JCTC Journal of Chemical Theory and Computation

Meta-Optimization of Evolutionary Strategies for Empirical Potential Development: Application to Aqueous Silicate Systems

Brian C. Barnes and Lev D. Gelb*

Department of Chemistry and Center for Materials Innovation, Washington University in St. Louis, St. Louis, Missouri 63130

Received April 12, 2007

Abstract: The use of evolutionary strategy optimizations in fitting empirical potentials against first-principles data is considered. Empirical potentials can involve a large number of interdependent quantities, the number varying with the complexity of the potential, and the optimization of these presents a challenging numerical problem. Evolutionary strategies are a general class of optimization methods that mimic natural selection by stochastically evolving a population of trial solutions according to rules that select for high values of some fitness function. In this work we apply a variety of evolutionary optimization methods to a representative "parametrization problem" in order to determine which such methods are well-suited to such applications. Prior work on the design of evolutionary strategies has generally focused on finding the extrema of relatively simple mathematical functions, and the findings of such studies may not be transferable to chemical applications of very high dimensionality. The test problem consists of parametrization of the Feuston-Garofalini all-atom potential developed for simulation of silicic acid oligomerization in aqueous solution (Feuston, B. P.; Garofalini, S. H. *J. Phys. Chem.* **1990**, *94*, 5351). "Meta-optimization" of the evolutionary method is first considered by fitting this potential against itself, using a wide variety of population sizes, recombination algorithms, mutation-size control methods, and selection methods. Simulated annealing is also considered as an alternative approach. Optimal choices of population size, recombination operator, mutation size control approach, and selection method are discussed, as well as the quantity of data required for the parametrization. It is clear from comparisons of multiple independent optimizations that, even when fitting this potential against itself, there are a considerable number of local extrema in the fitness function. Evolutionary methods are found to be competitive with simulated annealing and are more easily parallelized. Finally, the potential is reparametrized against reference data taken from a Car−Parrinello Molecular Dynamics trajectory of several relevant silicate species in aqueous solution, again using several variant algorithms.

1. Introduction

Empirical potentials (force fields) are widely used in molecular modeling and simulation and usually consist of analytic functions which have been parametrized to reproduce selected reference data. The functional forms are chosen to model specific intermolecular and intramolecular interactions thought to be important for a given application. For instance, in the potentials commonly used for studying the phase behavior of fluids one generally includes terms describing atomic-core repulsions, dispersion forces, bond angles, and torsions; if dipolar or charged species are present, then these may be described using point dipoles or distributions of point charges. By inclusion of higher multipoles and/or polariz-

* Corresponding author e-mail: gelb@wustl.edu.

10.1021/ct700087d CCC: $37.00
Published on Web 08/16/2007

abilities, such potentials can become quite complex. The design of effective potentials has been discussed extensively in the simulation literature, and the functional forms used vary considerably from problem to problem.[1−5]

Parameters may be fit to a wide range of data, including both experimental results and quantities calculated using first-principles or semiempirical electronic-structure methods. Experimental data often used for this purpose include, among others, crystal structures, thermophysical properties such as melting points and critical parameters, partial radial distribution functions, angular distributions, and diffusion constants. Parametrization against thermophysical quantities requires the use of simulations to determine the corresponding properties of trial parameter sets, which can be computationally expensive.

With first-principles methods one may calculate the energies and associated gradients for selected molecular configurations as well as charge distributions, multipole moments, and structural quantities. Such data may be obtained either for isolated molecules or in the condensed phase. The parametrization of empirical potentials against first-principles reference data is now a popular and widely used approach,[6−16] building on both the broad availability of software for high-quality electronic structure calculations and general interest in multiscale simulation methods.

In all cases, systematic parametrization of the chosen functional form presents a challenging numerical problem. This may be cast as the optimization of an objective function that measures the ability of the empirical potential to reproduce selected reference data and therefore as a minimization in some high-dimensional space where the dimensionality is equal to the number of parameters to be assigned. In general, the properties of the objective function will depend on the physical system under consideration, reference data, potential form, and metric used to compare model results with reference data. For a given parametrization problem there may well exist a multiplicity of possible solutions, as pointed out in the early literature in development of the central force model for liquid water.[17−19]

Many strategies for parametrization of empirical potentials are available, varying in both computational complexity and "philosophy".[1−5] One significant classification of these strategies is whether all parameters are considered simultaneously or if a sequential, one-parameter-at-a-time (or one-term-at-a-time) approach is used; the latter cases may also be iterated.

Iterated parameter-by-parameter optimizations correspond roughly to direction-set optimization methods[20] and therefore produce local minima of the objective function. Term-by-term optimizations (which may consider a few parameters at a time) are popular because they reflect the additivity of different interactions explicitly built into many potentials. For instance, one may parametrize a torsional motion independently of the associated angular terms by using an electronic structure program to scan over the torsional degree of freedom and then fit that data with some appropriately chosen function. The disadvantage of this approach is that the resulting torsion is then fit at particular values of the associated angles, and any dependence of the torsion on the associated angles will not be described well. To capture such interactions one must have both reference data that explores appropriate deformations of the molecule and additional terms in the potential that depend on both torsions and angles. In such a case, one may choose either to individually fit the torsion-only and angle-only terms and then fit the "cross" term or to fit all three parts simultaneously. The term-by-term approach allows for a better description of the isolated motions with inaccuracies concentrated in the cross-term, whereas the simultaneous fitting will spread inaccuracies more evenly among the three terms. Such issues become particularly important when extending previously developed potentials to include new atomic or molecular species. If the existing potential is not reparametrized to some degree, then the inaccuracies associated with the (necessarily imperfect) description of interaction with the new species will be concentrated in the added terms. Conversely, when all parameters are fit simultaneously this will not be the case, but parts of the potential may not be as accurate as the functional form allows, since global optimization may use them to compensate for some deficiency elsewhere in the functional form.

The ReaxFF family of reactive potentials,[6−11] for instance, is parametrized against small molecule calculations for bond distances and angles and experimental data for heats of formation. A local optimization technique of successive one-parameter optimization (line search) was used.[21] In an alternative approach, Voth et al. have used first-principles simulations of condensed phases to create potentials for water and hydrogen fluoride.[12,13] Their "force matching" technique uses a short-ranged cubic spline and a long-ranged Coulomb form to model site−site interactions. The linearly independent splines enable the use of singular value decomposition to exactly find parameters for a given configuration, and a final set of parameters is then determined by averaging over the results of many configurations.

For potentials describing a small number of degrees of freedom (and therefore either very small systems or species of low structural complexity) electronic structure calculations can be used to "scan" over the complete potential energy surface. These results can then be numerically interpolated, fit to analytical functions, or some combination of both, in order to obtain highly accurate potentials. Recent examples of such parametrizations include the water potential of Bukowski et al.[14] and the nine-dimensional potential for collisions of hydrogen gas and water monomers developed by Faure et al.;[15] there is a considerable literature on the development of such surfaces for use in reaction dynamics calculations.[22−25]

In the 1960s and 1970s, three groups developed independently numerical optimization methods which mimicked the process of evolution.[26,27] Rechenberg and Schwefel created a family of "evolutionary strategies" to solve real-valued problems.[28−32] Fogel researched artificial intelligence problems through an "evolutionary programming" technique.[33] Finally, Holland developed "genetic algorithms" as a general optimization method.[34] De Jong discusses all these methods under a unified framework of "evolutionary computation" and generalizes them as "evolutionary algorithms".[35]

A brief outline of an evolutionary strategy is as follows. First, a population of trial solutions, called parents, is created. Second, a *recombination* process creates a group of children by averaging or otherwise combining parts of the parents. Third, the children undergo *mutation*, consisting of small random changes. Fourth, those children are *evaluated*. Fifth, a *selection* process is used to select a new group of parents from the current population. The cycle is then repeated, starting with the recombination step.

One important difference between evolutionary strategies and genetic algorithms is in the representation of trial solutions: evolutionary strategies are phenotypic, and genetic algorithms are genotypic.[36] That is, in an evolutionary strategy the individuals are manipulated "as-is", whereas genetic algorithms operate on bitwise representations. This difference in representation requires different operators for recombination and mutation steps. In genetic algorithms, recombination operators exchange strings of bits between two parents in order to generate children, and the basic mutation operator is a random bit flip. For a continuous-valued problem represented phenotypically, the recombination step would involve choosing or averaging values from the parents to create a child, and the simplest mutation would be the random displacement of selected child parameters.

Genetic algorithms (GAs) have been used in potential development in a number of studies, mostly to extend semiempirical methods or to refine popular force fields. Cundari, Deng, and Fu used a GA to parametrize technetium interactions in the semiempirical PM3 method. Their results were fit against crystal structure geometries, and they found that their GA provided significantly better parameters than those obtained by interpolating parameters of the metals to the left and right of technetium in the periodic table.[37] Rossi and Truhlar used a GA to reparametrize the AM1 semiempirical method against quantum mechanical data in order to perform semiquantitative direct dynamics on the $Cl + CH_4$ potential energy surface.[38] Parameters for organic systems containing sodium and transition metals in the AM1 and PM3 methods have also been refit using GAs.[39,40] These targeted reparametrizations can allow semiempirical methods to give substantially improved structures for biochemically relevant systems. Ge and Head used dual genetic algorithms in a study of Si_xH_y clusters, with one GA tasked to iteratively reparametrize the AM1 method, and the other GA to search cluster geometries for a global minimum.[41] GAs have also been used in computer-aided molecular design.[42] As reviewed by Lameijer et al.,[43] in the area of drug design evolutionary algorithms have been applied to the design of molecule libraries, conformational analysis, molecule superposition and pharmacophore detection, quantitative structure−activity relationships (QSAR), ligand docking, de novo design, and "druglikeness" evaluation. In particular, Thomsen investigated the effects of variation operators and local-search hybrid methods on EA/GA performance for ligand docking.[44]

Strassner et al. performed one of the few studies of the influence of GA parameters in the context of developing empirical potentials. They examined the interaction of crossover rates, mutation rates, and selection methods on the overall GA performance for refitting of the MM3 force field for a rhenium complex.[16,45] In this study, different GA parameter sets were compared via the root-mean-squared deviation (rmsd) between experimental (or high-level theoretical) crystal structures and those obtained using the GA-parametrized force field; GAs which produced MM3 parameters with smaller rmsds were judged to be more effective. Results were averaged over only three different independent optimizations for each set of GA parameters, and definite trends in GA performance with different parameters were observed. The most efficient algorithm tested was a simple GA with a tournament selector, 90% crossover rate, and 20% mutation rate. Wolohan et al. reparametrized the MM3 force field for copper complexes[46] using the GA parameters recommended by Strassner et al.[16,45] Other efforts at reparametrizing force fields using GAs include partial reparametrization of the AMBER force field,[47] refitting of the BKS and TTAM potential forms,[48] and refitting of the Stillinger-Weber potential for silicon.[49]

With the exception of the work of Strassner et al.,[16,45] the actual performance of the GAs used in potential parametrization work has rarely been considered in any depth. Many previous studies of the efficiency of evolutionary strategies have considered only the optimization of relatively simple and low-dimensional mathematical functions.[26,27] The behavior of an ES for much more complex problems may be distinctly different.

