Available online at www.sciencedirect.com
SCIENCE DIRECT®
Advances in Water Resources 29 (2006) 760–775
Advances in Water Resources
www.elsevier.com/locate/advwatres

Pump-and-treat optimization using analytic element method flow models

L. Shawn Matott [a], Alan J. Rabideau [a,*], James R. Craig [b]

[a] *Department of Civil, Structural, and Environmental Engineering, 207 Jarvis Hall, University at Buffalo, Buffalo, NY 14260, United States*
[b] *Department of Civil Engineering, University of Waterloo, 200 University Avenue West Waterloo, Ont., Canada N2L 3G1*

Received 17 December 2004; received in revised form 14 July 2005; accepted 19 July 2005
Available online 12 September 2005

Abstract

Plume containment using pump-and-treat (PAT) technology continues to be a popular remediation technique for sites with extensive groundwater contamination. As such, optimization of PAT systems, where cost is minimized subject to various remediation constraints, is the focus of an important and growing body of research. While previous pump-and-treat optimization (PATO) studies have used discretized (finite element or finite difference) flow models, the present study examines the use of analytic element method (AEM) flow models. In a series of numerical experiments, two PATO problems adapted from the literature are optimized using a multi-algorithmic optimization software package coupled with an AEM flow model. The experiments apply several different optimization algorithms and explore the use of various pump-and-treat cost and constraint formulations. The results demonstrate that AEM models can be used to optimize the number, locations and pumping rates of wells in a pump-and-treat containment system. Furthermore, the results illustrate that a total outflux constraint placed along the plume boundary can be used to enforce plume containment. Such constraints are shown to be efficient and reliable alternatives to conventional particle tracking and gradient control techniques. Finally, the particle swarm optimization (PSO) technique is identified as an effective algorithm for solving pump-and-treat optimization problems. A parallel version of the PSO algorithm is shown to have linear speedup, suggesting that the algorithm is suitable for application to problems that are computationally demanding and involve large numbers of wells.
© 2005 Elsevier Ltd. All rights reserved.

Keywords: Groundwater; Optimization; Particle swarm; Analytic element; Pump-and-treat

1. Introduction

Remediation of contaminated groundwater is a continuing problem for the industrialized world. At many contaminated sites, a pump-and-treat system consisting of extraction and injection wells is installed to prevent further plume migration and remove contaminant mass. Designers typically determine the number, locations, and rates of extraction and injection wells such that plume containment is realized at the lowest cost. In this context, the *plume* is defined as a closed boundary delineating a fixed containment area (i.e., a region of groundwater contamination that must not be allowed to expand), and *plume containment* occurs when all groundwater initially residing within the containment area either remains within this boundary, or is extracted by a well during the remediation period.

Optimization of pump-and-treat systems using automated simulation-optimization techniques has been extensively studied over the past quarter century. Mayer et al. [37] provided a detailed review of this body of

* Corresponding author.
E-mail addresses: lsmatott@buffalo.edu (L.S. Matott), rabideau@buffalo.edu (A.J. Rabideau), jrcraig@civmail.uwaterloo.ca (J.R. Craig).

0309-1708/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.advwatres.2005.07.009

research and set forth a suite of community problems intended to serve as a unifying benchmark for future work. While previously published studies of pump-and-treat optimization (PATO) have used finite-difference (FD) or finite-element (FE) flow models, this paper investigates the use of groundwater flow models based on the analytic element method (AEM). Therefore, a major contribution of this study is that it is the first to assess the feasibility of AEM-based PATO. The purpose of this work is not to compare the merits of AEM versus FD/FE for PATO applications, but to examine the applicability of the AEM to this class of problems and develop guidelines for the appropriate PATO formulation.

Several major characteristics distinguish AEM groundwater models from their FD/FE counterparts. In AEM models, hydrologic features are treated as distinct model elements rather than distributed over a spatial grid or mesh. Continuously varying aquifer properties, such as hydraulic conductivity and recharge, are approximated using a set of parameter zones. While such zones are commonly assigned average or effective parameter values, solutions for continuous variation within a zone have been developed (e.g., the multi-quadric area sink [46]). Point features (such as wells) are directly represented in AEM models and their spatial coordinates can be varied as continuous functions. Conversely, such point features can be only approximately located in common FD and FE modeling software, and the effects of such features are averaged over the entire grid- or mesh-cell in which they are placed. Finally, while FD/FE techniques result in a set of discrete nodal or grid cell solutions, AEM flow solutions are inherently continuous over the entire model domain. Introductory treatments of AEM modeling are provided in [45,26], while [29,30] provide details on the high-order techniques underpinning the numerical engine used in this study. The particular AEM implementation used in this study assumes two-dimensional, steady state flow. Such assumptions are common among published PATO studies and should not be a limitation for most PATO applications. Notable exceptions are problems involving time-varying pumping rates and/or flexible management periods (e.g., [8,13–16,34]) and/or containing significant layering of aquifer permeability (e.g., [1,44]).

Given the fundamental differences between AEM and FD/FE flow modeling, the use of AEM introduces several unique possibilities with respect to PATO. For example, AEM-modeling allows for the coordinates of pump-and-treat wells to be directly represented as continuous design variables. Conversely, the majority of previous FD/FE-based PATO studies have optimized well coordinates using a discrete list of candidate well locations. Notable exceptions are found in [27,47], but in these studies potential well locations were nonetheless limited to the nodes of the FE mesh [47] or the cell center of a FD grid [27]. Limitations in the placement of FD/FE wells can be overcome via spatial grid adaptation methods, but such methods can be computationally expensive and result in a more complicated linkage between the simulation and optimization software.

Another benefit of AEM-based PATO arises when particle tracking techniques are used as plume containment indicators. Particle tracking, described by Mulligan and Ahlfeld [40], considers the advection of a series of particles, initially located inside or along the perimeter of the plume. Plume containment is indicated if all particles remain within the plume, or are extracted by pump-and-treat wells, at the conclusion of the remediation time frame. When particle tracking is applied to FD/FE model solutions, the accuracy of a particle track is sensitive to the level of grid or mesh discretization. Conversely, AEM flow solutions are grid-free and the corresponding particle tracks are not affected by domain discretization error.

In contrast to particle tracking, the hydraulic gradient control plume containment technique, introduced by Atwood and Gorelick [3], specifies a series of control locations along the perimeter of the plume. When the head gradient at all control locations is oriented toward the interior of the plume, plume containment is presumed to occur. In FD/FE models, head gradients are approximated using a pair of head values defined at each control location, with one member of the pair located inside the plume and the other outside of the plume. The separation distance between gradient points in such models is restricted to discrete multiples of the grid spacing, limiting the accuracy of the numerical gradient calculations. Conversely, AEM-based PATO allows for exact, analytic computation of head gradients, and provides a unique opportunity to examine the sensitivity of PATO to the inaccuracies of the numerically computed gradient control approach.

While previous PATO studies have used particle tracking and/or gradient control pairs, the present study includes consideration of an alternative plume containment indicator, designated here as the "zone outflux constraint". A complete mathematical description of the constraint is provided in Section 1.4, and is derived from the decomposition of flux (vertically integrated discharge) across an arbitrary transect into influx and outflux terms. A zone flux polygon is then assembled by connecting a series of transects. Plume containment is signified when the total outflux, summed over all transects in a plume-enclosing zone flux polygon, is identically zero.

In consideration of the possibilities suggested by AEM-based PATO, this study examines several AEM-based PATO formulations to identify those that are particularly efficient, effective and reliable. The following

subsections summarize the various facets of PATO considered in this paper: (1) optimization algorithm, (2) system cost, and (3) constraint integration. Mathematical formulations associated with system cost and constraint integration are included in Appendix A.

1.1. Optimization algorithms

Numerous algorithms have been applied to pump-and-treat and related hydraulic optimization; a thorough review and tabulated summary is provided by Mayer et al. [37]. More recent studies have used various heuristic approaches (genetic algorithm, simulated annealing, tabu search, and/or artificial neural networks) (e.g., [6,18,25,35,50]), nonlinear programming (e.g., [25]), and implicit filtering (e.g., [19,20]).

For this study, five algorithms were considered, with the goal of discovering those most effective at solving AEM-based PATO problems. The methods consisted of three heuristic algorithms (simulated annealing (SA) [33], particle swarm optimization (PSO) [32], and a real-coded genetic algorithm (GA) [22,49]), one nonlinear programming algorithm (Fletcher–Reeves conjugate-gradient (CG) [41]), and a random (RND) search algorithm.

The choice of GA and SA algorithms was motivated by the fact that they are heuristic (global) algorithms successfully applied in previous PATO studies (e.g., [2,6,7,17,18,25,27,35,42,48]). The CG algorithm is representative of nonlinear programming, a common approach in the PATO literature (e.g., [25,38,39,47,48]). The random search algorithm, where an optimal solution is selected from a random sample of solution sets, served as a control for the other algorithms.

Importantly, this paper represents the first application of the PSO algorithm to PATO problems. The algorithm was introduced by Kennedy and Eberhart [32], and is an outgrowth of attempts to simulate the cooperative–competitive nature of social behavior. The algorithm can effectively solve a variety of engineering optimization problems, and [9,21] demonstrated applications where PSO performance compared favorably with the GA and SA algorithms.

