JCTC Journal of Chemical Theory and Computation

Article

pubs.acs.org/JCTC

Optimization Algorithms in Optimal Predictions of Atomistic Properties by Kriging

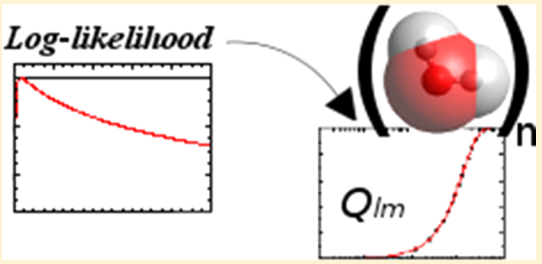
Nicodemo Di Pasquale, Stuart J. Davie, and Paul L. A. Popelier*

Manchester Institute of Biotechnology (MIB), 131 Princess Street, Manchester M1 7DN, Great Britain

School of Chemistry, University of Manchester, Oxford Road, Manchester M13 9PL, Great Britain

Supporting Information

ABSTRACT: The machine learning method kriging is an attractive tool to construct next-generation force fields. Kriging can accurately predict atomistic properties, which involves optimization of the so-called concentrated log-likelihood function (i.e., fitness function). The difficulty of this optimization problem quickly escalates in response to an increase in either the number of dimensions of the system considered or the size of the training set. In this article, we demonstrate and compare the use of two search algorithms, namely, particle swarm optimization (PSO) and differential evolution (DE), to rapidly obtain the maximum of this fitness function. The ability of these two algorithms to find a stationary point is assessed by using the first derivative of the fitness function. Finally, the converged position obtained by PSO and DE is refined through the limited-memory Broyden−Fletcher−Goldfarb−Shanno bounded (L-BFGS-B) algorithm, which belongs to the class of quasi-Newton algorithms. We show that both PSO and DE are able to come close to the stationary point, even in high-dimensional problems. They do so in a reasonable amount of time, compared to that with the Newton and quasi-Newton algorithms, regardless of the starting position in the search space of kriging hyperparameters. The refinement through L-BFGS-B is able to give the position of the maximum with whichever precision is desired.



INTRODUCTION

Condensed matter simulations are challenging: they demand the rapid yet accurate evaluation of the energy of a system of many thousands of atoms, varying over long time scales. Calculating the wave function on the fly severely restricts the size and length of the simulations that are feasible due to their huge computational cost. Force fields overcome this problem but at the cost of parametrization and possible vulnerability due to the mathematical shape of their equations. Typically, interactions are classified as either bonded or nonbonded, each with their own distinct energy penalty functions with respect to different reference states. One could question the physics behind this architecture. For example, why is there no Coulomb interaction between bonded atoms, or how should strong hydrogen bonds be treated? What if there is strong ligand−ligand interaction in a complex: should this interaction not acquire some bonded character? How do kinetic energy effects express themselves?

Classical force fields directly capture the link between the energy of a system and the nuclear positions of its constituent atoms. This directness is a blessing and a blight. It is a blessing because the equations are simple as they ignore the quantum mechanics underpinning the energy−position relationship. For example, a simple Hooke potential describes the energy variation upon compression or elongation of a bond near its equilibrium value. However, this potential is ignorant of the quantum mechanical cause for this energy variation. This ignorance is also a blight. An example concerns rotation barriers. Although torsion energy profiles can be simply represented by a short Fourier series, such analytical formulas do not "understand" the cause of this profile. Indeed, the torsion profile in ethane emerges from its underlying quantum contributions (intra-atomic energies, interatomic Coulomb, exchange, and correlation energies) in a way very different from that in formamide.[1] The question is then if one can design a force field that captures these underlying quantum contributions. The answer is yes, and one such initiative was started some time ago and developed under the acronym QCTFF.[2]

QCTFF is an energy predictor that sees atoms as malleable boxes, which interact through multipolar electrostatics,[3] while keeping track of the atoms' intra-atomic energies as well as their interatomic exchange and correlation energies. The atoms are defined according to the quantum theory of atoms in molecules,[4] while the knowledge of how they interact is stored in so-called kriging models. The machine learning method kriging[5] is rooted in geostatistics, where it achieves the prediction of geographical regions of a high concentration of some precious material based on a relatively low number of measurements. In the current context, kriging predicts a property of a given atom (e.g., charge, dipole moment, intra-atomic self-energy, exchange energy with its full atomic environment) as a function of the nuclear coordinates of the atoms surrounding this given atom. As such, QCTFF avoids the limitation of point charges, tackles

Received: October 1, 2015
Published: March 1, 2016

DOI: 10.1021/acs.jctc.5b00936
J. Chem. Theory Comput. 2016, 12, 1499−1513

polarization effects and charge transfer in a streamlined manner,[6] and accounts for the energetic behavior of any bond of any degree of covalency. In fact, the core of QCTFF's architecture does not differentiate between intra- and intermolecular interactions. This is because QCTFF is not formulated within the context of long-range Rayleigh−Schrödinger perturbation theory,[7] unlike other next-generation force fields mentioned below. Indeed, while targeting condensed matter, QCTFF is ultimately trained by supermolecular clusters in which (topological) atoms are always well-defined; the atoms interact as malleable boxes of finite volume, wherever they are, that is, within a single molecule or across two molecules.

The strategy described above is very different from that followed by alternative approaches that strive to enhance the accuracy and realism of classical force fields such as AMBER, CHARMm, OPLS, MM3, or GROMOS. Among the most developed of such approaches are SIBFA,[8] XED,[9] EFP,[10] AMOEBA,[11] NEMO,[12] and DMACRYS.[13] These approaches also recognize the value of multipolar electrostatics. The quest for a new model to increase the accuracy of current force fields has led to a wider use of machine learning techniques in computational chemistry. In 2007, Behler and Parrinello[14] proposed using neural networks to approximate potential energy surfaces and later extended neural networks to the *ab initio* description of sodium under high-pressure, high-temperature conditions.[15] Bartok et al.[16] developed the Gaussian approximation potential based on Bayesian inference to increase the accuracy of DFT schemes. Other machine learning techniques in predicting atomization energies were analyzed by Hansen et al.[17] The current article rigorously investigates the kriging training process on a number of water clusters. In particular, the so-called concentrated log-likelihood (which will be defined mathematically below) must be maximized in order to complete the training process. This maximization is achieved by machine learning methods (completely different from kriging) called particle swarm optimization (PSO)[18] or differential evolution (DE).[19] The main result is a proof that the maxima that PSO or DE find are confirmed by the analytical (nonmachine learning method) quasi-Newton algorithm L-BFGS-B.[20] The systems investigated are the water monomer, dimer, and hexamer.

COMPUTATIONAL METHODOLOGY

Kriging is a machine learning method that allows one to estimate the value of a generic property (e.g., one of the components of the atomic multipole moment) as a function of some input property using the values of the property in known locations (so-called training points). Intuitively, the kriging predictor assumes the continuity of physical phenomena by considering that two closer locations in space have a higher likelihood of showing similar values of a property when compared to that for far away points. Kriging is an interpolating predictor, i.e., it gives the exact value of the property in the training points and is defined as the best linear unbiased predictor with respect to minimizing the squared errors on the predictions. Unbiasedness ensures that the expected value of the predictor equals the expected value of the property under study.[21] In particular, Sacks et al.[21] showed, in their classic paper, that the kriging predictor is the best linear unbiased predictor (BLUP). The prediction of properties (e.g., atomic multipole moments) by means of kriging[22] requires the evaluation of the kriging parameters $\mathbf{p}$ and $\boldsymbol{\theta}$ by maximizing the log-likelihood function, which, starting from the likelihood function,[23] reads as

$$L(\boldsymbol{\theta}, \mathbf{p}, \sigma, \mu) = -\frac{N_t}{2}\ln(\sigma^2) - \frac{1}{2}\ln(|\mathbf{R}|) - \frac{(y - \mathbf{1}\mu)^T \mathbf{R}^{-1}(\mathbf{y} - \mathbf{1}\mu)}{2\sigma^2} \quad (1)$$

where $\mathbf{y}$ is the column vector of the modeled property evaluated at each of the N_t training points, $\mathbf{1}$ is a column vector of ones, $(\bullet)^T$ is the transpose of its argument (generating row vectors here), σ^2 is the variance, μ is the mean, $\mathbf{R}$ is the correlation matrix (defined below), and $|\mathbf{R}|$ is its determinant. By maximizing the log-likelihood, we obtain the most likely values of the kriging parameters $\mathbf{p}$ and $\boldsymbol{\theta}$ given the training points.

Although it is known that the maximization of the log-likelihood (ML) could result in the underestimation of the variance of the process described by kriging model,[24] it does not represent a problem in the situation described here because only an expected value of a prediction is considered, with the future goal of using the kriging model in an approximation of a deterministic computer experiment.

Model selection, i.e., selection of $\mathbf{p}$ and $\boldsymbol{\theta}$ values giving the best predictions through log-likelihood maximization, is not the only method available. Cross-validation (CV) estimation[25] is another widely known estimation method. Differences between the two methods, in terms of quality of the predictions, were considered in the work of Wahba[26] and Stein[27] for the estimation of the parameters for a smoothing spline. According to Wahba, CV is more robust when the model is not correct. A noncorrect model implies that the true probability distribution of the properties departs from the probability distributions that the model uses as an approximation. For the same case, Stein showed that the asymptiotic variance of an ML model is half of the variance of CV when the model is correct.

In a recent paper, Bachoc[28] analyzed the differences between ML and CV selection models for kriging and found that CV is more robust if the model is misspecified, i.e., predictions are better with hyperparameters obtained through cross-validation rather than likelihood maximization. However, when the model is correct, ML always performs better than CV. A discussion about the correctness of the model is beyond the scope of the work and will be left for future reports.

The matrix $\mathbf{R}$ is a $N_t \times N_t$ matrix describing the correlation between the points in the training set. Being a correlation matrix, it is both symmetric and positive semidefinite (see p 80 in ref 29). In this work, the correlation R_{ij} between two training points, referred to with indices i and j, is the power exponential correlation[29]

$$R_{ij} = \exp\left[-\sum_{k=1}^{N_f} \theta_k |x_k^i - x_k^j|^{p_k}\right] \quad (2)$$

where x_k^i is the kth component of the of the ith training point, and N_f is the number of features. Each component can have different dimensions (e.g., distance, angle); therefore, each θ_k will have a dimension (dependent on p_k) that assures that the argument of the sum is nondimensional. The latter property is manifest for any argument of an exponential function. Note that it is possible to work with nondimensional theta values, $\tilde{\theta}_k$, corresponding to normalized training points. However, it is

DOI: 10.1021/acs.jctc.5b00936
J. Chem. Theory Comput. 2016, 12, 1499−1513

always possible to transform one representation into the other, as demonstrated in the Appendix. The power exponential correlation function contains the two special cases: (1) Gaussian correlation when $p_k = 2$ for every k and (2) the exponential case when $p_k = 1$ for every k. The work presented here will exclusively use non-normalized coordinates because of preliminary results displaying slightly improved numerical stability. One of the problems in dealing with the concentrated log-likelihood is represented by the ill-conditioning of the correlation matrix, which may lead to numerical instabilities. The conditioning of a matrix is defined through the condition number

$$\kappa(\mathbf{R}) = \begin{cases} \|\mathbf{R}\|\,\|\mathbf{R}^{-1}\| & \text{if } \mathbf{R} \text{ is nonsingular} \\ \infty & \text{if } \mathbf{R} \text{ is singular} \end{cases}$$

$$\text{where } \|\bullet\| \text{ is any matrix norm} \tag{3}$$

which measures the relative error on $\mathbf{R}^{-1}$ when $\mathbf{R}$ is perturbed by an arbitrary infinitesimally small matrix. Therefore, a small condition number is preferable because each error in the training data will inevitably be amplified. This includes the situation presented here, where training points are obtained through numerical integration of certain functions, as will be described in following sections. It was found that a small spacing between training points increases the condition number of the correlation matrix.[30]

