

Krityakierne, Tupaluck; Baowan, Duangkamon

Article

Aggregated GP-based optimization for contaminant source localization

Operations Research Perspectives

Provided in Cooperation with:

Elsevier

Suggested Citation: Krityakierne, Tupaluck; Baowan, Duangkamon (2020) : Aggregated GP-based optimization for contaminant source localization, Operations Research Perspectives, ISSN 2214-7160, Elsevier, Amsterdam, Vol. 7, pp. 1-10, <https://doi.org/10.1016/j.orp.2020.100151>

This Version is available at:

<https://hdl.handle.net/10419/246422>

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.



<https://creativecommons.org/licenses/by-nc-nd/4.0/>



Aggregated GP-based Optimization for Contaminant Source Localization

Tipaluck Krityakierne^{*,a,b}, Duangkamon Baowan^{a,b}

^a Department of Mathematics, Faculty of Science, Mahidol University, Bangkok 10400, Thailand

^b Centre of Excellence in Mathematics, CHE, Bangkok 10400, Thailand



ARTICLE INFO

Keywords:

Contaminant source localization
Groundwater management
Expected improvement
Nonlinear programming
Simulation-based optimization

ABSTRACT

Recently a new simulation-based optimization benchmark of groundwater contaminant source localization problems has been introduced to the hydrogeological science community. Given information on contaminant concentration levels at each monitoring well and each time step, its objective is to identify the location of contaminant source. In this work, we analyze and look at the problem from different angles to gain more insights on this class of groundwater problems. To tackle the problem, a novel simulation-based optimization algorithm relying on an aggregated Gaussian process model, and the expected improvement criterion is introduced. Results from this study show that the proposed algorithm, though relying on an approximated Gaussian process model, demonstrates superior efficiency and reliability than a traditional expected improvement-based algorithm. The location of the monitoring wells was confirmed to play a crucial role in assisting the optimization algorithm to accurately localize the contaminant source. Additional monitoring wells, while adding more knowledge of the space-time mapping of concentration levels, could nevertheless slow down convergence of the algorithm due to the increase in problem complexity.

1. Introduction

Simulation experiments have been used extensively to study and model many natural systems from physics, astronomy, chemistry, biology, engineering etc. Not only is the effect of different values of input variables in the system, but oftentimes, the interest is to find the optimal value for input variables in terms of experiment outcomes. One could, of course, run exhaustive simulation experiments for all possible input variables and pick the best one; however, this is not always a practical choice due to model complexity or expensiveness.

In general, the formulation of an optimization problem involves two main aspects (1) defining an appropriate objective function or a functional form which expresses the aim of optimization, and (2) choosing efficient optimization method depending upon the objective function and some restrictions (constraints). It is an undeniable fact that both the function representation as well as the choice of optimization algorithm play a key role in achieving accurate results for the problem under investigation [4,19,36].

Objective functions can usually be formulated in more than one way, all of which representing the same goal but resulting in different complexity of functional landscapes: large basins of equal function value, a rough or noisy landscape, a deceptive landscape or one that bears a region of highly unattractive solutions which surround the

promising ones. Different landscapes due to different objective function representations directly affect the performance of a chosen optimization algorithm, and a representation with simpler landscape will be more favourable, assisting the algorithm in rapidly converging to a solution. This is particularly important when the objective function evaluations are obtained from computationally intensive simulation (e.g. taking minutes or hours per simulation). In simulation-based optimization, the approach starts with finding a good relationship between input and response variables for each simulated points. The response surface model will then be used to find the best input variables that produce desired outcomes in terms of response variables [1,6,9,13,17,22].

In essence, a response surface (also known as a surrogate model) provides an approximation of the objective simulation. The values from response surface will be used as part of optimization algorithm in place of the expensive evaluations. This class of methods is often called surrogate-based optimization. Examples of response surfaces are linear, quadratic approximations, as well as linear combinations of radial basis functions (RBF). There are different choices of basis functions, e.g. Gaussian, thin-plate splines, cubic splines and multi-quadratics.

Common to all surrogate-based optimization methods is the concept of iteratively selecting new points for evaluation, and updating the surrogate model with information gained from new sampling points

* Corresponding author.

E-mail address: tipaluck.kri@mahidol.edu (T. Krityakierne).

during the course of iterations. The selection criteria of new points usually balance the trade-off between local and global search. During the exploitation phase, the algorithm will aim at refining those points within the neighborhood of previously visited locations in order to improve their solution quality. During the exploration phase, as an RBF response surface does not provide uncertainty estimation of the prediction, the optimization algorithm explores the search space by trying to avoid those cluster points, and sample new points far away from already evaluated points.

Examples of RBF-based optimization methods are the method in [12] relying on sampling criteria which selects the point to maximize the smoothness of the interpolation, using some external global optimization algorithm. Several works are based on the idea of balancing local and global searches by weighting the global minimization of the surrogate model against the distance from previously selected points (see e.g. [18,27,28]).

Bayesian optimization (BO) differentiates itself from other surrogate-based optimization by using the model developed under Bayesian inferences to decide on the locations to sample next. The most commonly used model in BO is Gaussian process regression (GPR) where the output of a deterministic function is treated as a realization of a Gaussian process. While RBF approximation does not provide uncertainty estimation of the prediction, GP model provides also a measure of uncertainty along with its predicted value. To balance between exploration and exploitation, BO uses the expected improvement (EI) sampling criteria, along with the posterior at the sampled point, as a criteria to choose the next evaluation point. By relaxing the assumption made in EI's derivation that only those points previously evaluated can be returned as final solution, the knowledge-gradient (KG) acquisition function was proposed ([11,38]). Unlike EI sampling criteria which only considers the posterior at the points sampled, the KG sampling criteria considers the posterior over the entire search domain, and how the sample will change that posterior.

It is undebatable that a well established estimation for probabilistic functions like GPR may not be applicable for problems with deterministic data of computer experiments. In particular, the problem known as fixed-domain estimation of deterministic models is typical for application of Bayesian optimization algorithms [39–42]. Many high fidelity simulations arising in industry, however, can be very expensive to run, in terms of time, money, or resource. By effectively using all valuable information from limited number of expensive computer simulations, a class of BO algorithms and related methods turned out to be parsimonious and highly effective to find a “near-optimal” solution by performing only very few expensive simulation model evaluations in order to keep the overall optimization time acceptable.

Over the last decade, surrogate response surface has been used for modelling and optimizing a variety of groundwater applications. For example, wastewater treatment processes [21], water resources applications [26], coastal aquifer pumping optimization problem [7], optimal well placement for groundwater remediation [32], to name a few. Recently benchmark problems for simulation-based optimization driven by groundwater contaminant source characteristics have been introduced in Pirot et al. [24]. The localization of groundwater contaminant sources, a typical application in the field of groundwater environmental monitoring, aims at finding an effective and accurate positioning method to locate the source of contamination. Using this benchmark, the goal of this paper is twofold. Firstly, due to the expensiveness and highly non-linear nature of the groundwater objective functions, a simulation-based Bayesian optimization algorithm relying on an aggregated Gaussian process [31] and the expected improvement criteria is developed. The results demonstrated the superiority of such approach on this class of groundwater problems over classical Bayesian Optimization algorithm [14]. Secondly, it may be logical to think that as the level of contaminant information available to the algorithm increases, the algorithm will solve the problem faster. However, this is not the case for this class of groundwater problems. Using algorithm's

solving effort to define a measure of problem difficulty, the obtained results indicated that incorporating information from too many monitoring wells into the objective function could nevertheless increase the number of iterations needed for an algorithm to solve the problem, which in turn slows down the convergence of the optimization algorithm to the source of contaminant.

