A Novel Method for Well Placement Design in Groundwater Management: Extremal Optimization

Fleford Redoloza[a], Liangping Li[a,*]

[a]*Department of Geology and Geological Engineering, South Dakota School of Mines and Technology, Rapid City, 57701, USA*

Abstract

Well placement design refers to finding the optimal well locations to install with a set of constraints. This is important for both petroleum engineering and water resource management. This study presents a novel optimization method for well placement design in groundwater management. The proposed method, EO-WPP, is based on the Extremal Optimization (EO) algorithm. EO works by modifying the components of a solution that contribute the least to its overall performance. EO-WPP extends the EO algorithm to the fields of groundwater management and well field optimization for the first time. Groundwater Management program (GWM) is coupled with EO-WPP and used to rank wells in terms of pumping rate, given well locations. In the first testing phases of this work, EO-WPP was applied to a problem of simple geometry and a simple synthetic model in order to study its performance and its emergent spatial behaviors. Results show that the proposed method was faster than Particle Swarm Optimization (PSO) and the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithms. EO-WPP then was applied to a field problem involving the Aberdeen groundwater model in South Dakota. The results show that EO-WPP was able to generate a series of possible of well fields that can be used to pump effectively groundwater from the Elm aquifer.

Keywords: Extremal Optimization, GWM, Well Placement, Aberdeen

---

*Corresponding author
Email address: `liangping.li@sdsmt.edu` (Liangping Li)

Contents

1. Introduction

Well placement design refers to finding the optimal well locations to install with a set of constraints such as drawdown. This is a common problem found in many fields of natural resource management. In the petroleum industry, solving the well placement problem allows the design of optimal well fields that can efficiently and economically produce hydrocarbon in reservoirs (Sarma et al., 2008; Feng et al., 2012; Nwankwor et al., 2013; Nozohour-leilabady and Fazelabdolabadi, 2016). For water resource management, solving the well placement problem can lead to efficient well field design for pumping groundwater or for aquifer remediation (Park and Aral, 2004; Bayer et al., 2009; Elçi and Ayvaz, 2014; Wang and Ahlfeld, 1994).

In previous decades, many algorithms have been developed to solve the well placement problem (Minton, 2012). These optimization algorithms can be classified into two main categories: global search algorithms and local search algorithms.

Global search algorithms refer to optimization algorithms designed to seek the global minimum or maximum of a given optimization problem (Chong and Zak, 2013). Global search algorithms are the common type of methods used for well placement optimization (Minton, 2012). Examples of algorithms include differential evolution, particle swarm optimization, and genetic algorithms (Elçi and Ayvaz, 2014; Feng et al., 2012; Emerick et al., 2009). To improve performance, researchers have also developed hybrids of these methods (Nwankwor et al., 2013; Guyaguler et al., 2001). These algorithms gain popularity likely due to their ability to avoid local minimums by relying on stochastic methods and evaluating a population of solutions.

Unlike global search algorithms, local search algorithms are optimization algorithms that are susceptible to converging to sub-optimal solutions. But in exchange for the risk of getting trapped at local minimums, local search algorithms can reach an optimal solution faster than global search algorithms (Mahinthakumar and Sayeed, 2005; Humphries et al., 2014). Local search methods are faster because assumptions are usually made for the optimization problem that allows fewer evaluations of the objective function. Reducing the number of times for evaluating the objective function is a valuable technique for speed, especially when the objective function involves a large numerical model that is computationally expensive. Examples of local search algorithms include the Nelder-Mead method, the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm, gradient descent algorithms, and other pattern search algorithms (Nelder and Mead, 1965; Liu and Nocedal, 1989; Ruder, 2016; Torczon, 1997). To reduce risk of getting stuck on local minimums while still retaining the speed of requiring few objective function evaluations, researchers have developed hybrids of global and local search optimization methods (Mahinthakumar and Sayeed, 2005; Humphries et al., 2014). Our proposed method seeks a similar goal, however we approach the task using the unique perspective of extremal optimization.

Extremal optimization (EO) is an optimization algorithm introduced by Boettcher and Percus (1999). The main heuristic of EO is that in order to improve the performance of a given solution, simply identify the least performing component of a solution and replace it with something randomly generated. By iteratively changing the worst component of the solution, the performance of the overall solution will improve. After its introduction in 1999, EO was used in many disciplines of science and engineering. In mechanical engineering, De Sousa et al. (2004) used a variant of EO called generalized extremal optimization to design a heat pipe for satellite thermal control. In distributed computing, De Falco et al. (2015) used EO as a part of a load-balancing algorithm for clusters of multi-core processors. Additional applications include fractional order proportional-integral-derivative (PID) controllers, wind speed forecasting, and spin glass (e.g., Zeng et al., 2015; Chen et al., 2018; Boettcher, 2005)

Although EO has been used in a variety of applications, it has received less attention in hydrogeology. This is mainly because EO requires a fitness function that can rank the fitness of each of the components of a solution (Boettcher and Percus, 2002). Most optimization algorithms use an objective function that outputs a single value. However, EO also needs a function that determines how much each component of a solution contributes to the overall objective function. For many problems, such a function might be too ambiguous or impossible to define. Variants of EO, such as general extremal optimization (De Sousa et al., 2004) try to solve this problem by defining a general way to partition the objective function into components that correspond to components of a solution.

In this work, we introduce EO to the well placement problem in groundwater management for the first time and propose a novel component-based fitness function specific for the problem domain, termed as Extremal Optimization for the Well Placement Problem (EO-WPP). The EO-WPP algorithm will employ this new fitness function to allow the use of EO on well placement problems without significantly changing the structure of the original EO algorithm. We show that EO-WPP with its unique fitness function allows the algorithm to adopt both the local-minimum avoidance behavior of global search algorithms and the speed of local search algorithms. By the nature of the heuristic used to replace the worst-performing components, EO-WPP also displays emergent spatial behaviors that are useful for the design of well fields. A simple geometry and synthetic examples will be used to demonstrate the method. The method then will be applied in Aberdeen aquifer in South Dakota for a field example of the well placement problem.

2. Methodology

2.1. Groundwater Flow Equation

The governing equation for three-dimensional transient groundwater flow in heterogeneous and anisotropic conditions is given as follows (Anderson et al., 2015):

$$\frac{\partial}{\partial x}(K_x \frac{\partial h}{\partial x}) + \frac{\partial}{\partial y}(K_y \frac{\partial h}{\partial y}) + \frac{\partial}{\partial z}(K_z \frac{\partial h}{\partial z}) - W^* = S_s \frac{\partial h}{\partial t} \tag{1}$$

where K is hydraulic conductivity, h is hydraulic head, S_s is specific storag,e and t is time. W^* is a source or sink. In this study, MODFLOW(McDonald et al., 2003), a modular finite-difference flow model program developed by the U.S. Geological Survey (USGS), is used to solve the groundwater flow equation numerically.

A groundwater model is a conceptual representation of a real aquifer. When building a model, errors can be introduced through measurement, conceptual framework, or other sources (Anderson et al., 2015). This means that if a well-field configuration is optimized using a groundwater model, the optimal solution for the model could be different from the optimal solution for the real aquifer. For example, optimization algorithms may place wells next to constant head boundaries since there is effectively no limit on the flow rate. When interpreting any well field solution, ensure that the solution takes advantage of the underlying hydrogeological structure of the study area, instead of improbably using characteristics only unique to the computer model.

2.2. Extremal Optimization for Well Placement Problems (EO-WPP)

The EO-WPP algorithm is very similar to the original EO algorithm that was proposed by Boettcher and Percus (1999). The main difference is how the fitness function was defined and how the least fit component of the solution was adjusted.

The fitness function quantifies how much a given component of the solution contributes to the overall performance of the solution. Within the context of well placement problems, the fitness function determines how much a given pumping well contributes to the overall discharge of the well field. For EO-WPP, the fitness function evaluated at a well is defined to be the total volume of water produced by the well after operating at its optimal pumping rates for all time periods. Therefore, wells with a high fitness will produce a greater cumulative discharge than other wells. One of the main assumptions in EO-WPP is that the well which produces the most amount of water with the constraint of drawdown is the most fit well. The goal of EO is to adjust the components of a solution in order to maximize their fitness. Thus the goal of EO-WPP is to adjust the location of the wells such that their cumulative output is maximized.

The purpose of EO-WPP's fitness function is to determine optimal pumping rates, given the well locations. These optimal pumping rates are computed using a separate optimization method. For this study, EO-WPP's fitness function was implemented using a computer program called GWM (Ahlfeld et al., 2005) (see Section 2.4 for details about GWM). However, any other local optimization algorithm can be used. EO-WPP only uses the fitness function to identify the best and worst wells. Therefore, the accuracy of the optimal pumping rates only needs to be good enough to identify the best and worst wells. Approximations of the optimal pumping rates can be quickly reached by adjusting the convergence criterion of the optimization algorithm. This modification reduces the computational requirement for evaluating the fitness function.

In the original EO algorithm, the least fit component is replaced by a randomly generated component. In EO-WPP, the least fit well is removed and replaced with a new well that is randomly placed near the most fit well. This heuristic assumes that the best place to put a new well will likely be near the best well. The heuristic allows the EO algorithm to quickly converge toward an optimal solution, but it also generates a bias and makes the algorithm more susceptible to being trapped at local maximums. This can be resolved by implementing the τ-EO method introduced by Boettcher and Percus (1999).

2.3. EO-WPP Algorithm

The original EO algorithm was detailed in Boettcher and Percus (1999). The proposed EO-WPP has the following steps:

Step 1: *Initialize the solution matrix*

The algorithm begins by initializing the solution matrix, W. For EO-WPP, W is the matrix that contains the locations of all the wells that are to be optimized. When expanded, the locations of the wells can be encoded as such:

$$W = \begin{bmatrix} \vec{w}_1 \\ \vdots \\ \vec{w}_i \\ \vdots \\ \vec{w}_I \end{bmatrix} = \begin{bmatrix} x_1 & y_1 \\ \vdots & \vdots \\ x_i & y_i \\ \vdots & \vdots \\ x_I & y_I \end{bmatrix} \tag{2}$$

where I is the total number of wells, $\vec{w}_i$ is the location row vector of the ith well, and x_i, y_i are the row and column locations of the ith well. The goal of EO-WPP is to determine the W matrix that maximizes the objective function. It does this by starting with an initial, randomly generated W matrix, and then iteratively adjusting this matrix until it converges onto a solution.