1.2. System cost

Mayer et al. [37] summarized some of the more commonly used PATO cost formulations, which can be divided into three components: (i) C_{TOTQ}: total pumping rate as a surrogate for cost; (ii) C_{OPER}: operational costs only (such as energy, treatment, disposal and labor costs); and (iii) $C_{\mathrm{OPER+}}$: both operational and capital (well installation and pump) costs. The numerical experiments in this study examined each of the three cost formulations in order to demonstrate the feasibility of their use in an AEM-based PATO problem, and to assess the influence of the cost function on *algorithm effectiveness* (the ability of a given optimization algorithm to determine the minimum cost plume containment system). Details on the three cost formulations, as implemented in this study, are provided in Appendix A.

1.3. Integration of constraints

The pump-and-treat objective can be mathematically formulated as a combination of the system cost function (either C_{TOTQ}, C_{OPER}, or $C_{\mathrm{OPER+}}$) and a penalty function, P_{TOTAL}, which accounts for the cost of various constraint violations. Mayer et al. [37] provide a summary of the types of constraints that have been applied to pump-and-treat optimization, and this study demonstrates that AEM-based PATO is capable of handling commonly applied constraints on capacity, drawdown and plume containment. *Capacity* constraints limit total pumping so that the PAT system does not overload an established treatment facility, *drawdown* constraints prevent aquifer dewatering, and plume *containment* constraints enforce the remedial goals of the PAT system. Formulas for P_{TOTAL} and associated constraints are given in Appendix A.

Chan Hilton and Culver [7] discuss several techniques for combining cost and P_{TOTAL} to form the objective function, including the additive penalty method (APM), the multiplicative penalty method (MPM), and the exponential penalty method (EPM); formulations for these methods are provided in Appendix A. Following the reasoning of Mulligan and Ahlfeld [39], the APM was initially viewed as the preferred penalty method for all experiments in this study because the MPM and EPM methods tend to over-penalize constraint violations, biasing the optimization to favor constraint reduction over cost reduction. However, a set of initial experiments indicated that while the APM was effective in problems utilizing particle tracking constraints, the MPM performed better for gradient control and zone outflux constraints.

1.4. Zone outflux constraint

The zone outflux constraint enforces plume containment by utilizing the normal component of the vertically integrated discharge, Q_η [L^2/T], which may be expressed as

$$Q_\eta = \int_0^h q_\eta \, \mathrm{d}z, \tag{1}$$

where h [L] is the saturated thickness of the domain, q_η [L^3/T] is the specific discharge normal to some vertical plane through the aquifer, and z [L] is the vertical direction. Fig. 1 illustrates the decomposition of Q_η across an arbitrary transect into directional components, $Q_\eta^+(s)$ and $Q_\eta^-(s)$, that vary in a piecewise continuous manner

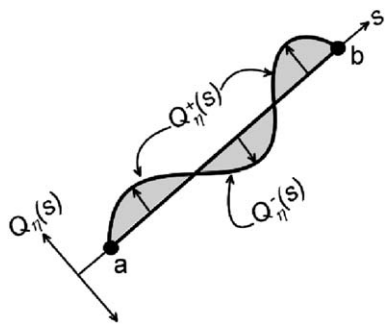

Fig. 1. Example of flux decomposition across an arbitrary transect.

between transect endpoints a and b. Total fluxes (F_{ab}^+ and F_{ab}^-) in each direction are computed via integration along transect ab, as

$$F_{ab}^+ = \int_a^b Q_\eta^+(s)\,\mathrm{d}s, \quad F_{ab}^- = \int_a^b Q_\eta^-(s)\,\mathrm{d}s, \tag{2}$$

where Q_η^+ and Q_η^- are strictly positive and $Q_\eta = Q_\eta^+ - Q_\eta^-$. A "zone flux polygon" can be assembled by connecting a series of transects aligned such that Q_η^+ terms for all transects correspond to outflow from the polygon and Q_η^- terms correspond to inflow. Using this convention, the total outward flux (F_{ZFP}^+) and total inward flux (F_{ZFP}^-) across the boundary of a given zone flux polygon are computed by summing the appropriate transect fluxes, as shown in Eq. (3).

$$F_{\mathrm{ZFP}}^+ = \sum_{i=1}^{N} F_i^+, \quad F_{\mathrm{ZFP}}^- = \sum_{i=1}^{N} F_i^-, \tag{3}$$

where, N is the number of transects in the given zone flux polygon, and F_i^+ and F_i^- are the respective total outflux and total influx across the ith transect boundary of the zone flux polygon.

For pump-and-treat optimization, a zone flux polygon is configured to match the plume boundary, and plume containment is enforced by constraining the total outward flux (F_{ZFP}^+) to be identically zero (i.e., no groundwater exits the plume, unless removed via a well). The zone outflux constraint may be considered equivalent to that of an infinite number of gradient control points placed along the plume boundary. For a given pump-and-treat system design, the zone outflux constraint is violated if the total outflux (F_{ZFP}^+) is nonzero. Such violations cause a penalty (proportional to F_{ZFP}^+) to be assessed to the pump-and-treat system cost.

Evaluation of the transect flux integrals in Eq. (2) is dependent on the groundwater modeling strategy employed. For grid-based models, the flux terms Q_η^+ and Q_η^- across each transect may be interpolated from the discretized flow solution and integrated numerically. Interpolation (and its associated numerical error) may be avoided if the geometry of the zone flux polygon is configured so that polygon sides coincide with cell faces. However, such a rectilinear geometry of the zone flux polygon may lead to an over-conservative constraint formulation. AEM-based models provide a continuous flow solution everywhere in the modeled domain and, regardless of zone flux polygon geometry, no interpolation is required to determine the flux across a given transect. Furthermore, the transect flux terms of an AEM-based flow solution can be integrated in two ways: strictly numerically or numerically with an analytic correction term (hereafter referred to as *semi-analytically*). For the semi-analytic method, the integrals in Eq. (1) are first evaluated numerically. Then the *net* flux through each transect, $F_{ab} = F_{ab}^+ - F_{ab}^- = \int_a^b Q_\eta(s)\,\mathrm{d}s$, is calculated analytically (as in [11, pp. 61–64] and [12, pp. 385–387]). If, for a given transect, flow is unidirectional (i.e., $F_{ab} = F_{ab}^+$ or $F_{ab} = -F_{ab}^-$), then the analytic expression is used. Otherwise, a correction is applied to the numerically calculated total fluxes in order to be consistent with the analytic net flux:

$$\begin{aligned}(F_{ab}^+)_{\mathrm{CORR}} &= F_{ab}^+\left[1 + \frac{F_{ab} - F_{ab}^+ + F_{ab}^-}{F_{ab}^+ + F_{ab}^-}\right],\\ (F_{ab}^-)_{\mathrm{CORR}} &= F_{ab}^-\left[1 - \frac{F_{ab} - F_{ab}^+ + F_{ab}^-}{F_{ab}^+ + F_{ab}^-}\right].\end{aligned} \tag{4}$$

Note that $(F_{ab}^+)_{\mathrm{CORR}} - (F_{ab}^-)_{\mathrm{CORR}} = F_{ab}$, as desired.

In the AEM modeling software utilized in this study, calculation of transect fluxes in a plume-enclosing zone flux polygon is carried out using the semi-analytical technique. Computation of the constraint is highly accurate, fairly inexpensive and convenient in that no decisions regarding the number and location of gradient pairs and/or particles are required. The enhanced accuracy of the semi-analytic technique may be superfluous in the context of a pump-and-treat constraint, as a single outflux term in the numerical integration will trigger a penalty. The analytic correction simply adjusts the penalty value to more accurately represent the severity of the constraint violation, which is unlikely to affect the overall optimization results. However, the additional computation required for the analytic correction is minor compared to the numerical integration step, and the zone flux feature is a standard component of the AEM software used in this work.

2. Methods

To investigate the performance of AEM-based PATO, two problem formulations were developed: (1) a "control" problem, selected from the community problems presented in [37], and (2) a "complex" problem, based on the twin-plume problem presented by Mulligan and Ahlfeld [39]. These problems were selected based on two criteria. First, they were readily adaptable to the AEM modeling software used in this study, which is limited to two-dimensional steady-state flow with

homogeneous conductivity and recharge zones. Second, the two problems differ significantly in terms of domain size, plume geometry, and boundary conditions, allowing for meaningful generalization of observed optimization trends.

In each of the test problems, the optimization design variables were the location and rates of extraction and injection wells. Most optimizations were carried out using well coordinates varied as continuous functions; a subset of the experiments limited well location coordinates to evenly spaced intervals. The optimal number of wells was determined implicitly by allowing algorithms to consider a large maximum number of wells. Of this maximum number, only wells operating above a threshold value (1.5 m^3/day) were retained in the solution.