The rest of this paper is organized as follows. Section 2 presents key concepts and a gentle introduction to the classical Efficient Global Optimization (EGO). A simulation-based optimization algorithm with an aggregated Gaussian process model (AEI) will also be discussed in this section. In Section 3, we introduce variants of the proposed AEI, which will be compared with the EGO on the contaminant source localization problems in Section 4. Numerical results as well as interesting findings on algorithm performance, effort and problem difficulty will also be touched upon in this section. Finally, conclusions will be given in Section 5.

2. Optimization Methodology

We consider a real-valued box-constrained optimization problem of the form

$$\min_{x \in \mathcal{D} \subset \mathbb{R}^d} f(x) \quad (1)$$

with unknown gradients. In particular, the objective function f , obtained from a simulation model, is blackbox and computationally expensive to simulate, in terms of money or time. The goal is to find the best possible function value for f when a limited number of function evaluations is allowed.

2.1. Gaussian Processes and Bayesian Optimization

Gaussian Process (GP) models, also known as kriging, have long been applied to solve engineering optimization problems such as model selection and hyperparameter optimization [23,34], and recently caught attention of hydrogeological science community due to model efficiency and flexibility [5,15,16,24]. The objective function is seen as a realization of a Gaussian process, employing the GP as a prior (in a Bayesian sense) over a function. Gaussian Process Regression (GPR) brings considerable conceptual and computational simplicity to the calculation, and therefore, has been applied as a model in derivative-free optimization known as Efficient Global Optimization (EGO) [14,20]. EGO is efficient and can be used when very little is known about the objective function, and particularly when the objective function is an expensive black-box function. Convergence as well as consistency properties of the algorithm was discussed in [3,37]. A brief refresher of these topics will now be given.

Assume that we have observed the vector of outputs

$$f(X_{\text{sim}}) = \{f(x_1), \dots, f(x_n)\}$$

of the simulator runs at locations $X_{\text{sim}} \subset \mathcal{D}$ (a set of n training points). By taking $\mu_0(\cdot) = 0$, one can update the prior belief about f at any new point x , and obtain a predictive distribution of a function value at a new point x :

$$f(x)|f(X_{\text{sim}}) \sim \mathcal{N}(\mu(x), \sigma^2(x)) \quad (2)$$

where

$$\begin{aligned} \mu_n(x) &= K(x, X_{\text{sim}})K(X_{\text{sim}})^{-1}f(X_{\text{sim}}) \\ \sigma_n^2(x) &= K(x, x) - K(x, X_{\text{sim}})K(X_{\text{sim}})^{-1}K(X_{\text{sim}}, x) \end{aligned} \quad (3)$$

are, respectively, the posterior mean and variance of a function value at x . For notation: given $X = \{x_1, \dots, x_m\} \subset \mathcal{D}$ and $Y = \{y_1, \dots, y_n\} \subset \mathcal{D}$, the object $K(X, Y)$ is defined as an $m \times n$ matrix whose elements are $k(x_i, y_j)$ where k is the prior covariance kernel. Also, $K(X) := K(X, X)$, assuming invertible here, is defined analogously. See [25] for more details.

In each iteration, EGO selects the next function evaluation point

Step 1. Construct a GP model using the training set

$$\{\mathbf{X}_n, \mathbf{f}_n\} = \{(x_1, f(x_1)), \dots, (x_n, f(x_n))\}$$

Step 2. $f_{\min}^n \leftarrow \min \mathbf{f}_n$

Step 3. Repeat until $n > N_{\max}$

1. Find x_{n+1} that maximizes the expected improvement:

$$x_{n+1} = \operatorname{argmax}\{x \in \mathcal{D} : \mathbb{E}_n[\max(0, f_{\min}^n - f(x))]\}$$

2. Simulate $f(x_{n+1})$
3. Update the training set $\mathbf{X}_{n+1} \leftarrow \mathbf{X}_n \cup \{x_{n+1}\}$, $\mathbf{f}_{n+1} \leftarrow \mathbf{f}_n \cup \{f(x_{n+1})\}$, $n \leftarrow n + 1$
4. Update the GP model, and $f_{\min}^n \leftarrow \min \mathbf{f}_n$

Algorithm 1. Bayesian Optimization

$x_{n+1} \in \mathcal{D}$ by maximizing the so-called *expected improvement* (EI) criterion, which depends both on the mean prediction $\mu_n(x)$ and on the associated uncertainty $\sigma_n^2(x)$ given by the GP model. More rigorously, for $n \geq n_0$, let $f_{\min}^n = \min\{f(x_1), \dots, f(x_n)\}$ be the best objective function value. EI is defined as the conditional expectation of the improvement brought by evaluating at a candidate point x and it has a closed-form expression:

$$\begin{aligned} \text{EI}_n(x) &= \mathbb{E}_n[\max(0, f_{\min}^n - f(x))] \\ &= (f_{\min}^n - \mu_n(x))\Phi\left(\frac{f_{\min}^n - \mu_n(x)}{\sigma_n(x)}\right) + \sigma_n(x)\phi\left(\frac{f_{\min}^n - \mu_n(x)}{\sigma_n(x)}\right) \end{aligned} \quad (4)$$

where \mathbb{E}_n is the conditional expectation given information $f(x_1), \dots, f(x_n)$ of all previous function evaluations, Φ and ϕ are the standard Gaussian cdf and pdf, respectively. The next function evaluation is performed at a point maximizing $\text{EI}_n(x)$, and the GP model is updated. Since EI values, as well as its first and second derivatives, can be computed quickly, optimizing EI function (Equation 4) can be done by using any off-the-shelf optimization solver such as a multistart algorithm with gradient ascent, or genetic algorithm with derivatives. Specific steps of EGO are outlined in Algorithm 1.

2.2. Bayesian Optimization with Aggregated Gaussian Process Model

An aggregated GP model [31] was proposed as an approximation to the exact GP model for a function in order to handle large number of input data points. The method is based on the idea of aggregating GP submodels that are cheaper to construct as follows. Let $X = \{x_1, x_2, \dots, x_n\} \subset \mathbb{R}^d$ be input training points whose value $Y(x_i)$ are known. The idea is to split the input data set X into p groups, X_i , $i = 1, \dots, p$. To make a prediction at a new point $x \in \mathcal{D}$, namely $Y(x)$, each GP submodel M_i is trained using a subset of the data X_i as the training set:

$$M_i(x) = E[Y(x)|Y(X_i)] = k(x, X_i)k(X_i, X_i)^{-1}Y(X_i), \quad (5)$$

where $k(\cdot, \cdot)$ is a given covariance function.