When initializing the solution matrix, a given number of wells are randomly placed within the model domain. This must be done in way that makes the constraint function return $True$, as shown below. The constraint function, $\mathbf{C}$, is the function that checks if a given solution matrix respects all constraints. For EO-WPP, the $\mathbf{C}$ function checks spatial constraints between wells and boundary conditions. Examples of spatial constraints may include minimum distances to the boundary or defining areas of the domain to avoid. Another important spatial constraint is that no two wells can have the same location or occupy the same cell:

$$\mathbf{C}(W) = \begin{cases} \mathbf{C}(W) = \mathit{True} & \text{if } W \text{ respects all constraints} \\ \mathbf{C}(W) = \mathit{False} & \text{if } W \text{ fails to meet all constraints} \end{cases} \tag{3}$$

The constraint function simply returns *True* if the solution matrix respects all constraints and returns *False* if it does not. When initializing the solution matrix, the generated matrix must satisfy $\big(\mathbf{C}(W_{l=0}) = \mathit{True}\big)$. Constraints for the drawdown and the pumping rates are handled by the fitness function.

Step 2: *Evaluate the fitness function*

Given the solution matrix, W, the corresponding fitness vector is calculated. The fitness vector, Q, is the vector that contains the fitness for all the components of the solution. For EO-WPP, Q is the vector that contains the cumulative volumes for each of the wells. The vector can be constructed as such:

$$Q = \begin{bmatrix} q_1 \\ \vdots \\ q_i \\ \vdots \\ q_I \end{bmatrix} \tag{4}$$

where q_i is the cumulative volume of water the ith well produces after operating through all time periods using its optimal pumping rates. Note that if only one stress period exists, then q_i can also represent the pumping rate of the ith well.

To calculate the fitness vector, the fitness function is applied to the solution matrix. The fitness function, $\mathbf{F}$, is the function that takes a solution matrix as its input and determines the corresponding fitness vector. For EO-WPP, $\mathbf{F}$ takes the well locations, W, and calculates their corresponding fitness,

Q:

$$Q = \mathbf{F}(W) \tag{5}$$

With every iteration of EO-WPP, the previous optimal values are discarded and new optimal values are recalculated. This is because the placement of new wells may affect the optimal values of adjacent wells. When implementing the **F** function, its computer code incorporates both the groundwater model and the optimization program that determines the optimal pumping rates. In this paper, the groundwater flow model was simulated by MODFLOW (McDonald et al., 2003) and the pumping rate optimization was performed by GWM (Ahlfeld et al., 2005). Note that the initial pumping rates are determined by the optimization program used. For this study, GWM initializes the pumping rates to 20% of their maximum pumping rate.

Step 3: *Remove the worst well*

With the new fitness vector, the worst well is identified. The worst well is the well that has the lowest fitness value:

$$\vec{w}_{worst} = \{\vec{w}_{i_{worst}} \in W : q_{i_{worst}} \leq q_i \ \forall q_i \in Q\} \tag{6}$$

After the worst well is identified, it is removed from the solution matrix. This is done by defining a new solution matrix, W', that contains everything but the worst well:

$$W' = \{\vec{w} \in W : \vec{w}_{worst} \notin W'\} \tag{7}$$

Step 4: *Insert a new well*

To replace the removed well, a new well is generated. The location of the new well $\vec{w}_{new}$ is dependent on the location of the best well, $\vec{w}_{best}$, the maximum distance between wells, d_{max}, and a random vector, $\vec{u}$:

$$\vec{w}_{best} = \{\vec{w}_{i_{best}} \in W : q_{i_{best}} \geq q_i \ \forall q_i \in Q\} \tag{8}$$

$$d_{max} = \text{maximum Euclidean distance between any two wells within } W' \tag{9}$$

$$\vec{u} = \text{random vector with a length within } (0,1] \text{ and the same dimensions as } \vec{w} \tag{10}$$

$$\vec{w}_{new} = \vec{w}_{best} + d_{max}\vec{u} \tag{11}$$

The new well is placed at a random location near the best well (Equation 11). The new well then is inserted into the solution matrix, W', to form a new solution matrix, W'':

$$W'' = \{\vec{w} : (\vec{w} \in W') \ or \ (\vec{w} = \vec{w}_{new})\} \tag{12}$$

Before moving on, the new solution matrix, W'' must satisfy all constraints $\big(\mathbf{C}(W'') = \mathit{True}\big)$. If it does not $\big(\mathbf{C}(W'') = \mathit{False}\big)$, then a new $\vec{u}, \vec{w}_{new}$ and W'' is generated and calculated until the new well field respects all constraints $\big(\mathbf{C}(W'') = \mathit{True}\big)$. After W'' passes all constraint checks, the temporary well field becomes accepted as the new well field configuration for the current iteration of the algorithm $\left(W'' \xrightarrow{\mathbf{C}(W'')=\mathit{True}} W\right)$.

Step 5: *Check if a new best solution is found*

To check the performance of the new solution, its objective function is evaluated. **O** is the objective function that EO-WPP tries to maximize. It is a function of the location of the wells, W, and can be calculated with the fitness function, **F**:

$$\mathbf{O}(W) = \mathbf{F}(W) \cdot \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}_{I \times 1} = \sum_{i=1}^{I} q_i \tag{13}$$

Unlike the fitness function, the objective function does not require a separate optimization process. The objective function simply takes the results of the fitness function, Q, and reports the sum of the fitness values of all the components. In other words, the objective function represents the cumulative volume of water a given well field produces, when their optimal rates are applied for all stress periods. The objective function of the new well-field configuration, W, is calculated and if the result is strictly greater than the best solution found so far $\big(\mathbf{O}(W) > \mathbf{O}(W_{Best})\big)$, then W is saved as the new best solution, W_{Best}.

Step 6: *Check if the stopping criterion is met*

Steps 2 to 5 are repeated for a set number of iterations, L. However, if computational power is not a limitation, then L should be set to the maximum value, $L_{Convergence}$. $L_{Convergence}$ is the number of EO-WPP iterations such that the performance of W_{Best} does not increase with iteration numbers greater than $L_{Convergence}$:

$$L = \left\{ 0 \leq L \leq L_{Convergence} : W_{Best_{L_{Convergence}}} = W_{Best_{L_{Convergence}+k}} \ \forall k \in \mathbb{N} \right\} \tag{14}$$

After performing L iterations, the algorithm simply reports the best solution found, W_{Best}, as the final result.

Figure 1 shows the flowchart of EO-WPP method. Its algorithm is shown on Algorithm 1.

Algorithm 1: Extremal Optimization for Well Placement Problems (EO-WPP)

begin
- Let: $L =$ Total number of iterations
- Let: $l =$ Current iteration of the algorithm
- Let: $W =$ The solution matrix (the set of all well locations)
- Let: $\vec{w}_i =$ The location of the ith well, $\vec{w}_i \in W$)
- Let: $q_i =$ The fitness of the ith well of solution W, $q_i \in Q$
- Let: $\mathbf{O}(W) =$ The objective function evaluated for solution W
- Let: $\mathbf{F}(W) =$ The fitness function evaluated for solution W
- Let: $\mathbf{C}(W) =$ The constraint function evaluated for solution W
- Let: $W_{Best} = \left\{ W_{Best} : O(W_{Best}) \geq O(W_l) \ \forall l \in \{0, 1, 2, \cdots, L\} \right\}$, i.e. the best solution found
- Set: $L = \left\{ 0 \leq L \leq L_{Convergence} : W_{Best_{L_{Convergence}}} = W_{Best_{L_{Convergence}+k}} \ \forall k \in \mathbb{N} \right\}$
- Set: $l = 0$
- Set: $W =$ Random initial configuration such that $\mathbf{C}(W) = \mathit{True}$
- Set: $W_{Best} = W$
- **while** $l \leq L$ **do**
  - Set: $l = l + 1$
  - Calculate: $Q = \mathbf{F}(W)$
  - Find: $\vec{w}_{worst} = \{\vec{w}_{i_{worst}} \in W : q_{i_{worst}} \leq q_i \ \forall q_i \in Q\}$
  - Find: $\vec{w}_{best} = \{\vec{w}_{i_{best}} \in W : q_{i_{best}} \geq q_i \ \forall q_i \in Q\}$
  - Let: $W' = \{\vec{w} \in W : \vec{w}_{worst} \notin W'\}$, i.e. remove $\vec{w}_{worst}$ from the solution
  - Let: $d_{max} =$ Maximum Euclidean distance between any two wells within W'
  - Let: $\vec{u} =$ Random vector with a length within $(0, 1]$ and the same dimensions as $\vec{w}$
  - Let: $\vec{w}_{new} = \vec{w}_{best} + d_{max}\vec{u}$
  - Let: $W'' = \{\vec{w} : (\vec{w} \in W') \ or \ (\vec{w} = \vec{w}_{new})\}$, i.e. add $\vec{w}_{new}$ to the solution
  - **while** $\mathbf{C}(W'') = \mathit{False}$ **do**
    - Create new: $\vec{u}$
    - Recalculate: $\vec{w}_{new} = \vec{w}_{best} + d_{max}\vec{u}$
    - Recalculate: $W'' = \{\vec{w} : (\vec{w} \in W') \ or \ (\vec{w} = \vec{w}_{new})\}$
  - Accept $W = W''$ unconditionally
  - **if** $\mathbf{O}(W) > \mathbf{O}(W_{Best})$ **then**
    - Set: $W_{Best} = W$
- **return** W_{Best}

2.4. Groundwater Management Program (GWM)

GWM is a Groundwater Management Process optimization program and its purpose is to determine the pumping rates which maximizes the overall output of a given well field while respecting a set of constraints. The objective function maximized by GWM can be described as (Ahlfeld et al., 2005):

$$\sum_{n=1}^{N}\beta_n Qw_n T_{Qw_n} + \sum_{m=1}^{M}\gamma_m Ex_m T_{Ex_m} + \sum_{l=1}^{L}\kappa_l I_l \tag{15}$$

where:

- β_n is the cost or benefit per unit volume of water withdrawn or injected at well site n;
- γ_m is the cost or benefit per unit volume of water imported or exported at external site m;
- κ_l is the unit cost or benefit associated with the binary variable I_l;
- Qw_n is the withdrawal or injection rate at well site n;
- Ex_m is the import or export rate at external site m;
- I_l is a binary variable at site l. $I_l = 1$ *or* 0;
- T_{Qw_n} is the total duration of flow at well site n;
- T_{Ex_m} is the total duration of flow at external site m;
- N, M, L are the total number of flow-rate, external, and binary decision variables;

Note that the objective function is composed of a summation term for the wells, a term for any external sources, and a term for any external sources with a binary attribute. For this work, only the summation term was used and the other two were disregarded (set to zero). This was done to simplify synthetic examples during testing. However, EO-WPP can operate with the entire objective function. To modify the objective function to give the cumulative water output, let β_n, $gamma_m$, and $\kappa_l = 1$.