The unknown values of σ^2 and μ must be estimated from the training points. This estimation adds an error to the predictions quantified by the mean-squared error (MSE) that will be defined later. Following Jones,[31] we choose the values of σ^2 and μ that maximize $L(\boldsymbol{\theta}, \mathbf{p}, \sigma, \mu)$. This is obtained by analytically obtaining the derivatives of the log-likelihood with respect σ^2 and μ and setting them to zero. The values of σ^2 and μ that maximize the log-likelihood will be indicated as $\hat{\sigma}^2$ and $\hat{\mu}$ and are equal to

$$\hat{\sigma}^2 = \frac{(\mathbf{y} - \mathbf{1}\hat{\mu})^T \mathbf{R}^{-1} (\mathbf{y} - \mathbf{1}\hat{\mu})}{N_t}$$

$$\hat{\mu} = \frac{\mathbf{1}^T \mathbf{R}^{-1} \mathbf{y}}{\mathbf{1}^T \mathbf{R}^{-1} \mathbf{1}} \tag{4}$$

By substituting the expression for $\hat{\sigma}^2$ in eq 4 back into eq 1, we obtain the concentrated log-likelihood $\hat{L}(\theta, \mathbf{p})$, which, ignoring constant terms, reads as

$$\hat{L}(\theta, \mathbf{p}) = -\frac{N_t}{2} \ln(\hat{\sigma}^2) - \frac{1}{2} \ln(|\mathbf{R}|) \tag{5}$$

The components of vectors $\boldsymbol{\theta}$ and $\mathbf{p}$ are bounded as follows, according to Rasmussen and Williams[29]

$$\begin{aligned} 0 &\le \theta_k & k &= 1, \ldots, N_f \\ 0 &< p_k \le 2 & k &= 1, \ldots, N_f \end{aligned} \tag{6}$$

Note that Jones defines[32] the boundaries of p_k as $1 \le p_k \le 2$. The dimension of $\hat{L}$ is given by the sum of the dimension of the $\mathbf{p}$ and $\boldsymbol{\theta}$ vectors, and in this case, it is $2N_f$. The function $\hat{L}$ is a high-dimensional function that requires the evaluation of a matrix ($\mathbf{R}$) and its inverse for a single function value, which makes the optimization computationally very expensive.

The ill-conditioning of the correlation matrix $\mathbf{R}$ cannot be avoided, especially during the initialization of the optimization algorithms, where it is possible that the initial points selected may lead to the $\mathbf{R}$ matrix being singular, up to machine precision. In fact, the initialization of the nonanalytic optimization algorithms (i.e., PSO and DE) is obtained by generating random numbers $r_\theta \sim U(0,1)$ (denoting the drawing from a uniform distribution) and assigning to each component of each point in θ-space the value

$$\theta_{i_k} = \theta_{\min} + (\theta_{\max} - \theta_{\min}) r_\theta \quad i = 1, \ldots, N;\ k = 1, \ldots, N_f \tag{7}$$

where N is the total number of points in θ-space, which is specific for each optimization algorithm and will be discussed in the next section. In this work, we use $\theta_{\min} = 20$ and $\theta_{\max} = 100$ for the monomer and $\theta_{\min} = 0$ and $\theta_{\max} = 1$ for the dimer and hexamer. Also, in this work, the value of $\mathbf{p}$ is fixed to 2 because it results in slightly better models[23] and an increase in the computational speed of the concentrated log-likelihood optimization. However, obtaining better results with fixed $\mathbf{p}$ is not a general result, and some other systems not studied here may benefit from an optimized $\mathbf{p}$, which is perfectly possible with the methodology developed here.

During the exploration of the θ–p space, the purpose is to avoid the points where the concentrated log-likelihood is not defined and instead continue the search in valid regions. Unfortunately, there is no *a priori* way to decide which regions have to be avoided. Recent investigations by Zimmerman[33] and Kok[34] present conflicting results: Zimmerman claims, under certain assumptions, that the concentrated log-likelihood goes to infinity as $\boldsymbol{\theta}$ goes to zero, but Kok disputes the generality of these assumptions by demonstrating, both analytically and numerically, that this concentrated log-likelihood behavior is not true in general. To prevent our optimization algorithms from moving through regions where the concentrated log-likelihood function is not defined, we used a slight redefinition of the concentrated log-likelihood itself

$$\bar{L}(\theta, \mathbf{p}, \sigma, \mu) = \begin{cases} \hat{L}(\theta, \mathbf{p}, \sigma, \mu) & \text{if } \mathbf{R} \text{ is not singular} \\ -100\,000 & \text{otherwise} \end{cases} \tag{8}$$

From now on, we refer to the concentrated log-likelihood as the fitness function.

Searching the global maximum of this function is affected by two issues:

(1) In general, the search landscape is not known, and an *a priori* prediction of the number and position of local maxima, or the location of the global maximum, is not feasible.
(2) An iterative procedure must deal with the fact that at every iteration a matrix (potentially sizable) must be built and inverted.

The first issue limits the use of direct optimization methods (conjugate gradients, Newton–Raphson methods) because, in order to be effective, these methods need to start in a position close to the global maximum. However, the second issue warrants a careful evaluation of which iterative method is best employed because it is desirable that the chosen method (i) converges fast toward the global maximum, (ii) is able to explore a wide region of feature space, and (iii) avoids being trapped in a local maximum. The selection of model parameters for a kriging model (i.e., the θ–p values), through the maximization of the log-likelihood, is affected by three main problems:[35] (a) ill-conditioned $\boldsymbol{R}$ matrix, (b) long-ridges in the log-likelihood, and (c) multimodality of the log-likelihood

DOI: 10.1021/acs.jctc.5b00936
J. Chem. Theory Comput. 2016, 12, 1499–1513

function. Point (a) has already been discussed in the derivation of the modified concentrated log-likelihood $\bar{L}$, whereas points (b) and (c) will be discussed in the following sections.

In this work, three different optimization methods are considered: PSO,[18] DE,[19] and the L-BFGS-B algorithm,[20,36] with the latter being the only analytical algorithm. The DE literature and the PSO literature use different names for the same quantities. For example, vectors in the multidimensional search space are referred to as "vectors" in DE context, as opposed to "particles" in PSO context, whereas a set of such vectors is called a "population" in DE and a "swarm" in PSO. In this work, we will adopt the respective names found in the literature for each method.

Particle Swarm Optimization. PSO is a population-based algorithm introduced by Kennedy and Eberhart in 1995,[18] named for the way it mimics the behavior of a swarm. Once a good candidate solution is found by a particle, the swarm tends to converge toward it, allowing a better exploration of the neighbors of the good candidate solution. The PSO optimization algorithm has already been considered for hyper-parameter selection[37] in the case of support vector machine (SVM) models.[38] A similar attempt to use PSO for kriging hyper-parameter selection is reported in Toal et al.,[39] where they proposed a hybrid PSO that uses reinitialization of some of the particles of the swarm in unexplored regions of the $\theta-p$ space. However, the approach presented by Toal et al.[39] increases the number of PSO free parameters that need to be optimized with respect to classic PSO. Although classic PSO is used here to obtain meaningful results, a possibly improved result could be achieved using an extended treatment, such as that by Toal et al.[39] However, a comprehensive comparison of various PSO algorithms is beyond the scope of this work. In PSO, the ith particle, which represents a point in $\theta-p$ space, is marked by a position vector $\mathbf{x}_i$. A displacement, indicated by $\mathbf{v}_i$, is associated with this particle and represents its movement at each iteration. For this reason, the quantity $\mathbf{v}_i$ is normally addressed as "velocity" by the PSO community for historical reasons,[40] and we decided to remain consistent with the literature.

$$\begin{aligned}\mathbf{v}_{t+1,i} &= \omega\mathbf{v}_{t,i} + c_1\mathbf{r}_{t,1} \circ (\mathbf{b}_{t,i} - \mathbf{x}_{t,i}) + c_2\mathbf{r}_{t,2} \circ (\mathbf{g}_{t,i} - \mathbf{x}_{t,i})\\ \mathbf{x}_{t+1,i} &= \mathbf{x}_{t,i} + \mathbf{v}_{t+1,i}\end{aligned} \tag{9}$$

where $\mathbf{r}_{t,1}$ and $\mathbf{r}_{t,2}$ are vectors of random real numbers uniformly chosen between 0 and 1 and their dependence on the iteration step t is used to indicate that these vectors are independently chosen at every iteration. In eq 9, the symbol $\circ$ represents the Hadamard product, such that $\mathbf{x} \circ \mathbf{y} = (x_1y_1, x_2y_2, ..., x_ny_n)$ for two given vectors with n components, that is, $\mathbf{x} = (x_1, x_2, ..., x_n)$ and $\mathbf{y} = (y_1, y_2, ..., y_n)$; ω is the inertia weight, and c_1 and c_2 are, respectively, the cognitive learning factor and the social learning factor (both sometimes called acceleration coefficients). In this work, they are both set to 1.494.[41] The inertia weight is set to 0.729,[41] but it is allowed to decrease linearly as the swarm converges toward a single point. Finally, $\mathbf{b}_{t,i}$ and $\mathbf{g}_t$ represent the private guide and global guide of each particle, respectively. The private guide represents the best position the particle has visited during all iterations, whereas the global guide represents the single best position found across the whole swarm so far.

The velocity of the particles is initialized through the "half diff" method[40]

$$\mathbf{v}_{1,i} = 0.5(\mathbf{u}_i - \mathbf{x}_{1,i}) \tag{10}$$

where each component of vector $\mathbf{u}$, $u_{i_k} \sim U(x_{\min}, x_{\max})$, is a number drawn from the uniform distribution between $x_{\min}$ and $x_{\max}$. If the problem is bounded (see eq 6), then a set of boundary conditions must be specified. Although in this work the value of the components of $\mathbf{p}$ are kept constant, they can vary in general and some boundary condition must be supplied. Hyperbolic boundary conditions can be used to bound the value of the components of $\mathbf{p}$ (eq 6) and are defined as follows

$$\begin{cases} v'_{t+1,i_k} = \dfrac{v_{t+1,i_k}}{1 + \left|\dfrac{v_{t+1,i_k}}{p_{\max} - p_{t,i_k}}\right|} \\ \quad \text{if } v_{t+1,i_k} > 0,\ k = 1, ..., N_f \\ v'_{t+1,i_k} = \dfrac{v_{t+1,i_k}}{1 + \left|\dfrac{v_{t+1,i_k}}{p_{t,i_k} - p_{\min}}\right|} \\ \quad \text{if } v_{t+1,i_k} \le 0,\ k = 1, ..., N_f \end{cases} \tag{11}$$

where p_{t,i_k} is used instead of x_{t,i_k}, v'_{t+1,i_k} is the new velocity at iteration $t + 1$ corrected by the boundary conditions, and $p_{\max}$ and $p_{\min}$ are the boundaries of the components of $\mathbf{p}$, which are set to 2 and 0, respectively.

The boundary conditions for the components of $\boldsymbol{\theta}$ are slightly modified hyperbolic boundary conditions, where only the lower boundary is considered

$$v'_{t+1,i_k} = \frac{v_{t+1,i_k}}{1 + \left|\dfrac{v_{t+1,i_k}}{\theta_{t,i_k} - \theta_{\min}}\right|} \quad \text{if } v_{t+1,i_k} \le 0,\ k = 1, ..., N_f \tag{12}$$

where again θ_{t,i_k} is used instead of x_{t,i_k}, v'_{t+1,i_k} is the new velocity at iteration $t + 1$ corrected by the boundary condition, and $\theta_{\min}$ is set to zero. In this case, only the lower boundary condition must be checked and reinforced. The dimension of the swarm, denoted by S, is defined as[40]

$$S = 10 + 2\sqrt{N_f} \tag{13}$$

Finally, the position of the maximum is considered to be found when the relative error between the best position at iteration t, $\mathbf{g}_t$, and the best position at iteration $t - 1$, $\mathbf{g}_{t-1}$, is under the tolerance of $\varepsilon = 10^{-8}$ for $C = 20$ iterations. These values were chosen in order to ensure the independence of the final results from ε and C and have previously been used successfully in our lab.