In this paper we evaluate the performance of a reasonable selection of evolutionary strategy algorithms applied to the problem of optimizing an empirical potential for molecular simulation applications. The process of finding the best algorithm for an optimization is termed a "meta-optimization". The empirical potential we consider is the all-atom, reactive potential for aqueous solutions of silicate oligomers developed by Feuston and Garofalini (FG).[50,51] Reparametrization of the FG potential is a useful test application because the short-ranged nature of the potential makes it inexpensive to evaluate, and optimization of the large number of parameters used poses a difficult numerical problem. The purposes of this work are to provide effective guidelines for future applications of evolutionary strategies in similar parametrization studies and to provide benchmarks for the behavior that can be expected of these algorithms.

2. Methodology

2.1. Evolutionary Strategy Optimizations. A complete evolutionary strategy implementation requires specification of initialization, recombination, mutation, evaluation, selection, and termination algorithms. In this work we evaluate the performance and behavior of a variety of recombination, mutation, and selection methods in the parametrization of an empirical potential against various reference data.

Individuals (parents and children) will be represented as vectors of real numbers $\mathbf{x} = \{x_i, \sigma_i\}$, $i = 1, ..., N$, where N is the number of parameters. The $\{x_i\}$ are the quantities to be optimized (in this case, parameters of an empirical potential), and the $\{\sigma_i\}$ are associated quantities that control the size of mutations applied to each parameter. The $\{\sigma_i\}$ may themselves be subject to evolution. The parts of the evolutionary strategy are presented below.

1. *Initialization.* In this step an initial population of m parents is created. Each parameter x_i of each parent is selected from a continuous uniform distribution within a constrained range, $x_i^{\min}$ to $x_i^{\max}$, which are part of the initial input. The initial values of the $\{\sigma_i\}$ are defined through scaling of an input parameter σ_0: $\sigma_i = \sigma_0 \cdot (x_i^{\max} - x_i^{\min})$. This reflects the fact that the absolute values of the x_i can vary by many orders of magnitude, depending on the units and functional forms used.

2. *Recombination.* Recombination is the process of combining parents to produce children. Following Schwefel, recombination operators are classified as *local* or *global* and also as *discrete* or *intermediate*.[52] Local operators generate a child entirely from two randomly selected parents. Global operators randomly select a new pair of parents *for each parameter* of every child. Discrete operators assign each (x_i, σ_i) pair for the child by setting them equal to the value of the corresponding (x_i, σ_i) pair in one of the randomly chosen parents. Intermediate operators instead assign the average value of the corresponding parent parameters to the child. Selections are made "without replacement", so that it is not possible to create a child from two "copies" of a single parent.

3. *Mutation.* Each parameter x_i in each child $\mathbf{x}$ is displaced with probability p by a random number chosen from a normal distribution of zero mean and standard deviation σ_i, $G(0, \sigma_i)$. This change is represented as

$$x_i = x_i + G(0, \sigma_i) \tag{1}$$

The σ_i control the size of mutations. As discussed in greater detail below, different mutation algorithms may independently evolve the $\{\sigma_i\}$ over the course of the optimization. Alternatively, the $\{\sigma_i\}$ may be controlled through a common reference σ, with $\sigma_i = \sigma \cdot (x_i^{\max} - x_i^{\min})$; various algorithms for evolving σ may then be applied.

4. *Evaluation.* The fitness of each new child is evaluated, as described in the next section.

5. *Selection.* In the selection step, the parents of the next generation are selected from the current population. Selection methods may be categorized according to (a) whether or not they allow overlapping generations and (b) their degree of *elitism*.

Evolutionary strategies are commonly labeled either (m, n)-ES or ($m + n$)-ES, where m is the number of parents and n is the number of children per generation.[53] An (m, n)-ES is nonoverlapping: the m parents of the next generation are chosen only from among the n children of the current generation. An ($m + n$)-ES is overlapping: the m parents of the next generation are chosen from the entire current population of $n + m$ individuals. This allows for the survival of individuals for more than one generation and potentially indefinitely.[52]

Elitism describes the importance placed on fitness when selecting parents. *Truncation* methods are the most elitist, and simply choose the best m individuals from the available population (n or $m + n$). A less elitist method is *binary tournament* selection, in which m random pairs are chosen from the available population, and the "winner" of each pair becomes one of the parents for the next generation.[54] With tournament methods, it is possible that the individual with the highest fitness is not selected. The tournament method may be extended to have competitions between an arbitrary number of children when creating a child, e.g., a three-way tournament instead of a binary (two-way) tournament. The truncation selection method is deterministic, while the tournament method is stochastic. We use the term *semi-overlapping* to refer to selection methods which, when choosing new parents (from either the children or from the full population), always include either the best current parent or the best current individual.

After selection, one generation is complete. The m selected individuals now become the parents, and the algorithm returns to the recombination step.

6. *Termination.* Common termination options include exiting after a certain fitness has been achieved, exiting when the fitness of the fittest individual becomes constant to within a specified tolerance, or exiting after a fixed number of generations. In the studies below, which compare various algorithms, termination criteria are chosen to ensure that the computational costs of the different methods are comparable. For algorithms with the same m and n, this corresponds to termination after a fixed number of generations, but for comparisons of algorithms with different m and n, optimizations are terminated after a fixed number of child evaluations, or "births".

2.2. Fitness Function. Our goal in potential parametrization is to have the empirical potential accurately reproduce some reference data, which we will call the *training set*. Here the training set will consist of the total energies of a series of N_{config} atomic configurations. The *fitness function* is defined as

$$\chi^2(\mathbf{x}) = \frac{1}{N_{\text{config}}} \sum_i^{N_{\text{config}}} [\{E_{\text{emp}}(\mathbf{R}_i, \mathbf{x}) - E_{\text{emp}}(\mathbf{R}_{\text{ref}}, \mathbf{x})\} - \{E_{\text{TS}}(\mathbf{R}_i) - E_{\text{TS}}(\mathbf{R}_{\text{ref}})\}]^2 \tag{2}$$

where $E_{\text{emp}}(\mathbf{R}_i, \mathbf{x})$ is the energy of configuration $\mathbf{R}_i$ determined using the empirical potential with parameters $\mathbf{x}$. $E_{\text{TS}}(\mathbf{R}_i)$ is the energy of configuration $\mathbf{R}_i$ determined using some high quality method, for instance Density Functional Theory (DFT). $\chi^2(\mathbf{x})$ is a measure of the mean-squared difference between the potential energy surfaces sampled by the training set and defined by the chosen empirical functional form and parametrization $\mathbf{x}$.

$\mathbf{R}_{\text{ref}}$ denotes a reference configuration, which is included in the definition of χ^2 because the empirical potential and reference method may differ in ways which make absolute comparisons of their energies impossible. For instance, the energies obtained from typical all-atom empirical potentials cannot be directly compared with the "raw" output of electronic structure calculations. This is because in electronic structure methods even isolated atoms have nonzero total energy due to their internal structure, which is generally not the case for empirical potentials. One possible solution to this problem is to use the energy at the dissociation limit (all atomic separations increased to infinity) to define the energy "zero" in each case, which corresponds to a particular choice of $\mathbf{R}_{\text{ref}}$. However, for many empirical potentials,

including nondissociable molecular potentials and potentials that include nonintegral charges, this is an awkward choice. In this work, we chose the lowest-energy configuration in the training set as the reference state $\mathbf{R}_{\mathrm{ref}}$. This choice is applicable regardless of the form of empirical potential used and requires no additional "reference" calculations. Furthermore, it has the appeal of directly including the differences in energy between "relevant" configurations of the reference system, which appear in the Boltzmann factors determining the thermodynamic properties of the system.

2.3. Application. Our test problem for meta-optimization of evolutionary strategies is a reparametrization of the Feuston and Garofalini (FG) potential for aqueous solutions of silicate oligomers.[50,51] The FG potential includes a modified Born-Mayer-Huggins[55,56] functional form and Rahman-Stillinger-Lemberg[18] (RSL) terms for two-body interactions, and three-body terms as introduced by Stillinger and Weber:[57]

$$V_2(\mathbf{r}_i, \mathbf{r}_j) = A_{ij} \exp\left(\frac{-r_{ij}}{\rho_{ij}}\right) + \frac{Z_i Z_j}{r_{ij}} \operatorname{erfc}\left(\frac{r_{ij}}{\beta_{ij}}\right) + \sum_{m=1}^{D_{ij}} \frac{a_{ij,m}}{1 + \exp(b_{ij,m}(r_{ij} - c_{ij,m}))} \quad (3)$$

$$V_3(\mathbf{r}_{ij}, \mathbf{r}_{ik}, \theta_{jik}) = \lambda_{jik} \exp\left[\frac{\gamma_{ij}}{r_{ij} - r_{ij}^{\mathrm{o}}} + \frac{\gamma_{ik}}{r_{ik} - r_{ik}^{\mathrm{o}}}\right] \times (\cos\theta_{jik} - \cos\theta_{jik}^{\mathrm{o}})^2 \quad (4)$$

The two-body part has a damped Coulomb potential, an exponential repulsion, and a soft (and short-ranged) attraction. Note that a different number D_{ij} of RSL terms are used for each type of two-body interaction involving hydrogen (Si−H, O−H, and H−H). The three-body term penalizes deviation from a specified angle $\theta_{jik}^{\mathrm{o}}$, controlled by parameters for cutoff distance, magnitude, and rate of decay. This is an all-atom, dissociable potential and can be used to study chemical reactions in solution, including the hydrolysis and condensation of siloxane bonds and the early stages of sol−gel processing.[50,51,58]

The FG potential was fit to thermophysical quantities including the radial distribution functions and angular distribution functions of melt-quenched silica. The short-ranged repulsive term was parametrized using a formula based upon ionic radii and charges. The other parameters were chosen based on hydrogen-bond energies, cluster geometries, and liquid-state properties extracted from molecular dynamics simulations, although how trial parameter sets were chosen for these simulations was not described.

The FG potential has two-body parameters for all combinations of the elements Si, O, and H, and parameters describing four different three-body combinations (Si−O−Si, H−O−H, O−Si−O, and Si−O−H), for a total of 55 adjustable parameters. In this work, 45 were optimized, and 10 were kept at fixed values because of physical arguments. The fixed parameters include the charges on each atom type, five three-body cutoff distances r_{ij}^{o}, and the four preferred angles $\theta_{jik}^{\mathrm{o}}$. The atomic charges were kept at their formal values (+1 for hydrogen, +4 for silicon, −2 for oxygen) so that dissociation produced ions with the correct integer charges. The three-body cutoff distances and angles ensure that all silicon and oxygen atoms prefer tetrahedral geometries, except for those oxygens in a waterlike environment, which prefer the experimental angle of 104.5° found in liquid water.

2.4. Training Sets. Two types of training set were used in this paper, both consisting of configurations sampled from molecular dynamics simulations of an aqueous solution of three silicate species. Each configuration in both sets contained one of each of silicic acid, disilicic acid, and cyclotrisilicic acid molecules and 64 water molecules, in a cubic box of 1.4014 nm edge length for a total density of 1.0 g/cm^3.