To aggregate the GP submodels, we define the vector

$$M(x) := (M_1(x), \dots, M_p(x))^T$$

which gathers all p submodels. Here, we consider $M_i(x)$ and $Y(x)$ as random vectors. It was shown that the $p \times 1$ covariance vector $k_M(x) := \text{Cov}[M(x), Y(x)]$ and the $p \times p$ covariance matrix $K_M(x) = \text{Cov}[M(x), M(x)]$ can be obtained analytically. Subsequently, the aggregated model $M_A(x)$ as well as the corresponding mean squared error $v_A(x)$ can be computed by

$$\begin{aligned} M_A(x) &= k_M(x)^T K_M(x)^{-1} M(x) \\ v_A(x) &= k(x, x) - k_M(x)^T K_M(x)^{-1} k_M(x). \end{aligned} \quad (6)$$

Observe a similarity between these two equations and the equations of the exact GP (Equation 3). The aggregated GP model M_A can be seen as an approximation to the exact GP model, and moreover it also enjoys many nice properties. The model has shown to be the best linear unbiased estimator of $Y(x)$ that writes $\sum_{i=1}^p \alpha_i M_i(x)$, where $\alpha = k_M(x)^T K_M(x)^{-1}$, with the mean squared error $E[(Y(x) - M_A(x))^2] = v_A(x)$. In addition, the aggregated model is interpolating, that is, if there is a model M_j such that $M_j(x_i) = Y(x_i)$, then we also have that $M_A(x_i) = Y(x_i)$ and in this case $v_A(x_i) = 0$. More details can be found in [31].

Thanks to these properties, by using the aggregated GP model as a predictor, we define corresponding expected improvement and the optimization algorithm in line with that of standard expected improvement in Equation 4 and Algorithm 1. In particular, by replacing the predictive mean and standard deviation μ_n and σ_n in Equation 4 with M_A and v_A , we obtain the expected improvement for the aggregated models.

We shall refer to this algorithm AEI: Aggregated GP-based optimization algorithm with Expected Improvement.

3. Experimental Study

3.1. AEI algorithm variants and alternatives

The aggregated GP model was fit using k-means to cluster the training data for the submodels in [2,31]. In this work, since the aggregated GP posterior mean and variance will later be used as parameter inputs for the AEI optimization algorithm, we in addition split and cluster the training data for the submodels randomly to compare and contrast the resulting expected improvement criteria. Therefore, two variants of AEI optimization algorithm will be considered depending upon how the training data is being gathered into p submodels:

- **Variant 1. AEI-kmeans:** the k-means is used to cluster training data points.
- **Variant 2. AEI-rand:** the training points are split into clusters randomly with roughly equal size.

In the experiment for Variant 2, in each iteration we divide the total number of observations n by a pre-defined number of clusters, p . If the quotient m of this fraction is not a whole number, for each of the first $p - 1$ clusters, we randomly pick $\lceil m \rceil$ points from the data, and then assign the remaining $n - (p - 1)\lceil m \rceil$ points to the final cluster.

Example of the two aggregated approximation models on a 2-dimensional Branin-Hoo function is given in Figure 1.

The AEI algorithm performance is compared with that of the standard Bayesian optimization algorithm, which relies on the standard GP, as defined in Algorithm 1. In the sequel, the initial design is an optimal

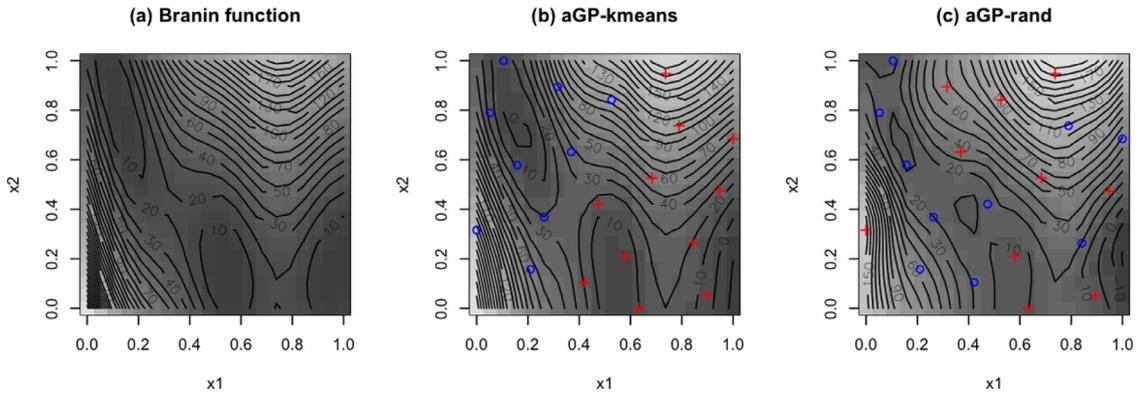


Fig. 1. Comparison of the true Branin-Hoo function surface and aggregated GP approximations. (a) Branin-Hoo function; (b) Variant 1 (k-means); (c) Variant 2 (rand).

n_0 -point Latin hypercube sampling (LHS) [8]. For all experiments, we utilized Gaussian process models with a Matern-3/2 covariance function. The EGO algorithm relies on the implementation in the R DiceOptim package [29], and the construction of the aggregated GP models are based on the code available in [30].

3.2. Preliminary Results

We now demonstrate the applicability of the algorithm on two benchmarking functions: Branin-Hoo function (2D) and Hartman6 function (6D) [10]. The domain for the first function is $[-5, 10] \times [0, 15]$ and for the Hartman6 is a unit cube $[0, 1]^6$.

Each experiment was repeated 30 times with different designs of experiments, and we represent the current best objective function value in terms of progress curve where we average the current best function value over 30 replications. Hereafter, plot legends “AEI-km”, “AEI-ra”, and “EI” will be used to refer to Bayesian optimization algorithms with aggregated GP model variant 1, variant 2, and the standard EGO, respectively.

Figure 2 shows the compared performances of the algorithms. On the left panel, the evolution of average current best function value is shown. At iteration 0, since all algorithms used the same initial design, all algorithms, and hence all plots start from the same average best function value. On the right panel, boxplots representing distribution of the best function values at the final iteration are shown.

For Branin-Hoo function, the size of initial LHS design in iteration 0 is set to 20 and the aggregated GP model uses 2 submodels. As the GP model used for predictions in EGO is an exact model while AEI variants are based on the approximated GP model, we did not expect the AEI method to work as well as EGO. From the progress plot, EGO indeed decreased the function value faster than AEI did. Nevertheless, the quality of the final value of AEI-kmeans is superior as can be seen from the boxplot.

As for Hartman6 function, we further explore the impact of the number of submodels used to fit the aggregated GP. We consider here 2, 5, and 10 submodels. The size of initial LHS design is set to 200. In this case, even though EGO seems to work best, the quality of the solution at the final iteration was outperformed by AEI-kmeans with 2 submodels. Indeed from modelling perspective, the number of submodels/clusters is directly related to the accuracy of the model (compared to the exact GP). The model with fewer number of clusters requires longer time to train (as it takes longer to fit a GP model with more points). The accuracy of the model, however, increases with decrease in the number of clusters. In practice, the number of clusters should be set by deciding on the maximum allowable number of observations in each cluster taking into account speed-accuracy tradeoff when training the model. The number of clusters can also differ from iteration to iteration, for example, to maintain the maximum number of points in each cluster.

3.3. Contaminant Source Localization Benchmark

The crux of the methodology behind the construction of the contaminant source localization optimization benchmark introduced in [24] is the use of inverse problem formulation to localize the contaminant source. Given information on the contaminant concentration levels at each of the groundwater monitoring wells $i = 1, \dots, 25$, and at times $t = 1, \dots, T$ days through *reference (observed) concentration mapping* $c_{\text{obs}}(i, t)$, the problem is to localize the source of the contaminant. An example of reference curves for the well number 2, 16, and 22 are given in Figure 3.