Differential Evolution. DE was first introduced by Storn and Price.[19] DE belongs to the category of evolutionary algorithms in which, at every generation, the decision to keep a new candidate solution is determined by a greedy criterion. The standard structure of the DE algorithm is as follows:

- **Mutation**: an "offspring" vector is created by randomly combining the difference between "parent" vectors. The literature provides several mutation strategies. In this work, we have considered five of them:[42,43]

DOI: 10.1021/acs.jctc.5b00936
J. Chem. Theory Comput. 2016, 12, 1499−1513

(0) DE/best/1

$$\mathbf{v}_{G+1,i} = \mathbf{x}_{G,\text{best}} + F(\mathbf{x}_{G,a1} - \mathbf{x}_{G,a2}) \tag{14}$$

(1) DE/current-to-best/2

$$\mathbf{v}_{G+1,i} = \mathbf{x}_{G,i} + F(\mathbf{x}_{G,\text{best}} - \mathbf{x}_{G,i} + \mathbf{x}_{G,a1} - \mathbf{x}_{G,a2}) \tag{15}$$

(2) DE/rand/1

$$\mathbf{v}_{G+1,i} = \mathbf{x}_{G,a3} + F(\mathbf{x}_{G,a1} - \mathbf{x}_{G,a2}) \tag{16}$$

(3) DE/best/2

$$\mathbf{v}_{G+1,i} = \mathbf{x}_{G,\text{best}} + F(\mathbf{x}_{G,a1} - \mathbf{x}_{G,a2} + \mathbf{x}_{G,a3} - \mathbf{x}_{G,a4}) \tag{17}$$

(4) DE/rand/2

$$\mathbf{v}_{G+1,i} = \mathbf{x}_{G,a5} + F(\mathbf{x}_{G,a1} - \mathbf{x}_{G,a2} + \mathbf{x}_{G,a3} - \mathbf{x}_{G,a4}) \tag{18}$$

where $\mathbf{x}_{G,i}$ is the Gth generation of the ith vector; a_1, a_2, a_3, a_4, and a_5 are integer numbers with $a_i \neq a_j$, if $i \neq j$, randomly drawn in the interval $[1, N_p]$, where N_p is the size of the population; $F \in [0, 2]$ controls the amplification of difference vectors $(\mathbf{x}_{G,i} - \mathbf{x}_{G,j})$; $\mathbf{x}_{G,\text{best}}$ is the best vector in the population at generation G (i.e., $\bar{L}(\mathbf{x}_{G,\text{best}}) > \bar{L}(\mathbf{x}_i)\ \forall\ i \in [1, N_p]$). For the rest of this article, the above five mutation strategies (MS) will be identified as MS0, MS1, MS2, MS3, and MS4.

•**Crossover**: The mutated offspring (also called the "trial vector") obtained in the previous step is randomly crossed with the parents in order to increase the variability of the offspring

$$u_{G+1,i_j} = \begin{cases} v_{G+1,i_j} & \text{if } (\text{rand}_{i_j} \leq CR) \text{ or } j = I_{\text{rand}} \\ x_{G,i_j} & \text{if } (\text{rand}_{i_j} > CR) \text{ and } j \neq I_{\text{rand}} \end{cases} \quad j = 1, \ldots, N_f \tag{19}$$

N_f is the number of features of the problem (i.e., the dimension of the concentrated log-likelihood function); $\text{rand}_{j,i} \sim U(0, 1)$, $CR \in [0, 1]$ is the crossover constant, and $I_{\text{rand}} \in [1, N_f]$ is a randomly chosen index, which ensures that $\mathbf{u}_{G+1,i} \neq \mathbf{x}_{G,i}$.

•**Selection**: Once vector $\mathbf{u}_{G+1,i}$ is created, the greedy criterion is applied

$$\mathbf{x}_{G+1,i} = \begin{cases} \mathbf{u}_{G+1,i} & \text{if } \bar{L}(u_{G+1,i}) > \bar{L}(x_{G,i}) \\ x_{G,i} & \text{otherwise} \end{cases} \tag{20}$$

The boundary conditions on $\mathbf{p}$ and $\boldsymbol{\theta}$ for the DE algorithm differ from those used for the PSO algorithm and are defined as follows

$$\begin{cases} \text{if } p_{G+1,i_k} > 2,\ p_{G+1,i_k} = 2 \\ \text{if } p_{G+1,i_k} < 0,\ p_{G+1,i_k} = 0 \end{cases} \quad i \in [1, N_p],\ j \in [1, N_f]$$
$$\text{if } \theta_{G+1,i_k} < 0,\ \theta_{G+1,i_k} = 0 \tag{21}$$

Although F and CR are used-defined values, we use the self-adapting parameter control proposed by Brest.[44] This algorithm has the advantage of defining the heavily problem-dependent parameters F and CR as a function of a different set of parameters, which, in contrast, are proved to be almost constant in a large range of different problems.

Following the work of Brest,[44] the following expressions for F and CR are obtained

$$F_{G+1,i} = \begin{cases} F_l + \text{rand}_1 \cdot F_u & \text{if } \text{rand}_2 < \tau_1 \\ F_{G,i} & \text{otherwise} \end{cases}$$
$$CR_{G+1,i} = \begin{cases} \text{rand}_3 & \text{if } \text{rand}_4 < \tau_2 \\ CR_{G,i} & \text{otherwise} \end{cases} \tag{22}$$

where F_l, F_u, τ_1, and τ_2 are the additional four parameters to be defined and $\text{rand}_i \sim U(0, 1)$, $i = 1, 2, 3, 4$. In the work of Brest, these parameters are set as $\tau_1 = \tau_2 = 0.1$, $F_l = 0.1$, and $F_u = 0.9$. These values can be used for every system, as the algorithm does not seem to be particularly sensitive to them.[44] The stopping criteria used for the DE algorithms are the same as those described for the PSO algorithm. The dimension of the population, that is, the number of DE vectors, is given by $10N_f$.[45]

Very similar to the PSO, the position of the maximum is considered to be found when the relative error between the best position $x_{G,\text{best}}$ at generation G and the best position at generation $G - 1$ is under the tolerance of $\varepsilon = 10^{-8}$ for $C = 20$ iterations.

As the concentrated log-likelihood must be calculated for each vector at each iteration of the algorithm, for a large value of N_f the linear scaling of the population dimension is undesirable, especially if we compare it to the square root scaling of the number of particles needed by PSO for a specific problem. For example, for a system where $N_f = 16$, the number of particles in PSO is 16, but the number of vectors needed by DE is $10N_f = 160$. Therefore, a dynamic reduction of population size is also implemented in the DE algorithm.[46] The reduction is performed at step G_R, which will be described later, and its frequency is controlled through the user defined parameter λ. The latter is set to ensure that the number of reductions does not exceed 5. It reads as follows

$$x_{G,i} = \begin{cases} x_{G,N_p/2+i} & \text{if } f(x_{G,N_p/2+i}) > f(x_{G,i}) \\ x_{G,i} & \text{otherwise} \end{cases} \quad i = 1, \ldots, \frac{N_{P_G}}{2}$$
$$N_{P_{G+1}} = \begin{cases} \frac{N_{P_G}}{2} & \text{if } G = G_R \\ N_{P_G} & \text{otherwise} \end{cases} \tag{23}$$

where it is assumed that N_{P_G} is even. The individuals of the first half of the current generation $x_{G,i}$ are compared with their equivalents in the second half of the current generation $x_{G,N_p/2+i}$, with the decision of which vector to retain based on the values of fitness function at the two positions (i.e., a greedy criterion). In case N_{P_G} is odd, the algorithm

DOI: 10.1021/acs.jctc.5b00936
J. Chem. Theory Comput. 2016, 12, 1499−1513

is straightforwardly changed by defining $N'_{P_G} = N_{P_G} + 1$ and putting $x_{G,N'_{P_G+1}} = x_{G,N_{P_G}}$, so eq 23 now reads as

$$x_{G,i} = \begin{cases} x_{G,N'_P/2+i} & \text{if } f(x_{G,N'_P/2+i}) > f(x_{G,i}) \\ x_{G,i} & \text{otherwise} \end{cases} \quad i = 1, \ldots, \frac{N'_{P_G}}{2} - 1$$

$$N'_{P_{G+1}} = \begin{cases} \frac{N'_{P_G}}{2} & \text{if } G = G_R \\ N_{P_G} & \text{otherwise} \end{cases} \tag{24}$$

The point with higher value of the fitness function is retained, whereas the other is discarded. G_R represents the step at which mod$(C, (kp_{max})) = 0$, where k is a counter that starts from 1 and is incremented by 1 each time the step G_R is met.

Derivative of the Concentrated Log-Likelihood Function. The L-BFGS-B algorithm is a quasi-Newton method for bounded problems that uses a limited amount of storage memory to approximate the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm. The bounded version of BFGS algorithm was chosen because of the constraints on $\boldsymbol{\theta}$ and $\mathbf{p}$. In order to use the L-BFGS-B algorithm, the analytical first derivative of the concentrated log-likelihood function with respect to $\boldsymbol{\theta}$ and $\mathbf{p}$ is needed. In the following derivation, the derivative with respect to $\boldsymbol{\theta}$ and $\mathbf{p}$ is reported, but because $\mathbf{p}$ is fixed to 2, the concentrated log-likelihood is optimized in $\boldsymbol{\theta}$ space only. Before deriving the first derivative, two useful properties of the derivative of a matrix are mentioned

$$\frac{\partial \ln|\mathbf{R}|}{\partial\gamma} = \mathrm{tr}\left(\mathbf{R}^{-1}\frac{\partial\mathbf{R}}{\partial\gamma}\right) \tag{25}$$

$$\frac{\partial\mathbf{R}^{-1}}{\partial\gamma} = -\mathbf{R}^{-1}\frac{\partial\mathbf{R}}{\partial\gamma}\mathbf{R}^{-1} \tag{26}$$

where $\frac{\partial\mathbf{R}}{\partial\gamma}$ is a matrix of elementwise derivatives and tr$(\bullet)$ is the trace of the matrix. The result shown in eq 25 has been derived in ref 47, and eq 26 can be obtained from letting the chain rule for differentiation operate on $\mathbf{1} = \mathbf{R}\mathbf{R}^{-1}$, yielding

$$\mathbf{0} = \frac{\partial\mathbf{1}}{\partial\gamma} = \left(\frac{\partial\mathbf{R}}{\partial\gamma}\right)\mathbf{R}^{-1} + \mathbf{R}\frac{\partial\mathbf{R}^{-1}}{\partial\gamma} \rightarrow \mathbf{R}^{-1}\left[\left(\frac{\partial\mathbf{R}}{\partial\gamma}\right)\mathbf{R}^{-1}\right]$$
$$= -\mathbf{R}^{-1}\left[\mathbf{R}\frac{\partial\mathbf{R}^{-1}}{\partial\gamma}\right] = -\frac{\partial\mathbf{R}^{-1}}{\partial\gamma}$$

where the latter equation is clearly identical to eq 26.