The first type of reference data, used below in the meta-optimization of the evolutionary strategy, consisted of configurations sampled according to the FG potential and the associated FG energies. These data were generated using a molecular dynamics trajectory thermostatted (via the Gaussian isokinetic method[2]) at 300 K, with configurations sampled at intervals of 2 ps. As in previous studies using this potential, interactions were truncated at 7 Å.

The second training set was generated using Car−Parrinello Molecular Dynamics (CPMD) simulations,[59] also in the canonical ensemble. In these calculations the Perdew-Burke-Ernzerhof (PBE) functional[60] was used with a plane-wave basis with 30 Rydberg cutoff for the wave function and 150 Rydberg cutoff for the density. Vanderbilt ultrasoft pseudopotentials were used for all atoms.[61] The silicon pseudopotential featured a nonlinear core correction. This level of theory was checked by comparing optimized bond distances, bond angles, and hydrogen bond strengths with similar data obtained with the same PBE functional and the 6-31G* and cc-pVTZ basis sets in Gaussian03.[62] The plane-wave results were closer to the 6-31G* basis results, giving bond lengths within 0.005 Å and similar hydrogen bond strengths.

Four visibly and temporally distinct configurations were selected from the first training set. These were used as the starting points for the CPMD simulations. For each configuration, the following procedure was followed. First, each configuration was optimized to a root-mean-square force of 0.005 au. Next, the configuration was relaxed through a series of 11 200-step CPMD simulations using a 3.0 au time step and a 400.0 au fictitious mass for the electrons. A velocity rescaling thermostat was used, with a target temperature of 300 K and rescaling whenever the temperature of the ions was more than 37.5 K away from the target value. After the first six 200-step simulations, the convergence criterion for the gradient of the wave function was tightened from 10^{-5} to 10^{-6} au. Between each 200-step simulation the electrons were quenched back to the Born−Oppenheimer surface. After the relaxation procedure was finished, the production CPMD run was started. The production run used a Nosé-Hoover thermostat for each degree of freedom.[63] The temperature was 300 K with a thermostat frequency of 2500.0 cm^{-1} for the ions and 10 000.0 cm^{-1} for the electrons. These simulations ran for 10 000 steps, giving a total of 242 fs of

data in each of the four CPMD simulations or nearly 1 ps total data. From these trajectories, 370 evenly spaced configurations were selected. Single point energies were then calculated for each configuration; these differ slightly from the CPMD energies because during the dynamics run the electrons are not quenched to the Born−Oppenheimer surface at each time step. These configurations and single-point energies make up the second training set. The program "CPMD version 3.9.2", was used for these calculations.[64]

2.5. Implementation. We have developed a computer code to optimize empirical potentials against training sets of the type described above. Our program implements several optimization techniques, including evolutionary strategies, a simple direct search minimizer, an unconstrained Powell line search algorithm, simplex simulated annealing,[65] and Metropolis simulated annealing.[66] Several potentials are implemented, including the Lennard-Jones model, central force water model, FG model, and a charge-transfer model.[67] Additional potentials may be easily added.

The program is parallelized in two ways. In evolutionary strategy optimizations, evaluation of the fitness of the n children in each generation is divided over many processors by assigning some number of children to each processor. In other optimization techniques, which do not involve the simultaneous evaluation of many trial solutions, the evaluation of a single χ^2 may be parallelized by the distribution of training-set configurations among multiple processors and the simultaneous evaluation of many of the $E_{emp}(\mathbf{R}_i)$ terms. Evolutionary strategy speedups were found to be nearly ideal using up to 16 processors, while the training-set decomposition approach is slightly less efficient due to the increased quantity of communication required. The parallel scalability is also different for the two approaches. For algorithms that only evaluate one trial solution at a time, the theoretical maximum number of processors that can be used is equal to the number of configurations in the training set. Evolutionary strategies, on the other hand, evaluate many individuals in parallel, with each processor handling an equal number of individuals. Therefore, if a very large number of processors is available (as is increasingly the case with modern multicore processors), cases where $n > N_{cpu} > N_{config}$ allow evolutionary strategies to scale higher than other methods. Finally, evolutionary strategies can be further parallelized by distributing the evaluation of each χ^2 among several processors (as in the single-evaluation methods), which could then be used even for $N_{cpu} > n$, and for all methods, even the evaluation of the energy of a single configuration could be spread across several processors using either domain-decomposition or replicated-data strategies.

3. Meta-Optimization of Evolutionary Strategies

The evolutionary strategy may itself be optimized for a particular class of problems by selection of appropriate population sizes, recombination methods, mutation size control schemes, and selection methods. In this study this will be accomplished by optimizing the FG functional form against reference data (training sets) generated using the FG potential itself. Since the functional form is unchanged, it is in principle possible for an optimization algorithm to reduce χ^2 to zero (within some numerical tolerance), which would occur at the exact FG parameters; $\chi^2(\mathbf{x}_{FG}) = 0$. Different ES algorithms will approach this limit more or less quickly and with different "profiles" of χ^2 vs generation.

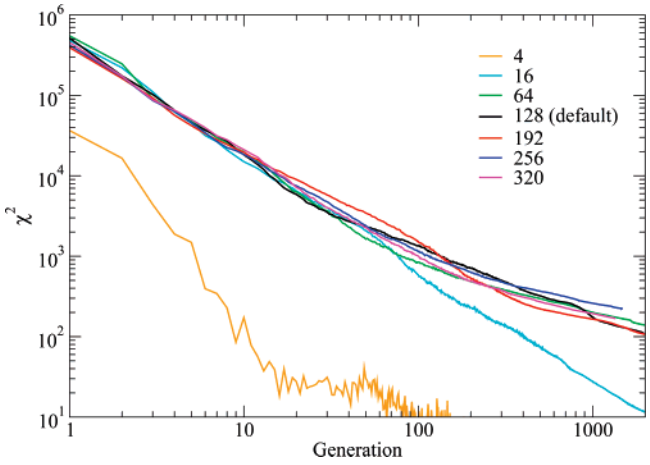

Figure 1. Variation of optimization profile with number of configurations in the training set. Training set sizes used ranged from 4 to 320 configurations. The quantity plotted is the fitness of the fittest (lowest χ^2) member of the current parent population at each generation, averaged over ten independent runs.

Testing different evolutionary strategies is accomplished here by first selecting a "default" combination of population size, recombination method, selection method, etc. and then considering and comparing several alternatives for each of these components. Note that this approach does not consider all possible combinations of methods but does allow for controlled comparisons of different variants of the same operator (for instance, mutation size control schemes).

The default options were selected based on a large number of preliminary trials and recommendations from the literature discussed above. They consist of populations of $m = 8$ and $n = 96$, local discrete recombination, mutation size control using evolving independent σ_i and an initial $\sigma_0 = 0.03$, and nonoverlapping truncation selection.

Unless otherwise stated, all individual optimizations were truncated after 192000 function evaluations, which took roughly 27 wall-clock hours running on two Opteron 250 (2.4 GHz) CPUs. The simulation code was parallelized using MPI. Near-linear scaling was observed in additional tests on up to 16 CPUs; all calculations were performed on a cluster of dual-processor nodes each with 2−4 GB of RAM and networked using Infiniband interconnects.

3.1. Preliminary Studies. In eq 2, each configuration provides only one energy datum. Therefore, the number of configurations in the training set must exceed the number of parameters to be optimized. Training set size may affect the reliability, speed, and smoothness of optimizations. These effects are shown in Figure 1, which compares training sets of varying sizes. Each optimization profile in Figure 1 consists of the lowest parent χ^2 at each generation, averaged over ten independent optimizations (see below). Two of the profiles, using 4 and 16 configurations, are for optimizations against too few independent data to be meaningful. These

optimizations have considerably different profiles than the others, rapidly finding parameter sets with very low χ^2, which is perhaps not surprising given that in these cases this can be satisfied in a large fraction of parameter space.

All the other traces are quite similar, both in the shape of the profile and the lowest χ^2 reached after the allotted simulation time. In Figure 1 all optimizations were run to between 1500 and 2000 generations. Based on the similarity of these data, a training set size of 128 configurations was chosen for use in all the calculations that follow. This is significantly greater than the number of free variables (45) and requires less CPU time than the larger sets of 192, 256, or 320 configurations while clearly retaining the same general properties.

Data are plotted in log−log form in this and subsequent figures. It is therefore important to note that the absolute decrease in χ^2 is much larger in the early generations than in later ones. The units of χ^2 are [(kJ/mol) per configuration]^2. The initial values of $\chi^2 > 10^5$ (kJ/mol)^2 correspond to the randomly generated parent populations described above, which are clearly of poor quality. The final values of χ^2 (for meaningfully large training sets) do not converge to zero in the allotted number of generations but instead tend to reach values near 100 (kJ/mol)^2. The meaning of this value can be assessed by performing simple perturbations of various parameters from their original FG values and measuring the resulting change in χ^2. This measure can then be averaged over perturbations of all the parameters. Single-parameter perturbations of ±0.1% increase χ^2 to 0.392 (kJ/mol)^2, on average. Deviations of ±1% increase χ^2 to 39.17 (kJ/mol)^2, on average, and deviations of ±10% increase it to 3812.5 (kJ/mol)^2, on average. Thus, final values near 100 (kJ/mol)^2 correspond roughly to parameters that have converged to within 1% of their optimal values. However, the sensitivity of χ^2 to such deviations varies considerably from parameter to parameter. Sensitive parameters include the ρ_{ij} parameters for the Buckingham exponential repulsions between oxygen and hydrogen atoms and oxygen and silicon atoms, and the position c_{ij} of the second RSL oxygen-hydrogen term (which is important for modeling hydrogen bonding).

In any single optimization run, χ^2 fluctuated strongly because the recombination and mutation steps are stochastic. In order to make meaningful comparisons of different ES algorithms, we therefore present χ^2 profiles averaged over multiple independent runs. "Independent" in this case means differently reseeding the random number generator for each run after generation of the initial population. The different runs therefore have the same "starting point". We determined that ten independent runs were sufficient to reliably profile different evolutionary strategy variants. This was done by performing 20 runs and then comparing the averaged profiles of two different sets of ten runs with the average profile of all 20 runs. As shown in Figure 2, the average of either set of ten runs is quite similar to the average of all 20 runs. Note that this is not the case for averages over only three independent runs, as used by Strassner et al.[16,45] Each of the 20 individual runs is also plotted in order to illustrate the magnitude of variation between runs. It is clear that the shape of the optimization profile can vary considerably from run

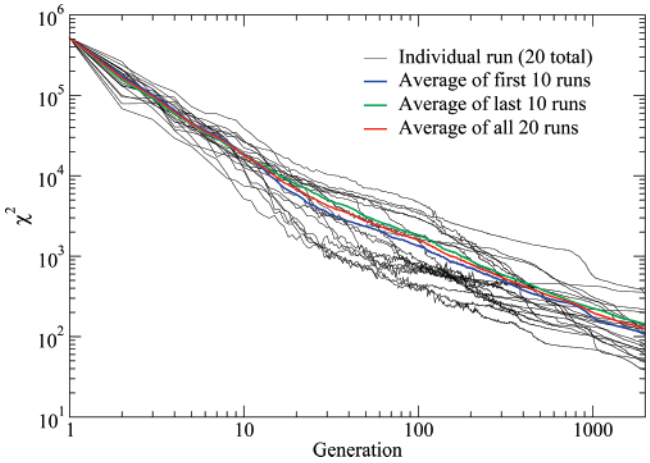

Figure 2. Variation of optimization profile with random number sequence. Twenty independent runs (starting from the same initial population) are shown, along with averages over the full set of 20, the first 10, and the last 10. Run conditions are the "default" algorithm, corresponding to the 128-configuration data shown in Figure 1.

to run and also that the final fitness values can vary by approximately one order of magnitude between runs started from the same initial population. As in Figure 1, all subsequent figures will show the χ^2 for the best parent in each generation averaged over ten runs, unless noted otherwise. Further analysis of the variations between individual runs will be presented in section 5, below.