The unknown location of the contaminant source is denoted as $x^* = (x_s, y_s)$, and $c_{\text{sim}}(x, i, t)$ denotes the simulated contamination level obtained at (i, t) when the source is located at x which can be obtained by running a computationally expensive (blackbox) simulator when the contaminant source is assumed to be at x . Given a pre-defined set of monitoring wells in the contributing area \mathcal{I} , the true location of the contaminant source (x^*) therefore can be found by solving a minimization problem whose objective function is a *misfit function*:

$$f(x) = \sum_{i \in \mathcal{I}} \sum_{t=1}^T |c_{\text{obs}}(i, t) - c_{\text{sim}}(x, i, t)|, \quad (7)$$

where x is in the search zone $Z \in [20, 170] \times [-75, 75]$ restricted to a discrete regular grid of 3m resolution, and the well location index set $\mathcal{I} \subseteq \{1, 2, \dots, 25\}$ as shown in Figure 3. Data and the R function used to generate benchmarks are available at GitHub through <https://github.com/gpirot/BGICLP>.

4. Optimization Results for Contaminant Source Localization

4.1. Main Results

In this section we solve a class of inverse formulation simulation-based optimization problems as detailed in Section 3.3. The objective function representing the mismatch concentration levels as described in Equation 7 is defined for eight well configuration settings with different numbers of monitoring wells and locations. We consider the case when the true contaminant source is located at location $(x_s, y_s) = (89, -36)$, and geological geometry 1 is assumed (as described in [24]). Since each expensive simulation takes approximately one hour, the algorithm that can solve the problem in fewest number of function evaluation calls is most desirable.

For each configuration, the locations of the wells were not selected in random fashion but instead they are chosen so as to form a line orthogonal to the main flow directions (Problems 1-6) or diagonal to the main flow directions (Problems 7-8). The identification of the wells for each configuration is given in Table 1, and the resulting objective function surface landscape is shown in Figure 4. Recall that different

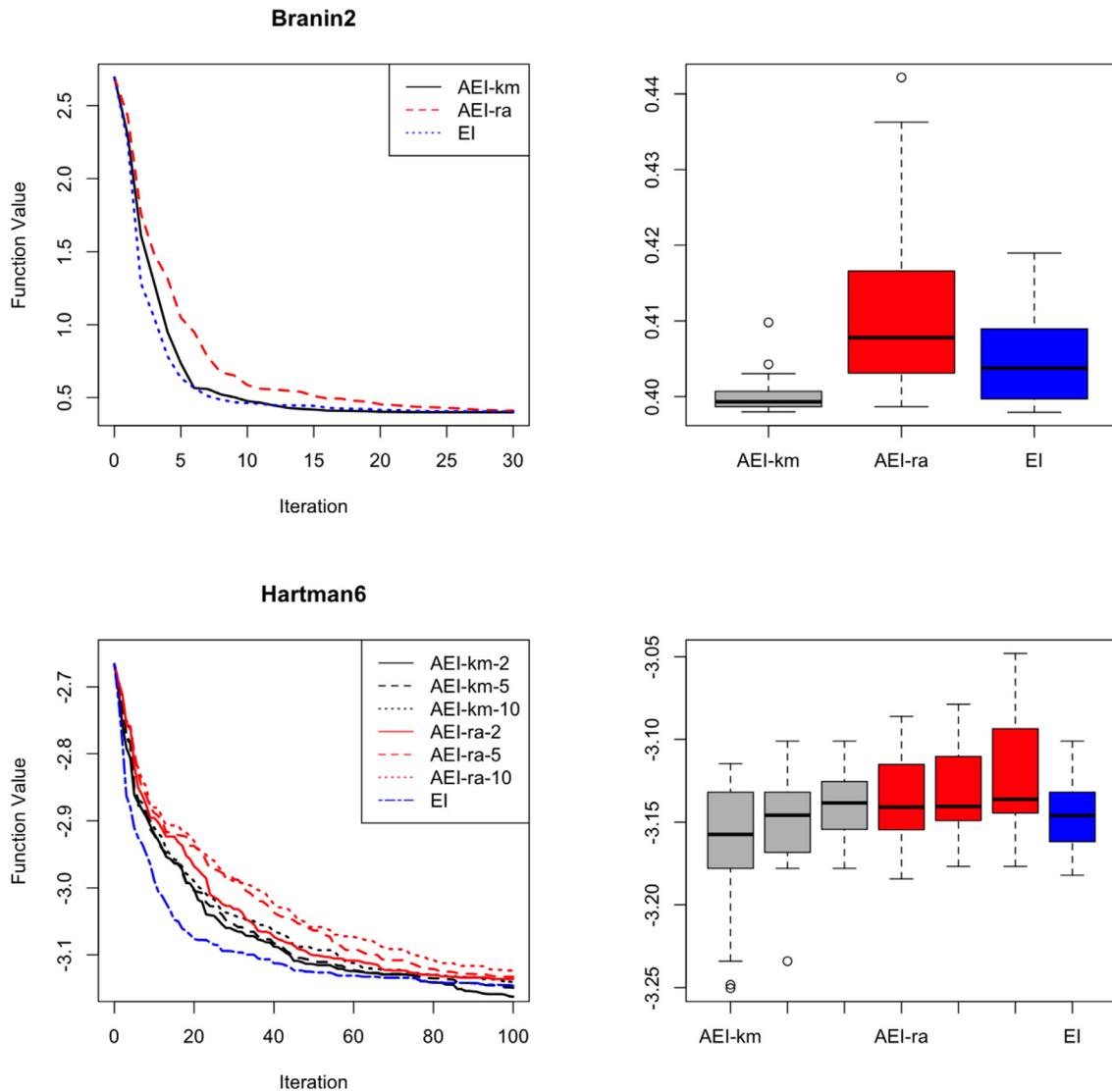


Fig. 2. Compared performances of AEI variants and the exact GP Bayesian optimization algorithms. **Left:** current best function value evolution (averaged over 30 trials); **Right:** distribution of the best function values at the final iteration.

shapes of landscapes are due to different representations of the same objective of source localization problem. Refer to Figure 3 for the location of each well.

Since the objective function was defined on a discrete domain, we adapt our algorithms in a straightforward way. In particular, the expected improvement function is maximized over a discrete domain instead of a continuous one.

The experiment was repeated and averaged over 30 trials. For each run, we start with a random initial Latin hypercube design of size 30, and let the algorithm run for 70 iterations using 2 submodels in AEI. The efficiency is again visualized by the progress curve; however, this time with the logarithmic scale on the y-axis for the sake of clarity. The obtained results are shown in Figure 5.

4.2. Algorithm Performance, Effort and Problem Difficulty

Using all 25 wells (Figure 5, Problem 6), surprisingly, both AEI algorithms turned out to work significantly better than EGO. Reducing number of wells to 20, 15 or 10 wells (Problems 5, 4, 3, respectively), although sometimes EGO could decrease function value somewhat faster at the beginning, it was shown that the other two AEI variants were able to catch up and succeed in getting ahead of EGO. Finally, when the number of wells was reduced to 5 or 3 (Problems 2 and 1,

respectively), although EGO and AEI-kmeans worked better than AEI-rand did, all the three algorithms could locate the contaminant source within 30 iterations. Now, as for the diagonal configuration (Problems 7 and 8), even though all the three algorithms succeeded in finding the source, AEI-rand outperformed the other two as it could overall get to the minimum value fastest (in less than 35 iterations).

