

Systematic comparison of Monte Carlo Annealing and Covariance Matrix Adaptation for the optimization of ReaxFF parameters

Ganna Shchygol,^{†,‡} Alexei Yakovlev,[†] Tomas Trnka,[†] Adri C.T. van Duin,[¶] and
Toon Verstraelen^{*,‡}

[†]*Software for Chemistry & Materials (SCM) B.V., De Boelelaan 1083, 1081 HV
Amsterdam, The Netherlands*

[‡]*Center for Molecular Modeling (CMM), Ghent University, Technologiepark 903, B-9052,
Ghent, Belgium*

[¶]*Department of Mechanical and Nuclear Engineering, The Pennsylvania State University,
University Park, PA 16802, United States*

E-mail: Toon.Verstraelen@UGent.be

Abstract

ReaxFF is a computationally efficient force field to simulate complex reactive dynamics in extended molecular models with diverse chemistries, if reliable force-field parameters are available for the chemistry of interest. If not, they must be calibrated by minimizing the error ReaxFF makes on a relevant training set. Because this optimization is far from trivial, many methods, in particular genetic algorithms (GAs), have been developed to search for the global optimum in parameter space. Recently, two alternative parameter calibration techniques were proposed, i.e. Monte-Carlo Force Field optimizer (MCFF) and Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES), which have the potential to find good parameters at a relatively low computational

cost. In this work, these two methods are tested, as implemented in ADF2018, using three ReaxFF training sets, which have previously been used to benchmark GAs. Even though MCFF and CMA-ES should not be considered as exhaustive global optimizers, they can find parameters that are comparable in quality to those obtained with GAs. We observe that CMA-ES leads to slightly better results and is less sensitive to the initial guess of the parameters. Concrete recipes are provided for obtaining similar results with new training sets.

Besides optimization recipes, a successful ReaxFF parameterization requires the design of a good training set. At every trial set of parameters, ReaxFF is used to optimize molecular geometries in the training set. When the optimization of some geometries fails easily, it becomes increasingly difficult to find the optimal parameters. We have addressed this issue by fixing several bugs in the ReaxFF forces and by improving the robustness of the geometry optimization. These improvements cannot eliminate all geometry convergence issues and we recommend to avoid very flexible geometries in the training set.

Both MCFF and CMA-ES are still liable to converge to sub- or near-optimal parameters, which we detected by repeating the calibration with different random seeds. The existence of distinct near-optimal parameter vectors is a general pattern throughout our study and provides opportunities to improve the training set or to detect overfitting artifacts.

1 Introduction

Molecular dynamics is a powerful tool to study the temporal evolution of various atomistic models (of materials, solute-solvent systems, biomolecular systems, etc.) under realistic conditions, e.g. at constant temperature and pressure. An essential ingredient in such simulations is a computationally efficient method to reliably compute (at every discrete time step) the forces acting on all atoms, which are needed to integrate Newton’s equation of motion. The quantum-mechanical (QM) treatment of the molecular electronic wavefunction

allows one compute these forces for any (reactive) chemical system, e.g. using density functional theory (DFT) methods. However, for long molecular dynamics simulations (nano- to microseconds) of extended atomistic models (up to millions of atoms), even DFT approximations becomes computationally prohibitive. Alternatively, one may use so-called force-field (FF) or molecular mechanics approximations, where the electronic structure calculation is replaced by a much cheaper and approximate model to compute forces acting on atoms. Many FFs model chemical bonds by simple springs with empirical parameters in Hooke’s law, which immediately reveals their major drawback, i.e. most force fields cannot describe chemical reactions. Reactive FFs overcome this limitation with a more complex mathematical expression that can describe reactive processes¹⁻⁴ and ReaxFF⁵⁻⁸ is one of the most popular models in this category. ReaxFF owes its popularity to a combination of unique advantages. It can be reparameterized for different combinations of chemical elements, which makes ReaxFF broadly applicable. ReaxFF is also computationally efficient compared (tight-binding) DFT approximations⁹ and neural-network potentials.¹⁰ As opposed to hybrid QM/MM methods, ReaxFF can describe complex systems where many reactive events occur simultaneously throughout the atomistic model. A traditional example is the simulation of hydrocarbon oxidation,^{11,12} where many chemical reactions quickly succeed each other. ReaxFF is also applied in situations where reactive events go beyond a reaction network of small molecules, e.g. to study the mechanical wear resistance of graphene.¹³

