Preprint of a manuscript submitted to:
Optimization and Engineering

A mixed-integer simulation-based optimization approach with surrogate functions in water resources management

Thomas Hemker

Simulation and Systems Optimization, Department of Computer Science, Technische Universität Darmstadt, Hochschulstraße 10, 64289 Darmstadt, Germany

Kathleen R. Fowler

Clarkson Center for the Environment, Department of Mathematics, Clarkson University, Potsdam, NY 13676-5815, United States

Matthew W. Farthing

Center for the Integrated Study of the Environment, Department of Environmental Sciences and Engineering, University of North Carolina at Chapel Hill, Chapel Hill, NC 27599-7431, United States

Oskar von Stryk

Simulation and Systems Optimization, Department of Computer Science, Technische Universität Darmstadt, Hochschulstraße 10, 64289 Darmstadt, Germany

Preprint of a manuscript submitted to:
Optimization and Engineering

Abstract. Efficient and powerful methods are needed to overcome the inherent difficulties in the numerical solution of many simulation-based engineering design problems. Typically, expensive simulation codes are included as black-box function generators; therefore, gradient information that is required by mathematical optimization methods is entirely unavailable. Furthermore, the simulation code may contain iterative or heuristic methods, low-order approximations of tabular data, or other numerical methods which contribute noise to the objective function. This further rules out the application of Newton-type or other gradient-based methods that use traditional finite difference approximations. In addition, if the optimization formulation includes integer variables the complexity grows even further. In this paper we consider three different modeling approaches for a mixed-integer nonlinear optimization problem taken from a set of water resources benchmarking problems. Within this context, we compare the performance of a genetic algorithm, the implicit filtering algorithm, and a branch-and-bound approach that uses sequential surrogate functions. We show that the surrogate approach can greatly improve computational efficiency while locating a comparable, sometimes better, design point than the other approaches.

Keywords: Mixed-integer nonlinear optimization, Computational engineering, Simulation-based optimization, Groundwater management, Surrogate optimization

1. Introduction and motivation

The focus of this work is the formulation of simulation-based optimal design problems in the context of water resources management. A considerable challenge in combining simulations and optimization for this scenario is that widely-used subsurface flow simulators, e.g., FEFLOW [15] and MODFLOW [36], are not designed for the specific needs of classical mathematical optimization methods. The development of new, closely tailored simulation and optimization codes is not the method of choice given the enormous effort required to provide such a package. Also, simulation packages are typically of a size or type (e.g., only closed source) such that automatic differentiation methods are not applicable. However, the demand for efficient handling of water resources problems will only be reached by applying optimization methods, as pointed out in [38], which is a common approach in other fields.

We consider two applications proposed in the literature as benchmark problems [33], a water supply problem and a hydraulic capture problem. Both require determining the appropriate number of wells, well locations, and pumping rates in a well-field to meet a manager's goal at a minimum cost. We are interested in how changes in the modeling of the optimization problem can change the range of applicable minimization methods and how the results can differ. The cost model considered here depends on the number of wells in the final design by including a fixed installation cost for each well in addition to an operational cost over the simulation period. Inclusion of fixed costs leads to a discontinuous objective function, and one must consider how to add or remove wells from the design space. The problems would naturally be formulated as mixed-integer, simulation-based, nonlinear programs, as it is mentioned in e.g., [8]. For an overview of applicable techniques see [23]. In the context of water resources, a method based on outer approximation for handling simulation-based constraints in the context of subsurface flow is described in [39].

These benchmark examples are part of a challenging class of problems that arise across computational engineering disciplines. In [35], these types of problems are studied with classical gradient-based methods for nonlinear programming and optimal control theory approaches. In particular, the coupling of numerical simulation and optimization yields an objective function that can be nonconvex, nondifferentiable, discontinuous, and contain undesirable local minima. The feasible region may also be disconnected. The black-box formulation and lack of gradient information implies that algorithms which use only objective function values to guide the minimization are a natural choice for this class of problems.

The computational cost of traditional mixed-integer approaches combined with expensive simulations and lack of gradient information is the motivation to consider alternate formulations. We apply three different optimization methods to the resulting problem formulations, chosen as subset of the available direct sampling, random sampling, and surrogate optimization approaches frequently used in the optimization community. None of the optimization methods need any additional information other than function values for minimization.

For the first approach, we introduce a mixed-integer problem formulation combined with an iterative stochastic modeling technique to build surrogate functions that approximate the simulation-based parts of the objective function. With this procedure, we can then use a branch-and-bound technique to solve the mixed-integer problem in contrast to methods working directly on the simulation results, which impedes relaxation of integer variables. For the second approach, we bypass the number of wells as a decision variable by defining an inactive-well threshold, and in a third approach proposed in the literature [37], we use penalty coefficients to transform the mixed-integer problem into a continuous one. For the two above formulations, we use the implicit filtering algorithm [22] to solve the resulting optimization problem. We compare the above approaches to results obtained with a genetic algorithm (GA), since GAs have gained popularity in the optimal design community and have been successfully applied in the groundwater engineering community, e.g., [27, 7, 31].

We describe the optimization application in §2 with details of the objective function and constraint in §3. The mixed-integer formulation applied to an approach based on sequential surrogate functions is given in §4, and the reference formulations from the literature are given in §5. We present promising numerical results on the benchmarking problems in §6. Finally, in §7 we discuss the results and point the way towards improvement and future work.

2. The community problems

The so-called *community problems* (CPs) posed in [33] were developed after an extensive literature search and serve as a suite of benchmarking applications for both the optimization and water management communities. The CPs consist of models, physical domains, objective functions, and constraints resulting in thirty design applications. In [34], there is a complete set of data to define the physical domains which range from simple homogeneous, confined aquifers to complicated, unconfined aquifers with hydraulic conductivities that are represented using correlated random fields corresponding to typical values observed in nature. The CPs provide an opportunity to apply recent advances in subsurface simulators, numerical methods, optimization algorithms, and computing capabilities to better understand the solutions to these applications.

2.1. Model problems

We consider two of the community problems, a well-field design problem and a hydraulic capture problem. The applications will be defined in terms of an objective function that measures the cost to install and operate a set of wells and constraints that define the goal of each application. These are described in detail in section 3 below. For both problems, the decision variables are the number of wells, n, the pumping rates, $\{Q_i\}_{i=1}^n$, and well locations, $\{(x_i, y_i)\}_{i=1}^n$.

The objective of the well-field design problem is to supply a specified amount of water while minimizing the cost to install and operate the set of wells. We consider the problem in two hydrological settings described in [33]. The first is a homogeneous confined aquifer, while the second is a homogeneous, unconfined aquifer, which leads to additional challenges and complications as is pointed out in [29].

The objective of the hydraulic capture application is to prevent an initial contaminant plume from spreading by using wells to control the direction of flow. Several approaches exist to control the migration of a contaminant plume including particle tracking advective control, flow based gradient control, and constraining a target concentration contour. For this work, we use the gradient control approach, which uses only flow information since it is the most straightforward to implement and is common in practice [1]. To capture the plume with the gradient control method, we impose constraints on hydraulic head differences at certain points around the plume. The hydrological setting for the hydraulic capture problem is the same homogeneous, unconfined aquifer used in the well-field problem. A detailed description about the groundwater flow model and hydrological settings can be found in [34, 19, 20, 18].