If the optimization problem is nonlinear, then GWM uses a technique called using Sequential Linear Programming (SLP) to maximize the objective function (Ahlfeld et al., 2005). SLP works by calculating the response matrix, and then using this matrix and the list of constraints to calculate how to adjust the parameters (such as pumping rates) to maximize the objective function. The response matrix, also termed the Jacobian matrix, is a matrix of partial derivatives of the objective function with respect to each of the parameters of interest. The elements of the response matrix are calculated by the finite-difference perturbation method. For an optimization problem with N parameters to adjust, the objective function (and so the groundwater model) runs N+1 iterations every time the response matrix is calculated. For linear optimization problems, the response matrix only needs to be calculated once. Unfortunately, most groundwater models contain rivers or other head-dependent boundaries, thereby making these optimization problems nonlinear. With nonlinear optimization problems, a new response matrix is calculated every time the parameters are adjusted. Compared to linear optimization problems, the need for repeated calculations of the response matrix makes optimizing nonlinear problems a computationally expensive process.

3. Demonstration of EO-WPP

3.1. Case 1: Simple Geometric Problem

To examine the spatial behaviors of EO-WPP, the algorithm was first tested on an optimization problem with simple geometry. The optimal solutions for these problems are simple and known, so these problems can give insight into how EO-WPP converges toward a solution.

3.1.1. Set-up of Problem

One of the geometry problems is a point target problem. Given a set of points randomly placed on a 2D plane, the goal of EO-WPP is to adjust the position of the points to be as close to the origin point as possible. The fitness function used by EO-WPP is just the distance from the point to the origin:

$$Fitness\ of\ \vec{w}_i = ||\vec{w}_i||_{L_2} = \sqrt[2]{x_i^2 + y_i^2} \tag{16}$$

Unlike the well placement problem, the goal for this optimization problem is to find a solution that minimizes the objective function. Simply multiplying the fitness function by negative one converts the minimization problem into a maximization problem. Otherwise, all other mechanisms of the algorithm remain the same. With every iteration of EO-WPP, points that are farthest from the origin have the lowest fitness and so will be removed. A removed point will be replaced by a point that is randomly placed near the point of highest fitness, which is the point that is closest to the origin. Points are free to be placed anywhere within the bounds of the domain. The location of these points are defined on a continuous 2D Cartesian plane that extends from -100 to 100 in both the x and y axis.

The parameter I, the number of points, was set to three, six, and twelve points during testing to observe how EO-WPP would respond with increasing numbers of points. Ten runs for each set of points were performed and the average performances with each set of runs were calculated and compared. Performance of the overall solution was measured by the average distance between the points and the origin.

To test for how the heuristic for placing a new well affects the performance of EO-WPP algorithm, a comparison of three different placement heuristics was performed with the simple geometric problem used as a benchmark. The first heuristic randomly places the new well anywhere within the domain. The second heuristic places the new well within a circle centered around the best well. The radius of this circle (the placement radius) is set to the distance between two different and randomly chosen wells. The third heuristic is similar to the second heuristic except that it sets the placement radius equal to the maximum distance between any two wells. This is also the heuristic used by the proposed EO-WPP algorithm. For each heuristic, 100 runs were performed, with each run consisting of 300 iterations of the EO-WPP algorithm. Each run was initialized with a random starting positions for the wells. The number of wells was set to six for all tests.

The EO-WPP algorithm was also compared against particle swarm optimization (PSO) and the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm. PSO was selected because it is a popular global search optimization algorithm. Likewise, the BFGS was also selected because it was a common local search algorithm. By comparing EO-WPP to PSO and BFGS, EO-WPP's performance can be compared to different modes of optimization. For each method, 100 runs were performed, with the goal of optimally placing six wells. The number of times for evaluating the simple geometric objective function was recorded to allow proper comparison among the three optimization methods.

3.1.2. Results

Figure 2 displays the results for running EO-WPP on the point target problem with various numbers of points, I. The results show that the performance of the EO-WPP algorithm is partially sensitive to the number of points to optimize for. For all values of I, the algorithm converged toward a solution that minimized the objective function. On average, EO-WPP quickly generated a solution with the lowest objective function value when $I = 6$. For values larger than $I = 6$, the algorithm took longer to converge toward a solution because each iteration of EO-WPP can only move one point. With larger numbers of points, more iterations are needed to adjust the entire set of points. For values smaller than $I = 6$, EO WPP initially outperformed the $I = 6$ curve. However, around 10 iterations, the $I = 3$ curve changes into slower rate, thereby losing to the $I = 6$ curve by iteration 20. This change of EO-WPP's performance for small point numbers was from premature convergence.

Figure 3 displays the results of the three different new-well placement heuristics. In the figure, the mode total fitness value (objective function value) is plotted against the number of iterations of the EO-WPP algorithm. The results show that among the three heuristics, the best heuristic is to set the placement radius equal to the maximum distance between any two wells within the well field. This is the same heuristic used by the proposed EO-WPP algorithm (Algorithm 1). For the heuristic of randomly placing the well within the domain, the algorithm converges slower than the other two methods. For the heuristic where the placement radius was set to the distant of two distinct and randomly chosen wells, it initially converged the fastest, but the algorithm plateaus and fails to converge any further after 25 evaluations.

Figure 4 displays the results for comparing EO-WPP to the PSO and BFGS optimization algorithms. The mode objective function value is plotted against the number of times the objective function was evaluated. The results indicate that EO-WPP performs better than both the PSO and BFGS algorithms, with EO-WPP achieving near full convergence after just 60 evaluations of the objective function. BFGS then follows up as the second best performer, leaving PSO as the slowest algorithm for this benchmark.

3.2. Case 2: Synthetic Groundwater Model

To test how the EO-WPP algorithm would perform on optimization problems with a groundwater model, a synthetic groundwater model was constructed. The synthetic example was built and based on the benchmark example provided by Ahlfeld et al. (2005) in the paper that was used to verify the GWM optimization algorithm.

3.2.1. Set-up of Synthetic Model

The modeling domain was one layer discretized by 25 by 30 grid of cells. All cells were squares and have a side length of 200 ft. The model was bounded by constant heads that varied from 86 to 100 ft at the top and bottom of the model with no-flow boundary conditions to the left and right. In the middle of the model was a river, composed of three stream segments, with flow from left to right. All stream segments were 20 ft wide and had a stream bed conductance of 20,000 ft^2/day. The main stream had a slope of 0.0025, whereas the tributary stream had a slope of 0.0010. Figure 5 shows details for the modeling domain. To test how EO-WPP handles constraints, four conditions for streamflow depletion were placed along the river. The streamflow depletion constraints were defined as such (Ahlfeld et al., 2005):

$$Qsd_r = (Qsf_r)^0 - Qsf_r \tag{17}$$

$$Qsd_r \leq Qsd_r^u \tag{18}$$

Streamflow depletion, Qsd_r, is defined as the difference between the initial streamflow at stream location r at the end of the stress period, $(Qsf_r)^0$, and the streamflow calculated at the location at the end of the stress period after implementation of the optimal pumping strategy, Qsf_r. The upper bound streamflow depletion constraint values, Qsd_r^u, and the times when the constraints are enforced were different for each site. This was done to test how EO-WPP handles constraint complexity across different stress periods. The transmissivity for the model was set to the synthetic heterogenous field shown on Figure 6.

For simplicity of analysis, the transmissivity was set to either 50 or 500 ft^2/day. Transmissivity was 500 ft^2/day across most of the model except for three regions of low transmissivity. The first region was at the top and bottom of the model where the constant-head boundaries were located. Any optimization algorithm can "cheat" in maximizing the objective function by pumping near constant-head boundaries (where nearly infinite flow is possible with little or no change of hydraulic head). To deter this behavior, low-transmissivity cells were placed near the boundaries to prevent EO-WPP from taking advantage of this edge-effect. The second region of low transmissivity was in the left-middle section of the model domain. This was done to see how EO-WPP would handle a situation in which a large region of the model would be a non-ideal area to place wells. The hope was that after placing a well in this region, the algorithm would quickly learn to avoid the area. The third low-transmissivity region was in the lower-right section of the model. Prior tests with this model have shown that the best place to put the wells was at the bottom right side of the model. By placing a region of low transmissivity in the same area, EO-WPP was forced to find a well-field solution that somehow navigated around this low-transmissivity region. The groundwater model simulated a three-year period, divided into 12 stress periods (one stress period for each season). The aquifer had a homogenous recharge at a rate of 0.005 ft/day in the winter, 0.002 ft/day in the spring, 0 ft/day in the summer, and 0.001 ft/day in the fall.

The goal for EO-WPP was to determine the best locations to place four wells. The wells ran at a single pumping rate for the entire three-year period. The pumping rates for the wells could vary between zero and 50,000 ft^3/day. The drawdown limit for all wells was set to 10 ft. The task of GWM was to determine the optimal rates that maximized the cumulative output of the field for a given well field configuration while respecting all constraints. For the tests, EO-WPP was given 128 iterations to find an optimal solution. The entire EO-WPP process was restarted 128 times with a random initial well-field solution each time. This was done to determine EO-WPP's average performance. The performance of the EO-WPP algorithm was measured by using the cumulative output of the optimal field. The unit and absolute value of the cumulative output was not important because these values were only compared to each other. Therefore, the cumulative output could be treated as the total fitness of a solution.