Even though a detailed description of the complete mathematical form of ReaxFF goes beyond the scope of this paper, it is instructive to review a few of its essential aspects. The ReaxFF model is a potential energy expression for an atomistic model that ultimately takes the Cartesian coordinates of atomic nuclei and a set of empirical ReaxFF parameters as input. Analytic differentiation of this energy towards the displacement of atomic positions yields the forces needed in a molecular dynamics simulation. The ReaxFF energy is a sum

of many contributions,

$$E_{\text{ReaxFF}} = E_{\text{bond}} + E_{\text{over}} + E_{\text{under}} + E_{\text{val}} + E_{\text{tors}} + E_{\text{vdW}} + E_{\text{charge}} + E_{\text{specific}} \quad (1)$$

most of which are covalent terms responsible for describing local chemical phenomena: bond breaking & formation (bond), over-coordination (over), under-coordination (under), valence angle bending (val), bond torsion (tors). The next two terms describe non-covalent interactions between all pairs of atoms within a cutoff distance, even when they are not chemically bonded. The van der Waals (vdw) interaction is similar to the Morse potential and captures all effects due to steric repulsion and dispersion interactions. The energy of the fluctuating charge model (charge) includes the pairwise screened Coulomb interaction and the polarization cost of the fluctuating charges. An important intermediate step needed for the local energy terms is the bond order for each pair of nearby atoms, which is derived from interatomic distances. Because these bond-orders are re-computed at every atomic configuration, ReaxFF is capable of describing chemical phenomena. Finally, ReaxFF contains several optional terms, which are not used in all parameterizations:

$$E_{\text{specific}} = E_{\text{lp}} + E_{\text{pen}} + E_{\text{coa}} + E_{\text{conj}} + E_{\text{trip}} + E_{\text{H-bond}} + E_{\text{lg}} \quad (2)$$

These terms can handle phenomena that are only of interest in specific cases: interaction with a lone pair (lp), penalty for valence angles between two double bonds (pen), conjugation correction for valence angles (coa), conjugation correction for bond torsions (conj), triple-bond stabilization (trip) and hydrogen bonding (H-bond). The low-gradient (lg) pairwise R^{-6} term was introduced to better describe long-range dispersion interactions¹⁴ but is rarely used in recent parameterizations. Another recent extension developed by van Duin is eReaxFF, adding explicit electron or hole particles that can interact with the atoms.¹⁵

As a consequence of its wide adoption, several ReaxFF implementations were developed next to the original “Standalone ReaxFF” by van Duin.^{5,8,11} The development of new

energy terms in ReaxFF were done in the original code and were later adopted in other implementations. One of the earliest alternative (and parallel) implementations was presented in LAMMPS.¹⁶ Other notable implementations of ReaxFF can be found in GULP,¹⁷ the implementation by Nomura¹⁸ and in PureMD.¹⁹ The latter two implementations strongly focus on efficient parallelization on high-performance clusters. One concern with various implementations is that they are not fully compatible with the original code by van Duin, e.g. because some were re-implemented from scratch. This introduces the risk that ReaxFF parameterizations from the literature, calibrated with one implementation, may no longer yield sensible results in another implementation. All results in this work were obtained with ReaxFF in an alpha version of ADF2018. This implementation is directly based on the original code by van Duin to assure that the functional form of the potential energy closely follows the original. Still, many changes were made to improve the efficiency and parallel scaling in ADF.²⁰ Even though the potential energy is compatible, we fixed several bugs in the force evaluation to make geometry optimization more robust, in the frame of this project. We noticed, as will be discussed below, that such bug fixes sometimes trouble the reproduction of literature results. Because fortuitous literature results that rely on bugs are generally not transferable to production simulations, we prefer to implement the forces as precisely as possible.