2.1. Control problem

The AEM setup of the control problem is illustrated in Fig. 2. The problem is based on Application III of the community problems posed by Mayer et al. [37], and is located in an unconfined aquifer with no-flow boundaries to the south and west and specified head boundaries (i.e., a head gradient of 0.001) to the north and east, resulting in a north–northeasterly flow direction near the plume. Following the guidelines for the community problem, aquifer porosity, hydraulic conductivity and recharge were set at 0.32, 4.33 m/day, and 0.00164 m/day, respectively. To facilitate comparison with [19], the plume containment boundary was delineated by the 50 ppb concentration contour resulting from five years of contaminant transport from the initial plume source. In this case, contaminant transport was simulated using Cardinal [11], a multi-species single-phase reactive-transport model that integrates an AEM flow solution with a finite-element or finite-difference transport model.

2.2. Complex problem

The AEM setup of the complex problem is illustrated in Fig. 3. The problem, adapted from a two-dimensional example developed by Mulligan and Ahlfeld [39], is an unconfined aquifer containing no-flow boundaries in the north and south, and specified heads in the west (head gradient of 3/3100) and east (head gradient of 3.17/3100), resulting in southeasterly flow through the plume. Following the description provided by Mulligan and Ahlfeld [39], hydraulic conductivity, recharge and porosity were set to 5 m/day, 0 m/day and 0.3, respectively. The plume is presumed to result from the transport of contaminants from two separate source zones, leading to the dual-lobe shape. For this problem, precise values for the specified head boundaries were not provided in the published literature and were therefore estimated using trial and error to achieve a flow-field closely resembling the descriptions provided by Mulligan and Ahlfeld [39].

2.3. Plume containment constraints

To assess the sensitivity of particle tracking and gradient control constraints to the number and distribution of particles and control pairs, several different formulations of these constraints were implemented. Particle tracking and gradient control constraints were divided into three size-classes, corresponding to "low", "medium" and "high" numbers of constraints, equal to 2, 5, and 10 times the maximum number of wells, respec-

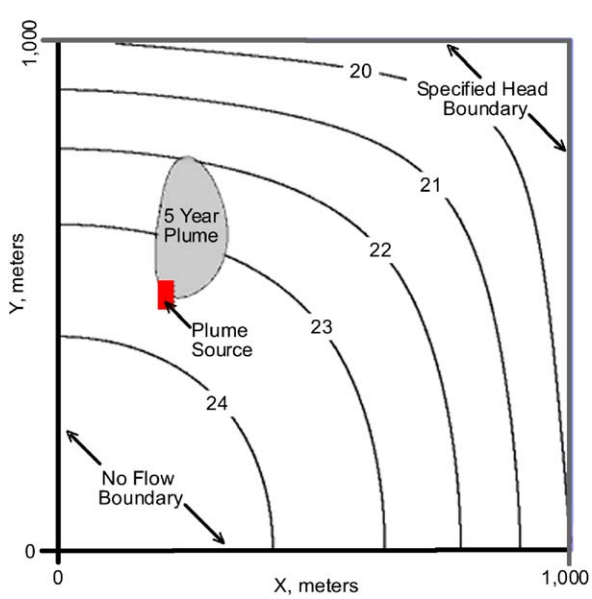

Fig. 2. Control model setup with simulated head contours (m).

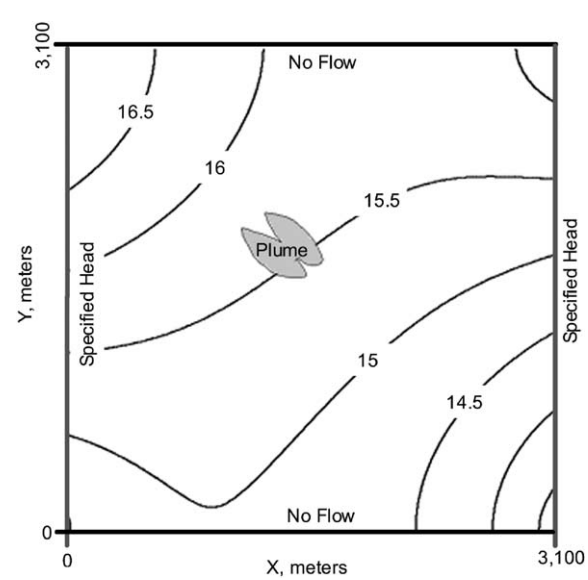

Fig. 3. Complex problem setup with simulated head contours (m).

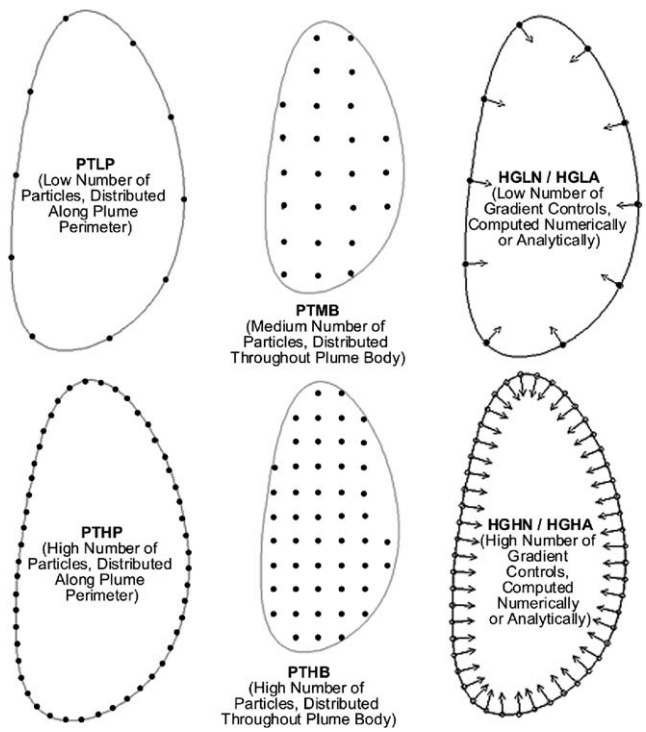

Fig. 4. Selected plume containment constraint formulations.

tively. Particle tracking constraints were further divided into two spatial categories: uniform placement of particles throughout the plume *body*, or uniform placement along the plume *perimeter*.

Hydraulic gradient constraints were always placed uniformly along the plume perimeter, but a distinction is made between whether the gradients were computed numerically (based on a distance of 10 m between head inside and outside the plume) or analytically, which is not possible with standard FD/FE methods. In all, the particle tracking, hydraulic gradient, and zone outflux constraints provided a total of 13 different plume containment constraint configurations. Fig. 4 illustrates selected constraints and Table 1 summarizes each of the formulations. In Fig. 4, the arrows associated with gradient control constraints indicate an inward flow direction perpendicular to the plume boundary.

2.4. Optimization algorithms

The performance of each of the optimization algorithms is dependent to some extent on the values assigned to various algorithmic parameters. Where possible, parameters were assigned values taken from published optimization studies, with preference given to references that specifically addressed pump-and-treat optimization. For example, PSO parameters values were taken from the tuning study of [5] and the guidelines of [50] were used in the assignment of SA and GA parameters. Algorithm parameters related to computation time (for example, maximum number of generations) were assigned so that each algorithm would run to completion using a similar number of objective function evaluations.

2.5. Software

Three software packages were utilized in the experiments, which were run on a workstation cluster comprised of 32 1-GHz Intel Pentium III processors with 256-kB cache, 1-GB of RAM, and the Red Hat Linux 7.3 operating system. Two of the software packages, Split [28] and Bluebird [10], are AEM flow modeling engines that incorporate high-order elements. For experiments with particle tracking or numerical gradient control constraints, the Split engine was used, while the Bluebird engine was used for experiments involving zone outflux or analytic gradient control constraints. The two flow engines use identical system representation and solution algorithms; selection between the two was based solely on the availability of desired output. Both models use an over-specified high-order representation

Table 1
Plume containment constraints

Acronym	Type	Number of constraints (control, complex)	Distribution or computation technique
PTLB	**P**article **T**racking	**L**ow (10, 38)	Uniformly distributed throughout the plume **B**ody
PTMB		**M**edium (25, 95)	
PTHB		**H**igh (50, 190)	
PTLP		**L**ow (10, 38)	Uniformly distributed along the plume **P**erimeter
PTMP		**M**edium (25, 95)	
PTHP		**H**igh (50, 190)	
HGLN	**H**ydraulic **G**radient control	**L**ow (10, 38)	Gradients computed **N**umerically
HGMN	(control points uniformly	**M**edium (25, 95)	
HGHN	distributed around plume perimeter)	**H**igh (50, 190)	
HGLA		**L**ow (10, 38)	Gradients computed **A**nalytically
HGMA		**M**edium (25, 95)	
HGHA		**H**igh (50, 190)	
ZONE	**ZONE** Outflux	n/a	n/a

of line elements [29,30]. For this work, the element segments were configured with an order of 12 and 18 control points, yielding maximum simulated head errors of 0.2 and 2 mm for the Control and Complex problems, respectively.

The automated optimizations were performed using Ostrich [36], a model-independent, multi-algorithmic optimization tool. For this purpose, a pump-and-treat optimization module was developed and is available with version 1.4 of the public domain software.

