, Q, x, y) = v_{\tilde{J}^o}(s, Q, x, y)\beta_{\tilde{J}^o} + Z_{\tilde{J}^o}(s, Q, x, y). \tag{10}$$

The first component, consisting of $v_{\tilde{J}^o}$ as a vector of basis functions and $\beta_{\tilde{J}^o}$ as real-valued vector, describes the global drift or trend. The second component $Z_{\tilde{J}^o}$ is a stationary Gaussian random function with a mean of zero, a covariance controlled by a correlation function R, and a variance σ^2. Z models the lack-of-fit between the trend and the interpolation points. For more details in theory of computer experiments, compare to [14]. The DACE-models here are evaluated by the Matlab Kriging Toolbox of Lophaven et.al. [15].

This consideration yields a surrogate objective function $\hat{J} = \tilde{J}^c + \hat{J}^o$, which no longer depends on simulation evaluations. Under the assumption, that $\hat{J}$ reflects the major characteristics of $\tilde{J}$, a minimum of $\hat{J}$ is a promising system design for the original problem.

In contrast to other investigated approaches we include the constraints on the flow direction (5) explicitly into the NLP-subproblems by generating DACE-models analog to (10) by

$$d \leq \hat{g} = v_g(s, Q, x, y)\beta_g + Z_g(s, Q, x, y), \tag{11}$$

with interpolation constraints for the set of N basis points

$$\hat{g}_k(s^{(j)}, Q^{(j)}, x^{(j)}, y^{(j)}) = h_k^{(1)}(s^{(j)}, Q^{(j)}, x^{(j)}, y^{(j)}) - h_k^{(2)}(s^{(j)}, Q^{(j)}, x^{(j)}, y^{(j)}),$$

for $j = 1, ..., N$, and $k = 1, .., M$. These NLP-subproblems at the different knots and leaves of the BB-tree are be solved via sequential-quadratic-programming (SQP) methods [10], because numerical noise is avoided by working on the surrogate functions. In our approach we use SNOPT for Matlab [9], which returns the best candidate found, and its merit function value to the BB as the result in a knot. The basis points for the DACE-model are not determined a priori by a space filling set of system designs [14]. We prefer an update strategy where new points improve the approximation quality of the DACE-model (10) iteratively. This way, all information obtained from earlier simulation runs is included in the decision for the new basis points.

The minimizer of the surrogate function $\hat{J}$ is added to the set of basis points for the DACE-model in the next iteration step. Such sequential procedures are discussed in greater detail in [23]. However, in contrast to the update strategies discussed therein, we use convergence to a previously determined system design as criteria to stop minimization on $\hat{J}$ and to search for a new simulation candidate. If the actual found minimizer is included in an ϵ-ball around a basis point $(s^{(j)}, Q^{(j)}, x^{(j)}, y^{(j)})$, $j = 1, ..., N$, the global approximation quality of $\hat{J}^o$, hence also for $\hat{g}$, is improved in regions of the design space, which are the "most unexplored". This is measured by the expected mean square error (MSE) of the DACE-model. The system design that is added to expand the set of basis points maximizes the MSE of $\hat{J}$. The initial basis points for the HC-problem are taken from the first iterations of the reference solution in [7]. The main procedure is summarized in a general setting in the following algorithm:

Algorithm 1 Surrogate function update for objective function f and variabel x:

1: Evaluate $f(x_i)$ for a set of basis points x_i, with $i = 1, ..., N$.
2: Build surrogate function $\hat{f}$ by running DACE with the set of basis points x_i, for $i = 1, ..., N$.
3: Search the minimizer $\hat{x}^*$ of $\hat{f}$.
4: If $\hat{x}^*$ is too close to a basis point, $|\hat{x}^* - x_i| \leq \epsilon$, $i = 1, ..., N$, go to 5, else to go 6.
5: Search for the maximizer $\hat{x}^{MSE*}$ of the $MSE(\hat{f})$.
6: Add $\hat{x}^*$ respectively $\hat{x}^{MSE*}$ to the set of basis points, stop or go to step 1.

The key feature of such surrogate function approaches is that the optimization process in each iteration forbears from running in a loop with the numerical simulation. The emerging computational costs to determine new candidates to be simulated can be neglected if the costs to run the simulation are taken into account.