The derivative for the concentrated log-likelihood can be calculated term by term. In the first part of the derivation, we will make use of a variable γ to indicate indifferently either θ_j or p_j. Differentiating both sides of eq 5, and using eq 25, leads to

$$\frac{\partial\hat{L}(\gamma)}{\partial\gamma} = \frac{\partial}{\partial\gamma}\left(-\frac{N_t}{2}\ln(\hat{\sigma}^2) - \frac{1}{2}\ln(|\mathbf{R}|)\right)$$
$$= -\frac{N_t}{2\hat{\sigma}^2}\frac{\partial\hat{\sigma}^2}{\partial\gamma} - \frac{1}{2}\mathrm{tr}\left(\mathbf{R}^{-1}\frac{\partial\mathbf{R}}{\partial\gamma}\right) \tag{27}$$

The right-hand side of eq 27 is composed of two terms, the first of which can be made more explicit. By using the chain rule on $\hat{\sigma}^2$ in eq 4, we obtain

$$\frac{\partial\hat{\sigma}^2}{\partial\gamma} = \frac{\partial}{\partial\gamma}\left(\frac{(\mathbf{y} - \mathbf{1}\hat{\mu})^T\mathbf{R}^{-1}(\mathbf{y} - \mathbf{1}\hat{\mu})}{N_t}\right)$$
$$= \frac{1}{N_t}\left[\frac{\partial(\mathbf{y} - \mathbf{1}\hat{\mu})^T}{\partial\gamma}\mathbf{R}^{-1}(\mathbf{y} - \mathbf{1}\hat{\mu}) + (\mathbf{y} - \mathbf{1}\hat{\mu})^T\frac{\partial\mathbf{R}^{-1}}{\partial\gamma}(\mathbf{y} - \mathbf{1}\hat{\mu}) + (\mathbf{y} - \mathbf{1}\hat{\mu})^T\mathbf{R}^{-1}\frac{\partial(\mathbf{y} - \mathbf{1}\hat{\mu})}{\partial\gamma}\right]$$
$$= \frac{1}{N_t}\left[-\left(\mathbf{1}\frac{\partial\hat{\mu}}{\partial\gamma}\right)^T\mathbf{R}^{-1}(\mathbf{y} - \mathbf{1}\hat{\mu}) + (\mathbf{y} - \mathbf{1}\hat{\mu})^T\frac{\partial\mathbf{R}^{-1}}{\partial\gamma}(\mathbf{y} - \mathbf{1}\hat{\mu}) - (\mathbf{y} - \mathbf{1}\hat{\mu})^T\mathbf{R}^{-1}\left(\mathbf{1}\frac{\partial\hat{\mu}}{\partial\gamma}\right)\right] \tag{28}$$

where we use the fact that $\mathbf{y}$ does not depend on γ (i.e., any θ_j or p_j) and thus $\frac{\partial\mathbf{y}}{\partial\gamma} = \mathbf{0}$. The only derivative not yet worked out in eq 28 is $\frac{\partial\hat{\mu}}{\partial\gamma}$. The derivation again needs a substitution from eq 4 but this time of $\hat{\mu}$, and the fact that $\frac{\partial\mathbf{1}}{\partial\gamma} = \mathbf{0}$, yielding

$$\frac{\partial\hat{\mu}}{\partial\gamma} = \frac{\partial}{\partial\gamma}\left(\frac{\mathbf{1}^T\mathbf{R}^{-1}\mathbf{y}}{\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1}}\right)$$
$$= \left[\left(\mathbf{1}^T\frac{\partial\mathbf{R}^{-1}}{\partial\gamma}\mathbf{y}\right)(\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1}) - (\mathbf{1}^T\mathbf{R}^{-1}\mathbf{y})\left(\mathbf{1}^T\frac{\partial\mathbf{R}^{-1}}{\partial\gamma}\mathbf{1}\right)\right]\frac{1}{(\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1})^2} \tag{29}$$

The derivative of the inverse of the $\mathbf{R}$ matrix is given in eq 26, which introduces $\frac{\partial\mathbf{R}}{\partial\gamma}$. Thus, everything in eqs 27–29 is now written as a function of $\frac{\partial\mathbf{R}}{\partial\gamma}$, which is the last factor to be made explicit. Starting with the derivative of $\mathbf{R}$ with respect to $\boldsymbol{\theta} = \{\theta_1, \theta_2, \ldots, \theta_{N_f}\}$ gives, using the fact that $\frac{\partial\theta_k}{\partial\theta_h} = \delta_{kh}$ and that the features x are independent of θ

$$\frac{\partial R_{ij}}{\partial\theta_h} = \frac{\partial}{\partial\theta_h}\left(\exp\left[-\sum_{k=1}^{N_f}(\theta_k|x_k^i - x_k^j|^{p_k})\right]\right)$$
$$= -\sum_{k=1}^{N_f}\left[\frac{\partial\theta_k}{\partial\theta_h}|x_k^i - x_k^j|^{p_k} + \theta_k\frac{\partial}{\partial\theta_h}|x_k^i - x_k^j|^{p_k}\right]\left(\exp\left[-\sum_{k=1}^{N_f}(\theta_k|x_k^i - x_k^j|^{p_k})\right]\right)$$
$$= -|x_h^i - x_h^j|^{p_h}\left(\exp\left[-\sum_{k=1}^{N_f}(\theta_k|x_k^i - x_k^j|^{p_k})\right]\right) \tag{30}$$

DOI: 10.1021/acs.jctc.5b00936
J. Chem. Theory Comput. 2016, 12, 1499–1513

whereas for $\mathbf{p} = \{p_1, p_2, ..., p_{N_f}\}$, we find that

$$\begin{aligned}\frac{\partial R_{ij}}{\partial p_h} &= \frac{\partial}{\partial p_h}\left(\exp[-\sum_{k=1}^{N_f}(\theta_k|x_k^i - x_k^j|^{p_k})]\right)\\ &= -\sum_{k=1}^{N_f}\frac{\partial}{\partial p_h}\left(\exp[\ln\theta_k + p_k\ln|x_k^i - x_k^j|]\right)\\ &\quad\left(\exp[-\sum_{k=1}^{N_f}(\theta_k|x_k^i - x_k^j|^{p_k})]\right)\\ &= -\sum_{k=1}^{N_f}\exp[\ln\theta_k + p_k\ln|x_k^i - x_k^j|]\left(\frac{1}{\theta_k}\frac{\partial\theta_k}{\partial p_h} + \frac{\partial p_k}{\partial p_h}\right.\\ &\quad\left.\times\ln|x_k^i - x_k^j| + p_k\frac{\partial}{\partial p_h}\ln|x_k^i - x_k^j|\right)\\ &\quad\left(\exp[-\sum_{k=1}^{N_f}(\theta_k|x_k^i - x_k^j|^{p_k})]\right)\\ &= -\exp[\ln\theta_h + p_h\ln|x_h^i - x_h^j|](\ln|x_h^i - x_h^j|)\\ &\quad\left(\exp[-\sum_{k=1}^{N_f}(\theta_k|x_k^i - x_k^j|^{p_k})]\right)\\ &= -\theta_h|x_h^i - x_h^j|^{p_h}(\ln|x_h^i - x_h^j|)\\ &\quad\left(\exp[-\sum_{k=1}^{N_f}(\theta_k|x_k^i - x_k^j|^{p_k})]\right)\end{aligned} \tag{31}$$

where we used the identity

$$\theta_k|x_k^i - x_k^j|^{p_k} \equiv \exp[\ln\theta_k + p_k\ln|x_k^i - x_k^j|] \tag{32}$$

and the new facts that $\frac{\partial p_k}{\partial p_h} = \delta_{kh}$ and $\frac{\partial\theta_k}{\partial p_h} = 0$ and that the features x are independent of p. More information about the behavior of the concentrated log-likelihood in the neighbor of the stationary point could be obtained by considering its second derivative with respect to the hyperparameters $\boldsymbol{\theta}$ and $\mathbf{p}$.

The derivative of the log-likelihood has been obtained by Park and Baek[48] and used with a quasi-Newton optimization algorithm. However, Park and Baek did not consider the full derivative of the log-likelihood by setting the mean of the process, μ, constant.[48]

The L-BFGS-B algorithm is implemented through a library,[49] and as it solves a minimization problem, the function provided to it is $\bar{L}'(\boldsymbol{\theta}, \mathbf{p}, \sigma, \mu) = -\bar{L}(\boldsymbol{\theta}, \mathbf{p}, \sigma, \mu)$ with the gradient changed accordingly. The boundaries used for $\boldsymbol{\theta}$ are

$$\begin{cases}\theta_{L,i} = 0\\ \theta_{U,i} = 100\,000\end{cases} \quad \text{for } i = 1, ..., N_f \tag{33}$$

This equation actually confines a hypercube in N_f dimensions, which is a subset of $\mathbb{R}^{N_f}$. The algorithm is terminated when one of the following two conditions is fulfilled[36,49]

$$\frac{(\bar{L}'_k - \bar{L}'_{k+1})}{\max(|\bar{L}'_k|, |\bar{L}'_{k+1}|, 1)} \le f\varepsilon \tag{34}$$

$$\|Pg_k\|_2 \le g(1 + |\bar{L}'_k|) \tag{35}$$

where $\|\bullet\|_2$ is the Euclidean norm, $\bar{L}'_k$ and $\bar{L}'_{k+1}$ are the values of the function at the iteration k and $k + 1$, and Pg_k is the projected[50] gradient of the concentrated log-likelihood function at iteration k. The projection[51] is necessary since we are considering a bounded problem, and it represents the unique projection of vector $\boldsymbol{\theta}$ onto the subset defined by eq 32. In eqs 33 and 34, ε is the machine precision and the two dimensionless parameters f and g control the rate of convergence. In this work, we used values of $f = 1.0$ and $g = 10^{-7}$.

Systems Analyzed. QTAIM-obtained multipole moments (up to hexadecapole) on the central oxygen atom of three different sized water clusters (monomer, dimer, and hexamer) were selected as the property with which to test and compare the optimization algorithms. QTAIM partitions a molecular system into a collection of space-filling nonoverlapping topological atoms.[4,52] Atom-centered multipole moments can then be used to accurately model the interatomic electrostatic interactions within the system. As the electron density of a molecule is a function of the configuration of the atoms it contains,[53] such multipole moments are well-suited to machine learning techniques such as kriging[54] and start showing a track record of being predicted accurately.[6,23,55]

To determine the robustness of the optimization algorithms on systems of varying dimensionality, the water monomer, a water dimer, and a water hexamer were evaluated with the number of features in each kriging model being equal to $3N_{atoms} - 6$. The water clusters were sampled by selecting the nearest neighbors of a selected water molecule at intermittent snapshots of a previous molecular dynamics simulation.[56] The simulations were completed as rigid-body simulations using the program DL_POLY_2.0[57] using a relatively low "quaternion tolerance" parameter in order to increase computational efficiency. This has allowed slight fluctuations in the bond lengths and angles of the individual monomers within each cluster, a situation we chose not to correct as we considered the added complexity of extra degrees of freedom a more thorough test for the optimization algorithms. Alternatively, the set of geometries pertaining to the monomer was obtained via random distortion of the molecule's normal modes of vibration, as outlined by Ochterski[58] and utilized by Fletcher et al.[55a] and Hughes et al.[59] A maximum bond and angle stretch factor of 1.2 was applied in order to ensure distorted structures were physically reasonable. The wave function for each cluster configuration was calculated using the program Gaussian 09[60] at the B3LYP/6-311++G(d,p) level of theory. The program AIMAll[61] was then used to obtain the atomic multipole moments using default settings and integration error control. Important cluster quantities are displayed in Table 1.

As stated, the configuration of the atoms in each cluster was used for the features of the kriging model. For generality, the configuration of the atoms was described through the atomic local frame (ALF), which was installed on the central oxygen atom. The ALF is a right-handed coordinate system centered on the oxygen of interest, from which every other atom's position can be defined using spherical polar coordinates. The first two features describing the atomic configuration correspond to the intramolecular OH bond lengths of the central molecule, whereas the third feature represents the central molecule's HOH angle. From the fourth feature onward, each atom A in the system is represented by three spherical coordinates, denoted d_A, ζ_A, and ϕ_A, where d_A is the distance from the central oxygen to atom A, ζ_A is the polar angle measured from the ALF's Z axis, and ϕ_A represents the azimuthal angle measured from the X axis. Finally, features corresponding to

DOI: 10.1021/acs.jctc.5b00936
J. Chem. Theory Comput. 2016, 12, 1499−1513

Table 1. Summary of Some of the Important Physical Properties of the Investigated Systems

		intramolecular OH length (Bohr)		intramolecular HOH angle (rads)		largest OO distance (Bohr)	
no. molecules	no. features	mean	SD	mean	SD	mean	SD
1	3	1.792	0.166	1.8555	0.0956	N/A	
2	12	1.811	0.001	1.8233	0.0006	5.1294	0.5959
6	48	1.811	0.001	1.8233	0.0006	6.3407	0.4470

Table 2. Number of Times the Fitness Function Is Calculated for Every System Considered and for Every Training Set Size[a]

	dimer						
	150	300	600	800	1000	2000	3000
DE-MS0	13238 ± 5515	10949 ± 2980	18771 ± 8784	16733 ± 9328	13853 ± 5045	19811 ± 10781	13895 ± 6863
DE-MS1	11693 ± 2506	16100 ± 9765	19813 ± 6659	21030 ± 14499	22618 ± 14128	17733 ± 9796	20904 ± 15550
DE-MS2	10151 ± 2891	12353 ± 5574	16688 ± 6352	17324 ± 3850	17746 ± 9826	19062 ± 6201	15091 ± 8629
DE-MS3	10755 ± 5973	11253 ± 2501	12207 ± 2164	11339 ± 5516	11838 ± 6541	11286 ± 4238	12797 ± 8435
DE-MS4	7387 ± 3278	8507 ± 2835	7653 ± 4538	6874 ± 4521	7253 ± 4193	7152 ± 3651	4712 ± 2732
PSO	3429 ± 374	3009 ± 744	3711 ± 1344	3244 ± 497	3524 ± 1282	3092 ± 395	3159 ± 344
	hexamer						
	200	500	700	1000	1500	2000	3000
DE-MS0	164382 ± 133288	125928 ± 82162	149010 ± 56724	119568 ± 58790	170916 ± 83262	118818 ± 57727	92100 ± 69433
DE-MS1	189276 ± 92591	177222 ± 112254	143031 ± 76992	103770 ± 43503	178734 ± 87822	130290 ± 54081	122190 ± 52324
DE-MS2	57768 ± 41448	38079 ± 16443	47526 ± 19849	38640 ± 12815	30378 ± 13618	32127 ± 11485	59640 ± 28095
DE-MS3	73470 ± 28453	64869 ± 46649	64683 ± 39224	75786 ± 37360	66483 ± 41985	68178 ± 44268	78471 ± 52223
DE-MS4	30708 ± 12030	20514 ± 5959	25614 ± 12497	28173 ± 8619	28398 ± 8492	32574 ± 11004	24030 ± 7732
PSO	58440 ± 13344	52744 ± 11871	57017 ± 14541	44645 ± 9760	43474 ± 12053	42845 ± 13669	34591 ± 11303

[a]The results are shown as a mean accompanied with the standard deviation averaged over 10 runs.

atoms of the same water molecule are kept adjacent in the training set, ordered by increasing OO distance.