2.2. The Simulation environment

To simulate groundwater flow, we use the U.S. Geological Survey code, MODFLOW [36]. MODFLOW is a block-centered finite difference code that is well-supported and widely used. In this context, MODFLOW is the black box simulator that provides the system state (hydraulic head values) of for a certain system design. At the end of the simulated time horizon, the hydraulic heads at installed wells and at monitoring wells are used to calculate the objective function and the constraints.

3. Objective function and constraints

If we take a closer look at the formulation of the underlying design problem, we see that a mixed-integer optimization application naturally arises. This can be viewed not only by the total number of wells installed in the system, which is an integer, but also in determining which well is to be de-installed in order to reduce the number of wells in the design. Such approaches were earlier proposed in the context of subsurface flow problems by [41, 47, 33]. For the sake of comparability to earlier work on the community problems, we use the notation proposed by Mayer [33] and also applied by Fowler [19, 20].

Part of this framework is that the general optimization problem is formulated as

$$\min_{z \in \Omega_z} f(z),$$

where $z = (u, w)$, Ω_w is the feasible domain for the system control parameter w, and Ω_u is the feasible domain for the system state u as simulation output. It follows $\Omega_z = \{(w, u) | u \in \Omega_u,\ w \in \Omega_w\}$.

We recognize a division between different types of infeasibility, determined by Ω_w and Ω_u, since two situations can arise based on simulation results for a set of parameters $w \in \Omega_w$. First, if the simulation fails to return reasonable (or any) output then the state u can not be determined. The output from these simulation runs can not be used quantitatively by an optimization approach, only as qualitative information that the parameter set leads to a infeasible system state u. The second case is that simulation returns a reasonable output for u, but a u that is not element of Ω_u, and therefore not feasible.

Kelley [5] described the simulation based constraints bounding of Ω_w as hidden constraints on the control parameter w. If only w can be varied by the user, then u is a function of w defined by the black-box simulator, thus the behavior of w is not understood a priori. The difference between the two cases becomes important if we take a closer look at the optimization methods that can be applied for such problems. If constraints can be included explicitly, then the returned system state u becomes additional information. To get the most use out of a small number of objective function evaluations, any information obtained from a simulation run should be used in order to reduce the overall computational effort.

We proceed by defining u, w, f, Ω_u, and Ω_w in terms of the water resources applications.

3.1. Decision variables

In our case the relevant system state u is provided by MODFLOW as hydraulic heads h at specified locations in the physical domain with $u = h$, and $h \in \mathbb{R}^{n+2M}$. Specifically, $h_i, i = 1, ..., n$ are the evaluated hydraulic heads at the position of all installed wells. The hydraulic heads at $2M$, monitoring positions $h_i,\ i = (n+1), ..., (n+2M)$, with $M \in \mathbb{N}$, are used to define additional hydraulic gradient constraints for the hydraulic capture application.

The decision variables are given by the vector of the control parameters w, but in a slightly different way than they are used in the reference approach from [19]. The position of each installed well is given by its x and y position on the domain, the operating rate of each well is given by Q, and the total number of wells in the system is n. In the reference formulation, the number of potential wells in the design is fixed and the decision of which wells are installed or de-installed is directly connected to the operation rate Q of each well. We will give more details about the reference approach in the subsequent sections. To conform to the numerical experiments that are made not only in [19] but also in [20] we introduce a switching vector s, with $s \in \{0,1\}^n$, that controls whether or a not a well is installed and ultimately included in the simulation.

It follows that our optimization variables are given by $w = (s, x, y, Q)$, with $\Omega_w \subset \{\{0,1\}^n \times \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n\}$. The constraints formulated in section 3.3 will define the boundary of the feasible domain for w explicitly and implicitly by constraints on the system state u. Further we can describe the system state h that is returned by the simulation as a unknown function of the optimization variables, $h = h(s, x, y, Q)$. Now the optimization problem is completely reduced to a problem of

s, x, y, and Q.

3.2. Objective function

The above definition of the optimization variables leads to a slightly different objective function in comparison to the proposed formulation of [33]. We define the total cost of a system design as a sum of the installation cost, f^c, and the operational cost, f^o, given by

$$f(h,s,x,y,Q) = \underbrace{\sum_{i=1}^{n} s_i c_0 d_i^{b_0} + \sum_{Q_i<0.0} s_i c_1 |1.5 Q_i|^{b_1} (z_{gs} - h^{min})^{b_2}}_{f^c} \tag{1}$$
$$+ \underbrace{\int_0^{t_f} \left(\sum_{i,Q_i \le 0}^{n} s_i c_2 Q_i (h_i - z_{gs}) + \sum_{i,Q_i \ge 0}^{n} s_i c_3 Q_i \right) dt}_{f^o},$$

where the total number of possible wells is given by n. Note that $Q < 0$ m^3/s for extraction wells and $Q > 0$ m^3/s for injection wells. In (1), the first term accounts for drilling and installing each well and the second term is an additional cost for an extraction well pump. In f^o the first term accounts for the cost to lift the water to surface elevation while the second term accounts for operation of the injection wells, which are assumed to operate under gravity feed. In (1), if $s = 1$ then the objective function matches the one in the reference approach and an alternate modeling approach for removing a possible well from the design space must be applied. The values for the constants used in (1) are given in table 1.

As pointed out above, f can be divided into two parts, one depending explicitly on the optimization variables,

$$f^{ex}(s,x,y,Q) = f^c + \int_0^{t_f} \left(\sum_{i,Q_i \ge 0}^{n} s_i c_3 Q_i \right) dt, \tag{2}$$

and one depending implicitly on the optimization variables given by,

$$f^{im}(h, s, x, y, Q) = \int_0^{t_f} \left(\sum_{i, Q_i \leq 0}^{n} s_i c_2 Q_i (h_i - z_{gs}) \right) dt. \quad (3)$$

Note that f^{im} depends on the simulation output via h and can be highly nonlinear due to the subsurface flow simulator. This term also leads us to a potentially noisy, black-box problem as mentioned in the introduction.

Table 1

3.3. Constraints

Constraints are enforced during the optimization so that the wells are located appropriately in the physical domain and are operating at reasonable levels. The location constraints and well capacities are defined by

$$(s_{min}, x_{min}, y_{min}, Q_{min}) \leq (s_i, x_i, y_i, Q_i) \leq (s_{max}, x_{max}, y_{max}, Q_{max}). \quad (4)$$

We also require that two wells can not be placed at the same grid cell,

$$\min(\max(|\overline{x}_i - \overline{x}_j|, |\overline{y}_i - \overline{y}_j|)) \geq s_i s_j \delta, \ \forall \ i, j = 1, ..., n, \ i < j, \quad (5)$$

where $\overline{x}_i = \text{int}(x_i/\delta)$, $\overline{y}_i = \text{int}(y_i/\delta)$ is a grid cell identifier along the x and the y axes from MODFLOW. We also bound the net pumping rate. For the well-field design problem the objective is to meet a specified demand of water and in the hydraulic capturing application the maximal net rate is bounded,

$$Q_T^{max} \leq Q_T = \sum_{i=1}^{n} s_i Q_i \leq Q_T^{min}. \quad (6)$$

Besides these explicit constraints on the the variables, we have implicit constraints for the system state h. For both applications, we bound hydraulic head by,

$$h_{min} \leq h_i \leq h_{max}, i = 1, ..., n, \tag{7}$$

to control the water level in the aquifer. In addition, for the hydraulic capture problem, we bound the hydraulic gradient to control the migration of the plume. This is given by

$$h_i - h_{i+M} \geq d, \quad i = n + j, \ j = 1, ..., M. \tag{8}$$

where i is the hydraulic head inside the plume and $i + M$ is a neighbor just outside the allowed plume boundary. The constraint values are given in the table below. The constraints (4) to (6) bound the feasible design space Ω_w, (7), and (8) the feasible system states domain Ω_u.