2.6. Numerical experiments

To assess different aspects of AEM-based PATO, a series of numerical experiments was designed using the control and complex AEM models in conjunction with various configurations of the PATO problem. The goals of the experiments were: (i) demonstrate that PATO problems can effectively make use of AEM flow models, (ii) compare the effectiveness of various optimization algorithms, as applied to PATO, (iii) compare the *reliability* (defined below, in Section 2.6.1) of various plume containment constraint formulations, and (iv) examine algorithm parallelization. To facilitate analysis of the experimental results, quantitative expressions for constraint reliability (R), algorithm effectiveness (F), and speedup (S) were developed, as discussed below.

2.6.1. Constraint reliability

Constraint reliability was computed following the completion of a given optimization run and measures the adequacy of the plume containment constraint formulation used in the optimization. Using the current optimal well-field design, the flow model was run with a very large number of verification particles distributed uniformly throughout the plume and forward-tracked over the remediation time frame. The reliability of the constraints was computed as the percentage of verification particles successfully contained by the given optimal design:

$$R = (N_{\text{con}}/N_{\text{total}}) \times 100\%, \tag{5}$$

where, N_{total} is the total number of verification particles and N_{con} is the number of these particles successfully contained.

2.6.2. Algorithm effectiveness

The measure of algorithm effectiveness was defined as the optimal objective function value discovered by a given algorithm, computed using the applicable additive or multiplicative penalty method (APM was used for particle tracking constraints, while MPM was used for gradient control and zone outflux constraints). As such, lower values indicate more effective algorithms.

2.6.3. Algorithm speedup

Algorithm speedup (S) was defined as the ratio of the wall-time required for optimization using a single processor (T_1) to the wall-time required when using p processors (T_p), where *wall-time* was the time required, from a human perspective, for an optimization to run to completion:

$$S = (T_1/T_p). \tag{6}$$

Highly parallel algorithms may be expected to produce linear speedup ($S \to p$).

3. Results

More than 80 optimizations were performed. In general, a typical flow simulation for both the control and complex problems consumed less than 1 s of computation time. However, the use of particle tracking constraints significantly increased the model computation time. On average, the tracking of particles increased the AEM model run-time by approximately 1 s for every 25 particles. However, the increase was affected by the complexity of the local flow field and varied considerably between model configurations. Because of the additional model computation, optimization using "high" numbers of particle tracking constraints required considerably more computer time than other constraint formulations. The Split modeling engine supports parallel particle tracking, and while not utilized in the present study, this feature may provide substantial benefit to more complex problems involving hundreds or thousands of particles.

3.1. Algorithm comparisons

To assess algorithm effectiveness, the CG, GA, PSO, RND, and SA algorithms were applied to the control AEM model. For each algorithm, separate optimizations were performed using the C_{TOTQ}, C_{OPER}, and $C_{\text{OPER+}}$ cost functions and the HGMN (medium number of numerically computed gradient control constraints) and PTMB (medium number of particle tracking constraints, placed throughout the plume body) plume containment constraint formulations. Additionally, two implementations of the conjugate gradient (CG) algorithm were examined, differentiated by the assignment of the initial well locations and rates. One CG run (CG/AR) used randomly assigned initial parameter values, while the other run (CG/EJ) used initial values assigned based on engineering judgment and a limited number of manual trial-and-error runs.

In Fig. 5a–c, the effectiveness of each algorithm is presented for the various cost and constraint formulations. With the notable exception of the CG/EJ runs, the relative effectiveness of each algorithm was consis-

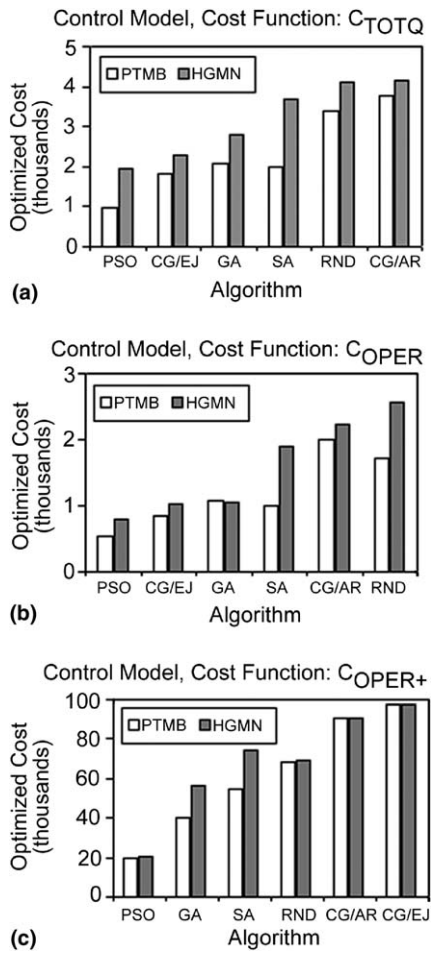

Fig. 5. Comparison of algorithms applied to the control model optimization using PTMB and HGMN constraints and: (a) C_{TOTQ} cost, (b) C_{OPER} cost, and (c) C_{OPER+} cost function.

tent across the different formulations. The particle swarm (PSO) algorithm was the most effective; the genetic algorithm (GA) also performed reasonably well across all formulations. For the C_{TOTQ} and C_{OPER} costs, the CG/EJ formulation was highly effective, but, as shown in Fig. 5c, the same algorithm was also the least effective at optimizing the C_{OPER+} cost formulation. It appears that the gradient-based CG algorithm was frustrated by the inclusion of capital costs in the C_{OPER+} cost function. These costs result in an optimization that is driven by the number of active wells in the system, a non-differentiable discrete parameter.

Based on their effectiveness at solving the control problem, the PSO and GA algorithms were applied to the complex problem. As shown in Fig. 6, the PSO algorithm was much more effective at solving the complex problem, which was optimized using the C_{OPER+} cost function. In the complex problem, the maximum number of wells was 19, a value much larger than required for optimal satisfaction of the plume containment constraints (only one well is required). An examination of the optimization behavior of the two algorithms

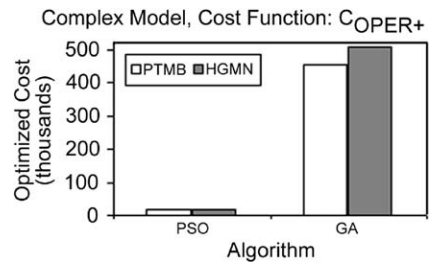

Fig. 6. Comparison of algorithms applied to the complex model optimization using PTMB and HGMN constraints and C_{OPER+} cost function.

revealed that the PSO algorithm was much more effective at eliminating the abundance of unnecessary wells. Given that the number of active wells drives the optimization of the C_{OPER+} cost function, the large difference between the effectiveness of the two algorithms is therefore attributed to the superior ability of the PSO algorithm to eliminate wells.

3.2. Constraint comparisons

The experiments described above focused on the ability of various algorithms to minimize one of three cost functions utilizing a baseline set of plume containment constraint formulations. Another set of experiments examined the ability of alternative constraint formulations to represent the goal of plume containment in a reliable fashion. Optimizations were performed on both the control and complex problems using the C_{OPER+} cost function, the PSO algorithm, and each of the 13 constraint formulations listed in Table 1. Following each optimization, constraint reliability (as defined in Section 2.6.1) was computed based on the tracking of 1000 particles initially located in and around the plume. Fig. 7a and b summarizes the resulting reliability measures for the various particle tracking and hydraulic gradient constraint formulations, respectively.

Among the particle tracking constraints, plume perimeter distributions were clearly more reliable than plume body distributions, but no configuration achieved 100% reliability (the "Control, PTHP" configuration had a 99.5% reliability). For plume-perimeter particle distributions, the "medium" setting yielded approximately 97% reliability, while "high" numbers of particles were required to achieve similar reliability for plume-body distributions in the complex model (all of the plume-body distributions in the control model have less than 90% reliability). Conversely, "low" numbers of gradient control constraints resulted in a range of 90–100% reliability while the "medium" and "high" gradient control settings were all between 97% and 100% reliable. Furthermore, the results show no difference in reliability between the analytic and discretized gradient constraint formulations.

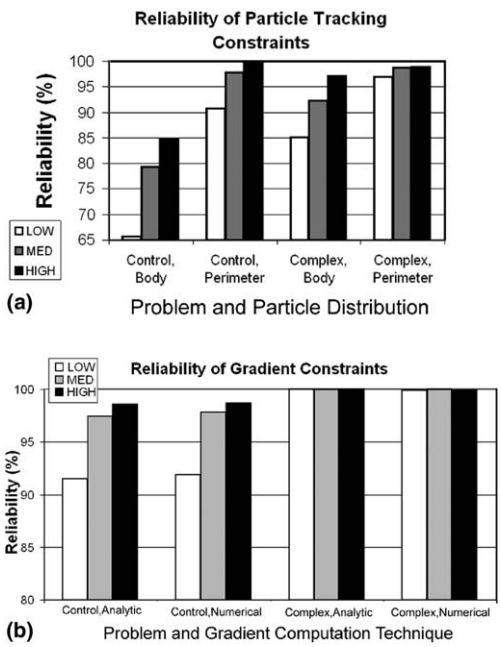

Fig. 7. Comparison of (a) particle tracking and (b) gradient control constraint reliabilities.