3.2.2. Results

A sample of a well field solution generated by EO-WPP is shown on Figure 7. The results of the test show that EO-WPP can converge toward optimal well solutions. On average, the well-field solutions involved wells that were placed close to the river (Figure 5). This is reasonable because of the high conductivity the river offered. EO-WPP often placed the wells on the bottom-right side of the model domain and next to the southeastern stream because the water in the model was flowing into that region. While converging, EO-WPP was able to generate well-field solutions that avoided the low transmissivity regions. This shows that relying on a global constant-drawdown constraint works as a method for EO-WPP to identify regions of low productivity. Many of the solutions also had wells that were placed far from streamflow constraint sites. Therefore, the EO-WPP algorithm generated well-field solutions that took constraint sites into consideration. For EO-WPP's overall performance, Figure 8 shows the statistics computed for all 128 runs. Case 2 results shows that the EO-WPP functions, and that it can statistically perform better than the best-out-of-N algorithm. In other words, on average it is computationally more efficient to run the EO-WPP algorithm N times than to randomly generate N well field solutions and report the best one. For this groundwater model, based on the 128 EO-WPP runs (Figure 8), a randomly generated well-field solution had a total fitness between 12,000 and 42,000 with a median of 30,000. With each iteration, the entire distribution of the solution fitness improves. By the 30^{th} iteration, the median solution fitness matched and exceeded the maximum fitness of the zeroth iteration of the solution. That means for this groundwater model, there is a 50% chance that running the EO-WPP algorithm for 30 iterations will yield a well-field solution that is better than what could be achieved by randomly placing wells in the model. This method of comparing with the best out of N algorithm is a valid technique that has been performed by other groups such as Feng et al. (2012). With each EO-WPP iteration, the groundwater model was evaluated 15 times. So, after 30 iterations the model ran a total of 450 times. Note that the number of times for evaluating the groundwater model is dependent on the optimization function used by the fitness function.

3.3. Case 3: Aberdeen Groundwater Model

After developing and testing EO-WPP with the synthetic example, the EO-WPP algorithm was applied to the Aberdeen aquifer, in South Dakota (for model details, see Valder et al. (2018)).

3.3.1. Set-up of Aberdeen Model

The City of Aberdeen is in Brown County in the northeastern part of South Dakota. The study area encompassed 490 mi^2 north of Aberdeen in the James River Lowland and Lake Dakota Plain physiographic provinces (Figure 9). The study area included the glacial aquifer system north of Aberdeen between Foot Creek and the James River, because that area supports the City's current municipal well field. Currently, most of the city's water is supplied from the Elm River. When the streamflow of the river becomes too low, water is pumped from a well field seven miles north of Aberdeen. These wells were completed in the Elm aquifer, a shallow alluvial aquifer system in hydraulic connection with the Elm River. Ideally, the EO-WPP algorithm paired with the Aberdeen groundwater model can provide insight on where to place new wells to efficiently use the Elm aquifer.

The Aberdeen groundwater model was presented in Valder et al. (2018). The Aberdeen model consisted of seven layers. Three layers were for the Elm aquifer, the Middle James aquifer and the Deep James aquifer, and the remaining four layers were confining layers that bound the three aquifers. The Elm aquifer (Layer 2 from the top) is of interest because it is the shallowest and most accessible aquifer. The average thickness of the Elm aquifer is 24 ft and the average depth to the aquifer is 30 ft. The model was discretized into a finite-difference grid consisting of 368 rows and 410 columns with a cell size of 200 by 200 ft. The model was bounded by recharge, river, drainage, and well boundary conditions. The model contained 99 stress periods that simulates the years 1975 to 2015. The revised model used the USGS finite-difference groundwater-flow model MODFLOW-NWT to calculate all water budgets and flows. Additional details for the model are in the report by Valder et al. (2018).

3.3.2. EO-WPP for the Aberdeen Model

The goal for EO-WPP was to determine the best way to place six wells. The number of wells used was inspired by the results of Case 2 (Figure 8). In the model, these wells ran at a constant pumping rate for one year (October 1974 to October 1975). All pumping wells were installed in the Elm aquifer (Layer 2) and all wells were subject to a drawdown constraint of 10 ft. To prevent EO-WPP from exploiting boundary conditions, a distant constraint was defined such that all wells were at least 600 feet away from rivers, boundaries, and each other. To deter "cheating," wells also were forced to be placed in a bounded region within the model domain. For the first optimization run of EO-WPP, the well locations are bounded by $10 \leq Row \leq 300$ and $100 \leq Column \leq 300$. For the remaining runs, the extent of the bounding region was reduced to $100 \leq Row \leq 300$. Four runs of the EO-WPP algorithm were performed. Each run involved initializing the solution with six randomly placed wells then iteratively improving the solution by applying EO-WPP for 100 iterations. Each run took approximately two days to complete when performed on a single Intel Core i7-6600U CPU running at 2.8GHz.

With each iteration, the majority of the time was spent on calculating the fitness function. The fitness function requires the optimal pumping rates for a given well-field solution. These pumping rates were determined with GWM, which was set to solve for the optimal pumping rates using SLP. To reduce computation time, the convergence criterion used by SLP was adjusted such that the SLP loop terminates early. Although this reduced the accuracy of the optimal values, it does not significantly affect the performance of the fitness function. The main purpose of the fitness function was to identify the well that will likely produce the least amount of water. Therefore, an approximation of the optimal pumping rates is enough. This method is similar to how τ-EO operates. Introduced by Boettcher and Percus (1999), τ-EO is a version of EO that randomly removes one of the low fitness components, instead of strictly removing the component of lowest fitness. This allows τ-EO to behave like a global search algorithm and avoid local minimums. By using approximately optimal pumping rates, EO-WPP exhibits the same behavior as τ-EO

3.3.3. Results of Aberdeen Model

Results of the four runs show that EO-WPP was able to optimize the well field and converge toward a solution. For all runs, EO-WPP was able to perform at least 90% of optimization progress within 50 iterations. The remaining iterations were spent on refining the solution. This agreed with results found with the synthetic examples in Case 2. An example of EO-WPP's optimization progress during a run is shown on Figure 10. With each iteration, the fitness of the best solution steadily increased, yet the fitness of the current solution either increased or decreased with each iteration. In Figure 10 during iteration 40 to 60, the fitness of the current solution dropped significantly before later recovering. This behavior was expected because removing the worst well and replacing it with a randomly placed new well did not guarantee an improvement of the total field output. Even without this guarantee, the fitness of the current solution still generally increased with increasing number of iterations. This indicates that the heuristic of strictly modifying the worst performing well allowed EO-WPP to generate new-well field solutions that were more likely to be better than previous solutions.

For all runs, the EO-WPP algorithm placed wells in locations that seemed to correlate with the horizontal hydraulic conductivity of the layer the wells were pumping from (Layer 2). The well-field solutions and the horizontal hydraulic conductivity are shown on Figure 11. In the first run, EO-WPP placed some wells close to the top boundary (Figure 11a). To ensure that EO-WPP was not taking advantage of boundary conditions, the remaining runs had the bounding region adjusted so that wells were placed below row 100. The effects of adjusting the bounding region affected the total output of the well field. Before the adjustment, the maximum well field output was $2.6 \times 10^8\ ft^3/yr$. After the adjustment, the well field output was reduced to a maximum of $1.3 \times 10^8\ ft^3/yr$.

Regardless of the bounding region, EO-WPP consistently placed a majority of the wells near or upon sites with high horizontal hydraulic conductivity. Recall that EO-WPP only uses pumping rates and drawdown at the wells. The algorithm does not use explicit knowledge of hydraulic conductivity. Yet for the Aberdeen groundwater model, the well-field solutions appear to correlate best with the horizontal hydraulic conductivity. This indicates that the horizontal hydraulic conductivity plays a crucial role when determining optimal well-field configurations. Well locations that deviate from peak horizontal hydraulic conductivity were caused by EO-WPP's consideration of other factors such as recharge, aquifer thickness, or vertical hydraulic conductivity.

4. Discussion

Within the EO-WPP algorithm, the placement of the new well is dependent on the location of the best well. This was done to introduce a clustering behavior into the EO-WPP algorithm. Though it seems like this placement heuristic may cause the EO-WPP algorithm to get stuck at local minimums, our results show that by abiding to certain guidelines, this can be avoided. For example, Figure 2 shows that with a low number of points, EO-WPP is more likely to display behavior that causes stagnation at local minimums. In Figure 2, this premature convergence behavior can be seen in the curve for $I = 3$. Note that by 10 iterations, the slope of the curve changes significantly. Yet for the other two curves, this change of slope does not exist. This is because with a low number of points, it becomes more likely for the points to become too close to each other and cause premature convergence. For a sufficiently large number of points, this behavior disappears. Based on these results, there must be at least six points to ensure EO-WPP does not exhibit this behavior.

The EO-WPP algorithm places the new well within a certain distance from the best well. This distance, called the placement radius, is set to be the maximum distance between any two wells within the well field. The results on Figure 3 show that this placement heuristic is ideal for the EO-WPP algorithm. If the placement radius was set too small, such as the distance between two random and distinct wells, then the clustering behavior becomes too strong and causes EO-WPP to converge prematurely. In Figure 3, this shows as an early plateau in the performance curve. If the placement radius was set too large, such as randomly placing the new well anywhere in the model domain, then EO-WPP converges too slowly towards the solution.

EO-WPP's placement heuristic introduces a clustering behavior that can be sensitive to the configuration of the initial solution. To ensure the initial solution does not have an influence in the shape of the final solution, EO-WPP must iterate a larger number of times. With a large number of iterations, EO-WPP's stochastic mechanisms allow the algorithm to properly explore the search space before converging towards a set of solutions. This was shown to be true in the results for the synthetic model (Figure 7) and the Aberdeen model (Figure 11). For both cases, the EO-WPP algorithm generated very similar solutions, even when going through the EO-WPP algorithm with 100 different, randomly generated initial solutions. Tests show that as EO-WPP's performance reaches its stall limit, the solutions begin to look similar to each other. This makes sense since the number of possible well field configurations decreases as the performance of these solutions approach the global optimum value. To gain greater confidence in the stability of the solutions, multiple instances of EO-WPP can be ran, with the iteration process terminated once all instances generate the same solution.

EO-WPP is essentially a combination of mechanisms from both global and local search algorithms. EO-WPP relies on a population of wells, a technique similar to the population mechanisms used by global search algorithms. EO-WPP also operates on a single well field solution and modifies the solution based on the information gained by the solution's components. This mechanism is similar to how local search algorithms operates. EO-WPP combines these techniques in a way that allows it to avoid local minimums and quickly converge towards a solution. Figure 4 shows that at least for the simple geometric case, EO-WPP converges faster than the PSO global search algorithm and the BFGS local search algorithm.