Table 2

4. Surrogate optimization for mixed-integer problems

Solving mixed-integer nonlinear problems is a challenging task, even when the objective is described completely analytically, because the process combines difficulties from both continuous and discrete optimization. A short overview of optimization methods for problems with integer and real valued variables is given in [4].

The first approach uses the objective function and the constraints as defined above, leading to a mixed integer optimization problem. The integer variable s is handled explicitly as an optimization variable. In [2] problems of this kind are defined as mixed variable problems. In this work, the use of a subsurface flow simulation impedes a relaxable formulation, which according to [4], would be the first step to apply a mixed-integer nonlinear programming method. This is the motivation to use a surrogate approach.

4.1. Sequential surrogate optimization

The idea behind sequential surrogate optimization for simulation-based and constrained mixed-integer problems is to approximate and replace the black-box parts of the objective function and the constraints of the original problem by a stochastic surrogate. The idea is based on approximating the whole objective function using design and analysis of computer experiments (DACE) [42]. The resulting surrogate problem consists of the analytic part given by the original objective function and an approximated surrogate function replacing the implicitly given, simulation-based part. Additionally, potential underlying noise induced by the numerical simulations is smoothed out. The resulting approximated problem leads to relaxable integer variables of the original problem and allows for gradient based mixed-integer nonlinear programming methods.

To build the surrogate, the simulation-based components of the problem are approximated by a multivariate DACE model $(\hat{f}^{im\ T}, \hat{g}^{im\ T})^T$. Under the assumption of a real-valued $s \in [0,1]^n$, these surrogate satisfies the interpolation constraints

$$\begin{pmatrix} \hat{f}^{im}(w^{(j)}) \\ \hat{g}^{im}(w^{(j)}) \end{pmatrix} = \begin{pmatrix} f^{im}(w^{(j)}) \\ g^{im}(w^{(j)}) \end{pmatrix}, \tag{9}$$

where $w^{(j)} = (s^{(j)}, Q^{(j)}, x^{(j)}, y^{(j)})$, and g^{im} stands for the implicit constraints given by (7) to (8). Here g^{im} is part of a vector

$$g = \begin{pmatrix} g^{ex} \\ g^{im} \end{pmatrix},$$

of functions that summarizes all non-box constraints, (5 to 8) directly transformed into $g(s, x, y, Q) \geq 0$. As proposed in [42] we assume a DACE model $(\hat{f}^{im}, \hat{g}^{im})$ with constant mean, and a covariance basing on the product R of one-dimensional correlations functions. The

DACE-models here are evaluated by the MATLAB Kriging Toolbox of Lophaven et.al. [30], and for more details in the theory of computer experiments, please refer to [28]. By using a surrogate approach, we get a completely analytic mixed-integer, nonlinear programming problem,

$$\min \hat{f} = f^{ex} + \hat{f}^{im}, \text{ s.t. } \hat{g} = \begin{pmatrix} g^{ex} \\ \hat{g}^{im} \end{pmatrix} \geq 0 \tag{10}$$

with the variables (s, x, y, Q) bounded by the box-constraint (4). The new problem no longer depends on simulation evaluations and under the assumption that the DACE-model reflects the major characteristics of the original problem, a minimum of (10) is a promising system design for the original problem. To avoid the influence of different sizes and ranges of the variables, the variable domain was scaled in all dimension to the interval $[0, 1]$.

As the first step of a sequential surrogate optimization procedure only a few points are used to build a DACE-model for the first surrogate problem. The sequential update starts by using additional basis points obtained from optimization results to improve the quality of the surrogate functions. In [43] different sequential update strategies are discussed, but in contrast to the update strategies discussed therein, this approach uses convergence to previously determined points as criteria to switch from searching for the point that minimizes the surrogate to searching for the point that maximizes the mean square error (MSE) of the surrogate function. The MSE information is a valuable byproduct from using the stochastic modeling approach DACE to determine a surrogate as described in [28] in detail. The optimization on the resulting surrogate problems during each iteration is performed by a basic Branch-and-Bound algorithm [23, 17] to guarantee an integer-valued s of new candidates, so that these candidates can be evaluated by the original objective function through simulation. The resulting nonlinear programming subproblems generated by the Branch-and-Bound algorithm are solved with a sequential quadratic programming code, in our case with SNOPT [21]. This procedure was also used successfully for mixed-integer black-box optimization problems in [25, 24].

The key feature of such surrogate optimization approaches is that the optimization process in each iteration dies not run 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 underlying simulation are taken into account. In addition to the DACE-Toolbox, SNOPT and MODFLOW are also called from MATLAB, where the complete sequential update procedure is implemented.

5. Reference approaches

The following two optimization formulations use the objective function directly as defined in (1), but minimization is subject only to the real-valued optimization variables. The constraints are included by a penalty function value f_{pen}. If a design set is not feasible with respect to the explicit constraints (5) or (6) or if no simulation run is obtained, then f_{pen} is used. For this work, f_{pen} is the function value of the initial design set plus 20%. The two reference approaches differ in how values for s are assigned. Implicit filtering (IF), described below, was used for the inactive-well threshold approach as well as for the penalty coefficient approach.

Preprint of a manuscript submitted to:
Optimization and Engineering

5.1. Inactive-well threshold

The reference approach from [19] to determine the number of wells in the design is to set an inactive-well threshold. If a well rate becomes low enough, the well is removed from the design space thereby avoiding any integer variables. For this work

$$s_i = \begin{cases} 0 & \text{for } |Q_i| < 10^{-6}\text{m}^3/\text{s}, \\ 1 & \text{otherwise.} \end{cases} \tag{11}$$

This means that the well rate is set to zero and well i is not included in the installation cost. Incorporating (11) 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, but s is removed from the optimization problem and only continuous variables have to be taken into account.

5.2. Multiplicative Penalty Coefficients

To reformulate a mixed-integer water management application as a continuous nonlinear program, the authors of [37] introduce a polynomial penalty coefficient method (PC). Here the penalty coefficient β is given by

$$\beta_i = Q_i/(Q_i + m), \tag{12}$$

where $0 < m << 1$ is a small number. This penalty term is then multiplied by the fixed costs for each well in f^c. Note that in (12), if $Q_i = 0$, then $\beta_i = 0$ and the fixed cost for well i does not contribute to the objective function at all. For this work we used $m = 10^{-6}$.

5.3. Optimization by implicit filtering

Implicit filtering (IF) was used for the Penalty Coefficient approach as well as for the Inactive-well threshold approach. It is a projected quasi-Newton method that uses a sequence of finite difference gradients [22]. 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 (Implicit Filtering For Constrained Optimization), with the symmetric rank one quasi-Newton update [6]. 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 [19, 20, 3].