RESULTS AND DISCUSSION

Multiple kriging models with different training set sizes were created for each of the three different systems analyzed. The training set sizes used consisted of 50, 150, 300, 600, 1000, 2000, and 3000 training points for the monomer; 150, 300, 600, 800, 1000, 2000, and 3000 for the dimer; and 200, 500, 700, 1000, 1500, 2000, and 3000 for the hexamer. This gives a total of 21 models to be optimized for the charge. We also considered the other components of the multipole expansion up to hexadecapole (i.e., 24 more systems) with 150, 300, 3000 training points for the dimer and 200 and 500 training points for the hexamer, for a total of 141 (= 21 + 3 × 24 + 2 × 24) models to be optimized. For the rest of this article, abbreviations will be used to identify the various systems with different training set sizes: MON*x* represents the monomer with a training set of size *x*, with DIM*x* representing the dimer and HEX*x* representing the hexamer.

Initially, the performance of the DE and PSO in terms of rate of convergence was compared. Here, rate of convergence means the number of times the fitness function is evaluated before the solution reaches the user-imposed tolerance. This metric, different from the commonly used metric of the number of solution iterations required to reach the tolerance, has been chosen as it accounts for the variable number of vectors in the DE algorithms and the difference between the number of particles and vectors between the PSO and DE algorithms, respectively.

Table 2 reports the total number of concentrated log-likelihood evaluations for the dimer and hexamer at each training set size. The results are averaged over 10 different runs with random initialization for each run. Table 2 shows that, in terms of the number concentrated log-likelihood evaluations, PSO outperforms DE, regardless of mutation strategy. The only exception is HEX with MS4, where PSO requires about twice as many evaluations, and MS2. MS2 mostly outperforms PSO as well. Similar results were obtained for MON, which are given in Table S1 of the Supporting Information. It was shown,[62] in a different numerical benchmark problem, that DE performs better than other optimization algorithms such as PSO, in terms of the number of iterations. Results shown here agree with these findings, if we consider the number of iterations only (Table S2). The number of iterations of DE is comparable to that of PSO. On average, the latter performs better for the monomer and for some training set sizes in the water dimer problem (e.g., DIM1000; Table S2). However, as expected, DE always outperforms PSO for higher dimensional problems. In this work, every time the concentrated log-likelihood is calculated, we need to build the **R** matrix (which we recall has dimension $N_t \times N_t$), perform a matrix inversion, and compute several vector and matrix multiplications. Therefore, the number of iterations is not indicative of the time spent on a single run, since the number of evaluations of the concentrated log-likelihood is the most expensive part of the calculation. This, in turn, is dependent on both the number of iterations and the number of vectors in DE or particles in PSO.

In this work, we used DE and PSO parameters suggested in the literature.[40,41,44,45,63] These values were tested on a number of different test functions to compare their performances in different situations that can arise in a real case (e.g., multiple local minima, large plateau, shallow global minima; see ref 62). Fine-tuning of these parameters is beyond the scope of this work but could potentially result in different rates of convergence. In fact, in the case of DE, the convergence rate seems to be highly dependent upon the choices of *F*, *CR*, and the

DOI: 10.1021/acs.jctc.5b00936
J. Chem. Theory Comput. 2016, 12, 1499−1513

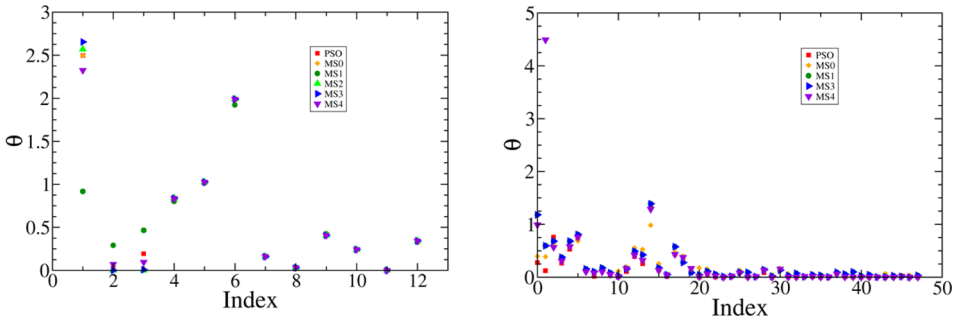


Figure 1. Comparison between results from PSO and DE (with different mutation strategies) in terms of θ values for the component Q_{00} of the multipole expansion on oxygen in DIM150 (left) and the component Q_{00} of the multipole expansion on oxygen in HEX500 (right).

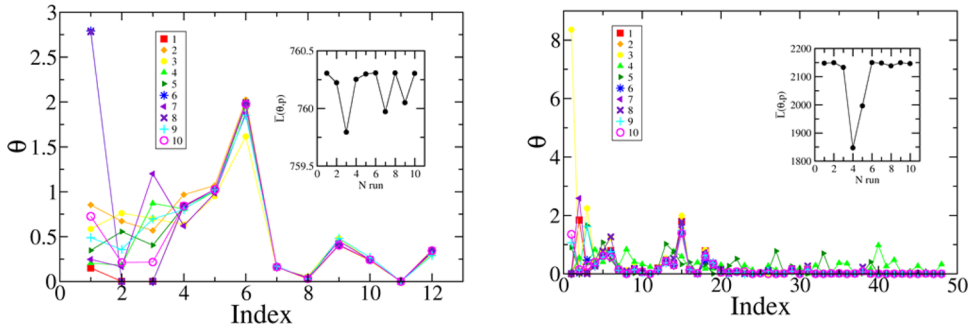


Figure 2. Theta values obtained for each of the 10 DE-MS1 runs for the component Q_{00} of the multipole expansion on oxygen in DIM150 (left) and the component Q_{00} of the multipole expansion on oxygen in HEX500 (right). The inset shows the final concentrated log-likelihood for each run. Lines connecting the data points serve to guide the eye.

mutation strategy.[64] The fine-tuning for each particular problem usually requires expensive trial and error estimation. However, in the recent literature, different choices of these parameters were proposed, and some of these methods are reported here, even if we do not mean to exhaustively review them. Suggested decision methods use fuzzy logic,[65] random generation,[63] self-adaptive design,[66] and many others.[64,67] Population size also has a great impact on the performance of DE. Different experiments with the population size N_p ranging between $3N_f$ and $40N_f$ are reported with different claims over rate of convergence and overall performances.[19,68] Different choices of parameter, moving strategies, and number of particles were also proposed for PSO, such as the landscape adaptive PSO[69] and ladder PSO,[70] or by considering the inertia weight and acceleration coefficients as functions of the local best and global best of the fitness function.[71]

One potential problem with a search algorithm is that the final point that it reaches could be different for the same training set, depending on the algorithm's initial position in search space. This behavior can be ascribed to a few causes, for example, the fitness function might be flat (relative to the user defined tolerance) and the algorithms might not be able to localize the stationary point, or the fitness function might be multimodal with the algorithms becoming trapped in local maxima. This behavior is not desirable. Indeed, a single answer should be obtained for any given training set no matter which search algorithm is used and which starting positions it has.

Figure 1 compares the results between PSO and DE for the component Q00 of the multipole expansion on oxygen in DIM150 (left panel) and for the component Q00 of the multipole expansion on oxygen in HEX500 (right panel). Results from PSO and DE show that final θ values can be very different between different runs, at least for some features (e.g., feature 1 for DIM and feature 2 for HEX in Figure 1). Therefore, even if the values of the concentrated log-likelihood are similar in each case, there is a large interval in which θ values can eventually vary. By looking at results in Figure 1 only, we cannot exclude the existence of more than one maximum for the concentrated log-likelihood.

First, we consider the value of the concentrated log-likelihood found and the corresponding θ values for DE-MS1, for the component Q00 of the multipole expansion on oxygen in DIM150, and the component Q00 of the multipole expansion on oxygen HEX500, which are shown in Figure 2. In the case of DIM150, 7 of the 10 runs (as can be counted from the inset of Figure 2) show similar results in terms of the maximum of the concentrated log-likelihood function. Second, all θ values, except for the first four features, are very close together for DIM150. In three cases (third, seventh, and ninth runs), the maximum of the concentrated log-likelihood is sensibly different from the other runs. For these three runs, most θ values are significantly different from those of the other runs. For HEX500, the 10 runs show more homogeneity, with the final concentrated log-likelihood value similar in almost all of the cases shown. In fact, for HEX500, only the fourth run shows a sensibly different value of the concentrated log-likelihood compared to the others. The fact that results for DIM150 seem to be less stable (in the sense of the final value of the concentrated log-likelihood) than those for HEX500 could be due to the higher number of dimensions in the HEX system.

The only rigorous way to prove that the final points, found by the DE and PSO optimization algorithms, are true stationary points is to invoke the analytical first derivative and check if it is zero. Here, we use the L-BFGS-B algorithm to find the true stationary point, using the positions given by the PSO and DE algorithms for initial points. As illustrated by feature 1 in

DOI: 10.1021/acs.jctc.5b00936
J. Chem. Theory Comput. 2016, 12, 1499−1513

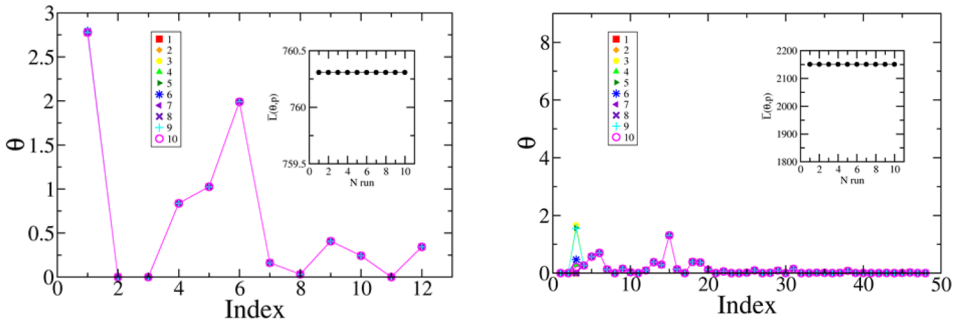


Figure 3. Theta values obtained through refinement with L-BFGS-B, for each of the 10 DE-MS1 runs for the component Q_{00} of the multipole expansion on oxygen in DIM150 (left) and Q_{00} on oxygen in HEX500 (right). The inset shows the final concentrated log-likelihood for each. Lines connecting the data points serve to guide the eye.