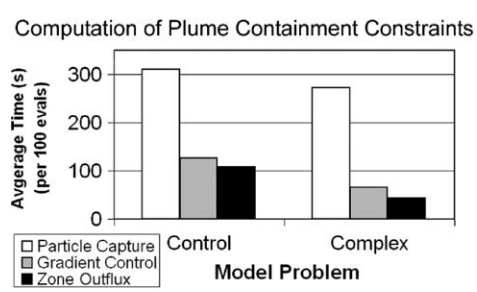

Fig. 8. Average model computation times for various plume containment constraints.

The zone outflux constraint (not shown) was found to be 100% reliable for both the control and complex problems. Average computation times required for the various constraint types (particle tracking, gradient control and zone outflux) are shown in Fig. 8, and illustrate a significant computational advantage when using zone outflux or gradient control constraints. The reliability of the zone outflux constraint combined with its computational efficiency makes this constraint an attractive and powerful technique for constraining AEM-based PATO problems.

3.3. Comparison with published solutions

Table 2 provides a summary of the solutions to the control problem, using the zone outflux constraint and the most reliable particle tracking and gradient control constraint formulations. Also included in the table are the solution found by Fowler et al. [19], using a finite difference flow model, and a solution found using the type curve procedures described by Javandel and Tsang [31] and Grubb [24]. The type curve analysis is based on the simplifying assumption of uniform flow through the plume. The most reliable solution is that of the zone outflux constraint, whose optimal design is roughly $5\frac{1}{2}\%$ less costly than the published FD-based solution. One possible explanation for the lower reliability of the published FD-based solution is that it used relatively few gradient control points (examination of Fig. 7b indicates that low numbers of gradient control points are associated with reduced reliability). A possible explanation for the increased cost of the FD-based solution is that the gradient control constraint was more restrictive than the one presented in the current study. In the FD-based study, the gradient control formulation constrained flow to be precisely directed toward an installed well, whereas the present study required flow to be towards the interior of the plume.

Table 3 provides a summary of solutions to the complex problem, using (i) the zone outflux constraint, (ii) the most reliable gradient control constraint formulation (HGHA), (iii) a particle tracking constraint formulation (PTMP) analogous to the constraint formulation used in the reference study [39], (iv) the optimal solution reported in [39], which used a finite difference flow model and a particle tracking containment constraint formulation consisting of 110 particles placed along the plume perimeter, and (v) the optimal two-well solution using type curve analysis (as with the control problem, this requires the assumption of uniform flow through the plume body).

Examination of the FD-based Complex solution and the analogous AEM-based solution (PTMP) reveals a close correspondence between the pumping rates of the

Table 2
Solutions to the control problem

Cost function	Constraint formulation (or literature reference)	Optimized cost [$]	Number of wells	Well-field configuration (Q [m^3/day], X [m], Y [m])	Reliability [%]
OPER+	PTHP	20,273	1	(178.1, 264, 726)	99.5
	HGHA	20,570	1	(219.3, 249, 711)	98.6
	ZONE	20,745	1	(241.6, 242, 705)	100.0
	Fowler et al. [19]	21,957	1	(371.5, 250, 650)	96.8
	Type Curve Analysis	20,107	1	(152.5, 289, 741)	86.6

Table 3
Solutions to the complex problem

Cost function	Constraint formulation (or literature reference)	Optimized cost [$] (total rate)	Number of wells	Well-field configuration (Q [m^3/day], X [m], Y [m])	Reliability [%]
TOTQ	HGHA	3385 (62.8)	3	(8.3, 1462, 1664), (28.2, 1432, 1759) (26.3, 1540, 1811)	100.0
	ZONE	3489 (64.7)	3	(14.6, 1441, 1670), (11.9, 1435, 1745), (38.2, 1520, 1819)	100.0
	PTMP	1694 (31.4)	2	(24.4, 1449, 1679), (7.0, 1569, 1702)	97.6
	Mulligan and Ahlfeld [39]	1565 (29.0)	2	(8.0, 1347, 1680), (21.0, 1573, 1720)	88.7
	Type curve analysis	1511 (28.0)	2	(14.0, 1504, 1676), (14.0, 1537, 1706)	89.9

two required wells. However, it is noted that the locations of the low-rate and high-rate wells for the two solutions are inverted. The low-rate (7 m^3/day) well in the AEM solution is associated with the eastern lobe of the plume, whereas the low-rate (8 m^3/day) well in the FD-based solution is associated with the western lobe of the plume. Furthermore, the reliability of the FD-based solution (computed by configuring the AEM-based complex model with the reported FD-based solution) is fairly poor (less than 90%). The discrepancies between the FD- and AEM-based solutions, when using comparable plume containment constraints, suggests differences in the aquifer properties of the two models (recall that the specified head boundary conditions of the AEM-based complex model were estimated using trial-and-error).

The type curve solutions for both problems are low cost, but also have low reliability. Two aspects of the type curve analysis account for its low reliability: (i) the simplifying assumption of uniform flow results in a less-than-optimal rate for plume containment, and (ii) the analysis applies to the steady-state capture zone (i.e., at infinite time), rather than the actual remediation time frames used in this study (5 and 30 years for the control and complex problems, respectively).

For the complex model, examination of the gradient control (HGHA) and zone outflux constraint (ZONE) solutions reveals that, relative to the particle tracking constraint formulation, one additional well is required and the total pumping rate is significantly higher (roughly double). This increase in cost and number of wells is likely due to the dual-lobed shape of the plume. As shown by the capture zones in Fig. 9, the gradient control and zone outflux constraints prevent the "clean" region between the two lobes from being contaminated during remediation, while the particle tracking solution drags contaminated groundwater through this region and then back into the confines of the plume boundary. Maintaining an inward flux along the plume boundary where the two lobes converge (thus preventing contamination of the clean region) is a more restrictive constraint, and necessitates an increased number of wells and increased total pumping rate.

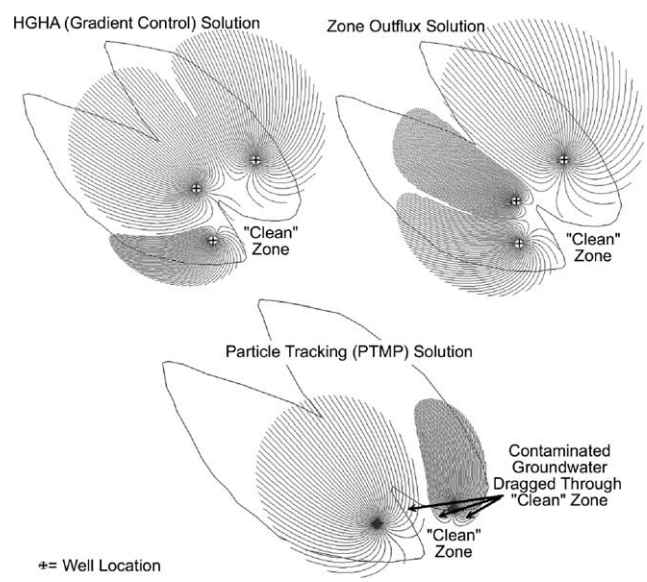

Fig. 9. Thirty-year capture zones for solutions to the complex problem.

3.4. Comparison of continuous and discretized well coordinates

Tables 4 and 5 summarize optimal solutions for various constraint formulations using continuous and discretized well coordinates. For discrete well-coordinates, candidate well locations were set at 10 m intervals in the x- and y-directions. Examination of Tables 4 and 5 reveals that discretization produces results nearly equivalent to those in which well coordinates are varied as continuous functions, both in terms of cost (the maximum cost difference is 4%) and well location. Since both methods provide equivalent results, the choice between discretized and continuous well coordinates (if continuous well coordinates can be handled by the given flow-modeling engine, and if the PATO problem is similar to those presented in this study) can be based on computational considerations. For example, conjugate-gradient algorithms require continuous variables, to facilitate computation of partial derivatives. Conversely, a binary-encoded GA is more amenable to the use of discretized (integer-based) values.

Table 4
Discretized versus continuous well-coordinate comparisons (control problem)

Cost function	Constraint formulation	Discretized?	Cost [$]	Number of wells	Well-field configuration (Q [m^3/day], X [m], Y [m])	Reliability [%]
C_{OPER+}	PTHP	No	20,273	1	(178.1, 264, 726)	99.5
		Yes	20,333	1	(186.0, 280, 730)	98.5
	HGHA	No	20,570	1	(219.4, 249, 711)	98.6
		Yes	20,588	1	(221.7, 250, 710)	98.8
	ZONE	No	20,745	1	(241.6, 242, 705)	100.0
		Yes	20,850	1	(253.9, 230, 710)	100.0

Table 5
Discretized versus continuous well coordinate comparisons (complex problem)

Cost function	Constraint formulation	Discretized?	Cost [$] (total rate)	Number of wells	Well-field configuration (Q [m^3/day], X [m], Y [m])	Reliability [%]
C_{TOTQ}	PTHP	No	2015 (37.4)	2	(33.8, 1507, 1728), (3.6, 1449, 1646)	99.1
		Yes	2018 (37.3)	2	(33.9, 1510, 1720), (3.4, 1440, 1640)	99.6
	HGHA	No	3385 (62.8)	3	(8.3, 1462, 1664), (28.2, 1432, 1759), (26.3, 1540, 1811)	100.0
		Yes	3526 (65.3)	3	(7.5, 1460, 1660), (30.8, 1430, 1760), (27.0, 1540, 1820)	99.9
	ZONE	No	3489 (64.7)	3	(14.6, 1441, 1670), (11.9, 1435, 1745), (38.2, 1520, 1819)	100.0
		Yes	3580 (66.3)	3	(17.8, 1420, 1660), (17.4, 1440, 1760), (31.1, 1540, 1820)	100.0

3.5. Algorithm parallelization

A final set of optimizations examined improvements in computation speed achievable via parallelization of the optimization algorithm. The PSO algorithm was parallelized for distributed, cluster-based processors using the message passing interface (MPI) (e.g., [23]). In the parallel PSO algorithm, a supervisor processor begins iteration by broadcasting a revised population set (particle swarm) to a set of subordinate processors. Each subordinate processor computes the objective function of one or more of the revised particles and sends the results back to the supervisor. The supervisor completes the iteration by revising algorithmic data (i.e., local and global memory) associated with each particle, and the entire process begins again with the next iteration.