5.4. A genetic algorithm for comparison

For comparison purposes, we also use a GA for both applications, since GAs are popular derivative-free approaches for black-box optimization problems. For this work we use the non-dominated sorting genetic algorithm (NSGA-II). The NSGA-II has been shown to perform well in comparison to a number of other genetic algorithms for multi-objective optimization problems [14, 48]. As its name suggests, NSGA-II is a multi-objective genetic algorithm based on non-dominated sorting [14] that includes elitism and does not require a sharing parameter for maintaining solution diversity in multi-objective problems. Although it is a multi-objective optimizer, we use the single-objective problem formulation introduced as equation (1) in order to facilitate comparison with the other optimization methods below.

As it is typical for GAs, NSGA-II uses a binary tournament operator for selection. It includes crossover and mutation operators for both real and binary-coded variables and uses simulated binary crossover for real-coded problems [11, 14]. Box constraints, like equation (4), are automatically enforced in the generation of candidate design variables, while constraints such as equations (5) to (8) must be reformulated as non-negative functions $g(s, x, y, Q) \geq 0$. These are then enforced through the GA tournament selection process in a straightforward manner without the use of penalty parameters [9, 12, 20].

Parameters like the population size, number of generations, as well as the probabilities and distribution indexes chosen for the crossover and mutation operators effect the performance of a GA [40, 33]. The population size used in the numerical experiments, 30, was the lower bound of the suggested range, while a maximum of 30 generations were allowed. The crossover and mutation operator parameters were chosen based on the performance of NSGA-II for a multi-objective test problem with several local Pareto-optimal fronts [13]. We performed a limited number of experiments with other crossover and mutation operator parameter settings, but found no combination that gave better performance across the test problems considered here. The values used are listed in Table 3. NSGA-II is implemented in C and is available for downloading from [10].

Table 3

Lastly, we note that the GA formulation differs in its treatment of the integer variable, s, in some respects. For the hydraulic capture problem, a switch $s_i \in \{0, 1\}$ is used to determine if well i was active or not.

In the well-field design problem, the active-inactive well information in s is collapsed into a single integer variable, p in the range $1, \ldots, p^{max}$. A value of $p \in 1, \ldots, n$ corresponds to shutting off the associated well, while a value greater than n means all wells are active.

6. Numerical Results

All of the above approaches provided feasible solutions with expected characteristics in the numerical experiments. The number of active variables is bounded by the number of maximal installed wells, which is four for the hydraulic capturing problem, and six for both well-field design problems. In the well-field numerical experiments, p^{max} is set to 8 for $n = 6$ in the GA formulation. The resulting unequal ranges of p associated with five well designs and six well designs skews the GA's formulation to favor five-well designs. This reflects a heuristic that installing the minimum number of wells is likely to be cheaper given the relative magnitudes of the installation and operational costs. In addition, the designs for both problems were also subject to the inactive-well threshold given in (11), regardless of the value of s.

For all benchmark examples, the number of installed wells is minimized and the results are comparable to those found in the literature. The solutions differ only in the real-valued variables which are of minor influence within the feasible domain for the total objective function value. Table IV shows the cost of the well design at the initial iterate and at the final design for each optimization model and for each application. The number of calls to the simulator is used to measure performance and is in parentheses. We include some general observations here and give more details for each application in the subsequent sections.

Table 4

The surrogate optimization method found system designs using the smallest number of simulation calls for the all three applications. However, the increasing number of points for interpolation and the computational time during each iteration should be considered, especially if many knots of the branch-and-bound tree have to be explored. We also observed in further numerical experiments that no improvement was made by increasing the stopping criteria to be larger than 150 simulation calls.

Two interesting observations can be seen with the PC and the inactive-well threshold formulations for the well-field design problems as solved with IFFCO. The penalty formulation originally applied by McKinney in [37] is not able to handle positive and negative pumping rates for a well, because the penalty coefficient becomes negative and drives the iterates of the optimization into a bad direction. IFFCO had nearly the same convergence pattern for both formulations because the stencil landed on an exactly zero well rate. Secondly, if the bounds on the pumping rate are varied so that 0 is not the middle of the range of possible pumping rates, the PC combined with IFFCO, is not able to find a solution where a well is de-installed from the design, in contrast to the threshold method which turns a well off. Additionally we tried to apply the "Pseudo Integer" (PI) approach from [37], but after extensive parameter testing not one optimization run could be performed that lead to a decrease of the objective function. In the reference PI approach of McKinney only switches for a larger number of fixed wells were included as decision variables, so that the resulting optimization problem was of a different characteristic than the ones considered here.

We used simulation calls as a measure for a comparison between the methods, but clearly parallel runs would have reduced the wall clock times. The GA would have found, in almost all cases, its solution first, but the objective function values of the GA are the highest compared to the other approaches.

Feasibility is a challenge for these applications and adds complexity in choosing an efficient optimization algorithm. Feasibility is not known until the simulation is complete and therefore computational effort is wasted on infeasible designs. The explicit inclusion of the constraints of the flow direction by surrogate functions could be a second advantage besides the direct handling of integer variables to explain the small number of simulation runs required by the surrogate approach.

As mentioned above we only provide numerical results for one initial system design, which was determined from an engineering perspective. In particular, we observed sensitivity to initial system designs for the unconfined aquifer problems. The GA was unable to find an optimal solution for a wide range of algorithmic parameters including population sizes and the number of generations, when the feasible reference system design was not included in the initial population. To evaluate the GA's sensitivity for the hydraulic capture problem, we considered 10 different random number seeds for the initial population. With the feasible reference design included in the initial population, the GA found optimal solutions in 9 out of 10 cases. Without inclusion of the reference design, the GA found suboptimal solutions with 9 out of 10 different random number seeds. This sensitivity was consistent with the fact that a small population size was used. For a larger population size of 100, the GA found an optimal solution after 30 generations independent of the random number seed.

However, we should note that the results for each of the optimizers are sensitive to both the initial design and the technique for handling infeasible points due to constraint violation.

Figure 1

6.1. Well-field design problem

For the well-field design problem all methods found solution where the number of wells is reduced to five for both hydrological settings. To fulfill the constraints on the total extraction rate (6), $Q = -0.0064 m^3/s$ for each of the five remaining wells.

Table 5

In Figure 1 and Figure 2 we see that all optimization methods reduce the objective function value by the installation costs of one well in less than 20 optimization iterations. But afterward, as is given in the subfigures, the rate of improvement varies for the different approaches. The characteristic is similar for both well-field problems.

Figure 2

Figure 1d and 2d illustrate the final well designs for each approach for the confined and unconfined aquifers. The final positions of the wells are nearly the same with the exception of the wells from the GA solution. Tables 5 and 6 give the specific points found and explains why the total costs barely differ.

Table 6

6.2. Hydraulic capture problem

Figure 3

The hydraulic capture example is the most challenging of this set of problems. The unconfined hydrological setting is sensitive to the positions of the wells with respect to the drawdown constraint (7), and there is the additional capture constraint (8) on the flow direction at the monitoring positions. The capture constraints are particularly sensitive to variations of the well positions within the catchment area, especially if only one well is placed on the whole domain. With only one well in the design, if the well moves outside the capture zone, the head gradient constraints are violated. Feasibility is a major difficulty to overcome, even in generating an initial feasible solutions to start the optimization. As in the two well-field problems the solution with the different optimization approaches are clustered around one location.