Genetic diversity is a measure of the difference between members of a population. If members of the population differ only slightly, then a population has low genetic diversity. We measure this through a radius of gyration R_g, defined as

$$x_i^{\text{ave}} = \frac{1}{m}\sum_{j=1}^{m} x_{i,j} \tag{5}$$

$$R_g{}^2 = \frac{1}{m}\sum_{j=1}^{m}\sum_{i=1}^{N}\left(\frac{x_{i,j}}{x_i^{\text{ave}}} - 1\right)^2 \tag{6}$$

where $x_{i,j}$ is the value of parameter i in parent j. Genetic diversity is an important quantity in ES optimizations. If there is too little genetic diversity, then the entire population will become trapped in a single minimum. While this is generally the end result of an evolutionary optimization, it is important that it not happen too early in the calculation, before a large part of parameter space has been explored. R_g data for the default ES strategy are shown in Figure 3. This is a strongly fluctuating quantity but shows clear structure. The initial R_g is large. After approximately ten generations (corresponding to a reduction of χ^2 from approximately 5×10^5 to around 10^4, see Figure 2) R_g drops to a plateau near 0.3, where it remains for approximately 250 generations. Over this period χ^2 decreases by another two orders of magnitude. After this, R_g begins to diminish quickly, becoming very small by the late generations.

3.2. Population. For an (m, n)-ES, a parent:child (m:n) ratio of 1:4 has been recommended,[26] although many studies use larger ratios.[68] Having a very high ratio of children to

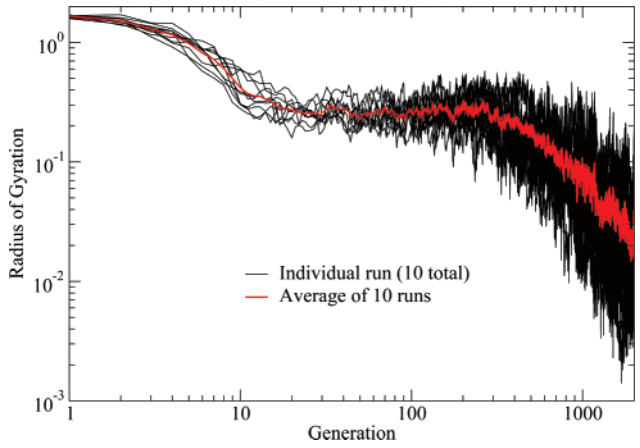

Figure 3. Radius of gyration for 10 individual runs and their average. This calculation corresponds to the 128-configuration data shown in Figure 1.

parents is considered inefficient, since the vast majority of computational time is spent evaluating individuals which do not survive to the next generation. However, in preliminary work we found that a *m*:*n* ratio of 1:12 seemed more effective. The effects of changing the numbers of children and parents, and the ratio *m*:*n*, are therefore of interest in further optimizing the ES approach.

In Figure 4, (*m*, *n*)-ES choices of (8,96), (8,16), (1,8), (48,-96), (8,384), and (8,48) are compared, labeled P-1−P-6, respectively. As explained above, each variant was terminated after a total of 192 000 fitness function evaluations, corresponding here to different numbers of generations. The best initial fitness value among the parents for any (*m*, *n*)-ES with the same number of parents is the same. The profile of P-3 (1,8) has a slightly worse initial best fitness than any $m = 8$ ES, while P-4 (48,96) has an initial best fitness over five times smaller than any $m=8$ ES. This is not surprising: a population with $m = 48$ instead of $m = 8$ has a much larger probability of generating an initial parent with low χ^2.

Comparing the P-1 (8,96) and P-4 (48,96) data shows the benefit of having a smaller parent:child ratio. In P-4, χ^2 actually increases over the first few generations. This can occur when the fittest parents are either not chosen in the recombination step or chosen so infrequently that a child more fit than those parents is not produced. As the selection method in the default strategy does not allow parents to survive to the next generation, the fitness of the best individual may increase from generation to generation.

P-3 is less effective than the other strategies throughout, but especially at early times. With only one parent, there cannot be recombination. Therefore, fitness can only be improved by random mutation of the single initial parent. Distinct jumps can be seen near generations 200, 600, and 1100, when especially productive mutations occurred. These data are again averaged over ten independent runs, and each of these jumps actually corresponds to a very large drop in χ^2 in an individual run.

Comparing strategies with $m = 8$ shows that an increase in the number of children leads to larger decreases in χ^2 per generation during the early stages of the optimization. P-5

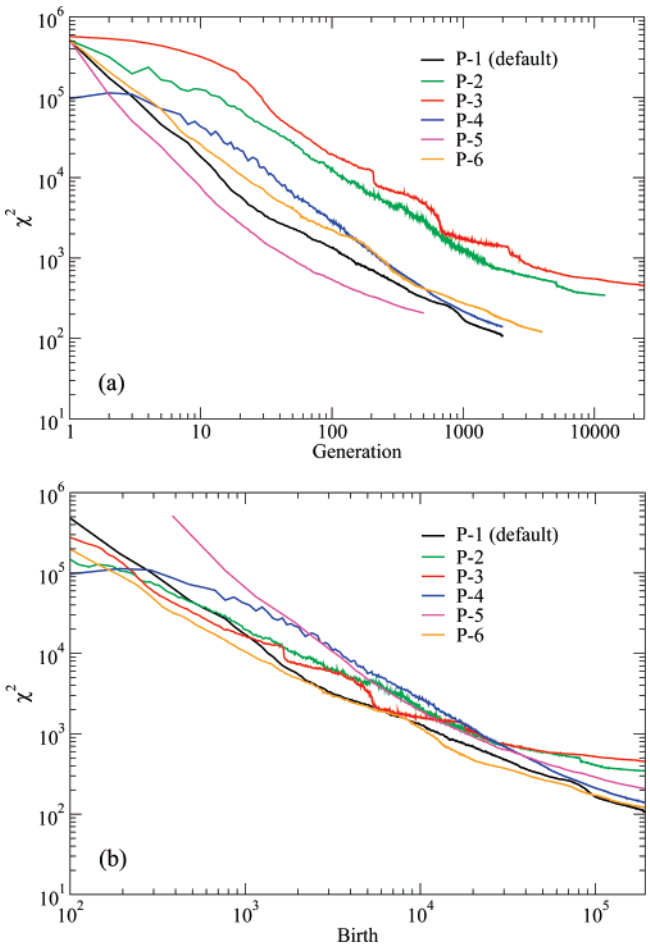

Figure 4. Variation of optimization profile with numbers of parents and children. Tested are (P-1) 8 parents and 96 children, (P-2) 8 parents and 16 children, (P-3) 1 parent and 8 children, (P-4) 48 parents and 96 children, (P-5) 8 parents and 384 children, and (P-6) 8 parents and 48 children. Top: optimization profiles vs number of generations. Bottom: optimization profiles vs number of births.

(8,384) has the largest initial decreases in χ^2 per generation, followed by P-1 (8,96), P-6 (8,48), and P-2 (8,16), in that order. However, the use of large numbers of children is generally avoided because it is both computationally more expensive (per generation) and it tends to more quickly reduce genetic diversity. This can be understood as follows. In the (8,384) optimization, there are only 36 unique pairs of parents, each of which will produce, on average, 10.67 children per generation. If the children of a single pair of parents are particularly fit and truncation selection is used (as is the default here), then the *entire* next generation of parents may consist of the offspring of that pair of parents and will have very low genetic diversity. As the ratio of *m* to *n* is increased, more of the current group of parents will likely contribute to the next generation, and genetic diversity will be preserved. Of the populations tested in Figure 4, P-1 (8,96) achieves the lowest χ^2 after the allotted time and appears to make the most effective compromise between genetic diversity and χ^2 reduction per generation. This finding has implications for the use of evolutionary methods on massively parallel computers. Increasing the number of children, *n*, may appear to be an efficient way to utilize many

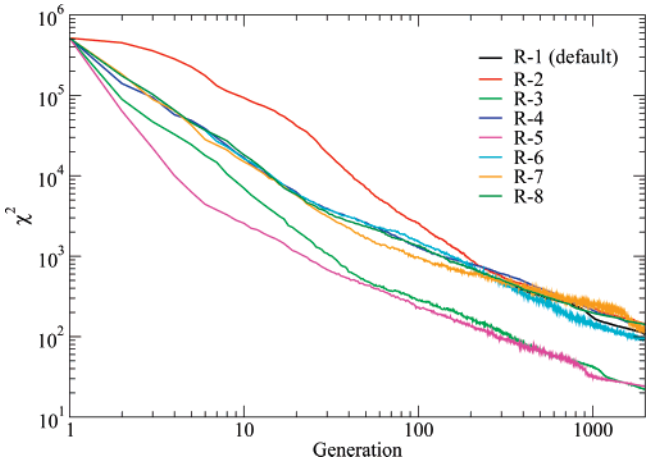

Figure 5. Variation of optimization profile with choice of recombination operator. Operators tested include (R-1) local discrete, (R-2) none, (R-3) local intermediate, (R-4) global discrete, (R-5) global intermediate, (R-6) local discrete for parameters and intermediate for σ, (R-7) global discrete for parameters and intermediate for σ, and (R-8) local discrete for the first 250 generations, none for the subsequent 1750.

processors in an optimization, but then m must likewise be increased to prevent loss of diversity. Furthermore, increasing both m and n does not necessarily improve the rate of convergence of the algorithm in a cost-effective way; this is easily seen in Figure 4b, wherein the performance of method P-6 measured against the number of births is clearly superior at nearly all times to the other algorithms, with P-1 pulling very slightly ahead after 10^5 births.

3.3. Recombination. By default we have used local, discrete recombination. This is the most commonly used recombination operator and is procedurally similar to the method used in genetic algorithms. Various recombination operators are compared in Figure 5. The two intermediate operators (local, R-3, and global, R-5) are seen to provide the most efficient recombination.

After approximately 250 generations, using no recombination at all (R-2) gave results equivalent to local discrete recombination (R-1). This was an unexpected result and suggests that recombination is most effective in the early generations of an optimization. After the first 250 generations, all the optimization profiles have similar slopes, suggesting that after this time the optimization is controlled by mutation instead of recombination. If recombination was still important in the later generations, we would expect the profiles in Figure 5 to differ significantly at late times. Intermediate operators (R-3 and R-5) produce better results overall due to their clear superiority during the early generations; these recombination operators eventually located parameter sets with χ^2 (again, averaged over ten independent runs) only 1/5 that of the typical result of the other operators.