Another advantage EO-WPP provides is its ability to find well field solutions with the wells close to each other. Figure 11 shows EO-WPP's clustering behavior found solutions where some wells are nearby each other (e.g. Run 2 and Run 3). This behavior is desirable for well field design since reducing distances between wells can reduce the amount of infrastructure needed to connect the wells together. What is interesting about this behavior is that it is not explicitly defined in the objective function or in the constraints. Instead, this spatial behavior emerges from the definition of the placement heuristic.

5. Conclusions

This paper introduced a novel well placement optimization algorithm, EO-WPP, which was inspired by the optimization algorithm called Extremal Optimization. EO-WPP works by removing the least productive wells and replacing them with new wells placed randomly near the most productive wells. By following this heuristic, EO-WPP can quickly generate well fields optimized for cumulative well-field output.

A simple geometric benchmark shows that EO-WPP was able to perform faster than common global search and local search methods. A synthetic groundwater model shows that with a large enough well count and number of iterations, EO-WPP was able to avoid local minimums and yield consistent well field solutions.

Results also verify that EO-WPP exhibits an emergent spatial behavior of clustering, a behavior that is useful during the design of optimal well fields. EO-WPP then was applied to the Aberdeen groundwater model. EO-WPP was able to generate multiple potential well field solutions that maximized total water discharge from the Elm aquifer while respecting drawdown and spatial constraints. The locations of the wells indicated that the horizontal hydraulic conductivity was an important factor when designing a well field for the region north of Aberdeen.

Although EO-WPP was applied only to a model built to help the City of Aberdeen, the methods introduced in this paper are applicable to groundwater management in general. EO-WPP can be used for designing well fields to use groundwater resources efficiently. Placement optimization problems extend beyond groundwater management, and the methods introduced by EO-WPP can be applied to other fields such as mining operations, petroleum production, groundwater monitoring, and more.

Acknowledgements. The authors acknowledges the financial support of the South Dakota Board of Regents through a Competitive Research Grant. This work also has been supported through a grant from the National Science Foundation (OIA-1833069). We are grateful to Dr. Arden Davis for his comments and edits. We thank the associate editor and three anonymous reviewers for their comments, which substantially improved the manuscript.

References

Ahlfeld, D.P., Barlow, P.M., Mulligan, A.E., 2005. GWM–A ground-water management process for the US Geological Survey modular ground-water model (MODFLOW-2000). US Department of the Interior, US Geological Survey.

Anderson, M.P., Woessner, W.W., Hunt, R.J., 2015. Applied groundwater modeling: simulation of flow and advective transport. Academic press.

Bayer, P., Duran, E., Baumann, R., Finkel, M., 2009. Optimized groundwater drawdown in a subsiding urban mining area. Journal of hydrology 365, 95–104.

Boettcher, S., 2005. Extremal optimization for sherrington-kirkpatrick spin glasses. The European Physical Journal B-Condensed Matter and Complex Systems 46, 501–505.

Boettcher, S., Percus, A.G., 1999. Extremal optimization: Methods derived from co-evolution, in: Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation-Volume 1, Morgan Kaufmann Publishers Inc.. pp. 825–832.

Boettcher, S., Percus, A.G., 2002. Optimization with extremal dynamics. complexity 8, 57–62.

Chen, J., Zeng, G.Q., Zhou, W., Du, W., Lu, K.D., 2018. Wind speed forecasting using nonlinear-learning ensemble of deep learning time series prediction and extremal optimization. Energy Conversion and Management 165, 681–695.

Chong, E.K., Zak, S.H., 2013. An introduction to optimization. volume 76. John Wiley & Sons.

De Falco, I., Laskowski, E., Olejnik, R., Scafuri, U., Tarantino, E., Tudruj, M., 2015. Extremal optimization applied to load balancing in execution of distributed programs. Applied Soft Computing 30, 501–513.

De Sousa, F.L., Vlassov, V., Ramos, F.M., 2004. Generalized extremal optimization: an application in heat pipe design. Applied Mathematical Modelling 28, 911–931.

Elçi, A., Ayvaz, M.T., 2014. Differential-evolution algorithm based optimization for the site selection of groundwater production wells with the consideration of the vulnerability concept. Journal of Hydrology 511, 736–749.

Emerick, A.A., Silva, E., Messer, B., Almeida, L.F., Szwarcman, D., Pacheco, M.A.C., Vellasco, M.M.B.R., et al., 2009. Well placement optimization using a genetic algorithm with nonlinear constraints, in: SPE reservoir simulation symposium, Society of Petroleum Engineers.

Feng, Q., Zhang, J., Zhang, X., Hu, A., 2012. Optimizing well placement in a coalbed methane reservoir using the particle swarm optimization algorithm. International Journal of Coal Geology 104, 34–45.

Guyaguler, B., Horne, R.N., et al., 2001. Uncertainty assessment of well placement optimization, in: SPE annual technical conference and exhibition, Society of Petroleum Engineers.

Humphries, T.D., Haynes, R.D., James, L.A., 2014. Simultaneous and sequential approaches to joint optimization of well placement and control. Computational Geosciences 18, 433–448.

Liu, D.C., Nocedal, J., 1989. On the limited memory bfgs method for large scale optimization. Mathematical programming 45, 503–528.

Mahinthakumar, G., Sayeed, M., 2005. Hybrid genetic algorithm—local search methods for solving groundwater source identification inverse problems. Journal of water resources planning and management 131, 45–57.

McDonald, M.G., Harbaugh, A.W., original authors of MODFLOW, 2003. The history of modflow. Groundwater 41, 280–283.

Minton, J.J., 2012. A comparison of common methods for optimal well placement. University of Auckland, research rep .

Nelder, J.A., Mead, R., 1965. A simplex method for function minimization. The computer journal 7, 308–313.

Nozohour-leilabady, B., Fazelabdolabadi, B., 2016. On the application of artificial bee colony (abc) algorithm for optimization of well placements in fractured reservoirs; efficiency comparison with the particle swarm optimization (pso) methodology. Petroleum 2, 79–89.

Nwankwor, E., Nagar, A.K., Reid, D., 2013. Hybrid differential evolution and particle swarm optimization for optimal well placement. Computational Geosciences 17, 249–268.

Park, C.H., Aral, M.M., 2004. Multi-objective optimization of pumping rates and well placement in coastal aquifers. Journal of Hydrology 290, 80–99.

Ruder, S., 2016. An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747
.

Sarma, P., Chen, W.H., et al., 2008. Efficient well placement optimization with gradient-based algorithms and adjoint models, in: Intelligent energy conference and exhibition, Society of Petroleum Engineers.

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

Valder, J.F., Eldridge, W.G., Davis, K.W., Medler, C.J., Koth, K.R., 2018. Revised groundwater-flow model of the glacial aquifer system north of Aberdeen, South Dakota, through water year 2015. Technical Report. US Geological Survey.

Wang, W., Ahlfeld, D.P., 1994. Optimal groundwater remediation with well location as a decision variable: Model development. Water Resources Research 30, 1605–1618.

Zeng, G.Q., Chen, J., Dai, Y.X., Li, L.M., Zheng, C.W., Chen, M.R., 2015. Design of fractional order pid controller for automatic regulator voltage system based on multi-objective extremal optimization. Neurocomputing 160, 173–184.

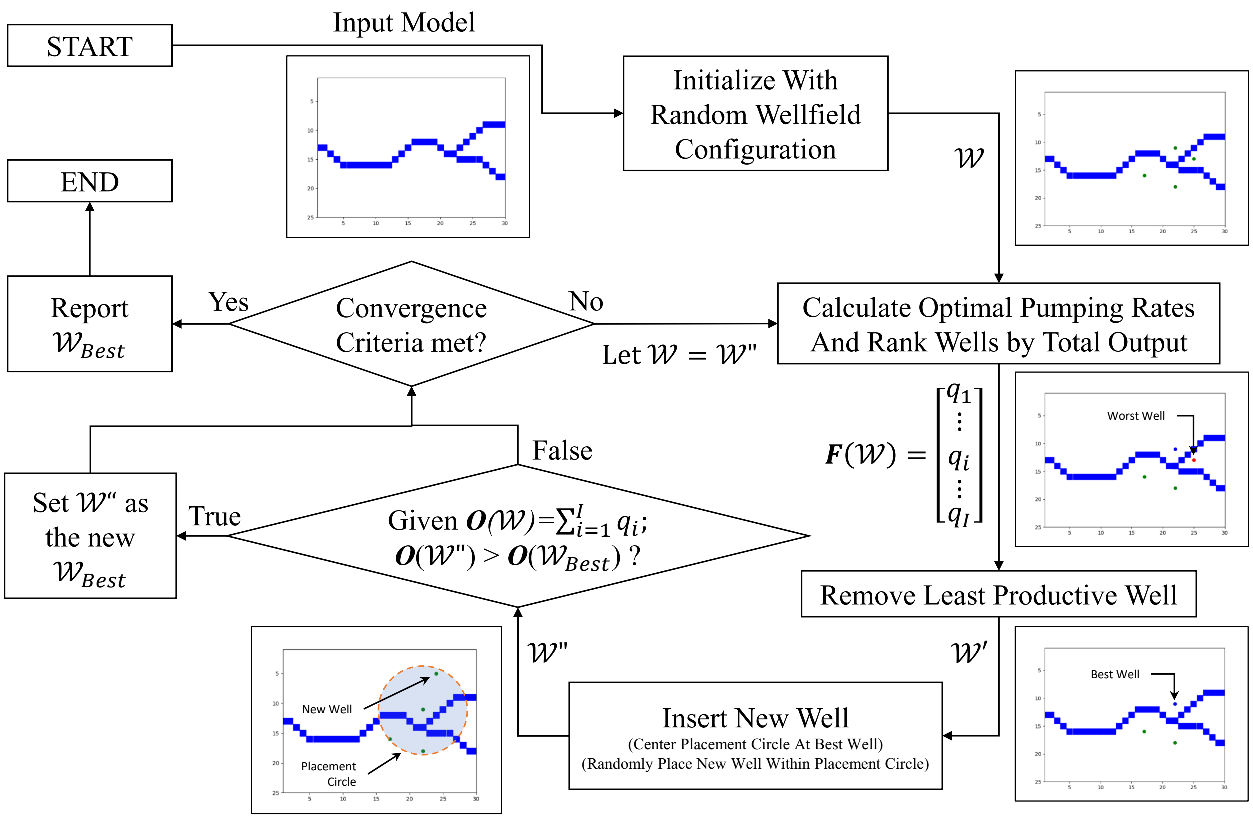


Figure 1: Flowchart of the EO-WPP algorithm.