One can see that the number, as well as the locations of the wells does play an important role on the number of iterations each algorithm needs in order to locate the contaminant source. In particular, while Problems 2, 7, and 8 are all using 5 monitoring wells, placing these wells orthogonal to the main flows simplify the problem as all three algorithms could locate the contamination source with smallest number of iterations.

Although AEI-km variant did not work as well as its counterpart AEI-rand algorithm, overall AEI-kmeans also outperformed EGO. It is however very interesting to observe the resemblance between the performance behavior of AEI-km and EGO. This can possibly be explained by the fact that the training data set for AEI-km is being grouped by the k-means clustering, and so the data points in a cluster are closer together. Each submodel can then be thought of as a fine resolution “local GP”. Merging each local GP into one aggregated GP model would, at least in theory, give a good approximation to the exact GP model leading to similarity of the performance of these two

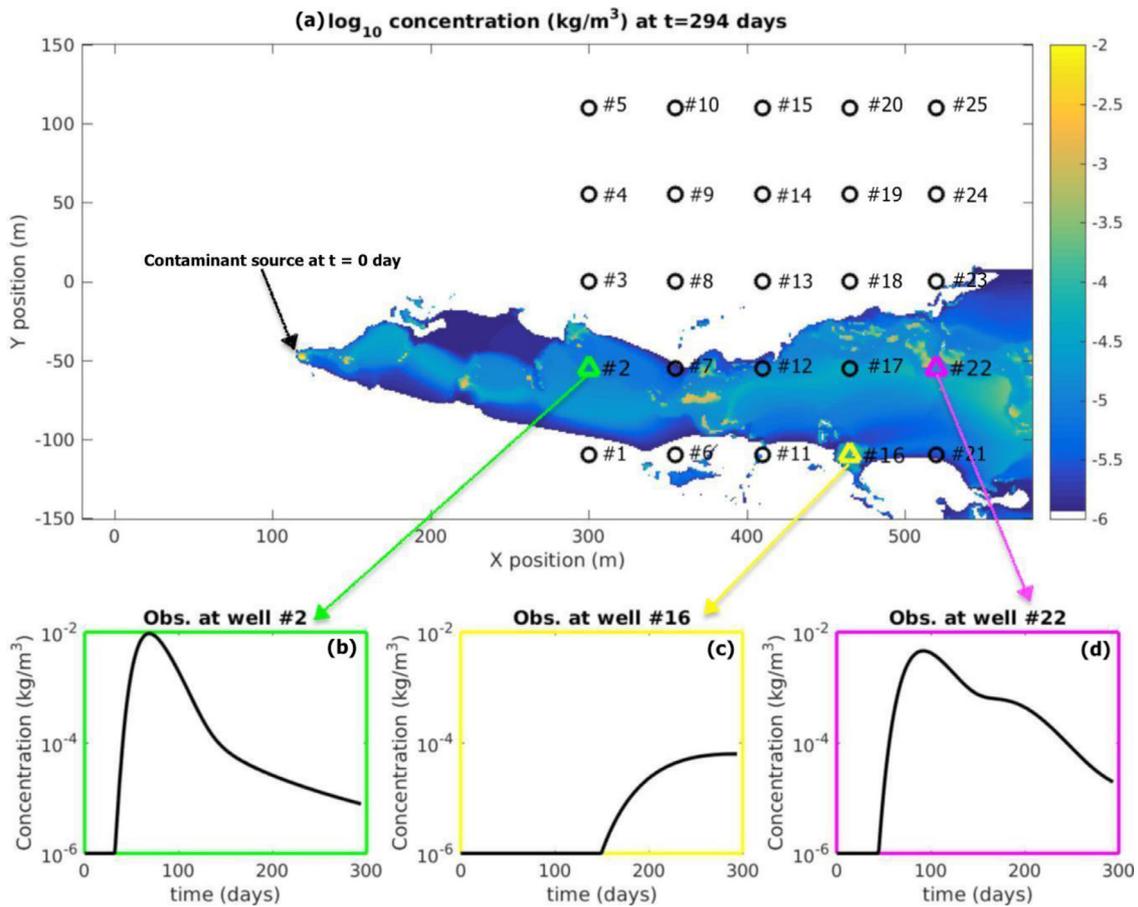


Fig. 3. The spreading of the contaminant concentration over time and concentration curves. (a) location of the 25 monitoring wells and source of contaminant at time 0; (b), (c), (d) concentration curves obtained at wells 2, 16, and 22, respectively.

Table 1
Description of the eight well configurations

Problem	Number of wells	Well ID
1	3	11,13,15
2	5	11,12,13,14,15
3	10	11,12,13,14,15,1,2,3,4,5
4	15	11,12,13,14,15,1,2,3,4,5,21,22,23,24,25
5	20	11,12,13,14,15,1,2,3,4,5,21,22,23,24,25,6,7,8,9,10
6	25	1 to 25
7	5 ✓	5,9,13,17,21
8	5 ✓	1,7,13,19,25

optimization algorithms. AEI-rand, on the other hand, is based on submodels whose training data points are randomly selected from the initial training set. Hence, each submodel can be viewed as a low resolution approximation model to the exact GP. And in our case, the findings suggest that merging these submodels into one and couple it with the expected improvement criteria lead to a more efficient and robust optimization algorithm.

It is worth mentioning that AEI-rand is the only algorithm that succeeded in locating the contaminant source in all 30 repetitions of all the eight configuration settings. In particular, the AEI-rand algorithm could solve all problems on average within 45 iterations. For the other two algorithms, some trials failed to converge to the contaminant source. Table 2 gives the number of trials and problems for which the algorithms did not converge within 70 iterations.

Given a problem, the level of problem difficulty could be measured by a success rate, i.e. computational performance, which reveals

problem difficulty as experienced by an algorithm. Moreover, a measure of effort (of an algorithm) could be defined as the number of computational iterations (running time) of an algorithm that solves a problem. We therefore propose using algorithm’s solving effort as a measure of problem difficulty.

Because each of these problems has the goal of localizing the same source of contaminant $((x_s, y_s) = (89, -36))$ in our case and the fact that AEI-rand could localize the source completely (success rate = 1), we focus our investigation on AEI-rand algorithm. In particular, for each problem, we will use the average number of iterations that required for AEI-rand algorithm to converge to the source of contaminant as a representative of a problem difficulty. The distribution of the number of iterations AEI-rand needed to solve each problem is shown in Figure 6.

Recall the mistfit objective function given in Eq. 7:

$$f(x) = \sum_{i \in I} \sum_{t=1}^T |c_{\text{obs}}(i, t) - c_{\text{sim}}(x, i, t)|.$$

By increasing the number of monitoring wells in the contributing area I , more information regarding contaminant levels $c_{\text{sim}}(x, i, t)$ from additional monitoring wells i ’s becomes available. As a result, we might be tempted to think that the algorithm will accurately localize the contaminant source quicker, this nevertheless was not the case here.