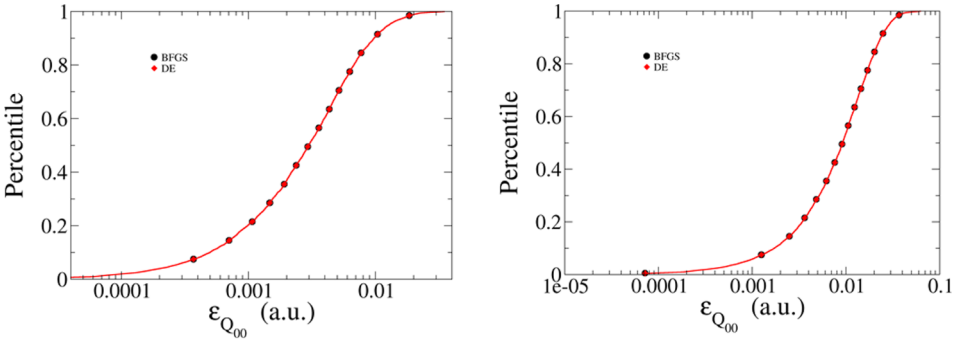


Figure 4. Absolute error in evaluation of the charge (Q_{00}) on oxygen in the water dimer (left) and hexamer (right). The predictions were obtained with theta values reported in Figure 2 for the seventh run for the dimer and for the fifth run for the hexamer, and the theta values shown in Figure 3. Predictions are obtained over 6000 points.

DIM150 (Figure 2, left panel), there are two possible situations. Either the stationary points to be located by L-BFGS-B (i) correspond to different values of θ and hence the concentrated log-likelihood has multiple local maxima or (ii) they are the same for each starting point (i.e., the output of DE or PSO optimization). Note that situation (ii) does not exclude the existence of local maxima of the concentrated log-likelihood; it simply means that both PSO and DE are able to actually "find" the same maximum (or arrive close to it at least), which is what we are interested in.

Figure 3 shows θ values obtained through the L-BFGS-B algorithm, using as initial points the system's θ values reported in Figure 2. For DIM150, all θ values converge toward the same point in each case considered (including, most notably, all of the runs with very different concentrated log-likelihood shown in Figure 2). For HEX500, all features but the third converge. Although for DIM150 it seems reasonable to assume that both DE and PSO are able to reach the same stationary points, for HEX500, there is still the possibility that more than one maximum exists. However, as shown in the inset of Figure 2 for the hexamer, in all cases refined through the L-BFGS-B algorithm, the final value of the concentrated log-likelihood is practically equal. We will come back to this issue later. Figures S1−S5 report the theta values obtained from PSO, DE-MS0, DE-MS2, DE-MS3, and DE-MS4 for DIM150, respectively. In each figure, the right panel shows the corresponding refinement through L-BFGS-B. Figures S6−S10 report the theta values obtained using PSO, DE-MS0, DE-MS2, DE-MS3, and DE-MS4, respectively, for HEX500, again with L-BFGS-B in the right panels. Figures S11−S16 report the theta values obtained using PSO, DE-MS0, DE-MS1, DE-MS2, DE-MS3, and DE-MS4 for MON50, respectively, with L-BFGS-B in the right panels.

As shown, the refinement using L-BFGS-B is able to eliminate (or at least reduce) the apparent arbitrariness in the chosen theta values for the prediction of the properties that we are interested in. However, nothing was said about the performance of the DE, PSO, or L-BFGS-B in terms of its ability to obtain correct predictions until now. In comparing the performances between the search algorithms, we define the absolute prediction error as follows

$$\varepsilon_y = |y(x) - \hat{y}(x)| \tag{35}$$

where $y(x)$ is the real value of the property at the point x (e.g., $y = Q_{00}$ the charge of the atom), whereas $\hat{y}(x)$ is the value of the property predicted by kriging at the same point.

Figure 4 reports the error on the charge (Q_{00}, in a.u.) on the oxygen atom in the DIM150 for the model obtained with DE (MS1) and after refinement with L-BFGS-B. It is seen that there is no appreciable difference between the quality of the predictions of the two models. This is due to the fact that both σ^2 and μ, shown in Table 3, are already very low for the DE

Table 3. Average and Variance of the Kriging Process Obtained through DE and Its Relative Refinement with L-BFGS-B.

	μ	σ^2
DIM150, DE-MS1	−1.1061	0.0005
DIM150, L-BFGS-B	−1.1063	0.0004
HEX500, DE-MS1	−1.1673	0.0006
HEX500, L-BFGS-B	−1.1586	0.0006

DOI: 10.1021/acs.jctc.5b00936
J. Chem. Theory Comput. 2016, 12, 1499−1513

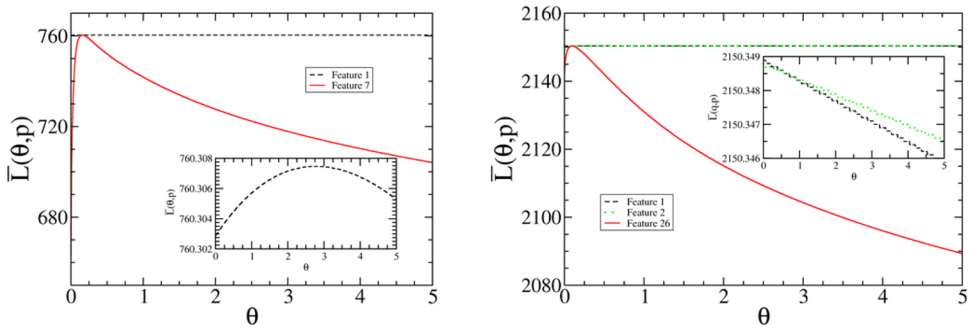


Figure 5. Concentrated log-likelihood for DIM150 (left) and HEX500 (right) as a function of a single feature. In the inset, the log-likelihood is shown only for feature 1 for DIM150 and features 1 and 2 for HEX500, with the *y* axis rescaled to appreciate the variation in the range considered.

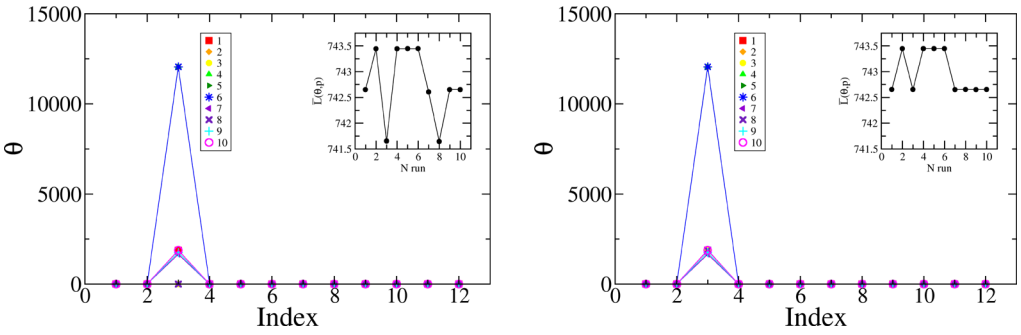


Figure 6. Theta values for each of the 10 PSO runs for DIM150 (left) and refinement of the same runs through L-BFGS-B (right) of the component Q_{11c} of the multipole expansion on oxygen in the water dimer. The inset shows the final concentrated log-likelihood for each run.

model and the further refinement obtained by the L-BFGS-B does not give substantial increase in the prediction performance.

These results, with no increase in the performance of predictions and with the residual arbitrariness on the first three features of HEX500, can be explained by considering the behavior of the concentrated log-likelihood function along the individual features. It can be seen in Figure 2 that the difference in the value of the concentrated log-likelihood between different optimization runs can be small despite certain features (e.g., feature 1, DIM150; Figure 2) being significantly varied. Conversely, all runs, including the run with the worst concentrated log-likelihood, return very similar values for other features (e.g., feature 3, DIM150; Figure 2). To investigate the variation of the concentrated log-likelihood across each feature, we froze all of the features but one and calculated the value of the concentrated log-likelihood as a single-variable function, starting from the maximum found by L-BFGS-B. Figure 5 displays the two features with the highest and lowest variance, respectively. Figure S17 shows the concentrated log-likelihood for DIM-150 as a function of a single feature (of a total of 12).

For DIM150, the concentrated log-likelihood in the θ_1 direction varies little relative to the θ_7 direction, which has a very sharp maximum. For HEX500, the situation is even worse: in the range of θ considered, the variation of θ_1 is 0.005% of the variation of θ_{26}. A similar plot is shown for HEX500, where the behavior of features θ_1, θ_2, and θ_{26} is reported and a similar conclusion can be drawn. The reason for the great variation of the final values of θ and $\bar{L}$ is not due to multimodality of the concentrated log-likelihood function but rather to the lack of ability of PSO and DE to reach the exact position of the stationary point of the function, which can be due to the fact that the concentrated log-likelihood has some directions in which it is almost flat. This makes the search very difficult even for the most elaborate algorithms.

Long ridges in the concentrated log-likelihood were first shown by Warnes and Ripley.[72] It is commonly acknowledged that a plateau in the log-likelihood leads to an increase in the difficulty in attaining the maximum for optimization algorithms.[24,35] In the case reported in Figure 5, the optimization algorithm used seems to be able to overcome the plateau problem. In the case of the charge, where a single maximum for the log-likelihood seems to exist, all of the algorithms considered were able to converge to same point, possibly with a refinement given by the L-BFGS-B.

The last problem in the optimization of the log-likelihood is related to its multimodality. Mardia and Watkins[73] studied multimodality of the concentrated log-likelihood (for a spherical correlation scheme) and stated that the multimodality could be due to lack of double differentiability of the kernel with respect to the hyperparameters. The power exponential kernel considered in this work is at least twice differentiable, with respect to either θ or p, thus the multimodality of the concentrated log-likelihood function will not occur as a problem for this reason. However, it seems that a definitive answer to the multimodality of the concentrated log-likelihood does not exist yet. Warnes and Ripley[72] and Ripley[74] reported an example where the log-likelihood calculated with the exponential kernel (i.e., with $p = 1$) is multimodal. Stein[24] disputed the correctness of these results for the exponential kernel and stated that, regardless of the examples of multimodality of the log-likelihood reported by Ripley,[74] the models give the same predictions.

There is still no way to know in advance if the log-likelihood in a specific problem will present multimodality, increasing the difficulty of this optimization problem. Figure 6 reports the theta values for each of the 10 runs of the dipole component

DOI: 10.1021/acs.jctc.5b00936
J. Chem. Theory Comput. 2016, 12, 1499−1513

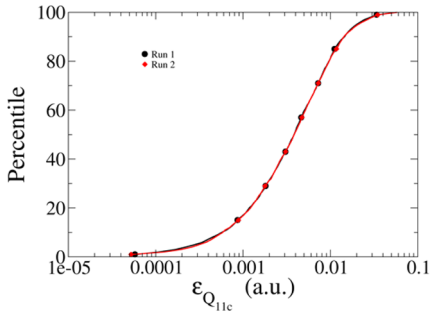


Figure 7. Absolute error in evaluation of the component Q_{11c} of the multipole expansion on oxygen in the water dimer. The predictions were obtained with theta values reported in Figure 6 for the first run (black) and second run (red). Predictions are obtained over 6000 points.

Q_{11c} of the multipole expansion obtained through PSO before being refined by L-BFGS-B. In the case reported here, the theta values obtained with PSO are different, as is the log-likelihood. Here, it is seen through refinement by BFGS that the log-likelihood is actually different (see inset of Figure 6, right), even if this difference is less than 0.002%. The difference between theta values in the case of Q_{11c} cannot be explained as long ridges in the log-likelihood.

As reported before, Stein[24] pointed out that small differences in maxima of the log-likelihood lead to kriging models that give almost the same predictions. Figure 7 reports the predictions obtained using the two sets of theta values shown in Figure 6, corresponding to the two maxima (by choosing the theta values corresponding to the first and second runs).