Method	Function-value [$]	x_{active} [m]	y_{active} [m]	Q_{active} [m^3/s]	Modflow runs	Infeasible Modflow runs
Inact.-well threshold	24,032	670	250	-0.00574999	363	118
Penalty coefficients	23,640	640	260	-0.00549999	574	186
Mixed-integer	23,908	620	260	-0.00568984	50	41

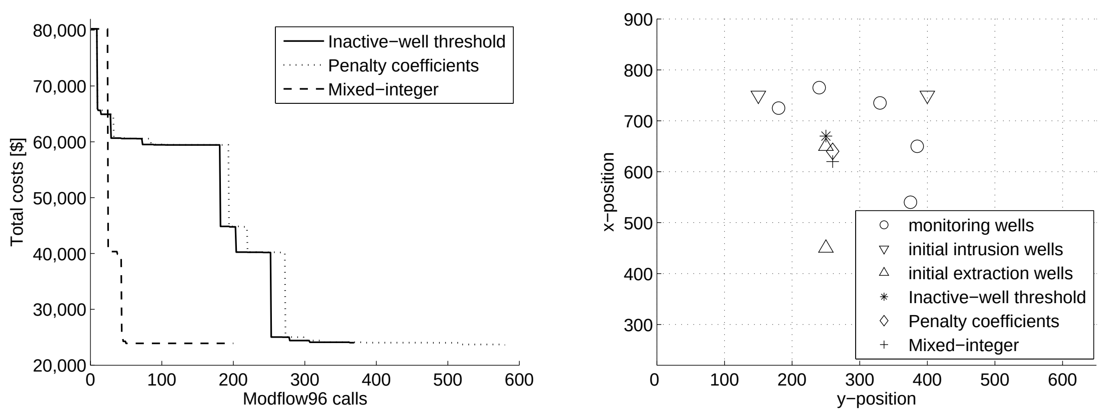
TABLE 1. Numerical results

5. NUMERICAL RESULTS AND DISCUSSION

We apply the three approaches above on the HC problem proposed in [20]. We point to [20, 7] for specific details in the model and just provide a general description here. The hydrologic setting is a $1000 \times 1000 \times 30[m]$ homogeneous, unconfined aquifer. Boundary conditions include no flow on the bottom, south, and eastern edges of the physical domain with prescribed head conditions in the north and west. A flux boundary condition on the top of the aquifer incorporates recharge. As described in [20], the plume is generated from a finite source for five years prior to the remediation period with a constant concentration of $1kg/m^3$ located at $[(200, 225), (475, 525), (h, h-2)][m]$ in the physical domain. We use MT3DMS, [24] to generate the plume and check for containment, and use the 5×10^{-5} contour line as the plume boundary.

All methods used an initial well design with two injection wells and two extraction wells set the the maximum allowable pumping rates, which is proposed by [7] in the reference solution. The cost of the initial design is $80,211. Table 1 shows the final cost, the final well design (which included only one extraction well), the number of calls to the simulator, and the number of infeasible trial designs for each optimization approach.

In Figure 1.(a) we show the total costs as a function of the number of simulation calls as a means of observing the progress of each optimizer. An impression of the proposed final system designs as well as the locations of the static installations on the domain is given by Figure 1.(b).



(a) Total costs vs. simulation runs (b) Locations of the different installations

FIGURE 1. Objective function reduction and final system designs

As a result we see that the reference method, the penalty coefficients method from [18] extended to handle moving wells, and also the introduced mixed-integer formulation combined with the proposed surrogate function approach are able to locate a system design with only one well, ultimately minimizing the fixed installation portion of the cost. Since the location and extraction rate from each optimizer is nearly the same, the final costs barely differ.

The two implicit filtering results show similar convergence properties since both formulations require a well rate to progress towards zero in the course of the optimization. Implicit filtering is able to handle the discontinuous formulation via the inactive-well threshold which offers a slight advantage in taking less function evaluations to reach a similar design than the penalty coefficient method, but with the highest final objective function value of all three approaches. The new mixed-integer formulation proves successful at improving the computational efficiency while simultaneously finding a reasonable solution of a quality between the results of the other tested approaches.

The feasibility aspect is an issue for this application, further adding to the complexity in choosing an optimization algorithm. As Table 1 indicates, during all tested optimization approaches a quota of infeasible system design are simulated. The explicit inclusion of the constraints of the flow direction by surrogate functions could be a second point beside of the direct handling of integer variables to explain the small number of simulation runs the third approach requires. For this work, we only provide numerical results for one initial system design, which was determined from an engineering perspective. However, we should note that the results of each optimizer are sensitive to both the initial design and the technique for handling infeasible points due to constraint violation.

The described observations lead to some aspects for future work in the area of simulation-based optimization, which include the further study on constraint formulation, prediction of feasibility for simulation-based constraints and the iterative improvement of them. Another point is the investigation of possible hybrid approaches to overcome the drawbacks of a single optimizer, for example, to alleviate the dependence on good initial data or failure due the vast regions of infeasibility. The continued study on benchmarking problems of increasing complexity including more realistic physical domains, more sophisticated well models and pumping strategies, and increasing problem dimensionality, could improve the acceptance of practitioners for new optimization approaches in this field.