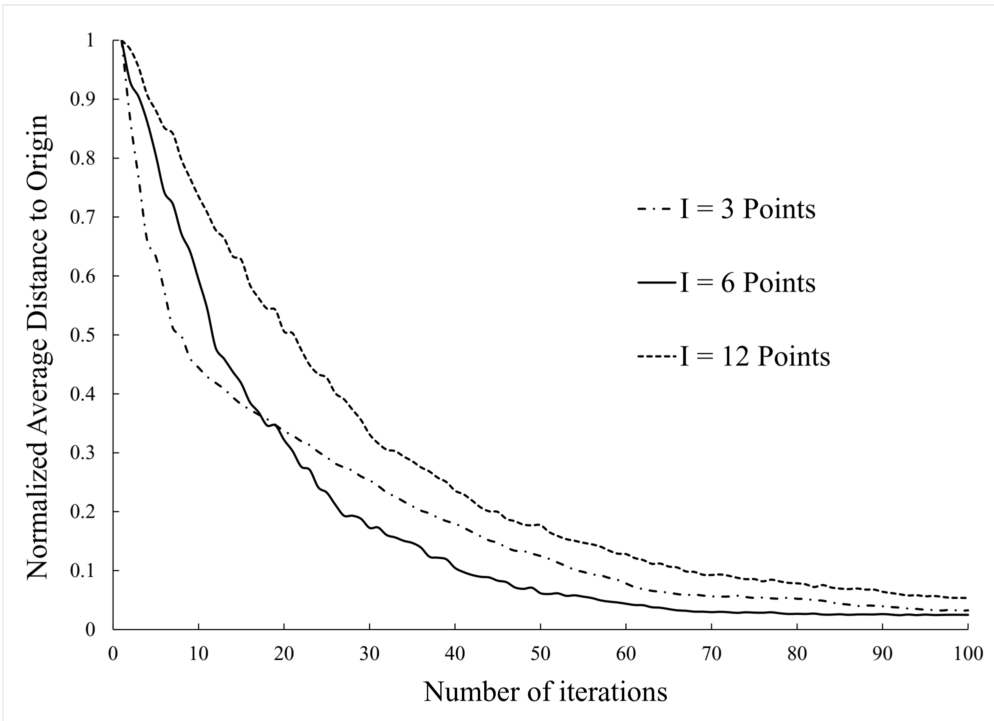


Figure 2: The average performances of the EO-WPP plotted for $I = 3$, 6, and 12. All curves were normalized by the initial values of the average distance, so all curves begin at 1.0 and approach zero as the EO-WPP minimizes the objective function. Note that EO-WPP achieved the lowest score when I = 6.

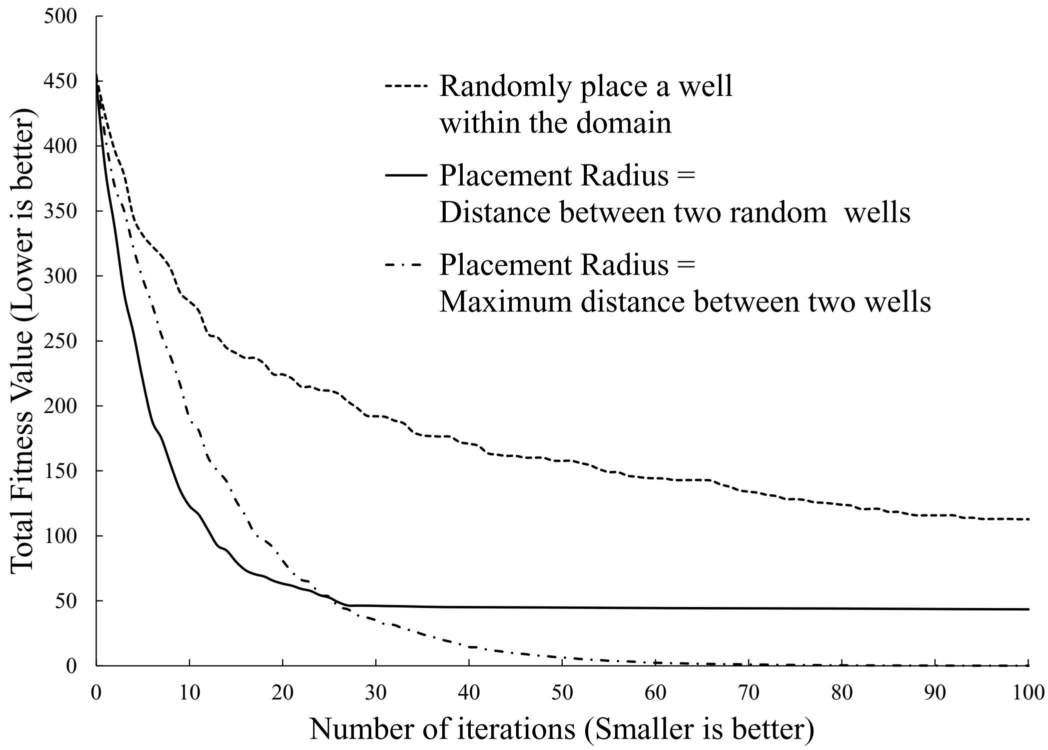


Figure 3: The average performance curve of the EO-WPP algorithm with various heuristics for placing a new well. Plotted is the mode objective function value plotted against the number of iterations of the EO-WPP algorithm. Note that the best performing heuristic is where the new well is placed within a circle centered at the best well with the radius of the circle (placement radius) is set to the maximum distance between any two wells. This is the heuristic used by the proposed EO-WPP algorithm.

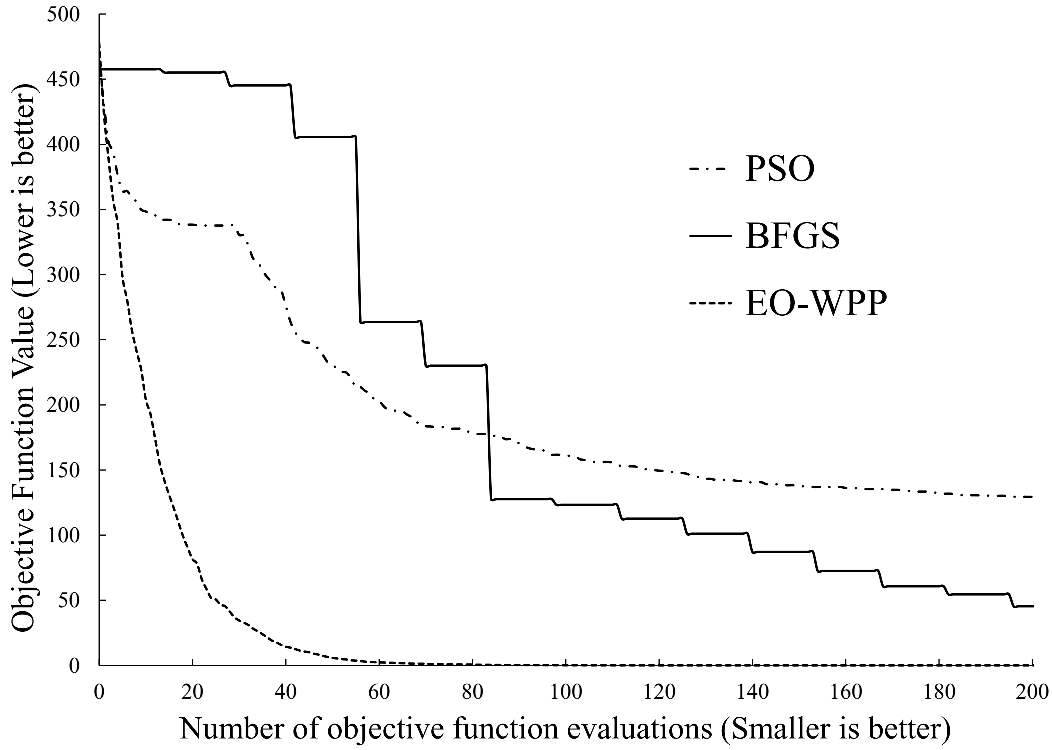


Figure 4: The average performance curve of three optimization algorithms on the simple geometric problem. The proposed method EO-WPP was compared against particle swarm optimization (PSO) and the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm. Plotted is the mode objective function value plotted against the number of times the objective function was evaluated. Note that EO-WPP converges onto the solution faster than PSO and BFGS for this benchmark.