As seen in Figure 7, the absolute errors in the prediction of Q_{11c} by the two runs (i.e., models) coincide. Despite there being different sets of hyperparameters, the models can be considered equivalent in practice (i.e., the predictions). The existence of different sets of theta values giving the same predictions, although correct in principle, introduces some arbitrariness in the construction of the kriging models. When kriging is used to build a model for a physical system, the theta values have a physical meaning: they represent the inverse of a characteristic length scale between two points, i.e., the maximum distance to which the influence of a point extends.[29] Thus, we would expect a unique set of optimal theta values for each problem. Drton and Richardson[75] apply log-likelihood parameter estimation to the seemingly unrelated regression (SUR) model[76] to prove that the multimodality of the log-likelihood disappears in the limit of large sample size. Despite Drton and Richardson's work being applied to the log-likelihood of the SUR model, the fact that the multimodality is more likely to occur with small samples suggests that such a situation is also likely to apply in the case analyzed here.

Figure 8 displays the results of 10 runs for the DIM300 system for the same multipole moment, Q_{11c}. There seems to be no multiple maximum anymore, and the PSO is again able to consistently find the same maximum in all 10 runs. It is important to highlight here that 150 training points for a 12-dimensional system is still a rather low number of training points for this kind of problem. Moreover, the multimodality disappears with just 300 training points, which is still a fairly low number.

Finally, in this work, we considered PSO and DE algorithms with literature-specified parameters. It is possible that a better tuning of those parameters (e.g., different acceleration coefficients for PSO to prevent the premature convergence of the swarm) could also increase the probability that PSO and DE will converge toward the same point regardless of initialization conditions. However, such analysis is beyond the scope of this article.

Other components of the multipole expansion, along with the predictions in the case of multimodality of log-likelihood and the comparison with the case DIM300 are reported in Figures S18−S32. DE algorithms showed similar behavior, with some difference given by the mutation strategy. The log-likelihood for the case DIM150 is reported in the Supporting Information, along with the refining with L-BFGS-B, in Figures S33−S57. The largest case we considered, i.e., the hexamer, is shown in Figures S58−S69. The hexamer is characterized by a higher number of features with respect to the dimer, and it is more sensitive to a low number of training points. As such, results show that the performance of PSO and DE seems to degrade for low training point models. While these results seem to suggest that use of L-BFGS-B is necessary to obtain a reasonable maximum for such high number-of-feature systems, note that the number of training points required for the degradation of PSO and DE to occur is lower than what would be usually recommended for systems of such size.

CONCLUSIONS

In this work we have shown how PSO and DE can be successfully used to optimize the concentrated log-likelihood function, which gives the parameters required to build the kriging model for the system that we are interested in. Here, the property of interest was the QTAIM-obtained multipole moments of the oxygen of a central water molecule in a water cluster, but the method is general and can be applied to other properties.

The use of PSO and DE for the training of kriging models is appealing: even the largest problem discussed here (i.e., water

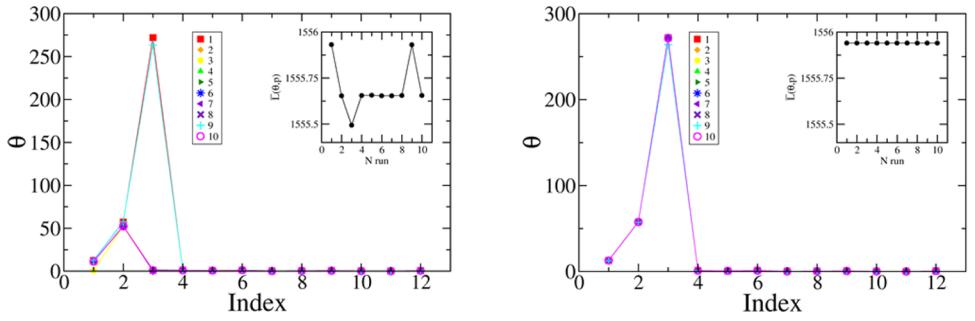


Figure 8. Theta values for each of the 10 PSO runs for DIM300 (left) and refinement of the same runs through BFGS (right) of Q_{11c} on oxygen in the water dimer. The inset shows the final concentrated log-likelihood for each run.

DOI: 10.1021/acs.jctc.5b00936
J. Chem. Theory Comput. 2016, 12, 1499−1513

hexamer, 48 dimensions, 3000 training points) can be easily modeled. The great advantage of PSO and DE is that they are able to explore the $\boldsymbol{\theta}-\boldsymbol{p}$ space quicker than an analytical algorithm and at the same time are able to reach the maximum of the concentrated log-likelihood.

Moreover, the approach described here is not confined to computational chemistry. In fact, the framework we presented is very general. From engineering to geostatistics, kriging is employed in a diverse array of fields, yet it struggles with the recurring problem of finding the best trade-off between computational time spent to train the kriging model and the accuracy of the model itself (which is roughly proportional to the training set size). We showed that PSO and DE give consistent results for each training set size considered, possibly refining the final solution with L-BFGS-B. Therefore, by using PSO and DE optimization techniques, the optimization of highly multidimensional problems that could need thousands of training points is no longer a concern.

One question that arises is whether analytical methods such as L-BFGS-B are really needed or if stochastic methods such as PSO and DE are able to correctly handle the search. Here, it is shown that stochastic methods can locate stationary points to a reasonable accuracy, with most uncertainty obtained for features that do not seem to be important for the model (i.e., the one for which the concentrated log-likelihood is practically constant over the domain). However, L-BFGS-B proved to be useful in this work to compare the performance of PSO and DE. As the maximum of the concentrated log-likelihood is not known *a priori*, there is no way to tell if the final position found by PSO and DE was actually the global maximum without other information, a situation complicated by the fact that both PSO and DE seem to give different answers in different runs with the same initial conditions. L-BFGS-B demonstrates that this difference was due to the flatness of the concentrated log-likelihood in some directions and that the results obtained by PSO and DE were still practically useful.

Therefore, we can conclude that PSO and DE are able to come close to the same maximum of the concentrated log-likelihood. However, they do not seem to be able to reach the exact stationary point without refinement through L-BFGS-B algorithm when the log-likelihood presents some long ridges or different maxima. Moreover, the ability of PSO and DE to efficiently explore all of the hyperparameter space is assessed. Moreover, the starting position for DE and PSO is not important because, after the refinement through L-BFGS-B, the same final position is reached.

Having assessed the correctness of the results of PSO and DE, future work will be devoted to deciding the optimum means of selecting training points in order to improve prediction performance. This work will also consider the chemical meaning of the theta values and their connection to the chemical environment of the problem that we are attempting to describe. In view of this, a refinement of the training method with a focus on dropping useless information will also be considered.

APPENDIX: NORMALIZED FEATURES

The normalized coordinates are defined as

$$\tilde{x}_i = \frac{x_i - x^m}{x^M - x^m} \tag{36}$$

where $x^M = \max_{i\in[1,\,N_i]}(x_i)$ and $x^m = \min_{i\in[1,\,N_i]}(x_i)$. We can then rewrite eq 2 in the main text as

$$\begin{aligned} R_{ij} &= \exp\left[-\sum_{k=1}^{N_f}\theta_k|(x_k^i - x_k^j + 0)\times 1|^{p_k}\right] \\ &= \exp\left[-\sum_{k=1}^{N_f}\theta_k\left|(x_k^i - x_k^j + x_k^m - x_k^m)\left(\frac{x_k^M - x_k^m}{x_k^M - x_k^m}\right)\right|^{p_k}\right] \\ &= \exp\left[-\sum_{k=1}^{N_f}\theta_k|x_k^M - x_k^m|^{p_k}\left|\frac{x_k^i - x_k^j + x_k^m - x_k^m}{x_k^M - x_k^m}\right|^{p_k}\right] \\ &= \exp\left[-\sum_{k=1}^{N_f}(\theta_k|x_k^M - x_k^m|^{p_k})\left|\frac{x_k^i - x_k^m}{x_k^M - x_k^m} - \frac{x_k^j - x_k^m}{x_k^M - x_k^m}\right|^{p_k}\right] \\ &= \exp\left[-\sum_{k=1}^{N_f}\tilde{\theta}_k|\tilde{x}_k^i - \tilde{x}_k^j|^{p_k}\right] \end{aligned} \tag{37}$$

ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge on the ACS Publications website at DOI: 10.1021/acs.jctc.5b00936.

Number of times the concentrated log-likelihood function is calculated for the water monomer, for each kriging hyperparameter search algorithm, and for every training set size considered; number of iterations for every system studied with each kriging hyperparameter search algorithm for every training set size considered; Q_{00} theta values for DIM150 PSO, DIM150 DE-MS0, DIM150 DE-MS2, DIM150 DE-MS3, DIM150 DE-MS4, HEX500 PSO, HEX500 DE-MS0, HEX500 DE-MS2, HEX500 DE-MS3, HEX500 DE-MS4, MON50 PSO, MON50 DE-MS0, MON50 DE-MS1, MON50 DE-MS2, MON50 DE-MS3, and MON50 DE-MS4; DIM150 theta values for Q_{10}, Q_{11c}, Q_{11s}, Q_{20}, Q_{21c}, Q_{21s}, Q_{22c}, Q_{22s}, Q_{30}, Q_{31c}, Q_{31s}, Q_{32c}, Q_{32s}, Q_{33c}, Q_{33s}, Q_{40}, Q_{41c}, Q_{41s}, Q_{42c}, Q_{42s}, Q_{43c}, Q_{43s}, Q_{44c}, and Q_{44s}; concentrated log-likelihood for DIM150 as function of a single feature; absolute error in the evaluation of components Q_{31c} and Q_{32s}; log-likelihood of 10 DE-MS0, DE-MS1, DE-MS2, DE-MS3, and DE-MS4 runs for DIM150, HEX200, and HEX500 for different multipole components (PDF)

AUTHOR INFORMATION

Corresponding Author

*E-mail: paul.popelier@manchester.ac.uk. Tel.: +44 161 3064511.

Funding

We are grateful to the EPSRC for the award of an Established Career Fellowship (EP/K005472), which funds all three authors.

Notes

The authors declare no competing financial interest.

REFERENCES

(1) Darley, M. G.; Popelier, P. L. A. *J. Phys. Chem. A* **2008**, *112*, 12954.

(2) Popelier, P. L. A. *Int. J. Quantum Chem.* **2015**, *115*, 1005.