Consider Problems 1-6 which correspond to the configuration placed along the vertical line perpendicular to the main flow when using 3, 5, 10, 15, 20, and 25 wells, respectively. Since the former well configuration setting is a subset of the latter ones, more information becomes available to the algorithm in the latter cases. We can see that adding information regarding contaminant levels for two more wells from 3 to 5 helps to speed up the algorithm in locating the source of

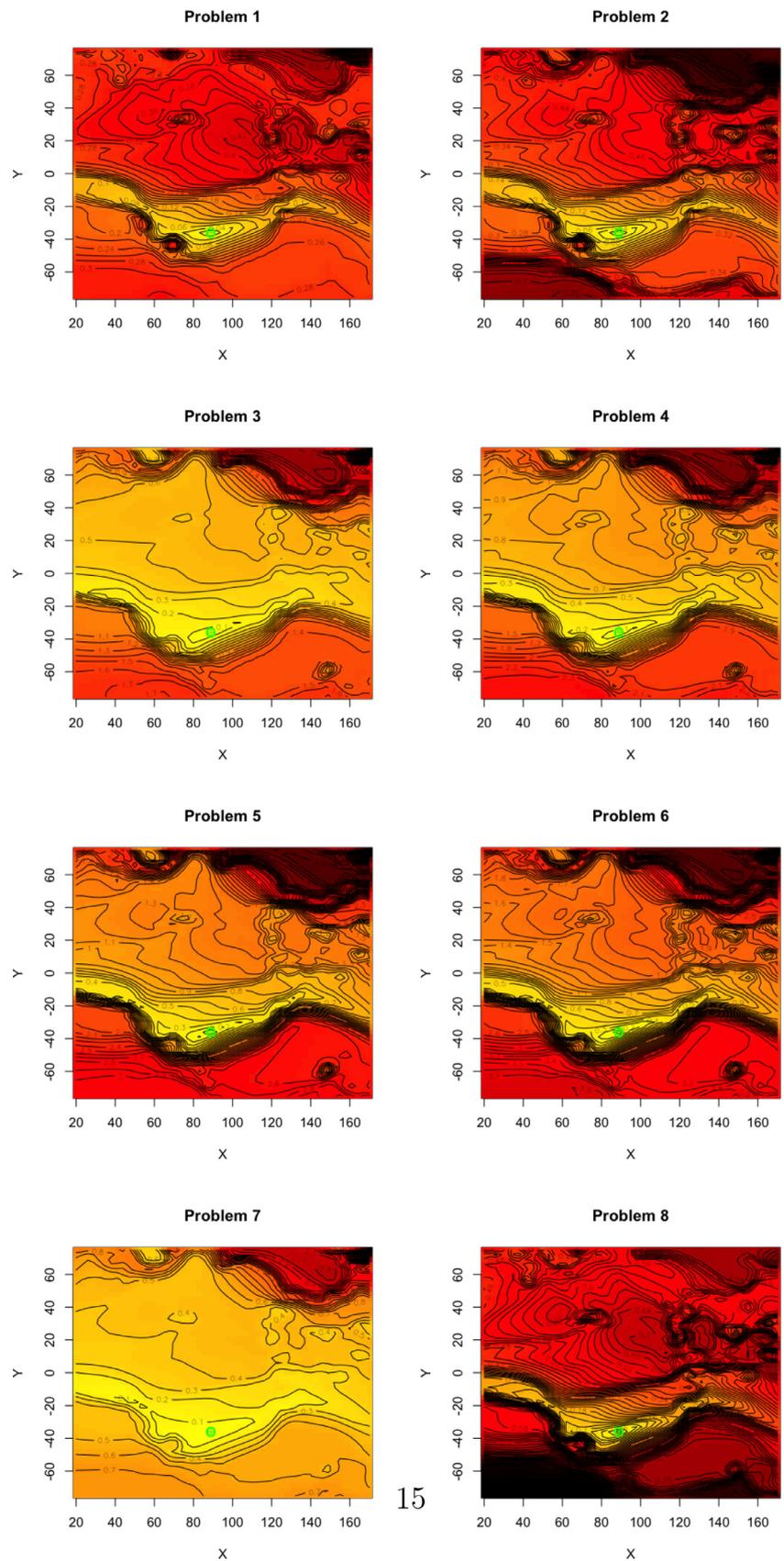


Fig. 4. Objective function surfaces for Problems 1-8. The location of the contaminant source (optimal point) at $(x_s, y_s) = (89, -36)$ is denoted by a green circle.

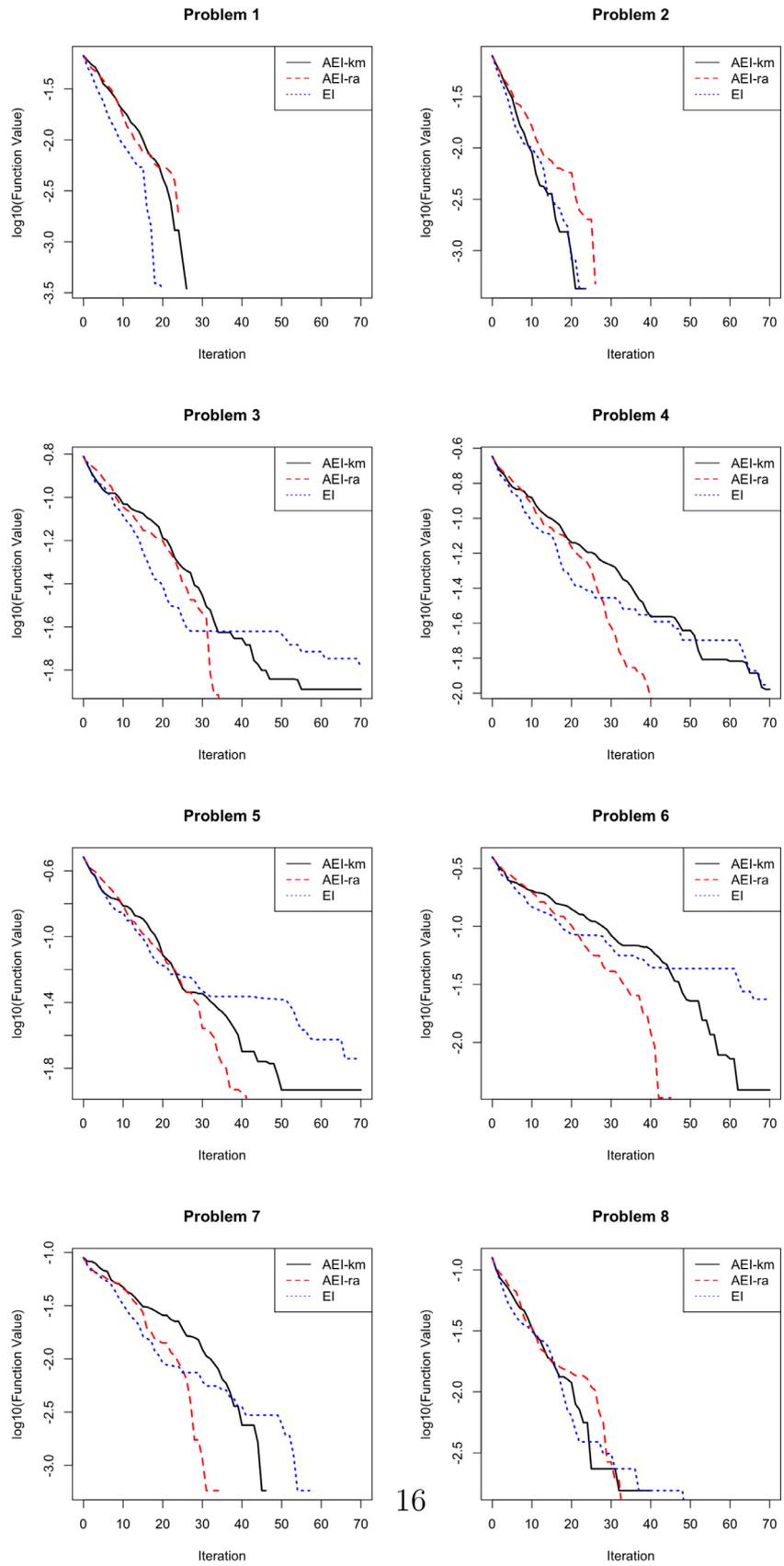


Fig. 5. Compared performances on source localization problems.