Table 7

7. Discussion and conclusion

All of the optimization methods used here are designed for black-box problems yet are not limited to water resource applications. Moreover, the benchmark set provides not only a framework for optimization on subsurface flow problems, but also serves as a set of problems for benchmarking any simulation-based optimization method. In particular, this set consists of black-box problems where parts of the underlying objective function and constraints are noisy and computationally expensive to evaluate.

A first conclusion drawn from this work is that the modeling of the analytical formulation is important regarding (1) the properties of the problem that has to solved and (2) the optimization methods that can be applied. The focus of this work is the different handling of the integer variables in the problem formulation, which are chosen appropriately for each of the considered optimization methods. The results of the surrogate optimization method in combination with a mixed integer problem formulation are promising in comparison to the reference approaches from the literature. Another point in this context is that if the initial number of candidate wells n is large, the complexity of the integer part of the optimization problem increases if the optimal number of wells is significantly smaller than n. Note that as the dimension of the real-valued variables of the problem gets bigger it will take more simulation calls for a derivative-free method to succeed.

Besides the formulation and modeling, new optimal designs for this set of benchmark problems were found and each applied optimization method has returned an acceptable system design of equal characteristic. The only difference between solutions is in the real-valued problem variables. However, despite the small differences in the final costs, the methods' iterations histories differed significantly in the number of function evaluations required for convergence. Although for these problem formulations we are only using flow information, optimization with fewer function calls is attractive when more sophisticated physical models and computationally expensive simulators must be used.

This study helped to identify specific challenges in the area of optimal well-field design. Future work will focus on improved methods for simulation-based, constrained black-box optimization problems with an emphasis on how to handle the "hidden" constraints. These nonconvex, nonlinear constraints were identified in [39] as problematic for efficient optimization methods and lead to a large number of infeasible simulation calls for this work as well. This is a fundamental difficulty, especially if "engineering-perspective" initial designs are not available as initial iterates for the optimizer. Better handling of these constraints would also lead to more robust methods that can find the same optimal point from a variety of starting points.

Finally, we would like to extend this work and explore hybrid optimization approaches to overcome the weaknesses of using a single search approach. Hybrid algorithms can include calling optimizers sequentially or using different optimizers together within a search procedure. See [46] for a classification of hybrid approaches. Coupled methods are beginning to emerge in the area of water resources management (see [26], [45],[16],[44],[32]), although there are open questions regarding *how* to optimally couple two different methods, to define switching criteria for changing from the initial to the second method, and to look for parallelization opportunities for processes that arise during the design a hybrid method. The hydraulic capture problem is a promising candidate for research is this area.

Acknowledgement

Parts of this research were supported by the the Computational Engineering Research Center of the Technische Universität Darmstadt, and a U.S. Department of Education GAANN Fellowship. Finally we would like to thank the unknown reviewers for their valuable comments to improve the quality of this contribution.

References

1. Ahlfeld, D. P. and A. E. Mulligan: 2000, *Optimal Design of Flow in Groundwater Systems*. San Diego: Academic Press.
2. Audet, C. and J. Dennis: 2001, 'Pattern search algorithms for mixed variable programming'. *SIAM Journal on Optimization* **11**, 573–594.
3. Battermann, A., J. Gablonsky, A. Patrick, C. T. Kelley, and K. R. Kavanagh: 2002, 'Solution of a Groundwater Control Problem with Implicit Filtering'. *Optimzation and Engineering* **3**, 189–199.
4. Bussieck, M. and A. Pruessner: 2003, 'Mixed-Integer Nonlinear Programming'. *SIAG/OPT Newsletter: Views & News* **14**(1), 19–22.
5. Carter, R. G., J. M. Gablonski, A. Pattrick, C. T. Kelley, and O. J. Eslinger: 2001, 'Algorithms for Noisy Problems in Gas Transmission Pipeline Optimization'. *Optimization and Engineering* **2**, 139–157.
6. Choi, T., P. Gilmore, O. J. Eslinger, A. Patrick, C. Kelley, and J. Gablonsky: 1999, 'IFFCO: Implicit Filtering for Constrained Optimization, Version 2'. Technical Report CRSC-TR99-23, Center for Research in Scientific Computation.
7. Cieniawski, S. E., J. W. Eheart, and S. Ranjithan: 1995, 'Using genetic algorithms to solve a multiobjective groundwater monitoring problem'. *Water Resources Research* **31**(2), 399–409.
8. Cuncha, M.: 1999, 'On Solving Aquifer Management Problems with Simulated Annealing Algorithms'. *Water Resources Management* **13**, 153–169.
9. Deb, K.: 2000, 'An efficient constraint handling method for genetic algorithms'. *Computer Methods in Applied Mechanics and Engineering* **186**(2–4), 311–338.
10. Deb, K.: 2003, 'KanGal Homepage'. Indian Institute of Technology Kanpur http://www.iitk.ac.in/kangal.
11. Deb, K. and R. B. Agrawal: 1995, 'Simulated binary crossover for continuous search space'. *Complex Systems* **9**, 115–148.
12. Deb, K. and T. Goel: 2001, 'Controlled Elitist Non-dominated sorting genetic algorithms for better convergence'. In: E. Zitler, K. Deb, L. Thiele, C. Coello-Coello, and D. Corne (eds.): *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization EMO 2001*. pp. 67–81.

13. Deb, K., A. Pratap, S. Agarwal, and T. Meyarivan: 2000, 'A Fast and Elitist Multi-Objective Genetic Algorithm: NSGA-II'. KanGal Report 200001, Indian Institute of Technology Kanpur, Kanpur, India.
14. Deb, K., A. Pratap, S. Agarwal, and T. Meyarivan: 2002, 'A Fast and Elitist Multi-Objective Genetic Algorithm: NSGA-II'. *IEEE Transactions on Evolutionary Computation* **6**(2), 182–197.
15. Diersch, H.-J.: 1998, 'Interactive, Graphics-Based Finite-Element Simulation System FEFLOW for Modeling Groundwater Flow Contaminant Mass and Heat Transport, Users Manual'. WASY GmbH, Berlin.
16. Espinoza, F., B. S. Minsker, and D. E. Goldberg: 2005, 'Adaptive Hybrid Genetic Algorithm for Groundwater Remediation Design'. *Water Resources and Planning Management* **131(1)**, 14–24.
17. Floudas, C. A.: 1999, *Deterministic Global optimization Theory, Methods and Applications*. Kluwer Academic Press.
18. Fowler, K., J. Reese, C. Kees, J. J.E. Dennis, C. Kelley, C. Miller, C. Audet, A. Booker, G. Couture, R. Darwin, M. Farthing, D. Finkel, J. Gablonsky, G. Gray, and T. Kolda: 2007, 'A Comparison of Derivative-free Optimization Methods for Water Supply and Hydraulic Capture Community Problems'. *Submitted to Advances in Water Resources*.
19. Fowler, K. R., C. T. Kelley, C. E. Kees, and C. T. Miller: 2004a, 'A Hydraulic Capture Application For Optimal Remediation Design'. In: Miller, Farthing, Gray, and Pinder (eds.): *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*. pp. 1149–1157.
20. Fowler, K. R., C. T. Kelley, C. T. Miller, C. E. Kees, R. M. Darwin, J. P. Reese, M. W. Farthing, and M. S. C. Reed: 2004b, 'Solution of a Well-Field Design Problem with Implicit Filtering'. *Optimzation and Engineering* **5**, 207–233.
21. Gill, P. E., W. Murray, and M. A. Saunders: 2006, '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.
22. Gilmore, P. and C. T. Kelley: 1995, 'An implicit filtering algorithm for optimization of functions with many local minima'. *SIAM Journal on Optimization* **5**, 269–285.