The ReaxFF energy expression contains many empirical parameters, which need to be fixed somehow, before ReaxFF can be used for production simulations. While a considerable number of tuned parameter sets are published in the literature,^{5,11,21-27} one must extend this effort whenever one wants to use ReaxFF for a new type of chemistry, which was not considered previously in parameter optimization studies. In such a scenario, on the order of 50 new parameters need to be estimated, by minimizing the deviations between ReaxFF predictions and corresponding reference data, i.e. the training set, which consists of (usually microscopic) properties of relevant small atomistic model systems. Finding the optimal parameters for a given training set is far from trivial. While this was traditionally a laborious

task where one parameter was optimized at a time,²⁸ more advanced and automated approaches were proposed over the last decade, most notably genetic algorithms (GAs)^{27,29–31} but also other methods, such as a Multi-Objective Evolutionary Strategy (MOES),³² a parallel local search method,³³ Taguchi method based optimization,³⁴ Monte-Carlo force-field optimizer^{25,35} (MCFF) and Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES).^{36,37} The goal of this work is to compare the efficiency of two parameter optimization strategies implemented in an alpha version of ADF2018: MCFF and CMA-ES. For this comparison, we will use the same reference data that was used in benchmarks and applications of the GAs developed by Hartke (Ogolem),³¹ to facilitate comparison to their work. GAs are also popular for the parametrization of other force fields^{38–43} and we expect that the comparison in this work is therefore also useful beyond the scope of ReaxFF.

The reference data used to optimize ReaxFF parameters are collected in a training set. This is a set of molecules or crystals for which properties, $x_{i,ref}$, were measured experimentally or computed with QM calculations. One also has to identify a set of ReaxFF parameters, p_j , which should be optimized to improve the ReaxFF predictions, $x_{i,calc}(\{p_j\})$, of the properties in the training set. The quality of a set of parameters is quantified by an error function, hereafter referred to as *the Error*. We will use the same least-squares Error as in the original work of van Duin:²⁸

$$\text{Error}(\{p_j\}) = \sum_{i=1}^n \left(\frac{x_{i,calc}(\{p_j\}) - x_{i,ref}}{\sigma_i} \right)^2 \quad (3)$$

where the sum runs over all items in the training set. For each reference property, also a tolerance, σ_i , is defined in the training set, which is an estimate of the acceptable deviations between a ReaxFF calculation and a reference value. In ADF2018, Error contributions due to dihedral angles take into account angular periodicity. For example, a calculated dihedral angle of -170 deg and a reference value of $+170$ deg only results in a difference of 20 deg.

Even though the Error is a simple sum of squares, it is in practice a very ill-behaved function of the parameters $\{p_j\}$, with significant numerical noise and many local minima,^{25,31} because (i) the ReaxFF energy is a very non-linear and not exactly smooth function of the

ReaxFF parameters and (ii) the predictions $x_{i,\text{calc}}$ often involve poorly converging geometry optimizations of the molecules and crystals in the training set. This poor convergence is caused by several effects, such as small discontinuities in the ReaxFF potential energy, very flexible molecules in the training set, poor optimization algorithms and bugs in the implementation of the ReaxFF atomic forces. While many problems in the latter two categories were fixed in the development of ADF2018, as part of this work, we decided to keep the ReaxFF potential energy as it is, except for an optional correction in the torsion term, which will be discussed below. We also tried to use training sets from the literature without further correcting them. (For one of the training set used, several corrections were unavoidable to obtain useful results, as will be discussed below.) Our conservative attitude is motivated by the need for compatibility with other ReaxFF implementations and to keep the assessment of MCFE and CMA-ES realistic: we would like to know how these algorithms perform for representative training sets from the literature.

Because the Error is such a rugged function of the ReaxFF parameters, with many local minima, it seems most appropriate to test global optimization algorithms. Several groups have designed GAs specifically for ReaxFF parameterization and showed that these can minimize the objective function equally well or even further than the successive one-parameter parabolic extrapolation (SOPPE) originally introduced by van Duin.^{31,44} Furthermore, these GAs no longer require manual intervention and human judgment while they minimize the Error. A drawback of GAs is their large computational cost, typically requiring a HPC infrastructure to perform the parameter optimization. The MCFE and CMA-ES algorithms tested in this work are not global optimization algorithms in the strict sense but they can handle the numerical noise in the Error and they can escape higher local minima to some extent. Barcaro et al. noted that MCFE requires much less computing power (or Error evaluations) than a GA⁴⁵ and the goal of this work is therefore to test how well MCFE and CMA-ES can still perform with relatively few Error evaluations. The tendency of these optimizers to get trapped in a local minimum will be tested by restarting the optimization

from the same initial guess or from different random initial parameter vectors.