As shown in Fig. 10, nearly linear speedup of the PSO algorithm was achieved using up to thirty parallel processors. The complex model appeared to benefit slightly more from the parallel processing, which is attributed to the use of a larger swarm size (i.e., number of candidate solutions) in solving the complex problem. Evaluation of this larger swarm by the subordinate processors necessarily consumed more computation time and therefore diminished the relative amount of time spent performing MPI communication.

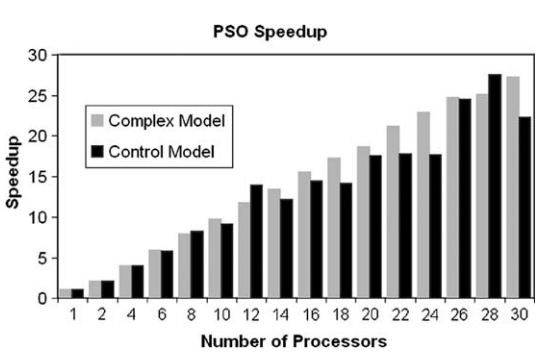

Fig. 10. Parallel speedup of the PSO algorithm.

4. Discussion and conclusion

In this paper, the use of AEM flow models in the automated optimization of pump-and-treat systems has been demonstrated for the first time. Two hydraulic control problems were adapted from published literature and numerous optimizations were successfully performed using a variety of algorithms, cost functions, penalty techniques, and plume containment constraints. Although the chosen problems are amenable to solution by elementary type-curve analysis, they are also rather complex from the viewpoint of a general-purpose optimizer. When expressed in a constrained optimization framework, the problems are highly nonlinear and contain a multitude of design variables, and are therefore a useful means to benchmark AEM-based PATO against other simulation-optimization approaches.

Of the algorithms investigated, the particle swarm (PSO) algorithm performed most effectively and exhibited linear parallel speedup. Of particular note is the ability of the PSO to reduce the number of extraction wells when the maximum number is highly over-estimated. This paper also marks the first application of

the PSO algorithm to PATO problems, and its superior performance suggests that it should be evaluated for more complex environmental optimization problems.

The use of analytic element (AEM) flow models for pump-and-treat optimization provides several interesting capabilities that are either not possible or not yet utilized within standard finite difference (FD) and finite element (FE) flow models. For example, the optimizations in this paper varied well coordinates as continuous functions, an approach that is not possible with grid-based FD/FE models. Furthermore, while AEM models can readily incorporate commonly used particle tracking and gradient control plume containment constraints, the method also supports the use of a zone outflux constraint (not presently implemented in FD/FE models) and the analytic computation of gradient control constraints (not possible in FD/FE models). The zone outflux constraint was shown to be extremely reliable and, like the gradient control constraint, required significantly less computation time than particle tracking techniques. Conversely, in the context of pump-and-treat optimization, the results fail to demonstrate that other features of AEM (such as well coordinates varied as continuous functions and analytically computed head gradients) offer a particular advantage.

Limitations associated with the AEM modeling software used in this study include the assumptions of steady-state two-dimensional flow with zoned heterogeneity (i.e., piecewise constant regions of aquifer heterogeneity). Other public domain AEM software (such as Tim^{ML} [4]) can overcome some of these limitations, suggesting that the AEM-based PATO concepts discussed in this paper may be applicable to more complex aquifer systems. Alternatively, the particular advantages associated with the use of AEM-based PATO could be utilized in a preliminary optimization exercise, to be followed by optimization of a more complex (e.g., three-dimensional, time-varying and/or heterogeneous) FD/FE model. Such a preliminary model could be used to identify candidate well locations and estimate the required number of wells and total pumping rate.

Acknowledgements

This research was supported by Grant # R82-7961 from the US Environmental Protection Agency's (EPA) Science to Achieve Results (STAR) program. This paper has not been subjected to any EPA review and therefore does not necessarily reflect the views of the Agency, and no official endorsement should be inferred. L. Shawn Matott and James R. Craig were supported by the National Science Foundation Integrated Graduate Education and Research Training (IGERT) program in Geographic Information Science.

Appendix A

This appendix provides more information on the pump-and-treat cost and constraint formulations, as well as a list of acronyms. Throughout the appendix, numbers in parentheses following variable definitions correspond to the values used in this study and brackets following these numbers define the associated units. In cases where two numbers are provided, the first number applies to the control problem, and the second to the complex problem.

A.1. List of Acronyms

AEM analytic element method
APM additive penalty method
CG conjugate gradient algorithm
CG/AR conjugate gradient algorithm, with initial parameters values randomly assigned
CG/EJ conjugate gradient algorithm, with initial parameter values assigned based on engineering judgment and a limited number of manual trial-and-error runs
C_{OPER} cost function that considers operational costs only (such as energy, treatment, disposal and labor costs)
$C_{\text{OPER}+}$ cost function that considers operational and capital (well installation and pump) costs
C_{TOTQ} cost function that uses total pumping rate as a surrogate for cost
EPM exponential penalty method
FD finite difference
FD/FE finite difference and finite element
FE finite element
GA genetic algorithm
HGHA hydraulic gradient plume containment constraint formulation, using a high number of control points, distributed along the plume perimeter and computed analytically
HGHN hydraulic gradient plume containment constraint formulation, using a high number of control points, distributed along the plume perimeter and computed numerically
HGLA hydraulic gradient plume containment constraint formulation, using a low number of control points, distributed along the plume perimeter and computed analytically
HGLN hydraulic gradient plume containment constraint formulation, using a low number of control points, distributed along the plume perimeter and computed numerically
HGMA hydraulic gradient plume containment constraint formulation, using a medium number of control points, distributed along the plume perimeter and computed analytically

HGMN hydraulic gradient plume containment constraint formulation, using a medium number of control points, distributed along the plume perimeter and computed numerically
MPI message passing interface
MPM multiplicative penalty method
PAT pump-and-treat
PATO pump-and-treat optimization
PSO particle swarm optimization algorithm
PTHB particle tracking plume containment constraint formulation, using a high number of particles distributed throughout the plume body
PTHP particle tracking plume containment constraint formulation, using a high number of particles distributed along the plume perimeter
PTLB particle tracking plume containment constraint formulation, using a low number of particles distributed throughout the plume body
PTLP particle tracking plume containment constraint formulation, using a low number of particles distributed along the plume perimeter
PTMB particle tracking plume containment constraint formulation, using a medium number of particles distributed throughout the plume body
PTMP particle tracking plume containment constraint formulation, using a medium number of particles distributed along the plume perimeter
P_{TOTAL} a penalty function that accounts for the cost of various constraint violations (drawdown, capacity and plume containment)
RND random search algorithm
SA simulated annealing algorithm
ZONE zone outflux plume containment constraint formulation

A.2. System cost

Three cost functions were considered in this study: (i) total pumping rate (C_{TOTQ}), (ii) operational costs only (C_{OPER}), and (iii) both operational and capital costs ($C_{\mathrm{OPER+}}$). The equations for these costs are derived from the formulations provided in [37,43].

A.2.1. Total pumping rate

Minimization of the total pumping rate is often used as a surrogate for a rigorous cost minimization objective. For problems in which the system will be operated in the long term, this approximation is often adequate because operational costs (which are largely a function of pumping rate) will dominate capital costs.

$$C_{\mathrm{TOTQ}} = \alpha_{\mathrm{ext}} \sum_{i=1}^{n_{\mathrm{ext}}} |Q_{i,\mathrm{ext}}| + \alpha_{\mathrm{inj}} \sum_{i=1}^{n_{\mathrm{inj}}} |Q_{i,\mathrm{inj}}|, \tag{A.1}$$

where, C_{TOTQ} is the cost as a function of total pumping rate; α_{ext}, the cost conversion factors for the total extraction rate (10.59, 53.98) [\$ day/m^3]; α_{inj}, the cost conversion factors for the total injection rate (0.26, 1.59) [\$ day/m^3], n_{ext}, the number of extraction wells; n_{inj}, the number of injection wells; $Q_{i,\mathrm{ext}}$, the rate of extraction of the ith extraction well; and $Q_{i,\mathrm{inj}}$, the rate of injection of the ith injection well.