These findings are consistent with the genetic diversity data of Figure 3, where a substantial drop-off in genetic diversity is observed after approximately 250 generations. Once a population is sufficiently inbred, it is unlikely that recombination can lead to substantial improvements in fitness, since the parents are already all very similar. This is investigated by performing an optimization using the default ES parameters (as in R-1) but then disabling all recombination after 250 generations. These results (R-8) overlap with those obtained with the default (R-1) until roughly 1000 generations, after which the default improves very slightly over the modified version, as shown in Figure 5. This behavior is consistent with the hypothesis that recombination is not a substantial contributor to further improvement in fitness after the drop-off in genetic diversity.

It has been suggested that using a discrete operator for the parameters x_i and an intermediate operator for the σ_i is more effective than using either fully discrete or fully intermediate operators.[52] Our results show that this is not the case in this application and that the use of an intermediate operator for the parameters x_i is the key factor. Fully intermediate operators R-3 and R-5 are clearly much more efficient than operators R-6 and R-7, which apply discrete recombination to the x_i and intermediate recombination to the σ_i. The similarity between R-3 and R-5 after the first 50 generations suggests that there is no substantial difference between local and global recombination operators in this application.

3.4. Mutation Size Control. Mutation operators must be included in ES optimizations because recombination operators alone cannot fully search the available parameter space. For instance, when using intermediate operators, the averaging of parameters would mean that children with x_i values outside of the largest and smallest x_i in the current group of parents would never be generated. Likewise, when using discrete recombination operators, the only children that could be created would be combinations of parameters already in the population.

While all mutations involve Gaussian perturbations, the size of these perturbations may be controlled in various ways. It is considered advantageous to have large mutations at the beginning of the optimization, which helps to search across the range of allowed values. However, at later times smaller mutations may be desirable as they can allow near-optimal parents to produce children that are "refinements" of themselves; this is analogous to the very small steps taken by conventional optimization techniques as they approach an extrema. Therefore, the absolute size of mutations should be gradually reduced.[26] The method used for this may also attempt to promote genetic diversity.

The default method used here, labeled M-1, is to use an independent σ_i for each parameter x_i. Following Beyer and Schwefel,[26] the σ_i are generated through a recombination process (as above) and then mutated via

$$\sigma_i^{\text{child}} := \sigma_i^{\text{child}} \cdot \mathcal{J}_g \cdot \mathcal{J}_i \tag{7}$$

where the two mutation operators $\mathcal{J}_g$ and $\mathcal{J}_i$ are

$$\mathcal{J}_g = \exp(\tau_g \cdot G(0, 1)) \quad \tau_g = \frac{1}{\sqrt{2N}} \tag{8}$$

$$\mathcal{J}_i = \exp(\tau_l \cdot G(0, 1)) \quad \tau_l = \frac{1}{\sqrt{2\sqrt{N}}} \tag{9}$$

$\mathcal{J}_g$ is calculated independently for each child and used for all the σ_i; this acts as a global scaling of mutation size, while

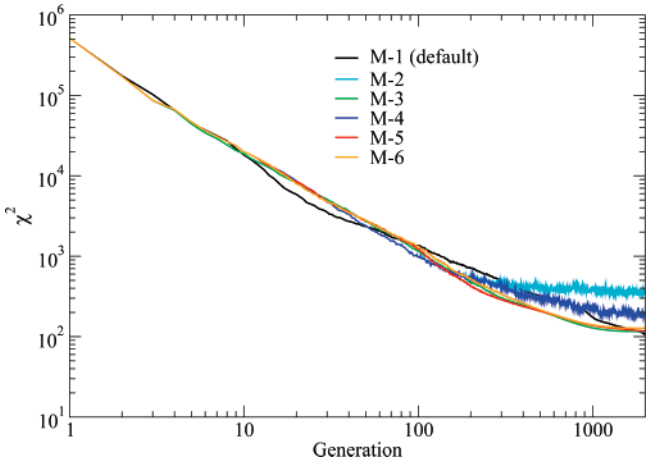

Figure 6. Variation of optimization profile with mutation size control algorithm. Algorithms tested include (M-1) independent $\{\sigma_i\}$, (M-2) constant σ, (M-3) annealing σ by a constant factor, (M-4) adjustment of σ relative to early χ^2, (M-5) a history-dependent, diversity-preserving algorithm, and (M-6) an alternative history-dependent, diversity-preserving algorithm.

the σ_i are calculated independently for each i for each child, allowing for variations in mutation size between parameters.

The simplest mutation size control operator is to fix σ for the entire length of the optimization. Method M-2 demonstrates such a constant global σ.

Method M-3 is referred to as "simple annealing". Here, a global σ is reduced by a constant factor every generation: $\sigma := \sigma \cdot c_\sigma$ where $0 < c_\sigma < 1$. For the profile in Figure 6, $c_\sigma = 0.995$. Note that M-2 may be considered a special case of M-3.

Method M-4 introduces history dependence. It sets σ by scaling σ_0 by the square root of the current average value of the parents' fitness divided by the average value of the parents' fitness after an initial equilibration period. This equilibration period is determined as the end of the initial rapid decrease in χ^2. Specifically, for generation $g > 100$, once $\langle\chi^2\rangle(g) \geq 0.9\langle\chi^2\rangle(g - 100)$, we set $\langle\chi^2\rangle_{\text{ref}} = \langle\chi^2\rangle(g)$ and proceed according to

$$\sigma = \sigma_0 \times \left(\frac{\langle\chi^2\rangle_{\text{parents}}}{\langle\chi^2\rangle_{\text{ref}}}\right)^{(1/2)} \tag{10}$$

where σ_0 is the initial value for σ.

Method M-5 is also history-dependent and attempts to promote genetic diversity while still allowing small mutations near the end of a run. To do this, M-5 compares χ^2_{min} (the lowest χ^2 of the current population) with the χ^2 averaged over the last 100 generations. It uses the following quantities:

$$\chi^2_{\text{scale}} = \left(\frac{10}{g} + 1\right) \cdot \chi^2_{\text{min}}(g) \tag{11}$$

$$\langle\langle\chi^2\rangle\rangle_{100}(g) = \frac{1}{100}\sum_{i}^{i-100}\langle\chi^2\rangle(g) \tag{12}$$

For every tenth generation, if $\langle\langle\chi^2\rangle\rangle_{100}(g) > \chi^2_{\text{scale}}$, then σ is reduced by a multiplicative factor c_σ; else σ is increased by the inverse of the factor c_σ. In this work $c_\sigma = 0.95$. Furthermore, if $\chi^2_{\text{min}}(g) = \chi^2_{\text{min}}(g - 100)$, then we assume that the minimum has been approximately located and reduce σ by c_σ^2. Note that this equality can only be satisfied using overlapping or semioverlapping selection methods.

Last, mutation size control method M-6 uses a history-dependent adjustment of σ which is similar in motivation to M-5 but with a different criterion for changing σ. M-6 tracks the average of the last 10 changes in χ^2_{min} by defining a quantity $\langle\Delta\chi^2_{\text{min}}\rangle_{10}(g)$, which is the average over the 10 most recent nonzero changes in χ^2_{min}. This measures the "step size" of progress toward an optimum solution. Then, if $\langle\langle\chi^2\rangle\rangle_{10}(g) > 4 \cdot \langle\Delta\chi^2_{\text{min}}\rangle_{10}(g)$, then σ is reduced by a multiplicative factor c_σ; else σ is increased by the inverse of c_σ. As in M-5, $c_\sigma = 0.95$, and if $\chi^2_{\text{min}}(g - 100) = \chi^2_{\text{min}}(g)$, then σ is reduced by a factor c_σ^2.

The performance of these different mutation operators is shown in Figure 6. There is no significant impact of mutation size control until roughly 250 generations. It was argued above that recombination methods only had a significant effect in the first 250 generations. It appears that after 250 generations the populations are sufficiently homogeneous that mutation becomes the dominant method of search.

Keeping a constant mutation size prevents parameters from being optimized to values any more precise than the size of Gaussian mutations being applied. This is shown by the fluctuating yet flat fitness of the constant-σ method M-2 from generation 300 onward. The flat fitness profile occurs because the default selection method is nonoverlapping, and the best parent is not carried forward to the next generation. Method M-4 gives results similar to keeping σ constant in the later generations, which is surprising. The scaling factor in M-4 should allow for drops in χ^2 to produce relatively greater drops in σ when the optimization is in its later generations. However, this is not observed, and σ never became small enough to reach the χ^2 values achieved in other methods.

History-dependent, diversity-promoting methods M-5 and M-6 produce results similar to simple annealing, algorithm M-3. Methods M-5 and M-6 did have the desired impact on the genetic diversity of the parent population, but the effect only became noticeable after roughly 1300 generations. At that point, the population had already converged on a single minima, and the diversity was quite low. The likely explanation for the observed behavior is that the diversity-enhancing mutations that were accepted tended to be in the parameters upon which χ^2 did not depend sensitively, such that they would increase the radius of gyration but not lower the fitness. These mutations, therefore, would not contribute strongly to the location of new, lower-χ^2 minima. For such methods to have a significant effect on the optimization, they would have to be tuned to become active closer to the point when mutation takes over from recombination as the dominant form of search, near 250 generations. The default algorithm M-1 performed well but has a somewhat "wavier" profile than the other variants, possibly caused by sporadic large reductions in χ^2 in one of the independent runs. This algorithm ends up very slightly outperforming the other mutation size control algorithms tested.

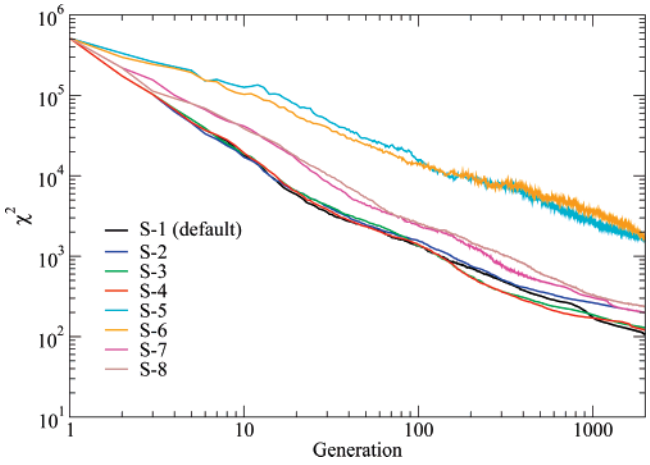

Figure 7. Variation of optimization profile with choice of selection operator. Operators tested include (S-1) nonoverlapping truncation, (S-2) overlapping truncation, (S-3) nonoverlapping truncation plus best parent, (S-4) nonoverlapping truncation plus best-ever individual, (S-5) nonoverlapping 2-way tournament, (S-6) overlapping 2-way tournament, (S-7) nonoverlapping 8-way tournament, and (S-8) overlapping 8-way tournament.