Table 2

Number of trials (out of 30 trials) that did not converge to the source of contaminant after 70 iterations. For other problems, both AEI-kmeans and EI converged for all 30 trials. AEI-rand algorithm converged for all problems all trials.

Algorithm	Problem 3	Problem 4	Problem 5	Problem 6
AEI-kmeans	8	5	4	1
EI	11	5	6	6

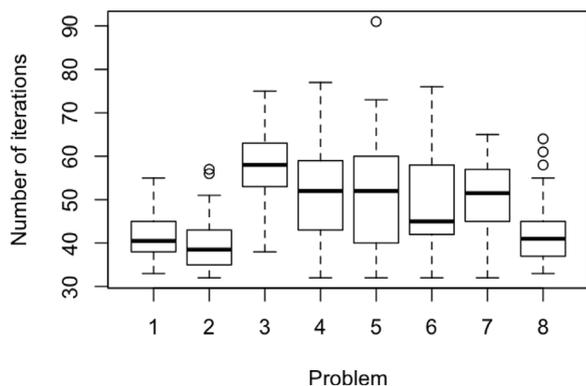


Fig. 6. Distribution of the number of iterations needed for AEI-rand algorithm to locate the contaminant source on each problem configuration (30 replications).

contaminant as it requires fewer number of iterations to converge. However, though seemingly counterintuitive, beyond this point, including more information provided by additional wells, the number of iterations needed for AEI-rand to converge grows which implies the harder the problem became.

In reverse-chronological order, we now provide the results of Leave-one-out cross validation on the initial design of the model to reveal the predictive capabilities of aggregated GP models. We show the boxplots of the distribution of mean-squared cross-validation errors in **Figure 7**. We can see that both models fit the data quite well, with the aggregated GP-rand version being slightly better. From another aspect, it is clearly observed that the problem becomes harder to model when the number of wells grows from 3 to 25 (Problems 1 to 6) as the value of the LOO error becomes larger. This further supports the conclusion made earlier.

We also tested the genetic algorithm (GA) ([33]) on the contaminant source localization problem to get some baseline performance. Here, we let the algorithm run until it converged, with population size equal 100 in each generation. The average number of generations required by the algorithm until converging are 17.90, 19.48, 20.47, 21.59, 22.40, 22.58, 22.53, 21.70, respectively for Problems 1-8. Note that since each GA population carries some solutions

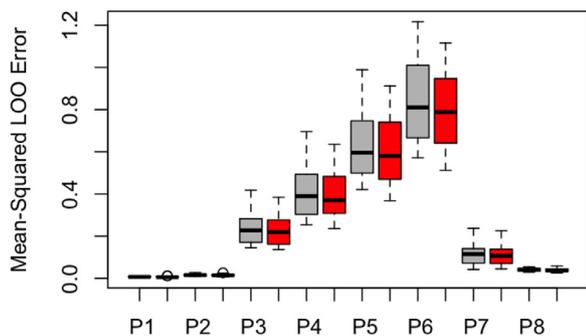


Fig. 7. Distribution of the mean leave-one-out error of the aggregated GP models on the initial design (30 replications) for Problems 1-8. Grey: aggregated GP with k-means, Red: aggregated GP with rand.

over the generation, we did not have to recompute these simulations every time. Counting only the number of unique simulations required over the course, the average number (over 30 trials) of expensive simulations needed are 484.37, 509.87, 522.60, 529.02, 543.46, 549.76, 548.14, 533.97, respectively. These numbers are to be directly compared with the results of AEI-rand algorithm with the average number of simulations ranging between 40-60 as reported in **Figure 6**. We can see that if we run the algorithm long enough, the minimum can also be located by GA, though at a higher price. In particular, since each simulation takes roughly 1 hour, ignoring other negligible computational overhead, AEI-rand could achieve solution much more quickly, and save several hundreds of hours of wall-clock time. This indeed highlights the potential benefits of using BO over metaheuristic methods on the groundwater applications. In addition, the results obtained from GA algorithm also confirms our findings that incorporating information from too many monitoring wells may result in an increase in problem complexity.

From a computational perspective, since the computationally expensive part comes from running simulations, it follows that the expense is roughly the same for each of the benchmark problems, as long as they run the same number of iterations, no matter how many wells are contributing the information. Therefore, efficient groundwater monitoring network can substantially assist the algorithm in successfully localizing the source of contaminant, or even accelerating the process. From an operational perspective, not only will the groundwater monitoring wells that were improperly constructed or installed slow down the convergence speed of the algorithm, they may need to be reclaimed or replaced, resulting in significant additional costs.

5. Conclusions

In this work, we studied and analyzed some interesting properties of a class of groundwater contaminant source localization problems under various well configuration settings using a Bayesian optimization algorithm relying on an aggregated GP model. While we showed the applicability of such approach on groundwater applications, we strongly believe that the method will prove very beneficial in solving other classes of optimization problems especially once we move ourselves to a higher dimension and therefore more data points are needed beyond the feasibility of the standard GP models.

As strange as it may seem, the proposed simulation-based optimization with an approximate model demonstrates superior reliability than those of the exact GP model. The reason might be that model overfitting occurs with the exact Gaussian process regression: it fits too good to the peculiarities of the data, rather than summarizing the underlying behavior of a highly complex non-linear objective function. Because the AEI-rand algorithm has successfully located the source of contaminant in all problems, and all trials, we subsequently defined a problem difficulty level through an algorithm’s solving effort using the number of iterations required for the algorithm to converge.

As the groundwater simulation is computationally expensive (can sometimes take longer than an hour per simulation), the optimization problem that requires fewest number of iterations to converge to the contaminant source is considered most cost-effective. Thus from both an algorithmic and economic points of view, the evidence from this study suggests that no less important to finding the configuration setting (number and location of wells) was the optimization methods that are robust and time efficient in localizing the contaminant source. It is nevertheless worthwhile to keep in mind that the groundwater monitoring system in practice involves also the minimum number of monitoring wells necessary to meet the performance standards issued by the protection agency [35]. This interesting topic could be the subject for further investigation and future works.

Declaration of interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRedit authorship contribution statement

Tipaluck Krityakierne: Conceptualization, Methodology, Software, Visualization, Investigation, Writing - original draft, Writing - review & editing. **Duangkamon Baowan:** Investigation, Supervision, Writing - review & editing.

Acknowledgments

The authors wish to thank the editor and the anonymous reviewers for their valuable comments. We thank Poopool Buathong for assistance with the complementary results of genetic algorithm to use as baseline performance for comparison. This work was supported by the Thailand Research Fund under Grant No.: MRG6080208, the Faculty of Science, Mahidol University, and Centre of Excellence in Mathematics, CHE, Thailand.