23. Grossmann, I. E.: 2002, 'Review of Nonlinear Mixed-Integer and Disjunctive Programming techniques'. *Optimization and Engineering* **3**, 227–252.
24. Hemker, T., K. R. Fowler, and O. von Stryk: 2006a, 'Derivative-Free Optimization Methods for Handling Fixed Costs in Optimal Groundwater Remediation Design'. In: *Proc. of the CMWR XVI - Computational Methods in Water Resources*.
25. Hemker, T., M. Glocker, H. De Gersem, O. von Stryk, and T. Weiland: 2006b, 'Mixed-integer simulation-based optimization for a superconductive magnet design'. In: *Proceedings of the Sixth International Conference on Computation in Electromagnetics*. Aachen, Germany.
26. Hsiao, C. and L. Chang: 2002, 'Dynamic optimal Groundwater Management with Inclusion of Fixed Costs'. *Journal of Water Resources Planning and Managment* **128**, 57–64.
27. Huang, C. and A. S. Mayer: 1997, 'Pump-and-treat optimization using well locations and pumping rates as decision variables'. *Water Resources Research* **33**(5), 1001–1012.
28. Koehler, J. R. and A. Owen: 1996, 'Computer Experiments'. *Handbook of Statistics* **13**, 261–308.
29. Lall, U. and M. D. Santini: 1989, 'An optimization model for unconfined stratified aquifer systems'. *Journal of Hydrology* **111**, 145–162.
30. Lophaven, N., H. B. Nielsen, and J. Sondergaard: 2002, 'DACE, A Matlab Kriging Toolbox'. Technical Report IMM-TR-2002-12, IMM.
31. Maskey, S., A. Jonoski, and D. P. Solomatine: 2002, 'Groundwater Remediation Strategy Using Global Optimization Algorithms'. *Journal of Water Resources Planning and Management* **128**(6), 431–440.
32. Mattot, S. L., S. L. Bartelt-Hunt, A. J. Rabideau, and K. R. Fowler: 2006, 'Application of Heuristic Optmization Techniques and Algorithm Tuning to Multi-Layered Sorptive Barrier Design'. *Accepted to appear in Environmental Science and Technology*.
33. Mayer, A. S., C. T. Kelley, and C. T. Miller: 2002a, 'Optimal Design for Problems Involving Flow and Transport in Saturated Porous Media'. *Advances in Water Resources* **12**, 1233–1256.

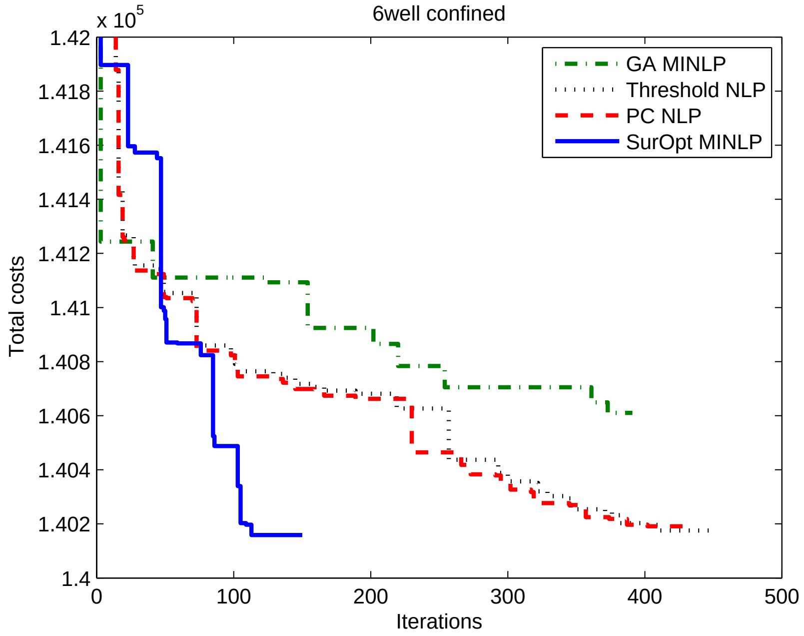
34. Mayer, A. S., C. T. Kelley, and C. T. Miller: 2002b, 'Optimal Design for Problems Involving Flow and Transport in Saturated Porous Media, Electronic Supplement'. *Advances in Water Resources* **12**, 1233–1256.
35. Mays, L. W.: 1997, *Optimal Control of Hydrosystems*. Marcel Dekker.
36. McDonald, M. and A. Harbaugh: 1988, 'A Modular Three Dimensional Finite Difference Groundwater Flow Model'. *U.S. Geological Survey Techniques of Water Resources Investigations*.
37. McKinney, D. C. and M.-D. Lin: 1995, 'Approximate mixed-integer nonlinear programming methods foroptimal aquifer remediation design'. *Water Resources* **31**, 731–740.
38. Nicklow, J. W.: 2000, 'Discret-Time optimal Control for Water Resources Engineering and Management'. *International Water Resources Association* **25**(1), 89–95.
39. Papadopoulou, M. P., G. F. Pinder, and G. Karatzas: 2003, 'Enhancement of the outer approximation method for the solution of concentration constrained optimal-design groundwater-remediation problems'. *Water Resources Research* **39**(7), 1185.
40. Reed, P., B. Minsker, and D. E. Goldberg: 2000, 'Designing a competent simple genetic algorithm for search and optimization'. *Water Resources Research* **36**(12), 3757–3761.
41. Ritzel, B. J., J. W. Eheart, and S. Ranjithan: 1994, 'Using genetic algorithms to solve a multiple objective groundwater pollution containment problem'. *Water Resources Research* **30**(5), 1589–1604.
42. Sacks, J., S. B. Schiller, and W. J. Welch: 1989, 'Design for Computer Experiments'. *Technometrics* **31**, 41–47.
43. Sasena, M. J.: 2002, 'Flexibility and Efficiency Enhancements for Constrained Global Design Optimization with Kriging Approximations'. Ph.D. thesis, University of Michigan.
44. Sayeed, M. and G. Mahinthakumar: 2005, 'Efficient Parallel Implementation of Hybrid Optmization Approaches for Solving Groundwater Inverse Problems'. *Journal of Computing in Civil Engineering* **19(4)**, 329–340.
45. Shieh, H. and R. Peralta: 2005, 'Optimal In Situ Bioremediation Design by Hybrid Genetic Algorithm and Simulated Annealing'. *Water Resources and Planning Management* **131(1)**, 67–78.