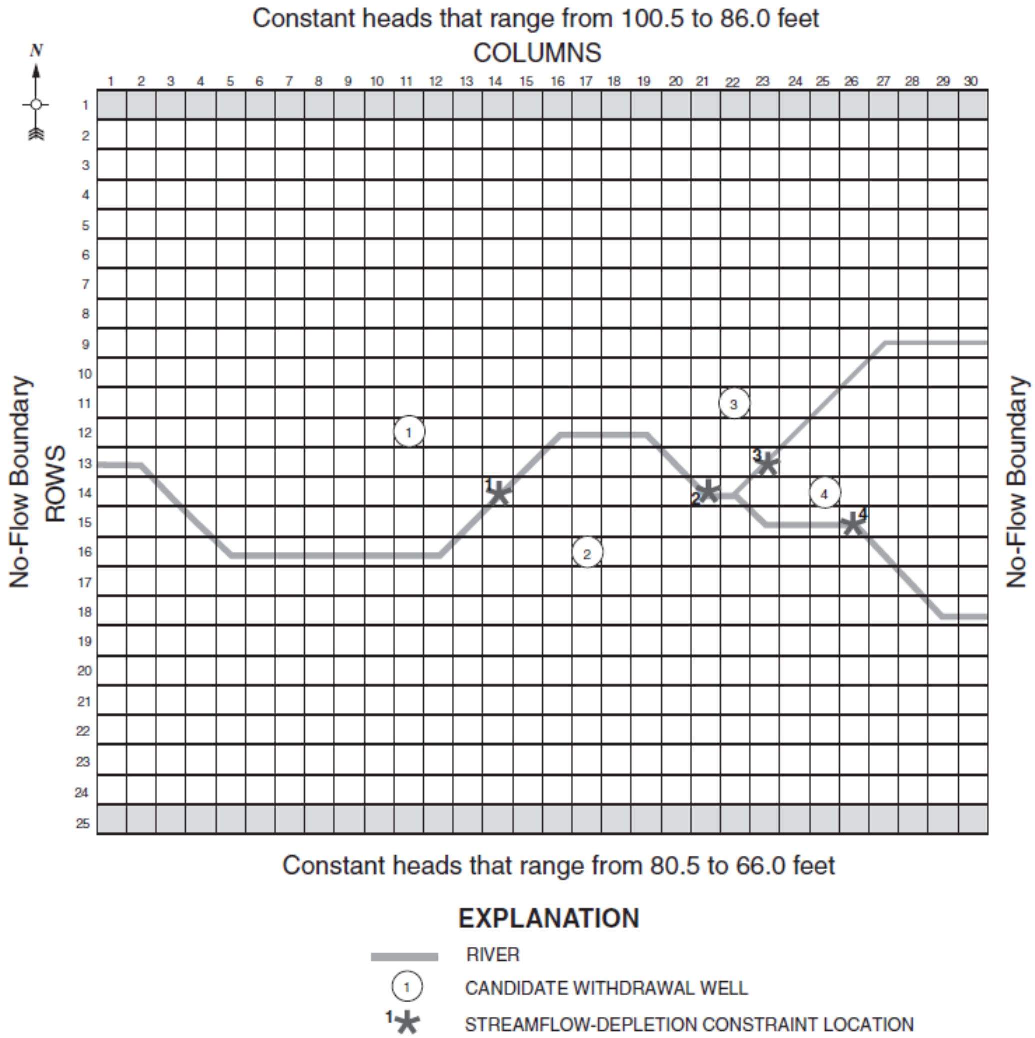


Figure 5: Diagram of the size and dimensions of the synthetic modeling domain. The domain was a 25 by 30 grid of square cells with a side length of 200 feet. The model was bounded by constant heads at the top and bottom of the model, with no-flow boundary conditions to the left and right. In middle of the model was a river with flow from left to right. Four constraints for stream-flow depletion were placed along the river. Marked locations for wells are from Ahlfeld et al. (2005), but were not used in this work. For the optimization problem, the locations of the wells will be constantly changing. This figure is from the SUPPLY example by Ahlfeld et al. (2005).

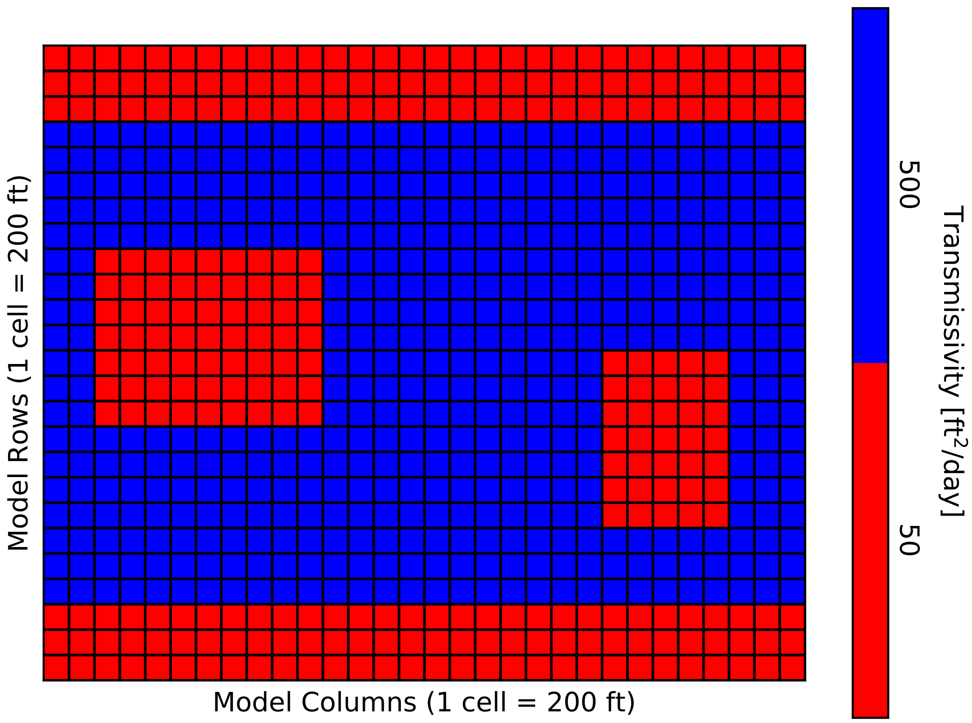


Figure 6: Transmissivity of all the cells of model. The orientation of the grid is the same as in Figure 5. Transmissivity is either 50 or 500 ft^2/day. There are four regions of low hydraulic conductivity. The first two region are the at top and bottom of the model where there are constant head boundaries. The third region is at the left-middle side of model, and the fourth region is at the lower-right side of the model.

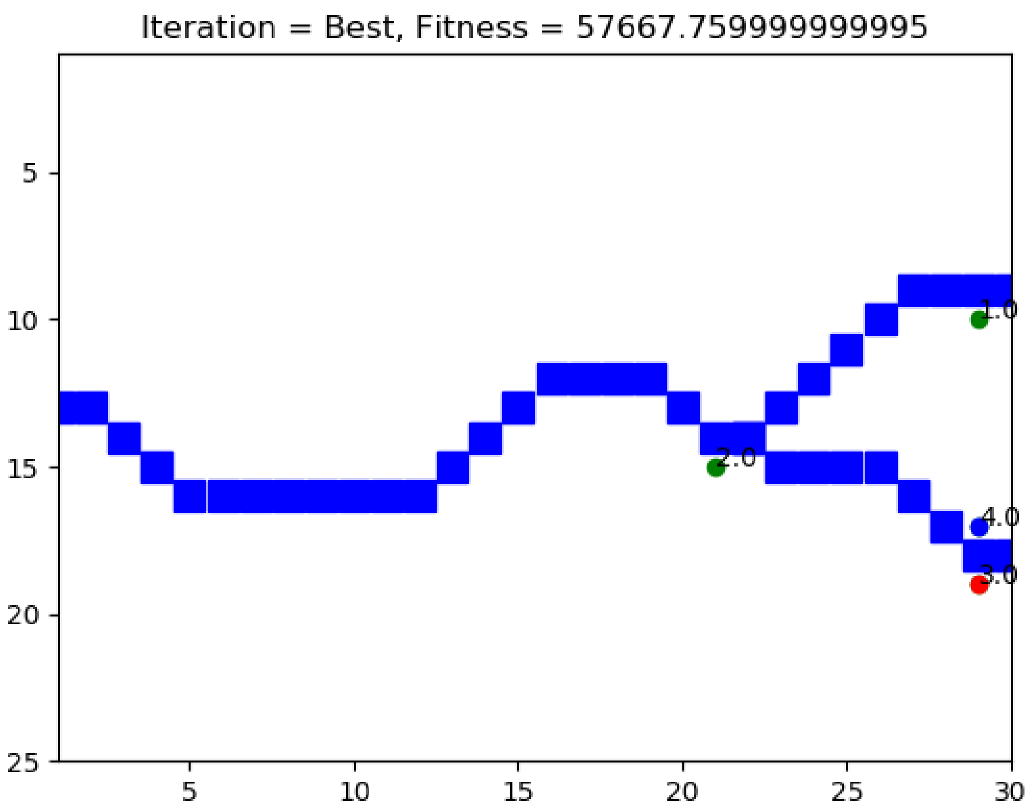


Figure 7: A well-field solution EO-WPP generated after running for 128 iterations. The blue squares indicate the river cells. The four circles indicate the location of the four wells. The wells are annotated with their index number. Their color indicates their rank of fitness. The blue circle is the well with the highest fitness, the red circle indicates the well with the lowest fitness, and the green circles indicate a fitness that is between the best and the worst. In this well field solution, well 3.0 has the lowest fitness and well 4.0 has the highest fitness. Figures 3 and 4 show the model set-up.

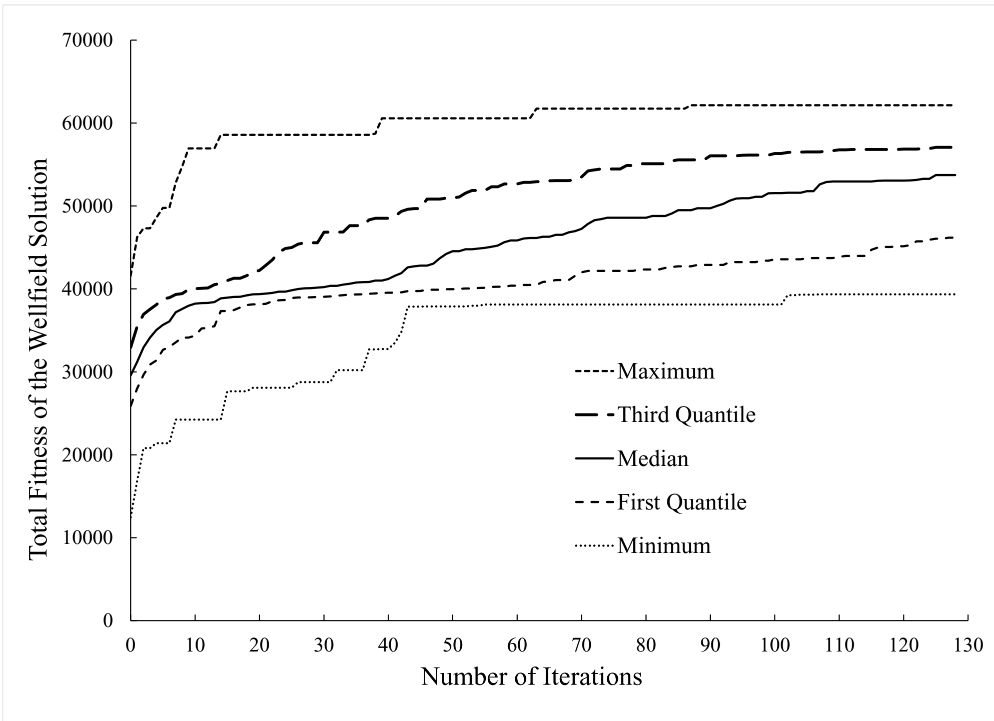


Figure 8: The total fitness of the well-field solution plotted against the iteration number. For all runs, the solution fitness increased with the number of iterations. With additional iterations, the rate of fitness improvement decreased because of EO-WPP convergene towards the optimal solution. Note that the median crosses the maximum fitness of the zeroth iteration by the 30^{th} iteration.