3.5. Selection. Selection methods are compared in Figure 7. The default selection method used, S-1, was the (m, n) nonoverlapping truncation method; S-1 is deterministic, choosing the best m out of n children to be the parents for the next generation. This is compared against overlapping (S-2) and semioverlapping (S-3 and S-4) truncations, and all combinations of overlapping and nonoverlapping two-way and eight-way tournament methods (S-5−S-8). S-1, S-3 and S-4 clearly outperformed all other options in the selection tests. S-1 and S-2 performed similarly until roughly 350 generations into the optimization. S-1 provided a final result with a χ^2 almost 50% better than S-2. Tournament methods are less elitist than truncation methods and also less effective. The two-way tournament methods S-5 and S-6, also called binary tournaments, do not approach the χ^2 value of other methods. Increasing the number of participants in a tournament increases the method's elitism, which makes this method more flexible than truncation methods. However, even eight-way tournament selection methods S-7 and S-8 still lag behind truncation methods.

3.6. Simulated Annealing. For comparison with the evolutionary strategies, we also considered an efficient simulated annealing (SA) algorithm.[66] Simulated annealing is similar to (1+1)-ES, though with different selection and mutation size control operators.

In our SA implementation, a new trial solution (child) is generated by applying Gaussian mutations to parameters of the parent. As this is only done for one child per cycle, we refer to births instead of generations. With probability 0.2 we mutate each parameter x_i by addition of a Gaussian random number $G(0, \sigma_i)$, where σ_i is a global σ scaled by the allowed range of parameter i, as in most of the ES mutation size control variants. Another change made beyond a typical simulated annealing algorithm is that acceptance and rejection of trial solutions are tracked over the past 512 births. If fewer than 20% of children are accepted, then σ is decreased by a factor of $c_\sigma = 0.995$. If more than 20% are accepted, then σ is increased by a factor of $1/c_\sigma$. This is a simple version of the "1/5 rule" sometimes used in (1,1) evolutionary strategies and Monte Carlo simulations.[52] The algorithm has a "temperature" T (with initial value 175.0585 (kJ/mol)^2) which is annealed by a factor $c_T = 0.99994$ after each birth. The child replaces the parent if $U(0,1) \leq \exp(-(\chi^2_{\text{child}} - \chi^2_{\text{parent}})/T)$ where U(0,1) is a uniform random number on the interval [0, 1].

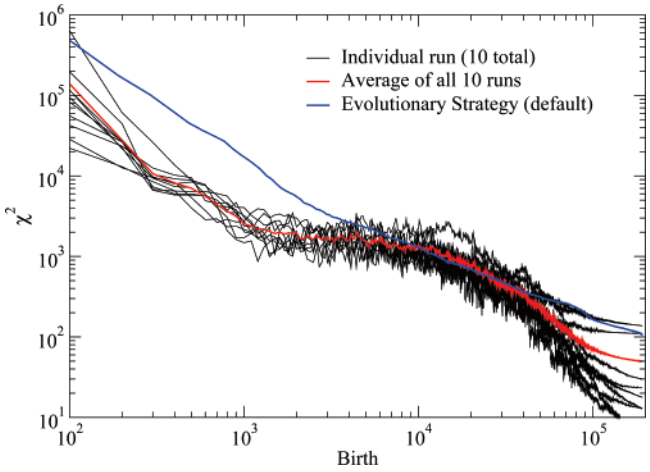

Figure 8. Simulated annealing optimizations. As in Figure 2, ten independent runs (starting from the same point) are shown, as well as their average.

As shown in Figure 8 the shape of the convergence profile in simulated annealing is substantially different from that displayed by the evolutionary strategies tested. After an initial rapid improvement, a period of slow searching occurs. The rapid feedback of simulated annealing—only considering one child per generation before choosing a new parent—may explain the advantage of SA in the first 1000 births or so. The advantage of SA toward the end of the simulation is probably related to the "1/5 rule" which allows mutation size to be adjusted on-the-fly. Interestingly, the profile of SA optimizations at very late times is still different in shape than that of any of the ES mutation size control variants, even though they are designed to have similar effects.

As the simulated temperature is lowered, the algorithm becomes trapped in a single minimum. Different annealing runs produce fitness values varying over about 1 order of magnitude, much as do the independent ES optimizations of Figure 2. The cooling schedule used here was chosen to allow the optimization to reach low temperatures, characterized by fluctuations in χ^2 much smaller than $\mathcal{O}(1)$, within the same number of function evaluations that the evolutionary strategies were allowed. There may be less variation between final fitness values when using a slower cooling schedule. Nevertheless, simulated annealing is very effective in finding a good solution.

4. Parametrization against CPMD Reference Data

Using combinations of ES options that were found to be effective in the meta-optimization study, we then ran several optimizations of the FG potential against the second training

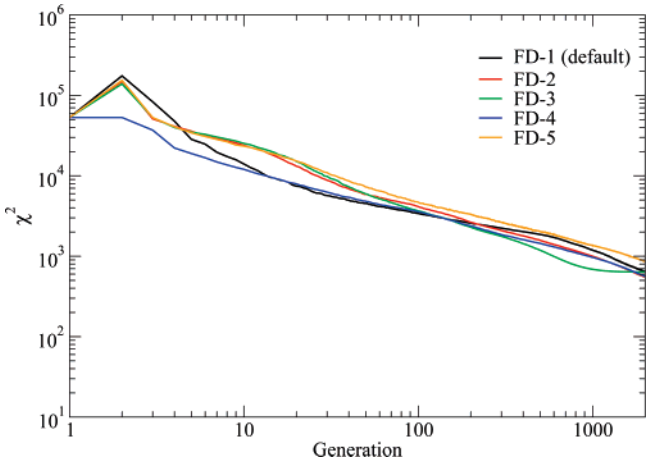

Figure 9. Fitting the FG functional form to the CPMD training set. FD-1 is the default method in the meta-optimization tests. FD-2 uses local, intermediate recombination, and other options are as in FD-1. FD-3 uses local, intermediate recombination and simple annealing mutation size control, and other options are as in FD-1. FD-4 uses local, intermediate recombination and nonoverlapping truncation plus best-ever individual selection, and other options are as in FD-1. FD-5 uses local, intermediate recombination and nonoverlapping 8-way tournament selection, with other options as in FD-1.

set, composed of DFT data. These calculations fit the FG functional form against data which it cannot perfectly reproduce, and so the minimum possible χ^2 will no longer be equal to zero. These optimizations were initialized with the original FG potential parameters as one of the parents.

These results are shown in Figure 9. FD-1 was the default method used in the meta-optimization study. FD-2 used local, intermediate recombination. FD-3 used local, intermediate recombination and simple annealing for mutation size control. FD-4 used local, intermediate recombination and semioverlapping truncation selection from the population $m + n$. FD-5 used local, intermediate recombination and 8-way tournament selection.

The FG parameters are better than almost any random guess. The use of nonoverlapping selection then creates a "spike" at the second generation in four of the five methods tested, since recombination and mutation create children with a larger χ^2 than the FG parameters while the FG potential is not carried over to the second generation.

FD-2 and FD-4 performed the best and have extremely similar profiles for the last 1000 generations of the optimization. Against this training set, the effects of recombination are observed much further into the optimization than the 250 generations usually seen during the meta-optimization study. FD-1 and FD-2 develop similar slopes after generation 1000. FD-3, using simple annealing, performs strongly until just after generation 1000, when σ became too small to make further significant improvements in fitness. Last, FD-5 lagged consistently behind the other options, showing that for this problem and the population size used even large tournament sizes may not be sufficiently elitist. Except for FD-3, all of these methods displayed optimization profiles similar to those seen in the meta-optimization study, suggesting that the approach of fitting an empirical potential to itself is a reasonable choice of a test problem for investigation of ES behavior.

The parameter sets obtained from these calculations are shown in Table 1; these are the fittest individual results from the ten independent runs using each evolutionary strategy variant. All five parametrizations are dramatically fitter (closer to the CPMD reference data) than the original FG parameters, though we should note that this does not a priori indicate that they will be more suitable for modeling a particular system or property. The obtained χ^2 values of $\sim$500 $(\text{kJ/mol})^2$ correspond to an rms deviation of 0.1 kJ/mol per atom in the energy of any given configuration relative to the reference configuration. The average hydrogen bond strength in liquid water is about 20 kJ/mol. Since hydrogen bonding is expected to dominate the energy differences between configurations, we expect that these important interactions should be described well by these parameter sets, at least to within the accuracy of the density functional theory used. The different sets vary considerably in the actual values of particular parameters, with some, such as the λs, varying over a fairly large range, while others, such as β(O−O), are very similar from one set to the next. In a few cases (γ(Si−O−Si), for example) parameters have converged to one side of their "allowed range," which suggests that better fits could be obtained by expanding these ranges.

5. Discussion

All of the optimization profiles shown above are averaged over ten independent runs. In a typical run, for instance as shown in Figure 3, the radius of gyration R_g of the population at the endpoint had a value near to 0.03, indicating that the members of the population were all very similar to each other and that the algorithm had converged into a single minimum of the fitness function. However, the R_g value measured for the ten best solutions obtained from the ten independent runs is 1.49, approximately 2 orders of magnitude larger. Comparing the two values suggests that independent optimization runs are finding different minima of the fitness function; inspection of the actual parameter sets given in Table 1 (which is a different calculation, but with similar convergence properties) supports this. While evolutionary methods are often touted as globally convergent, it appears that for "reasonable" run conditions, performing multiple independent runs is probably a good strategy.

The number of minima, and the "shape" of the fitness function χ^2, are of interest in this regard. Given the high dimensionality of the parameter space, one might suppose that the many different solutions found in these optimizations arise from the relatively small number (128) of configurations used in the training set: the fewer conditions there are to satisfy, the more ways there should be to do so. However, this appears to not be the case. The R_g values for the ten independent optimal solutions for each of the different training set sizes of Figure 1 are all between 1.33 and 1.68, with no correlation with training set size. That is, adding additional data beyond 128 configuration energies does not bring the many locally optimal parameter sets any closer to each other. Likewise, the corresponding R_g values for the