DOI: 10.1021/acs.jctc.5b00936
J. Chem. Theory Comput. 2016, 12, 1499−1513

(3) Cardamone, S.; Hughes, T. J.; Popelier, P. L. A. *Phys. Chem. Chem. Phys.* **2014**, *16*, 10367.
(4) Bader, R. F. W. *Atoms in Molecules: A Quantum Theory*; Oxford University Press: Oxford, 1990.
(5) Cressie, N. *Statistics for Spatial Data*; Wiley, New York, 1993.
(6) Mills, M. J. L.; Hawe, G. I.; Handley, C. M.; Popelier, P. L. A. *Phys. Chem. Chem. Phys.* **2013**, *15*, 18249.
(7) Stone, A. *The Theory of Intermolecular Forces*; Oxford University Press: Oxford, 2013.
(8) Gresh, N.; Cisneros, G. A.; Darden, T. A.; Piquemal, J.-P. *J. Chem. Theory Comput.* **2007**, *3*, 1960.
(9) Vinter, J. G. *J. Comput.-Aided Mol. Des.* **1994**, *8*, 653.
(10) Gordon, M. S.; Slipchenko, L.; Li, H.; Jensen, J. H. *Annu. Rep. Comput. Chem.* **2007**, *3*, 177.
(11) Ponder, J. W.; Wu, C.; Ren, P.; Pande, V. S.; Chodera, J. D.; Schnieders, M. J.; Haque, I.; Mobley, D. L.; Lambrecht, D. S.; DiStasio, R. A. J.; Head-Gordon, M.; Clark, G. N. I.; Johnson, M. E.; Head-Gordon, T. *J. Phys. Chem. B* **2010**, *114*, 2549.
(12) Holt, A.; Boström, J.; Karlström, G.; Lindh, R. *J. Comput. Chem.* **2010**, *31*, 1583.
(13) Price, S. L.; Leslie, M.; Welch, G. W. A.; Habgood, M.; Price, L. S.; Karamertzanis, P. G.; Day, G. M. *Phys. Chem. Chem. Phys.* **2010**, *12*, 8478.
(14) Behler, J.; Parrinello, M. *Phys. Rev. Lett.* **2007**, *98*, 146401.
(15) Eshet, H.; Khaliullin, R. Z.; Kuhne, T. D.; Behler, J.; Parrinello, M. *Phys. Rev. B: Condens. Matter Mater. Phys.* **2010**, *81*, 184107.
(16) (a) Bartok, A. P.; Gillan, M. J.; Manby, F. R.; Csanyi, G. *Phys. Rev. B: Condens. Matter Mater. Phys.* **2013**, *88*, 054104. (b) Bartok, A. P.; Payne, M. C.; Kondor, R.; Csanyi, G. *Phys. Rev. Lett.* **2010**, *104*, 136403.
(17) Hansen, K.; Montavon, G.; Biegler, F.; Fazli, S.; Rupp, M.; Scheffler, M.; von Lilienfeld, O. A.; Tkatchenko, T.; Mueller, K. *J. Chem. Theory Comput.* **2013**, *9*, 3404.
(18) Kennedy, J.; Eberhart, R. C. *Proceedings of the IEEE Int. Conf. on Neural Networks* **1995**, *4*, 1942.
(19) Storn, R.; Price, K. *J. Global Opt.* **1997**, *11*, 341.
(20) Byrd, R. H.; Nocedal, J.; Schnabel, R. B. *Mathematical Programming* **1994**, *63*, 129.
(21) Sacks, J.; Welch, W. J.; Mitchell, T. J.; Wynn, H. P. *Statistial Science* **1989**, *4*, 409.
(22) (a) Zhang, H.; Wang, Y. *Environmetrics* **2010**, *21*, 290. (b) Simpson, T. W.; Mauery, T. M.; Korte, J. J.; Mistree, F. *AIAA J.* **2001**, *39*, 2233.
(23) Kandathil, S. M.; Fletcher, T. L.; Yuan, Y.; Knowles, J.; Popelier, P. L. A. *J. Comput. Chem.* **2013**, *34*, 1850.
(24) Stein, M. L. In *Interpolation of Spatial Data: Some Theory for Kriging*; Springer-Verlag: New York, 1999.
(25) Kohavi, R. 14th International Joint Conference on Artificial Intelligence (IJCAI), Montreal, Quebec, Canada, August 20−25, 1995; p 1137.
(26) Wahba, G. *Annals of Statistics* **1985**, *13*, 1378.
(27) Stein, M. L. *Annals of Statistics* **1990**, *18*, 1139.
(28) Bachoc, F. *Computational Statistics and Data Analysis* **2013**, *66*, 55.
(29) Rasmussen, C. E.; Williams, C. K. I. *Gaussian Processes for Machine Learning*; MIT Press: Cambridge, MA, 2006.
(30) Davis, G. J.; Morris, M. D. *Math. Geol.* **1997**, *29*, 669−683.
(31) Jones, D. R. *J.Global Optim.* **2001**, *21*, 345.
(32) Jones, D. R.; Schonlau, M.; Welch, W. J. *J.Global Optim.* **1998**, *13*, 455.
(33) Zimmerman, R. *J. Appl. Math.* **2010**, *2010*, 17.
(34) Kok, S. EngOpt 2012: 3rd International Conference on Engineering Optimization, Rio de Janeiro, Brazil, July 1−5, 2012.
(35) Martin, J. D.; Simpson, T. W. *AIAA J.* **2005**, *43*, 853.
(36) Byrd, R. H.; Lu, P.; Nocedal, J.; Zhu, C. *SIAM J. Sci. Comput.* **1995**, *16*, 1190.
(37) (a) Yang, H.; Zhang, S.; Deng, K.; Du, P. *J. China Univ. Min. Technol.* **2007**, *17*, 473. (b) de Souza, B. F.; de Carvalho, A. C. P. L. F.; Calvo, R. I.; Ishii, R. P. Sixth International Conference on Hybrid Intelligent Systems, Rio de Janeiro, Brazil, Dec 13−15, 2006; p 31. (c) Lee, B.; Kim, S.; Seok, J.; Sangchul, W. 2006 SICE-ICASE International Joint Conference, Busan, Korea, Oct 18−21, 2006; p 5614. (d) Yang, Y.; Chen, R. S.; Ye, Z. B. *MIcrowave and Optical Technology Letters* **2006**, *48*, 141.
(38) Christianimi, N.; Shawe-Taylor, J. *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*; Cambridge University Press: Cambridge, 2000.
(39) Toal, D. J. J.; Bressloff, N. W.; Keane, A. J.; Holden, C. M. E. *Engineering Optimization* **2011**, *43*, 675.
(40) Panigrahi, B. K.; Hiot, L. M.; Shi, Y. *Handbook of Swarm Intelligence: Concepts, Principles and Applications*; Springer: Berlin, 2010.
(41) (a) Eberhart, R. C.; Shi, Y. *Proc.of the 2000 Congress on Evolut.Comp.* **2000**, 84. (b) Clerc, M. Proceedings of the 1999 Congress on Evolutionary Computation, CEC 99, Washington, DC, July 6−9, 1999.
(42) Das, S.; Suganthan, P. N. *IEEE Transactions on Evolutionary Computation* **2011**, *15*, 4.
(43) Price, K. V.; Storn, R. M.; Lampinen, J. A. *Differential Evolution: A Practical Approach to Global Optimization*; Springer-Verlan: Berlin, 2005.
(44) Brest, J.; Greiner, S.; Boskovic, B.; Mernik, M.; Zumer, V. *IEEE Trans.* **2006**, *10*, 646−657.
(45) Storn, R. On the Usage of Differential Evolution for Function Optimization. *NAFIPS-96* **1996**, 519−523.
(46) Brest, J.; Maucec, M. S. *Appl. Intell.* **2008**, *29*, 228.
(47) Golberg, M. A. *American Mathematical Monthly* **1972**, *79*, 1124.
(48) Park, J.; Baek, J. *Comput. Geosci.* **2001**, *27*, 1.
(49) Zhu, C.; Byrd, R. H.; Lu, P.; Nocedal, J. *ACM Transactions on Mathematical Software* **1997**, *23*, 550.
(50) Calamai, P. H.; More, J. J. *Math. Programming* **1987**, *39*, 93.
(51) Bertsekas, D. P. *IEEE Trans. Autom. Control* **1976**, *21*, 174.
(52) Popelier, P.; Aicken, F.; O'Brien, S. *Chemical modelling: applications and theory* **2000**, *1*, 143.
(53) Koch, U.; Popelier, P. L. A.; Stone, A. J. *Chem. Phys. Lett.* **1995**, *238*, 253.
(54) Handley, C. M.; Hawe, G. I.; Kell, D. B.; Popelier, P. L. A. *Phys. Chem. Chem. Phys.* **2009**, *11*, 6365.
(55) (a) Fletcher, T.; Davie, S. J.; Popelier, P. L. A. *J. Chem. Theory Comput.* **2014**, *10*, 3708. (b) Hughes, T. J.; Kandathil, S. M.; Popelier, P. L. A. *Spectrochim. Acta, Part A* **2015**, *136*, 32. (c) Mills, M. J. L.; Popelier, P. L. A. *Theor. Chem. Acc.* **2012**, *131*, 1137. (d) Mills, M. J. L.; Popelier, P. L. A. *Comput. Theor. Chem.* **2011**, *975*, 42.
(56) Liem, S. Y.; Popelier, P. L. A.; Leslie, M. *Int. J. Quantum Chem.* **2004**, *99*, 685.
(57) Smith, W.; Forester, T. R. *J. Mol. Graphics* **1996**, *14*, 136.
(58) Ochterski, J. W. *Vibrational Analysis in Gaussian*, 1999. http://www.gaussian.com/g_whitepap/vib.htm.
(59) Hughes, T. J.; Cardamone, S.; Popelier, P. L. A. *J. Comput. Chem.* **2015**, *36*, 1844.
(60) Frisch, M. J.; Trucks, G. W.; Schlegel, H. B.; Scuseria, G. E.; Robb, M. A.; Cheeseman, J. R.; Scalmani, G.; Barone, V.; Mennucci, B.; Petersson, G. A.; Nakatsuji, H.; Caricato, M.; Li, X.; Hratchian, H. P.; Izmaylov, A. F.; Bloino, J.; Zheng, G.; Sonnenberg, J. L.; Hada, M.; Ehara, M.; Toyota, K.; Fukuda, R.; Hasegawa, J.; Ishida, M.; Nakajima, T.; Honda, Y.; Kitao, O.; Nakai, H.; Vreven, T.; Montgomery, J. A., Jr.; Peralta, J. E.; Ogliaro, F.; Bearpark, M.; Heyd, J. J.; Brothers, E.; Kudin, K. N.; Staroverov, V. N.; Kobayashi, R.; Normand, J.; Raghavachari, K.; Rendell, A.; Burant, J. C.; Iyengar, S. S.; Tomasi, J.; Cossi, M.; Rega, N.; Millam, J. M.; Klene, M.; Knox, J. E.; Cross, J. B.; Bakken, V.; Adamo, C.; Jaramillo, J.; Gomperts, R.; Stratmann, R. E.; Yazyev, O.; Austin, A. J.; Cammi, R.; Pomelli, C.; Ochterski, J. W.; Martin, R. L.; Morokuma, K.; Zakrzewski, V. G.; Voth, G. A.; Salvador, P.; Dannenberg, J. J.; Dapprich, S.; Daniels, A. D.; Farkas, O.; Foresman, J. B.; Ortiz, J. V.; Cioslowski, J.; Fox, D. J. *Gaussian 09*; Gaussian, Inc.: Wallingford, CT, 2009.
(61) Keith, T. A. *AIMAll*, version 13.10.19; TK Gristmill Software: Overland Park, KS, 2013. aim.tkgristmill.com.

DOI: 10.1021/acs.jctc.5b00936
J. Chem. Theory Comput. 2016, 12, 1499−1513

(62) Vesterstrom, J.; Thomsen, R. Congress on Evolutionary Computation, CEC2004, Portland, OR, June 19−23, 2004; p 1980.
(63) Das, S.; Konar, A.; Chakraborty, U. K. GECCO '05: Proceeedings of the 7th Annual Conference on Genetic and Evolutionary Computation, Washington, DC, June 25−29, 2005.
(64) Mallipeddi, R.; Suganthan, P. N.; Pan, Q. K.; Tasgetiren, M. F. *Applied Soft Computing* **2011**, *11*, 1679.
(65) Liu, J.; Lampinen, J. *Soft Computing* **2005**, *9*, 448.
(66) Qin, A. K.; Huang, V. L.; Suganthan, P. N. *IEEE Transtactions on Evolutionary Computation* **2009**, *13*, 398.
(67) Neri, F.; Tirronen, V. *Artificial Intelligence Review* **2010**, *33*, 61.
(68) (a) Ronkkonen, J.; Kukkonen, S.; Price, K. V. The 2005 IEEE Congress on Evolutionary Computation, Edinburgh, Scotland, Sept 2−5, 2005; p 506. (b) Gamperle, R.; Muller, S. D.; Koumoutsakos, P. WSEAS International Conference on Advances in Intelligent Systems, Fuzzy Systems, Evolutionary Computation, 2002; p 293.
(69) Yisu, J.; Knowles, J.; Hongmei, L.; Yizeng, L.; Kell, D. B. *Applied Soft Computing* **2008**, *8*, 295.
(70) Chen, D.; Zhao, C. *Applied Soft Computing* **2009**, *9*, 39.
(71) Arumugam, M. S.; Rao, M. V. C. *Applied Soft Computing* **2008**, *8*, 324.
(72) Warnes, J. J.; Ripley, B. D. *Biometrika* **1987**, *74*, 640.
(73) Mardia, K. V.; Watkins, A. J. *Biometrika* **1989**, *76*, 289.
(74) Ripley, B. D. *Statistical Inference for Spatial Processes*; Cambridge University Press: New York, 1988.
(75) Drton, M.; Richardson, T. S. *Biometrika* **2004**, *91*, 383.
(76) Zellner, A. *J. Am. Stat. Assoc.* **1962**, *57*, 348.

DOI: 10.1021/acs.jctc.5b00936
J. Chem. Theory Comput. 2016, 12, 1499−1513