A.2.2. Operational costs

Pump-and-treat operational costs reflect the ongoing cost of operating the pump-and-treat system. Like the total pumping rate cost, operational costs do not account for capital costs, making this an appropriate cost function for long-term operations-dominated systems. Unlike the C_{TOTQ} formulation, the operational cost is a nonlinear function of pumping rates (due to the additional consideration of lift), and may also be a function of the total number of wells (if maintenance, analytic and/or labor costs are included):

$$C_{\mathrm{OPER}} = [C_{\mathrm{L}} + C_{\mathrm{E}} + C_{\mathrm{A}} + C_{\mathrm{D}} + C_{\mathrm{M}}]T, \tag{A.2}$$

where, C_{OPER} is the operational cost, C_{L}, C_{E}, C_{A}, C_{D}, and C_{M} are the annual labor, energy, analytic, disposal and maintenance costs, respectively; and T is the remediation time frame (5 and 30 years for the control and complex problems, respectively). To be consistent with the community problems [37], only energy costs are considered in this paper (i.e., $C_{\mathrm{L}} = C_{\mathrm{A}} = C_{\mathrm{D}} = C_{\mathrm{M}} = 0$), and the formulation of C_{E} is

$$C_{\mathrm{E}} = \beta_0 \sum_{i=1}^{n_{\mathrm{ext}}} |Q_{i,\mathrm{ext}}|(h_i - Z_{\mathrm{gs},i}) + \beta_1 \sum_{i=1}^{n_{\mathrm{inj}}} |Q_{i,\mathrm{inj}}|, \tag{A.2a}$$

where, β_0 is the annualized energy cost conversion factor for extraction (0.11) [\$ day/year m^4]; β_1, the annualized energy cost conversion factor for injection (0.05) [\$ day/year m^3]; h_i, the head at extraction well i; and $Z_{\mathrm{gs},i}$, the ground surface elevation at the ith extraction well.

A.2.3. Operational and capital costs

Operational and capital costs account for fixed (one-time) installation costs as well as ongoing operational costs. As such, the formulation of $C_{\mathrm{OPER+}}$ adds capital cost terms to the operational cost defined previously:

$$C_{\mathrm{OPER+}} = C_{\mathrm{OPER}} + \beta_2(\mathrm{NW}) + \beta_3(n_{\mathrm{ext}}), \tag{A.3}$$

where, β_2 is the per-well drilling cost (15,258, 13,511) [\$/well]; NW, the number of active wells; and β_3, the pump cost, assuming a design pumping rate of 553 m^3/day (4028, 3832) [\$/pump].

A.3. Constraints

The total constraint violation (P_{TOTAL}) is computed as the sum of the violations of the three types of constraints considered in this study, namely capacity

(P_{CPCY}), drawdown (P_{DRAW}) and plume containment (P_{PLUME}):

$$P_{\mathrm{TOTAL}} = P_{\mathrm{CPCY}} + P_{\mathrm{DRAW}} + P_{\mathrm{PLUME}}. \tag{A.4}$$

A.3.1. Capacity constraints

Capacity constraints reflect a limit on the total pumping rate that can be handled by an existing treatment system:

$$P_{\mathrm{CPCY}} = \begin{cases} \beta_{\mathrm{CPCY}}(Q_{\mathrm{tot}} - Q_{\max}), & Q_{\mathrm{tot}} > Q_{\max}, \\ 0, & Q_{\mathrm{tot}} \leqslant Q_{\max}, \end{cases} \tag{A.5}$$

where, β_{CPCY} is the cost-conversion factor (dollars per L^3/T of excess) (100,000) [\$ day/m^3]; Q_{tot}, the net extraction rate (total extraction minus total injection); and $Q_{\max}$, maximum allowable extraction rate (2765, 3000) [m^3/day].

A.3.2. Drawdown constraints

Drawdown constraints prevent the pump-and-treat system from dewatering the aquifer, and are assessed by examining the head near each well. If the head near a given well is reduced (via pumping) below some minimum allowable head level, a penalty is assessed:

$$P_{\mathrm{DRAW}} = \beta_{\mathrm{DRAW}} \sum_{i=1}^{\mathrm{NW}} \max(h_{\min} - h_i, 0), \tag{A.6}$$

where β_{DRAW} is the cost-conversion factor (dollars per length of excess) (100,000) [\$/m]; and $h_{\min}$, the minimum allowable head (10, 1.5) [m].

A.3.3. Plume containment constraints

Plume containment constraints use flow-model output to gauge the effectiveness of a given well-field design at controlling plume migration. Each experiment in this study considered one of three plume containment constraints: hydraulic gradient control, particle tracking, or zone outflux:

$$P_{\mathrm{PLUME}} = \beta_{\mathrm{HGRAD}} \sum_{i=1}^{m} P_{\mathrm{HGRAD},i} + \beta_{\mathrm{PTRK}} \sum_{i=1}^{n} P_{\mathrm{PTRK},i} + \beta_{\mathrm{ZONE}} P_{\mathrm{ZONE}}, \tag{A.7}$$

where, β_{HGRAD}, β_{PTRK}, and β_{ZONE} are the cost-conversion factors (100,000) [\$/m, \$/m^2, \$ day/m^3]; m, the number of gradient control constraints; n, the number of particle tracking constraints; $P_{\mathrm{HGRAD},i}$, the penalty of the ith gradient control constraint; $P_{\mathrm{PTRK},i}$, the penalty of the ith particle tracking constraint; P_{ZONE}, the zone outflux penalty.

A.3.3.1. Gradient control constraints. Gradient control constraints examine the head gradient at a given control location, and assess a penalty if the gradient is oriented away from the plume interior. For numerical gradient computations, a pair of control points, one outside and one inside the plume is considered, and the constraint requires that the head outside the plume be greater than the head inside the plume, implying flow direction is towards the plume interior.

$$\begin{aligned} P_{\mathrm{HGRAD},i} &= \max(0, h_{\mathrm{in},i} - h_{\mathrm{out},i}) && \text{if numerical,} \\ P_{\mathrm{HGRAD},i} &= \max(0, dh_i) && \text{if analytic,} \end{aligned} \tag{A.7a}$$

where, $h_{\mathrm{in},i}$ and $h_{\mathrm{out},i}$ are the inside and outside head of the ith control pair and dh_i is the analytically computed gradient at the ith control location (a positive value of dh_i indicates flow away from the plume, and results in a constraint violation).

A.3.3.2. Particle tracking constraints. Particle tracking constraints advect a given particle from an initial location within the plume body or along the plume perimeter. If the particle is not within the plume boundary or captured by a well at the conclusion of the remediation time frame, then a penalty is assessed:

$$P_{\mathrm{PCAP},i} = \begin{cases} d_i^2, & \text{particle}_i \text{ outside plume,} \\ 0, & \text{particle}_i \text{ inside plume,} \end{cases} \tag{A.7b}$$

where d_i is the distance from the particle to the plume boundary.

A.3.3.3. Zone outflux constraints. The zone outflux constraint examines the total outflux (F^{+}_{ZFP}, as defined in Eq. (3)) across the perimeter of the plume and assigns a penalty if this outflux is nonzero:

$$P_{\mathrm{ZONE}} = \max(F^{+}_{\mathrm{ZFP}}, 0). \tag{A.7c}$$

A.3.4. Objective functions

The functional forms of the additive (F_{APM}), multiplicative (F_{MPM}), and exponential (F_{EPM}) penalty methods, as applied in this study, are given below:

$$F_{\mathrm{APM}}(\mathrm{NW}, \overline{Q}, \overline{X}, \overline{Y}) = \mathrm{Cost} + P_{\mathrm{TOTAL}}, \tag{A.8}$$

$$F_{\mathrm{MPM}}(\mathrm{NW}, \overline{Q}, \overline{X}, \overline{Y}) = \max(\mathrm{Cost}, P_{\mathrm{TOTAL}})(1 + P_{\mathrm{TOTAL}}), \tag{A.9}$$

$$F_{\mathrm{EPM}}(\mathrm{NW}, \overline{Q}, \overline{X}, \overline{Y}) = \max(\mathrm{Cost}, P_{\mathrm{TOTAL}}) \exp(P_{\mathrm{TOTAL}}), \tag{A.10}$$

$\overline{Q} = [Q_1, Q_2, \ldots, Q_{\mathrm{NW}}]^{\mathrm{T}}$, vector of pumping rates; $\overline{X} = [x_1, x_2, \ldots, x_{\mathrm{NW}}]^{\mathrm{T}}$, vector of x-coordinates; $\overline{Y} = [y_1, y_2, \ldots, y_{\mathrm{NW}}]^{\mathrm{T}}$, vector of y-coordinates; Cost: one of C_{TOTQ}, C_{OPER} or $C_{\mathrm{OPER+}}$.

A.4. Parameter bounds

Tables A1 and A2 provide parameter bounds used in solving the PATO problems.

Table A1
Control problem parameter bounds

Parameter	Min value	Max value	Units
Q_i	−553	553	m^3/day
X_i	189	328	m
Y_i	495	769	m
NW	1	5	n/a

Table A2
Complex problem parameter bounds

Parameter	Min value	Max value	Units
Q_i	0	550	m^3/day
X_i	1112	1624	m
Y_i	1619	2031	m
NW	1	19	n/a

References

[1] Ahlfeld DP, Page RH, Pinder GF. Optimal ground-water remediation methods applied to a superfund site: from formulation to implementation. Ground Water 1995;33(1):58–70.