Table 1. Feuston-Garofalini Reparametrizations by Evolutionary Strategies[a]

parameter	FD-1	FD-2	FD-3	FD-4	FD-5	FG
A (H−H), $\times 10^{-9}$ ergs	0.03103	0.02106	0.03571	0.02257	0.021513	0.0340
ρ (H−H), Å	0.2827	0.1784	0.2573	0.1786	0.2206	0.35
β (H−H), Å	1.319	1.3790	1.3526	1.3496	1.3727	2.10
a_1 (H−H), $\times 10^{-12}$ ergs	−5.335	−6.3800	−5.3370	−5.7848	−5.3192	−5.2973
b_1 (H−H), Å^{-1}	5.117	4.7664	4.7996	5.2802	5.4553	6.0
c_1 (H−H), Å	1.2663	1.2006	1.2770	1.2207	1.2542	1.51
a_2 (H−H), $\times 10^{-12}$ ergs	0.2009	0.2632	0.4197	0.2993	0.3546	0.3473
b_2 (H−H), Å^{-1}	1.8539	2.0173	1.3476	2.1513	2.2582	2.0
c_2 (H−H), Å	3.2085	3.1084	2.5569	3.0789	3.0109	2.42
A (O−H), $\times 10^{-9}$ ergs	0.3360	0.3838	0.4018	0.3882	0.3848	0.3984
ρ (O−H), Å	0.2992	0.2773	0.2695	0.2757	0.2787	0.29
β (O−H), Å	1.7270	1.7978	1.7405	1.9038	1.9026	2.26
a_1 (O−H), $\times 10^{-12}$ ergs	−2.2366	−1.2288	−1.8019	−1.7787	−1.4016	−2.0840
b_1 (O−H), Å^{-1}	10.2427	21.4197	19.0815	20.9755	17.0696	15.0
c_1 (O−H), Å	1.1064	1.1605	1.1855	1.1760	1.1541	1.05
a_2 (O−H), $\times 10^{-12}$ ergs	6.8043	7.1150	7.1936	8.4660	7.8496	7.6412
b_2 (O−H), Å^{-1}	2.8448	3.2279	3.2265	2.7840	3.0235	3.2
c_2 (O−H), Å	1.4358	1.6233	1.5092	1.5852	1.5941	1.50
a_3 (O−H), $\times 10^{-12}$ ergs	−0.8008	−1.1142	−0.8619	−0.8341	−1.0400	−0.8336
b_3 (O−H), Å^{-1}	3.8372	5.3733	4.9270	5.1868	5.1650	5.0
c_3 (O−H), Å	1.7244	1.9072	1.8161	1.9928	1.8755	2.00
A (O−O), $\times 10^{-9}$ ergs	0.6204	0.9318	0.7086	1.0126	0.6314	0.7250
ρ (O−O), Å	0.1536	0.2258	0.2316	0.1685	0.1815	0.29
β (O−O), Å	1.6597	1.7056	1.7057	1.7451	1.7893	2.34
A (Si−H), $\times 10^{-9}$ ergs	0.03488	0.04092	0.05571	0.05767	0.05520	0.0690
ρ (Si−H), Å	0.3333	0.1732	0.2241	0.1868	0.2076	0.29
β (Si−H), Å	1.7574	1.8393	1.8692	1.8520	1.9144	2.31
a_1 (Si−H), $\times 10^{-12}$ ergs	−5.9716	−5.9754	−6.2415	−6.0339	−6.3399	−4.6542
b_1 (Si−H), Å^{-1}	3.6173	3.7601	3.7488	3.7710	3.7888	6.0
c_1 (Si−H), Å	2.1270	2.1799	2.2019	2.1767	2.1761	2.20
A (Si−O), $\times 10^{-9}$ ergs	4.3049	2.0904	2.3021	2.1387	2.3477	2.9620
ρ (Si−O), Å	0.2320	0.3052	0.3041	0.3058	0.3070	0.29
β (Si−O), Å	1.2277	1.5972	1.6715	1.6305	1.7657	2.34
A (Si−Si), $\times 10^{-9}$ ergs	2.0641	2.0021	2.2312	1.7762	2.1179	1.8770
ρ (Si−Si), Å	0.3035	0.1890	0.2862	0.2197	0.1855	0.29
β (Si−Si), Å	1.1892	1.4321	1.4610	1.4137	1.5670	2.29
λ (O−Si−O), $\times 10^{-11}$ ergs	11.3068	10.1754	19.44	9.6978	19.1985	19.0
γ (O−Si−O), Å	4.1957	3.8445	3.1944	4.1697	3.9531	2.8
λ (Si−O−Si), $\times 10^{-11}$ ergs	0.4496	0.4483	0.3136	0.4447	0.4439	0.3
γ (Si−O−Si), Å	1.0005	1.0052	2.0065	1.0021	1.0067	2.0
λ (Si−O−H), $\times 10^{-11}$ ergs	4.8690	3.1015	5.1819	2.7365	3.9802	5.0
γ (Si−O−H: Si−O), Å	1.6022	1.0161	1.9427	1.0495	1.7518	2.0
γ (Si−O−H: O−H), Å	1.5203	1.7038	1.3923	1.7058	1.5326	1.2
λ (H−O−H), $\times 10^{-11}$ ergs	31.9566	25.3210	32.1643	38.3666	32.3834	35.0
γ (H−O−H), Å	1.4741	1.3718	1.4345	1.4649	1.4264	1.3
χ^2 (kJ/mol)^2	352.4	430.7	501.3	459.8	560.7	52963.0

[a] The fittest parameter sets from Figure 9 are shown as well as the original FG parametrization. Parameter names and units are as given in ref 51. Only "fitted" parameters are given in the table; other parameters (cutoffs, reference angles, and formal charges) are kept fixed at their literature values.[51]

runs of Figure 4, which vary in m and n, are all between 1.49 and 1.73 and likewise do not exhibit any trend with population parameters. It therefore appears that the many local minima in this objective function result from the potential itself and the particular definition of χ^2 used, rather than the size of the training set or other, more arbitrary parameters.

A significant feature observed in many of the optimization profiles in this study was an apparent crossover, at about 250 generations, from behavior dominated by recombination to behavior dominated by mutation. This crossover was remarkably robust to changes in the various operators involved, and therefore its appearance may be anticipated in related problems.

Since most of the computational effort is expended after the crossover, in order to more quickly locate optimized parameter sets one should make the mutation operator as efficient as possible. However, of the considerable number of mutation operators tested in this work there were no clearly superior ones, and significant further improvements

may be difficult. One possible alternative could be a composite (or "memetic"[69]) optimization strategy, in which, once the ES algorithm "slows down", one switches over to a different, locally convergent, method which is good at "refining" an approximately located solution. The radius of gyration R_g introduced above is an effective signature for the ES crossover and could be monitored to trigger the change to another method. We note, in this regard, that rapidly converging methods such as conjugate-gradient optimization or Newton−Raphson root-finding are not very well suited to parameter optimization problems, since it is preferable to not have to implement derivatives of the energy with respect to the potential parameters. However, such derivatives could be efficiently estimated by using parallelized one-way finite or centered difference methods, which could provide a cost-effective route to the precise location of χ^2 minima; the effectiveness of this approach would depend on the roughness of the χ^2 function and the stability of the optimizer with respect to numerical precision. It should be noted that when applied to ligand docking, a prior study did not find local optimization to be beneficial.[44]

Based on the results of the meta-optimization study, we recommend the use of intermediate recombination operators for both the parameters $\{x_i\}$ and mutation size control variables $\{\sigma_i\}$. No substantial difference is observed between global intermediate and local intermediate recombination at long times, though at short times the global variant appears preferable. Of the mutation operators considered, the self-adaptive, independent-σ method M-1 is at least as effective as any of the others considered and lacks any "adjustable" parameters. We note that "simple annealing" is nearly as effective with one adjustable parameter (here chosen arbitrarily) and considerably simpler to implement. Finally, nonoverlapping or semioverlapping truncation methods are clearly preferred for selection, as the tournament methods appeared to not have enough selection pressure, and overlapping methods exhibited slowdowns in the later stages of optimization.

Simultaneous parametrization of all parts of a potential has the advantage of providing more uniform "quality" between different terms but greatly increases the complexity of the numerical problem to be solved. Even in fully automated parametrizations one must still provide initial estimates of the magnitude (and, likely, the allowed range) of each parameter, which requires at least some physical insight into the problem. In applications where an existing potential is to be extended, such estimates are straightforward, but for the parametrization of a new functional form or previously unstudied chemical species they may be more difficult to obtain. For very large problems, preliminary parametrization of groups of related parameters against subsets of the available reference data may also be a viable strategy.

ES methods are inherently parallelizable. While evolution of the objective (fitness) function used in this work can also be parallelized over a reasonable number of processors, the ES approach has a considerable advantage in this regard and therefore should be of particular interest when wall-clock time is a limiting factor. This suggests that ES is particularly suitable for work involving a large number of parametrizations, for instance comparisons of different functional forms, comparisons of potentials based on different reference data, or even the (common) extension of an existing potential to treat some new chemical species.

As shown in Figure 8, the efficient simulated annealing method used in this study generally outperformed the evolutionary strategies when fitting the FG potential to the FG training set. Simulated annealing can be parallelized either through distribution of configurations in the training set or by performing multiple independent runs. As discussed earlier, evolutionary strategies may spread the evaluation of groups of children across available processors. This is a significant advantage: the number of CPU cores available in modern supercomputers or clusters is increasing at a greater rate than the performance per core. We also note that the adaptive mutation algorithm in the simulated annealing optimizations may have been superior to the mutation algorithms used in the evolutionary strategy, as no equivalent to the "1/5 rule" was available for ES runs.

Finally, we note that the type of reference data used (configurational energies) and definition of the fitness function as a least-squares-like criterion are themselves arbitrary choices, and there are certainly other possibilities. Force (gradient) data could also be used in the fitness function (as in the "force-matching" studies described above,[12,13] for instance), and a "minimax" criteria could be used to define the fitness function, so that the final optimized value would limit the maximum deviation in selected quantities between the model system and the reference data. Although a wide variety of ES algorithms were considered and the potential function studied is representative of a large class of related models, the reference data used in the parametrizations described a single aqueous solution of silicate species. It is therefore possible that the ES variants selected for the parametrization against CPMD data might not transfer well to a different physical system, although one hopes that they should at least serve as a useful starting point and reference.

In summary, we have presented guidelines for the selection of ES operators and training set sizes suitable for the parametrization of empirical potentials against reference data generated using electronic-structure methods. Such parametrizations are considerably higher in dimension and complexity than the typical problems used in development of evolutionary strategies, and algorithms optimized for these different problem classes differ in nonobvious ways. The ES approach is highly parallelizable and may therefore be suited to "large" optimization problems. However, ES exhibits relatively slow convergence at later generations that may warrant changeover at late times to an alternate method which converges rapidly once a solution has been approximately located.

Acknowledgment. We acknowledge financial support from the National Science Foundation (CHE-0241005, CHE-0443501, and CHE-0626008) and the Washington University Center for Materials Innovation and NCSA (DMR030033N) and the Washington University Center for Scientific Parallel

Computing for computer resources. We also thank Rafael Salazar Tio for useful discussions.

References

(1) Gray, C. G.; Gubbins, K. E. *Theory of Molecular Fluids*; Clarendon Press: Oxford, 1984; Vol. 1, pp 27−142.

(2) Allen, M. P.; Tildesley, D. J. *Computer Simulation of Liquids;* Clarendon Press: Oxford, 1987; pp 6−22.

(3) Schlick, T. *Molecular modeling and simulation: an interdisciplinary guide;* Springer-Verlag: New York, 2002; pp 225−304.

(4) Jensen, F. *Introduction to Computational Chemistry*, 2nd ed.; John Wiley & Sons, Ltd.: Chichester, U.K., 2007; pp 22−79.

(5) MacKerell, A. D., Jr.; Brooks, B.; Brooks, C. L., III; Nilsson, L.; Roux, B.; Won, Y.; Karplus, M. CHARMM: The Energy Function and Its Parameterization with an Overview of the Program. In *The Encyclopedia of Computational Chemistry*; Schleyer, P. v. R., Ed.; Wiley: Chichester, 1998; Vol. 1.

(6) van Duin, A. C. T.; Dasgupta, S.; Lorant, F.; Goddard, W. A., III *J. Phys. Chem. A* **2001**, *105*, 9396−9409.

(7) van Duin, A. C. T.; Strachan, A.; Stweman, S.; Zhang, Q.; Xu, X.; Goddard, W. A., III *J. Phys. Chem. A* **2003**, *107*, 3803- 3811.