We hope to increase interest in the presented benchmark problem and point the reader to the community problems webpage, `http://www4.ncsu.edu/eos/users/c/ctkelley/www/community.html`. There, the HC application is entirely packaged with subroutines for the objective function and constraints, all the corresponding Modflow96 data files for the flow simulation, and directions to install the simulator. The hope is that interested researchers can easily consider the HC application with new optimization approaches.

Acknowledgement

This work was in part supported by the DFG-Research-Training-Group “Modeling, Simulation and Optimization of Engineering Applications” (GK 853), the Computational Engineering Research Center of the TU Darmstadt, and a U.S. Department of Education GAANN Fellowship.

References

[1] Ahfeld, Page, and Pinder, *Optimal ground-water remediation methods applied to a superfund site: from formulation to implementation*, Ground Water **33** (1995), no. 1, 58–70.

[2] Ahlfeld and Mulligan, *Optimal design of flow in groundwater systems*, Academic Press, San Diego, 2000.

[3] Battermann, Gablonsky, Patrick, Kelley, and Kavanagh, *Solution of a groundwater control problem with implicit filtering*, Optimzation and Engineering **3** (2002), 189–199.

[4] Bussieck and Pruessner, *Mixed-integer nonlinear programming*, SIAG/OPT Newsletter: Views & News **14** (2003), no. 1, 19–22.

[5] Choi, Eslinger, Gilmore, Patrick, Kelley, and Gablonsky, *IFFCO: Implicit Filtering for Constrained Optimization, Version 2*, Tech. Report CRSC-TR99-23, Center for Research in Scientific Computation, July 1999.

[6] Floudas, *Deterministic global optimization theory, methods and applications*, Kluwer Academic Press, 1999.

[7] Fowler, Kelley, Kees, and Miller, *A hydraulic capture application for optimal remediation design*, Proceedings of the 15th International Conference on Computational Methods in Water Resources (CMWR XV), 13-17 June 2004, Chapel Hill, North Carolina, Computational Methods in Water Resources, Volume 2, Developments in Water Science 55 (Miller, Farthing, Gray, and Pinder, eds.), Elsevier Science, Amsterdam, The Netherlands, 2004, pp. 1149–1157.

[8] Fowler, Kelley, Miller, Kees, Darwin, Reese, Farthing, and Reed, *Solution of a well-field design problem with implicit filtering*, Optimzation and Engineering **5** (2004), 207–233.

[9] Gill, Murray, and Saunders, *User's guide for snopt 7.1: a fortran package for large-scale nonlinear programming*, Report NA 05-2, Department of Mathematics, University of California, San Diego., 2006.

[10] Gill, Murray, and Wright, *Practical optimization*, Academic Press, 1981.

[11] Gilmore and Kelley, *An implicit filtering algorithm for optimization of functions with many local minima*, SIAM Journal on Optimization **5** (1995), 269–285.

[12] Grossmann, *Review of nonlinear mixed-integer and disjunctive programming techniques*, Optimization and Engineering **3** (2002), 227–252.

[13] Hemker, Glocker, De Gersem, von Stryk, and Weiland, *Mixed-integer simulation-based optimization for a superconductive magnet design*, Proceedings of the Sixth International Conference on Computation in Electromagnetics (Aachen, Germany), 4-6 April 2006.

[14] Koehler and Owen, *Computer experiments*, Handbook of Statistics **13** (1996), 261–308.

[15] Lophaven, Nielsen, and Sondergaard, *DACE, a matlab kriging toolbox*, Tech. Report IMM-TR-2002-12, IMM, August 2002.

[16] Matott, Rabideau, and Craig, *Pump-and-treat optimization using analytic element method flow models*, Adv. in Water Resources (2005), In Press.

[17] McDonald and Harbarbaugh, *A modular three dimensional finite difference groundwater flow model*, U.S. Geological Survey Techniques of water Resources Research, 31 ed., 1988.

[18] McKinney and Lin, *Approximate mixed integer nonlinear programming methods for optimal aquifer remdiation design*, Water Resources Research **31** (1995), 731–740.

[19] Meyer, Kelley, and Miller, *Electronic supplement to "optimal design for problems involving flow and transport in saturated porous media"*, Adv. in Water Resources **12** (2002), 1233–1256.

[20] ———, *Optimal design for problems involving flow and transport in saturated porous media*, Adv. in Water Resources **12** (2002), 1233–1256.

[21] Ren and Minsker, *Which groundwater remediation objective is better, a realistic one or a simple one*, Journal of Water Resources and Planning Management **131(5)** (2005), 351–361.

[22] Sacks, Schiller, and Welch, *Design for computer experiments*, Technometrics **31** (1989), 41–47.

[23] Sasena, *Flexibility and efficiency enhancements for constrained global design optimization with kriging approximations*, Ph.D. thesis, University of Michigan, 2002.

[24] Zheng and Wang, *MT3DMS: A modular 3-D multispecies transport model for simulation of advection, dispersion, and chemical reactions of contaminants in groundwater systems*, 1999.