[2] Aly AH, Peralta RC. Comparison of a genetic algorithm and mathematical programming to the design of groundwater cleanup systems. Water Resour Res 1999;35(8):2415–25.

[3] Atwood DF, Gorelick SM. Hydraulic gradient control for groundwater contaminant removal. J Hydrology 1985;76:85–106.

[4] Bakker M. Tim^{ML}: A multiaquifer analytic element model, version 2.1. University of Georgia, Athens, GA, 2004. Available from: www.engr.uga.edu/~mbakker/TimML.html.

[5] Beielstein T, Parsopoulos KE, Vrahatis MN. Tuning PSO parameters through sensitivity analysis. Technical Report, Reihe Computational Intelligence CI 124/02. Collaborative Research Center (Sonderforschungsbereich), Department of Computer Science, University of Dortmund, 2002.

[6] Chan Hilton AB, Culver TB. Groundwater remediation design under uncertainty using genetic algorithms. J Water Resour Plann Manage 2005;131(1):25–34.

[7] Chan Hilton AB, Culver TB. Constraint handling for genetic algorithms in optimal remediation design. J Water Resour Plann Manage 2000;126(3):128–37.

[8] Chang LC, Shoemaker CA, Liu PL. Optimal time-varying pumping rates for groundwater remediation: application of a constrained optimal control algorithm. Water Resour Res 1992; 28(12):3157–73.

[9] Coelho JP, Oliveira PM, Cunha JB. Greenhouse air temperature control using the particle swarm optimisation algorithm. In: Proceedings of 15th triennial world congress of the international federation of automatic control (IFAC 2002), Barcelona, Spain, 2002.

[10] Craig JR, Matott LS. Visual bluebird user's manual: version 2.0. Department of Civil, Structural, and Environmental Engineering, University at Buffalo, Buffalo, NY, 2005. Available from: www.groundwater.buffalo.edu.

[11] Craig JR. Reactive Contaminant transport modeling using analytic element flow solutions. PhD dissertation, Department of Civil, Structural, and Environmental Engineering, University at Buffalo, Buffalo, NY, 2005.

[12] Craig JR, Rabideau AJ. Discretization of analytic element flow solutions for transport modeling. Proceedings of the XV international conference on computational methods in water resources. Chapel Hill (NC): Elsevier; 2004. p. 381–91.

[13] Culver TB, Shenk GW. Dynamic optimal ground water remediation by granular activated carbon. J Water Resour Plann Manage 1998;124(1):59–64.

[14] Culver TB, Shoemaker CA. Dynamic optimal control for groundwater remediation with flexible management periods. Water Resour Res 1992;28(3):629–41.

[15] Culver TB, Shoemaker CA. Optimal control for groundwater remediation by differential dynamic programming with quasi-Newton approximations. Water Resour Res 1993;29(4):823–31.

[16] Culver TB, Shoemaker CA. Dynamic optimal ground-water reclamation with treatment capital costs. J Water Resour Plann Manage 1997;123(1):23–9.

[17] Dougherty DE, Marryott RA, Stollar RL. Optimal groundwater management, 2, Application of simulated annealing to a field-scale contamination site. Water Resour Res 1991;29(4): 847–60.

[18] Environmental Security Technology Certification Program (ESTCP). Cost and performance report: Application of flow and transport optimization codes to groundwater pump-and-treat systems. US Department of Defense, ESTCP Report CU-0010, 2004. Available from: www.estcp.org/documents/techdocs/CU-0010.pdf.

[19] Fowler KR, Kelley CT, Kees CE, Miller CT. A hydraulic capture application for optimal remediation design. Proceedings of the XV international conference on computational methods in water resources. Chapel Hill (NC): Elsevier; 2004. p. 1149–57.

[20] Fowler KR, Kelley CT, Miller CT, Kees CE, Darwin RW, Reese JP, et al. Solution of a well-field design problem with implicit filtering. Optim Eng 2004;5:207–34.

[21] Gies D, Rahmat-Samii Y. Particle swarm optimization for reconfigurable phase-differentiated array design. Microw Opt Technol Lett 2003;38(3):168–75.

[22] Goldberg DE. Genetic algorithms in search, optimization and machine learning. Reading (MA): Addison-Wesley; 1989.

[23] Gropp W, Lusk W, Skjellum A. Using MPI: Portable parallel programming with the message passing interface. 2nd ed. Cambridge (MA): MIT Press; 1999.

[24] Grubb S. Analytical model for estimation of steady-state capture zones of pumping in confined and unconfined aquifers. Ground Water 1993;31(1):27–32.

[25] Guan J, Aral MM. Optimal design of groundwater remediation systems using fuzzy set theory. Water Resour Res 2004;33(5): 1001–12.

[26] Haitjema H. Analytic element modeling of ground water flow. San Diego (CA): Academic Press; 1995.

[27] Huang C, Mayer AS. Pump-and-treat optimization using well location and pumping rates as decision variables. Water Resour Res 1997;33(5):1001–12.

[28] Jankovic I. SPLIT: Win32 computer program for analytic-based modeling of single-layer groundwater flow in heterogeneous aquifers with particle tracking, capture zone delineation, and parameter estimation. Department of Civil, Structural, and Environmental Engineering, University at Buffalo, Buffalo, NY, 2005. Available from: www.groundwater.buffalo.edu.

[29] Jankovic I. High-order analytic elements in modeling groundwater flow. PhD thesis, University of Minnesota, Minneapolis, MN, 1997.

[30] Jankovic I, Barnes R. High order line elements in modeling two-dimensional ground water flow. J Hydrology 1999;226:211–23.

[31] Javandel I, Tsang C. Capture-zone type curves: a tool for aquifer cleanup. Ground Water 1986;24(5):616–25.

[32] Kennedy J, Eberhart RC. Particle swarm optimization. In: Proceedings of IEEE international conference on neural networks, Piscataway, NJ, 1995. p. 942–8.

[33] Kirkpatrick S, Gelatt JC, Vecchi M. Optimization by simulated annealing. Science 1983;220:671–80.

[34] Lee S-I, Kitanidis PK. Optimal estimation and scheduling in aquifer remediation with incomplete information. Water Resour Res 1991;27(9):2203–17.
[35] Maskey S, Jonoski A, Solomatine DP. Groundwater remediation strategy using global optimization algorithms. J Water Resour Plann Manage 2002;128(6):431–40.
[36] Matott LS. Ostrich: an optimization software tool, documentation and user's guide, Version 1.4. Department of Civil, Structural, and Environmental Engineering, University at Buffalo, Buffalo, NY, 2004. Available from: www.groundwater.buffalo.edu.
[37] Mayer AS, Kelley CT, Miller CT. Optimal design for problems involving flow and transport phenomena in subsurface systems. Adv Water Resour 2002;25:1233–56.
[38] McKinney DC, Lin M. Pump-and-treat ground-water remediation system optimization. J Water Resour Plann Manage 1996; 122(2):128–36.
[39] Mulligan AE, Ahlfeld DP. Optimal plume capture design in unconfined aquifers. In: Smith J, Burns S, editors. Physicochemical groundwater remediation. Dordrecht: Kluwer Academic; 2001. p. 23–44.
[40] Mulligan AE, Ahlfeld DP. Advective control of groundwater contaminant plumes: model development and comparison to hydraulic control. Water Resour Res 1999;35(8):2285–94.
[41] Press WH, Teukolsky SA, Vetterling WT, Flannery BP. Numerical recipes in C. 2nd ed. New York: Cambridge University Press; 1995.
[42] Rogers LL, Dowla FU. Optimization of groundwater remediation using artificial neural networks with parallel solute transport modeling. Water Resour Res 1994;30(2):457–81.
[43] RS Means ECHOS. Environmental remediation cost data assemblies. RS Means Company, Kingston, MA, 2004.
[44] Sawyer CS, Ahlfeld DP, King AJ. Groundwater remediation design using a three-dimensional simulation model and mixed-integer programming. Water Resour Res 1995;31(5):1373–85.
[45] Strack ODL. Ground water mechanics. Englewood Cliffs (NJ): Prentice-Hall; 1989.
[46] Strack ODL, Jankovic I. A multi-quadric area-sink for analytic element modeling of groundwater flow. J Hydrology 1999; 226(3–4):188–96.
[47] Wang W, Ahlfeld DP. Optimal groundwater remediation with well location as a decision variable; Model development. Water Resour Res 1994;30(5):1605–18.
[48] Yoon J-H, Shoemaker CA. Comparison of optimization methods for ground-water bioremediation. J Water Resour Plann Manage 1999;125(1):54–63.
[49] Yoon J-H, Shoemaker CA. Improved real-coded GA for groundwater bioremediation. J Comput Civil Eng 2001;15(3): 224–31.
[50] Zheng C, Wang PP. MGO: A modular groundwater optimizer incorporating MODFLOW and MT3DMS; documentation and user's guide. University of Alabama and Groundwater Systems Research Ltd., 2003.