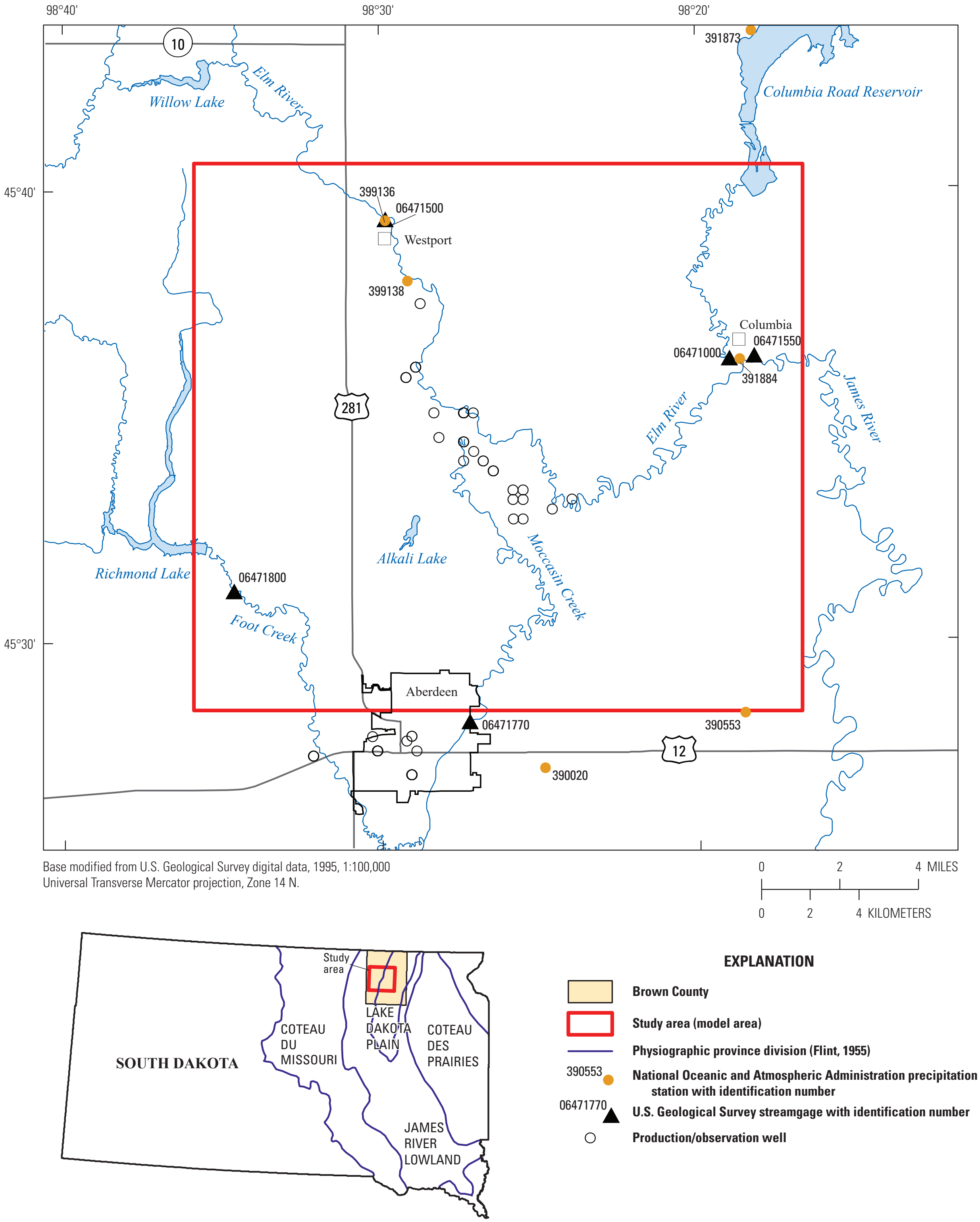


Figure 9: Locations of study area (model area), stream-gages, precipitation stations, and production/observation wells. Inset shows model area location in Brown County and physiographic provinces in eastern South Dakota (From Figure 1 of Valder et al. (2018)).

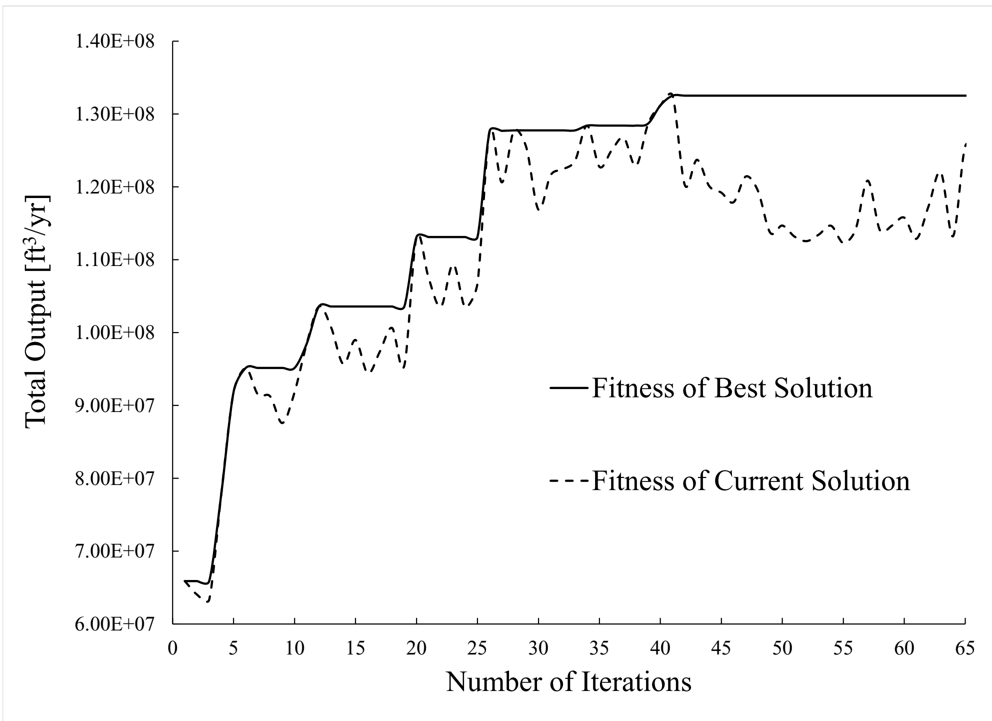
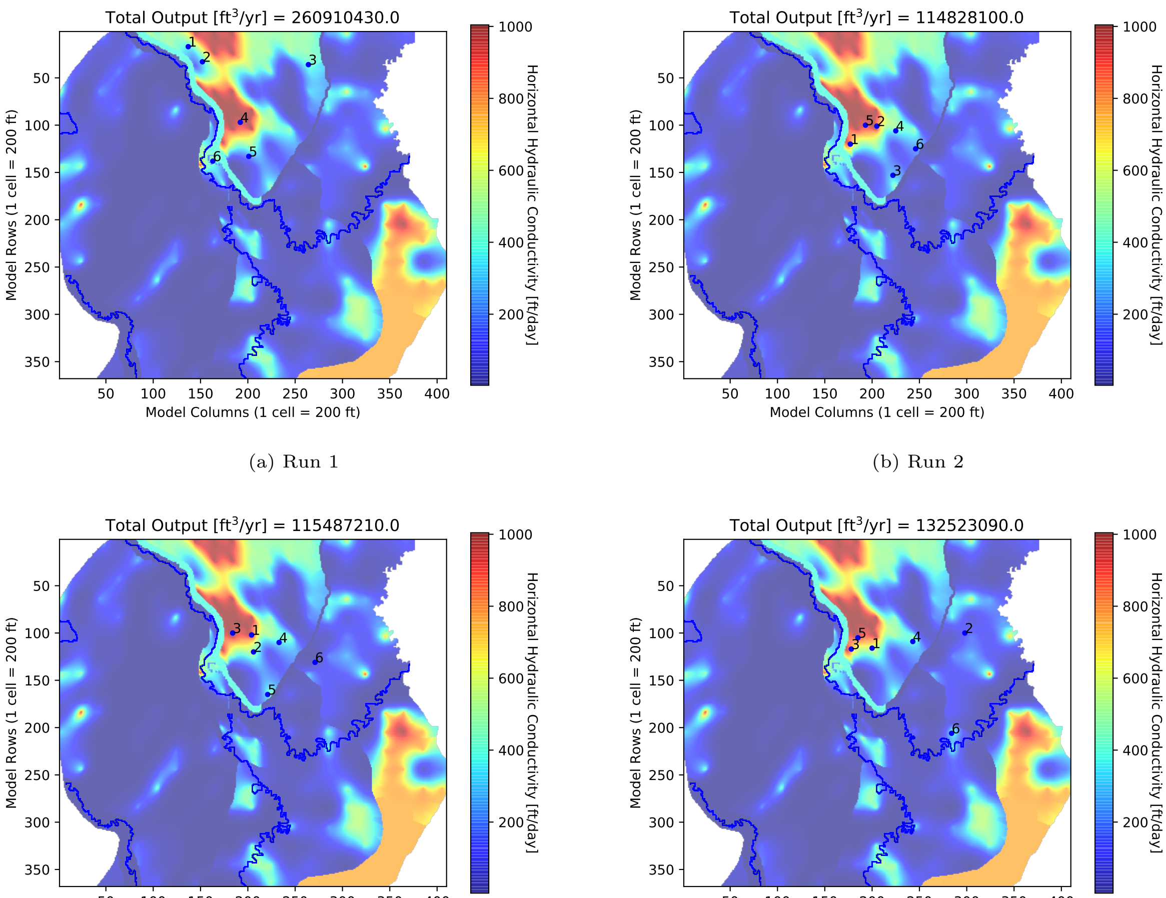


Figure 10: The total fitness (cumulative output) of the well-field solution plotted against the iteration number for Run 4 (Figure 11d). Plotted is the fitness of the best solution found (solid line), and the fitness of the current solution (dashed line), for a given iteration. Notice that the fitness of the current solution erratically increased.



(a) Run 1

(b) Run 2

(c) Run 3

(d) Run 4

Figure 11: The best well-field solutions from each of the EO-WPP runs plotted against the horizontal hydraulic conductivity for Layer 2. Wells are plotted with colored dots, where blue dots are the most productive wells, red dots are the least productive wells, and green dots show wells with intermediate performance. The wells also are annotated with their fitness rank, where ”1” indicates the most productive well and ”6” indicates the least productive well. Notice that EO-WPP places wells near or on sites with high horizontal hydraulic conductivity.