46. Talbi, E.: 2002, 'A Taxonomy of Hybrid Metaheuristics'. *Journal of Heuristics* **8**, 541–563.

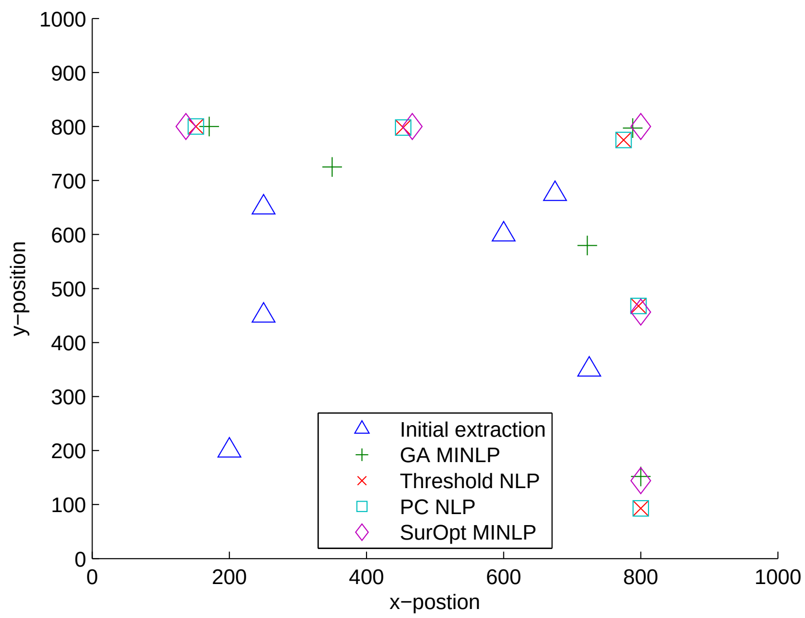
47. Watkins Jr, D. and D. C. McKinney: 1998, 'Decomposition methods for water resources optimization models with fixed costs'. *Advances in Water Resources* **21**(4), 261–324.

48. Zitzler, E., K. Deb, and L. Thiele: 2000, 'Comparison of Multiobjective Evolutionary Algorithms: Empirical Results'. *Evolutionary Computation Journal* **8**(2), 173–195.

**Preprint of a manuscript submitted to:
Optimization and Engineering**

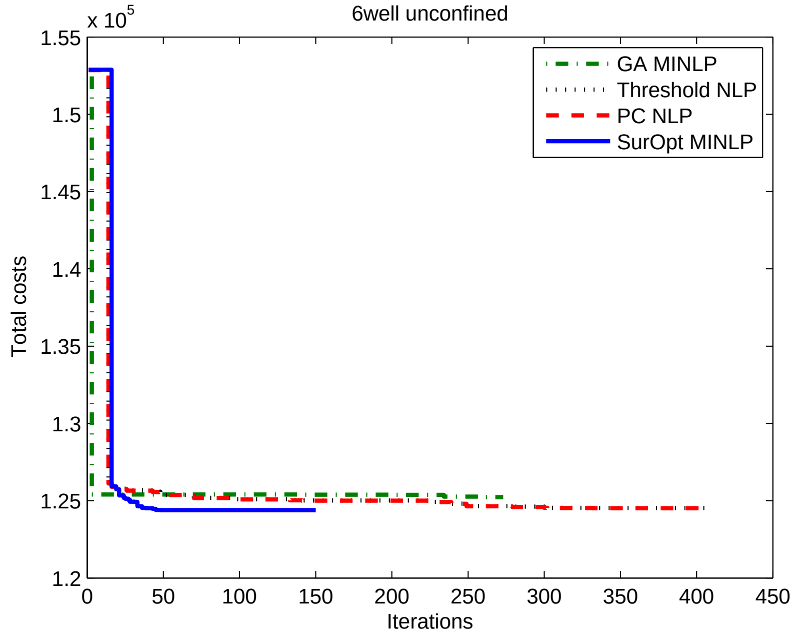


a. Objective function value versus number of simulations calls

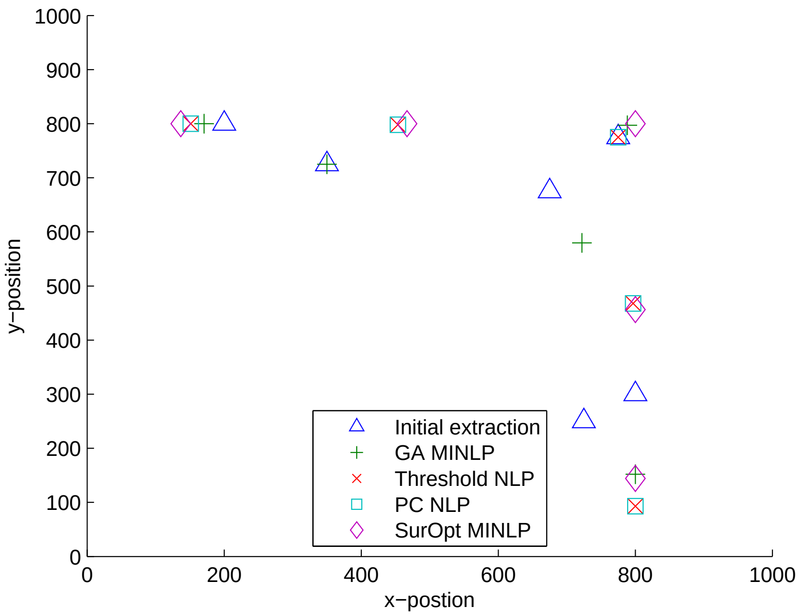


b. Initial and final well positions

Figure 1. Objective function reduction versus iterations for confined well-field problem and a comparison of the final positions