(8) Nielson, K. D.; van Duin, A. C. T.; Oxgaard, J.; Deng, W. Q.; Goddard, W. A., III *J. Phys. Chem. A* **2005**, *109*, 493−499.

(9) Cheung, S.; Deng, W. Q.; van Duin, A. C. T.; Goddard, W. A., III *J. Phys. Chem. A* **2005**, *109*, 851−859.

(10) Han, S. S.; van Duin, A. C. T.; Goddard, W. A., III; Lee, H. M. *J. Phys. Chem. A* **2005**, *109*, 4575−4582.

(11) Han, S. S.; Kang, J. K.; Lee, H. M.; van Duin, A. C. T.; Goddard, W. A., III *J. Chem. Phys.* **2005**, *123*, 114703.

(12) Izvekov, S.; Parrinello, M.; Burnham, C. J.; Voth, G. A. *J. Chem. Phys.* **2004**, *120*, 10896−10913.

(13) Izvekov, S.; Voth, G. A. *J. Phys. Chem. B* **2005**, *109*, 6573−6586.

(14) Bukowski, R.; Szalewicz, K.; Groenenboom, G. C.; van der Avoird, A. *Science* **2007**, *315*, 1249−1252.

(15) Faure, A.; Valiron, P.; Wernli, M.; Wiesenfeld, L.; Rist, C.; Noga, J.; Tennyson, J. *J. Chem. Phys.* **2005**, *112*, 221102.

(16) Strassner, T.; Busold, M.; Herrmann, W. A. *J. Comput. Chem.* **2002**, *23*, 282−290.

(17) Lemberg, H. L.; Stillinger, F. H. *J. Chem. Phys.* **1975**, *62*, 1677−1690.

(18) Rahman, A.; Stillinger, F. H.; Lemberg, H. L. *J. Chem. Phys.* **1975**, *63*, 5223−5230.

(19) Stillinger, F. H.; Rahman, A. *J. Chem. Phys.* **1978**, *68*, 666−670.

(20) Press, W. H.; Teukolsky, S. A.; Vetterling, W. T.; Flannery, B. P. *Numerical Recipes in Fortran 77*, 2nd ed.; Cambridge University Press: Cambridge, 1992; pp 387−406.

(21) van Duin, A. C. T.; Baas, J. M. A.; van de Graaf, B. *J. Chem. Soc. Faraday Trans.* **1994**, *90*, 2881−2895.

(22) Collins, M. A. *Theor. Chem. Acc.* **2002**, *108*, 313−324.

(23) Moyano, G. E.; Collins, M. A. *J. Chem. Phys.* **2004**, *121*, 9769−9775.

(24) Netzloff, H. M. *J. Chem. Phys.* **2006**, *124*, 154104.

(25) Guo, Y.; Harding, L. B.; Wagner, A. F.; Minkoff, M.; Thompson, D. L. *J. Chem. Phys.* **2007**, *126*, 104105.

(26) Beyer, H.-G.; Schwefel, H.-P. *Nat. Comput.* **2002**, *1*, 3−52.

(27) Fogel, D. B. *IEEE Trans. Neural Networks* **1994**, *5*, 3−14.

(28) Rechenberg, I. *Cybernetic solution path of an experimental problem*; Technical Report; Royal Aircraft Establishment: Farnborough, 1965; Library Translation 1122.

(29) Schwefel, H.-P. Cybernetic evolution as a strategy for experimental research in fluid mechanics. Master's Thesis, Technical University of Berlin, 1965.

(30) Rechenberg, I. Evolutionary strategies: optimizing technical systems with principles of biological evolution. Thesis, Technical University of Berlin, Department of Process Engineering, 1971.

(31) Rechenberg, I. *Evolutionary strategies: optimizing technical systems with principles of biological evolution*; Frommann-Holzboog Verlag: Stuttgart, 1973.

(32) Schwefel, H.-P. Evolutionary strategies and numerical optimization. Thesis, Technical University of Berlin, 1975.

(33) Fogel, L.; Owens, A.; Walsh, M. *Artificial Intelligence through Simulated Evolution;* John Wiley and Sons: 1966.

(34) Holland, J. *Adaptation in Natural and Artificial Systems;* University of Michigan Press: 1975.

(35) De Jong, K. A. *Evolutionary Computation: A Unified Approach;* MIT Press: Cambridge, MA, 2006; pp 49−69.

(36) De Jong, K. A. *Evolutionary Computation: A Unified Approach;* MIT Press: Cambridge, MA, 2006; pp 72−73.

(37) Cundari, T. R.; Deng, J.; Fu, W. *Int. J. Quantum Chem.* **2000**, *77*, 421−432.

(38) Rossi, I.; Truhlar, D. G. *Chem. Phys. Lett.* **1995**, *233*, 231−236.

(39) Mohr, M.; McNamara, J. P.; Wang, H.; Rajeev, S. A.; Ge, J.; Morgado, C. A.; Hiller, I. H. *Faraday Discuss.* **2003**, *124*, 413−428.

(40) Brothers, E. N.; Merz, K. M., Jr. *J. Phys. Chem. B* **2002**, *106*, 2779−2785.

(41) Ge, Y.; Head, J. D. *Int. J. Quantum Chem.* **2003**, *95*, 617−626.

(42) Clark, D. E.; Westhead, D. R. *J. Comput.-Aided Mol. Des.* **1996**, *10*, 337−358.

(43) Lameijer, E.-W.; Bäck, T.; Kok, J. N.; Ijzerman, A. P. *Nat. Computing* **2005**, *4*, 177−243.

(44) Thomsen, R. *BioSystems* **2003**, *72*, 57−73.

(45) Strassner, T.; Busold, M.; Radrich, H. *J. Mol. Model.* **2001**, *7*, 374−377.

(46) Wolohan, P.; Yoo, J.; Welch, M. J.; Reichert, D. E. *J. Med. Chem.* **2005**, *48*, 5561−5569.

(47) Wang, J.; Kollman, P. A. *J. Comput. Chem.* **2001**, *22*, 1219−1228.

(48) Mallik, A.; Runge, K.; Cheng, H.-P.; Dufty, J. *Mol. Sim.* **2005**, *31*, 695−703.

(49) Globus, A.; Menon, M.; Srivastava, D. *Comput. Model. Eng. Sci.* **2002**, *3*, 557−574.

(50) Feuston, B. P.; Garofalini, S. H. *J. Chem. Phys.* **1988**, *89*, 5818−5824.

(51) Feuston, B. P.; Garofalini, S. H. *J. Phys. Chem.* **1990**, *94*, 5351−5356.

(52) Bäck, T.; Hoffmeister, F.; Schwefel, H.-P. A Survey of Evolutionary Strategies. In *Proceedings of the 4th International Conference on Genetic Algorithms*; Belew, R. K., Booker, L. B., Eds.; Morgan Kaufmann: San Franscisco, CA, 1991.

(53) De Jong, K. A. *Evolutionary Computation: A Unified Approach;* MIT Press: Cambridge, MA, 2006; p 27.

(54) De Jong, K. A. *Evolutionary Computation: A Unified Approach;* MIT Press: Cambridge, MA, 2006; pp 128−130.

(55) Born, M.; Mayer, J. E. *Z. Phys.* **1932**, *75*, 1−18.

(56) Huggins, M. L.; Mayer, J. E. *J. Chem. Phys.* **1933**, *1*, 643−646.

(57) Stillinger, F. H.; Weber, T. A. *Phys. Rev. B* **1985**, *31*, 5262−5271.

(58) Rao, N. Z.; Gelb, L. D. *J. Phys. Chem. B* **2004**, *108*, 12418−12428.

(59) Car, R.; Parrinello, M. *Phys. Rev. Lett.* **1985**, *55*, 2471−2474.

(60) Perdew, J. P.; Burke, K.; Ernzerhof, M. *Phys. Rev. Lett.* **1996**, *77*, 3865−3868.

(61) Vanderbilt, D. *Phys. Rev. B* **1990**, *41*, 7892−7895.

(62) Frisch, M. J.; Trucks, G. W.; Schlegel, H. B.; Scuseria, G. E.; Robb, M. A.; Cheeseman, J. R.; Montgomery, J. A., Jr.; Vreven, T.; Kudin, K. N.; Burant, J. C.; Millam, J. M.; Tyengar, S. S.; Thomasi, J.; Barone, V.; Mennucci, B.; Cossi, M.; Scalmani, G.; Rega, N.; Petersson, G. A.; Nakatsuji, H.; Hada, M.; Ehara, M.; Toyota, K.; Fukuda, R.; Hasegawa, J.; Ishida, M.; Nakajima, T.; Honda, Y.; Kitao, O.; Nakai, H.; Klene, M.; Li, X.; Knox, J. E.; Hratchian, H. P.; Cross, J. B.; Adamo, C.; Jaramillo, J.; Gomperts, R.; Stratmann, R. E.; Yazyev, O.; Austin, A. J.; Cammi, R.; Pomelli, C.; Ochterski, J. W.; Ayala, P. Y.; Morokuma, K.; Voth, G. A.; Salvador, P.; Dannenberg, J. J.; Zakrzewski, V. G.; Dapprich, S.; Daniels, A. D.; Strain, M. C.; Farkas, O.; Malick, D. K.; Rabuck, A. D.; Raghavachari, K.; Foresman, J. B.; Ortiz, J. V.; Cui, Q.; Baboul, A. G.; Clifford, S.; Cioslowski, J.; Stefanov, B. B.; Liu, G.; Liashenko, A.; Piskorz, P.; Romaromi, I.; Martin, R. L.; Foz, D. J.; Keith, T.; Al-Laham, M. A.; Peng, C. Y.; Nanayakkara, A.; Challacombe, M.; Gill, P. M. W.; Johnson, B.; Chen, W.; Wong, M. W.; Gonzalez, C.; Pople, J. A. *Gaussian*; Gaussian, Inc.: Pittsburgh, PA, 2003.

(63) Hoover, W. G. *Phys. Rev. A* **1985**, *31*, 1695−1697.

(64) CPMD. Copyright IBM Corp 1990−2003, Copyright MPI für Festkörperforschung Stuttgart 1997−2001.

(65) Press, W. H.; Teukolsky, S. A.; Vetterling, W. T.; Flannery, B. P. *Numerical Recipes in Fortran 77*, 2nd ed.; Cambridge University Press: Cambridge, 1992; pp 436−448.

(66) Kirkpatrick, S.; Gelatt, C. D., Jr.; Vecchi, M. P. *Science* **1983**, *220*, 671−680.

(67) Alavi, A.; Alvarez, L. J.; Elliot, S. R.; McDonald, I. R. *Philos. Mag. B* **1992**, *65*, 489−500.

(68) De Jong, K. A. *Evolutionary Computation: A Unified Approach;* MIT Press: Cambridge, MA, 2006; pp 77−78.

(69) Moscato, P.; Norman, M. G. Parallel Computing and Transputer Applications. In *Chapter A "Memetic" Approach for the Traveling Salesman Problem Implementation of a Computational Ecology for Combinatorial Optimization on Message-Passing Systems*; IOS Press/CIMNE: 1992; pp 177−186.

CT700087D