References

- [1] Amaran S, Sahinidis NV, Sharda B, Bury SJ. Simulation optimization: a review of algorithms and applications. *Annals of Operations Research* 2016;240(1):351–80.
- [2] Bachoc F, Durrande N, Rullière D, Chevalier C. Some properties of nested kriging predictors. arXiv preprint arXiv:170705708 2017.
- [3] Bect J, Bachoc F, Ginsbourger D. A supermartingale approach to gaussian process based sequential design of experiments. arXiv preprint arXiv:160801118 2016.
- [4] Burke EK, Silva JL. The influence of the fitness evaluation method on the performance of multiobjective search algorithms. *European Journal of Operational Research* 2006;169(3):875–97.
- [5] Candelieri A, Perego R, Archetti F. Bayesian optimization of pump operations in water distribution systems. *Journal of Global Optimization* 2018:1–23.
- [6] Carson Y, Maria A. Simulation optimization: methods and applications. *Proceedings of the 29th conference on Winter simulation*. IEEE Computer Society; 1997. p. 118–26.
- [7] Christelis V, Mantoglou A. Pumping optimization of coastal aquifers assisted by adaptive metamodelling methods and radial basis functions. *Water Resources Management* 2016;30(15):5845–59.
- [8] Damblin G, Couplet M, Iooss B. Numerical studies of space-filling designs: optimization of latin hypercube samples and subprojection properties. *Journal of Simulation* 2013;7(4):276–89.
- [9] Dehghanimohammadabadi M, Keyser TK. Intelligent simulation: Integration of simio and matlab to deploy decision support systems to simulation environment. *Simulation Modelling Practice and Theory* 2017;71:45–60.
- [10] Dixon LCW. The global optimization problem. an introduction. *Toward global optimization* 1978;2:1–15.
- [11] Frazier P, Powell W, Dayanik S. The knowledge-gradient policy for correlated normal beliefs. *INFORMS journal on Computing* 2009;21(4):599–613.
- [12] Gutmann H-M. A radial basis function method for global optimization. *Journal of global optimization* 2001;19(3):201–27.
- [13] Ilievski I, Akhtar T, Feng J, Shoemaker CA. Efficient hyperparameter optimization for deep learning algorithms using deterministic rbf surrogates. *Thirty-First AAAI Conference on Artificial Intelligence*. 2017.
- [14] Jones DR, Schonlau M, Welch WJ. Efficient global optimization of expensive black-box functions. *Journal of Global optimization* 1998;13(4):455–92.
- [15] Kavetski D. Parameter estimation and predictive uncertainty quantification in hydrological modelling. *Handbook of hydrometeorological ensemble forecasting* 2019:481–522.
- [16] Kopsiaftis G, Protopapadakis E, Voulodimos A, Doulamis N, Mantoglou A. Gaussian process regression tuned by bayesian optimization for seawater intrusion prediction. *Computational intelligence and neuroscience* 2019;2019.
- [17] Krityakierne T, Akhtar T, Shoemaker CA. Sop: parallel surrogate global optimization with pareto center selection for computationally expensive single objective problems. *Journal of Global Optimization* 2016;66(3):417–37.
- [18] Krityakierne T, Shoemaker CA. Soms: Surrogate multistart algorithm for use with nonlinear programming for global optimization. *International Transactions in Operational Research* 2017;24(5):1139–72.
- [19] Lehman J, Stanley KO. Exploiting open-endedness to solve problems through the search for novelty. *ALIFE*. 2008. p. 329–36.
- [20] Mockus J. The bayesian approach to local optimization. *Bayesian Approach to Global Optimization*. Springer; 1989. p. 125–56.
- [21] Nair AT, Makwana AR, Ahammed MM. The use of response surface methodology for modelling and analysis of water and wastewater treatment processes: a review. *Water science and technology* 2014;69(3):464–78.
- [22] Nguyen A-T, Reiter S, Rigo P. A review on simulation-based optimization methods applied to building performance analysis. *Applied Energy* 2014;113:1043–58.
- [23] Pautrat R, Chatzilygeroudis K, Mouret J-B. Bayesian optimization with automatic prior selection for data-efficient direct policy search. *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE; 2018. p. 7571–8.
- [24] Pirot G, Krityakierne T, Ginsbourger D, Renard P. Contaminant source localization via bayesian global optimization. *Hydrology & Earth System Sciences* 2019;23(1).
- [25] Rasmussen CE. *Gaussian processes in machine learning*. Advanced lectures on machine learning. Springer; 2004. p. 63–71.
- [26] Razavi S, Tolson BA, Burn DH. Review of surrogate modeling in water resources. *Water Resources Research* 2012;48(7).
- [27] Regis RG, Shoemaker CA. Constrained global optimization of expensive black box functions using radial basis functions. *Journal of Global optimization* 2005;31(1):153–71.
- [28] Regis RG, Shoemaker CA. A stochastic radial basis function method for the global optimization of expensive functions. *INFORMS Journal on Computing* 2007;19(4):497–509.
- [29] Roustant O, Ginsbourger D, Deville Y. Dicekriging, diceoptim: Two r packages for the analysis of computer experiments by kriging-based metamodelling and optimization. *Journal of Statistical Software, Articles* 2012;51(1):1–55. <https://doi.org/10.18637/jss.v051.i01>.
- [30] Rullière D, Durrande N, Bachoc F., Chevalier C.. Code: Fast computation of best linear predictors when the dataset is large. 2016b. URL <http://www.clementchevalier.com/index.php/r-packages>.
- [31] Rullière D, Durrande N, Bachoc F, Chevalier C. Nested kriging predictions for datasets with a large number of observations. *Statistics and Computing* 2016:1–19.
- [32] Sbai MA. Well rate and placement for optimal groundwater remediation design with a surrogate model. *Water* 2019;11(11):2233.
- [33] Scrucca L, et al. Ga: a package for genetic algorithms in r. *Journal of Statistical Software* 2013;53(4):1–37.
- [34] Shahriari B, Swersky K, Wang Z, Adams RP, De Freitas N. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE* 2015;104(1):148–75.
- [35] Shipman J, Liu F, Wilcox A, Haut R, Yuan B, Liu A, et al. Comprehensive overview of us environmental regulations; implementation and impact on the industry. *SPE Western Regional Meeting*. Society of Petroleum Engineers; 2019.
- [36] Sipper M., Urbanowicz R.J., Moore J.H.. To know the objective is not (necessarily) to know the objective function. 2018.
- [37] Vazquez E, Bect J. Convergence properties of the expected improvement algorithm with fixed mean and covariance functions. *Journal of Statistical Planning and Inference* 2010;140:11:3088–95.
- [38] Wu J, Frazier P. The parallel knowledge gradient method for batch bayesian optimization. *Advances in Neural Information Processing Systems*. 2016. p. 3126–34.
- [39] Xu W, Stein ML. Maximum likelihood estimation for a smooth gaussian random field model. *SIAM/ASA Journal on Uncertainty Quantification* 2017;5(1):138–75.
- [40] Zhigljavsky A, Žilinskas A. Selection of a covariance function for a gaussian random field aimed for modeling global optimization problems. *Optimization Letters* 2019;13(2):249–59.
- [41] Žilinskas A, Calvin J. Bi-objective decision making in global optimization based on statistical models. *Journal of Global Optimization* 2019;74(4):599–609.
- [42] Žilinskas A, Zhigljavsky A. Stochastic global optimization: a review on the occasion of 25 years of informatica. *Informatica* 2016;27(2):229–56.