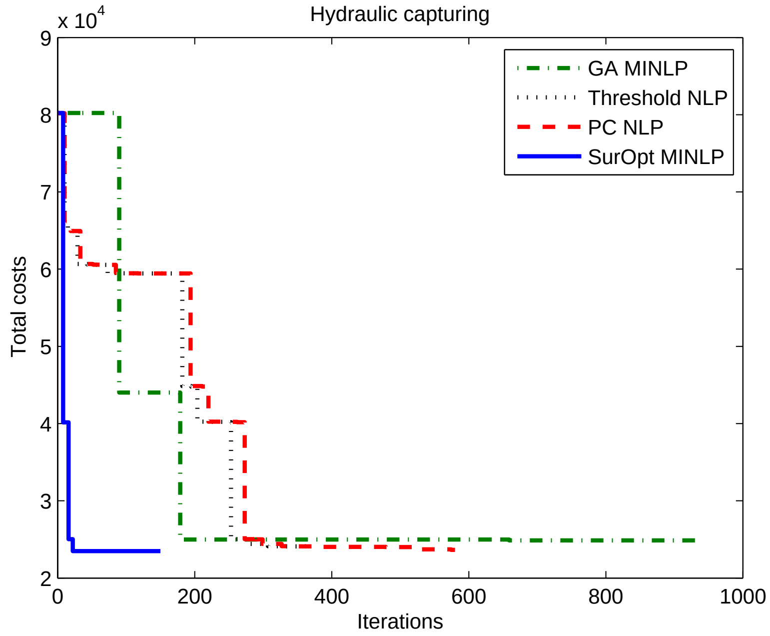


a. Objective function value versus number of simulations calls

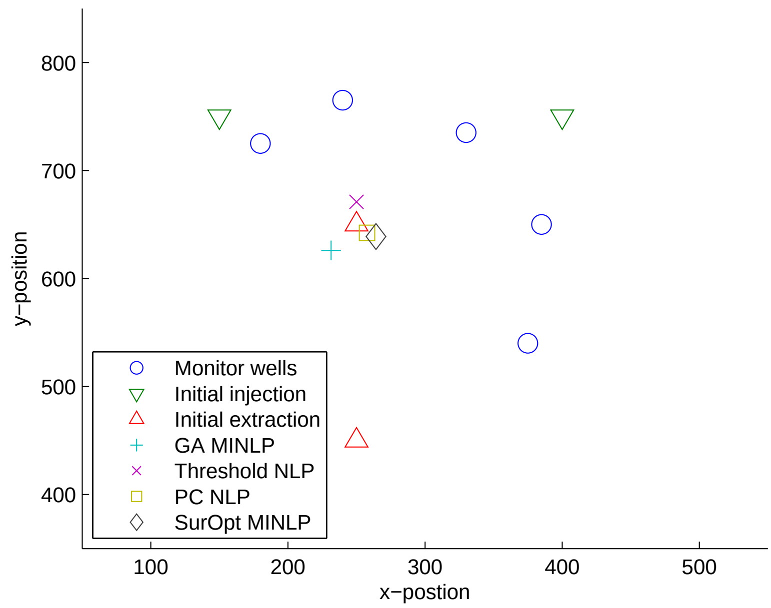


d. Initial and final well positions

Figure 2. Objective function reduction versus iterations for unconfined well-field problem and a comparison of the final positions



a. Objective function value versus number of simulations calls



d. Initial and final well positions

Figure 3. Objective function reduction versus iterations for hydraulic capturing problem and a comparison of the final positions

Table 1. Constants included in the objective function f.

constant	well-field confined	well-field unconfined	hydraulic capturing	units
b_0	0.3	0.3	0.3	-
b_1	0.45	0.45	0.45	-
b_2	0.64	0.64	0.64	-
c_0	5.5×10^3	5.5×10^3	5.5×10^3	$\$/\mathrm{m}^{b_0}$
c_1	5.75×10^3	5.75×10^3	5.75×10^3	$\$/[(\mathrm{m}^3/\mathrm{s})^{b_1} \cdot \mathrm{m}^{b_2}]$
c_2	2.9×10^{-4}	2.9×10^{-4}	2.9×10^{-4}	$\$/\mathrm{m}^4$
c_3	1.45×10^{-4}	1.45×10^{-4}	1.45×10^{-4}	$\$/\mathrm{m}^3$
d_i	60	30	30	m
z_{gs}	60	30	30	m
t_f	5	5	5	years
h^{min}	40	10	10	m

Table 2. Additional constants in use for the constraints.

constant	well-field confined	well-field unconfined	hydraulic capturing	units
s_{min}	0	0	0	-
s_{max}	1	1	1	-
x_{min}	0	0	0	m
x_{max}	800	800	800	m
y_{min}	0	0	0	m
y_{max}	800	800	800	m
Q_{min}	-6.4×10^{-3}	-6.4×10^{-3}	-6.4×10^{-3}	m^3/s
Q_{max}	6.4×10^{-3}	6.4×10^{-3}	6.4×10^{-3}	m^3/s
Q_T^{max}	n.d.	n.d.	-3.2×10^{-2}	m^3/s
Q_T^{min}	-3.2×10^{-2}	-3.2×10^{-2}	n.d.	m^3/s
δ	20	20	10	m
h^{max}	60	30	30	m
d	n.d.	n.d.	10^{-4}	m/s
M	n.d.	n.d.	5	-

Table 3. GA parameters for simulations.

30	size of population
30	max number of generations
0.9	crossover probability
0.1	real-coded mutation probability
20	distribution index for real-coded crossover
10	distribution index for real-coded mutation
0.5	binary-coded mutation probability

Table 4. Final objective function values and obtained simulation calls in brackets.

problem formulation	optimization method	well-field confined	well-field unconfined	hydraulic capturing
Initial	-	$170,972	$152,878	$80,211
MINLP	NSGA-II	$140,610 (391)	$125,226 (273)	$24,854 (659)
Threshold/NLP	IFFCO	$140,175 (362)	$124,527 (320)	$24,032 (363)
PC/NLP	IFFCO	$140,190 (402)	$124,512 (316)	$23,640 (580)
MINLP	Sur Opt	$140,159 (113)	$124,387 (87)	$23,491 (22)

Table 5. Solutions for confined well-field problem.

location (meters)	initial	NSGA-II MINLP	IFFCO Threshold	IFFCO PC	Sur opt MINLP
x_1, y_1	350, 725	366.0, 791.8	341.0, 798.1	350.0, 798.1	135.6, 800
x_2, y_2	775, 775	788.6, 799.0	799.4, 775.0	799.4, 775.0	800, 22.7
x_3, y_3	675, 675	769.6, 676.5	656.6, 794.6	668.9, 772.5	399.8, 800
x_4, y_4	200, 200	181.1, 376.4	102.5, 797.2	102.5, 797.2	800, 20
x_5, y_5	725, 350	788.3, 302.6	792.0, 300.0	792.0, 300.0	800, 800
x_6, y_6	600, 600	674.4, 602.3	600.0, 600.0	600.0, 600.0	800, 567.7
integer decision	-	$s = 5$	$q_6 = 0$	$q_6 = 0$	$s_2 = 0$

Table 6. Solutions for the unconfined well-field problem.

location (meters)	Initial	NSGA-II MINLP	IFFCO Threshold	IFFCO PC	Sur opt MINLP
x_1, y_1	350, 725	350.1, 725.0	453.6, 798.1	453.6, 798.1	466.9, 800.0
x_2, y_2	775, 775	788.3, 797.1	775.0, 775.0	775.0, 775.0	800.0, 800.0
x_3, y_3	675, 675	722.2, 579.7	796.9, 467.8	796.9, 467.8	800, 456.2
x_4, y_4	200, 800	170.8, 800.0	151.3, 800.0	151.3, 800.0	136.7, 800
x_5, y_5	725, 250	710.0, 324.2	725.0, 250.0	725.0, 250.0	800, 144.2
x_6, y_6	800, 300	800.0, 152.0	800.0, 92.8	800.0, 92.8	800, 143.7
integer decision		$p = 6$, $q_5 \leq 10^{-6}$	$q_5 \leq eps$	$q_5 = 0$	$s_6 = 0$

Table 7. Solutions for hydraulic capturing problem.

location (meters)	Initial	NSGA-II MINLP	IFFCO Threshold	IFFCO PC	Sur opt MINLP
x_1, y_1	150, 750	183.5, 342.2	165.3, 750.0	165.3, 750.0	10.0, 746,8
x_2, y_2	400, 750	143.3, 528.8	400.0, 750.0	400.0, 750.0	411.9, 717.4
x_3, y_3	250, 650	272.2, 652.2	250.0, 671.0	257.7, 642.3	264.3, 638.9
x_4, y_4	250, 450	231.5, 626.1	250.0, 450.0	250.0, 450.0	266.9, 413,9
well rates (m^3/s)					
q_1, s_1	6.4E-3, 1	6.4E-3, 0	0.0, -	0.0, -	6.4E-3, 0
q_2, s_2	6.4E-3, 1	-3.362E-3, 0	0.0, -	0.0, -	6.2969E-3, 0
q_3, s_3	-6.4E-3, 1	-6.185E-3, 0	-5.7499E-3, -	-5.4999E-3, -	-5.3984E-3, 1
q_4, s_4	-6.4E-3, 1	-6.306E-3, 1	0.0, -	0.0, -	-0.0064, 0

Preprint of a manuscript submitted to:
Optimization and Engineering

List of Figures

List of Tables