When simply minimizing the Error in Eq. (3), there is a significant risk to overfit the parameters. Such overfitted parameters have a low Error for the training set but they may still produce very unphysical results when they are used in realistic ReaxFF molecular dynamics simulations.^{46,47} While we recognize the importance of this problem, this work solely focuses on the testing the efficiency of the MCFF and CMA-ES parameter optimization algorithms. To avoid overfitting, one may introduce a so-called test set, in addition to the training set. Overfitting can then be detected by an increase of the Error on the test set while the error on the training set still decreases.²⁷

The remainder of this paper is structured as follows. In the Methods section, we describe ReaxFF geometry optimization details, training sets, parameter optimization algorithms with their settings and the evaluation criteria used to compare MCFF and CMA-ES. The performance of the algorithms is described and compared in section 3, where we also discuss some implications on the design of an effective training set. Finally, section 4 summarizes the main conclusions of this work and gives an outlook on future work.

2 Methods

2.1 Geometry optimization

Prior to the evaluation of the Error [Eq. (3)] with a set of trial parameters, the energy of many (if not all) molecules in the training set needs to be minimized, resulting in the optimal geometries for the trial parameters. The properties of these optimal geometries are then used to evaluate the Error. Only when forces of non-equilibrium geometries are used in the training set, the geometry optimization is not performed. Because the convergence of the geometry optimization affects the value of the Error, and thus also the optimal parameters, we used relatively stringent convergence settings and tuned the line-search algorithm in the L-BFGS geometry optimizer to handle cases with negative curvature more gracefully. A

geometry optimization is considered to be converged if the maximum force on any of the atoms drops below $0.1 \text{ kcal mol}^{-1} \text{ \AA}^{-1}$. In some difficult cases, convergence is still not reached after 3000 steps, after which the algorithm is stopped and the last geometry is used instead of the optimal one. Such convergence issues may appear when the optimal geometry is near a point on the ReaxFF energy surface where the forces have discontinuities, or when the Hessian matrix, representing the curvature of the potential energy surface, is ill-conditioned.

2.2 Training sets

The MCFF and CMA-ES algorithms were compared and tested for the optimization of ReaxFF parameters using three training sets from the literature. In order of increasing number of parameters, there is one training set for solid and liquid cobalt and defects in cobalt crystals,²⁴ one for silica clusters and (porous) crystals²³ and one focusing on disulfide mechanochemistry.²⁷ Key properties of the three training sets are compared in Table 1, which shows that these sets are fundamentally different in terms of types of information they contain. For example, the Cobalt set only contains energy data, while the Disulfide set is the only one containing atomic forces.

Table 1: Overview of ReaxFF training sets used in this work. The number of data points in each training set is broken down into five categories: C (atomic charges), G (geometry, internal coordinates), F (Cartesian atomic forces), P (cell parameters) and E (reaction energies). Note that unused geometries in the `geo` files were not counted and similarly non-existing geometries in the `trainset.in` file were ignored. For the Disulfide training set, also the literature value for the absolute minimum of the Error is included, for which the parameters were reportedly overfitted.

Label	N_{par}	N_{geo}	Number of data points						Literature Error	ADF2018 α Error	
			C	G	F	P	E	Total		Default	Torsion2013
Cobalt	12	146					144	144	1444	1459	1459
Silica	67	302	5	26		13	265	309	3196	17369	17484
Disulfide (overfitting)	87	231		255	4401		219	4875	12393 7574	15577 n/a	16271

In the first place, a training set provides information to evaluate the Error, see Eq. (3). This is the primary information and consists of a list of molecule or crystal geometries for

which reference data is provided, which may include atomic charges, equilibrium geometries (internal coordinates), Cartesian atomic forces, cell parameters and reaction energies. The original papers from which the training sets were taken, also contain secondary information: the final optimal parameters and constraints that were imposed during the parameter optimization. We followed these constraints as closely as possible. In each training set, many parameters are kept fixed at values from older ReaxFF parameterizations, for mainly two reasons. First, a subset of the parameters is usually taken from an older force field, for which the new training set contains no relevant data. This guarantees compatibility with older models and simplifies the extension of ReaxFF with new chemical elements. Second, a list of global ReaxFF parameters (not associated with chemical elements) is kept fixed, also to maintain compatibility with older models. For each variable parameter, an interval of allowed values is also provided, which will be denoted by $[p_i^{\min}, p_i^{\max}]$ for parameter p_i in the remainder of the text. In case of the Silica training set, we have slightly modified some of these intervals, as explained in section S1 of the supporting information.

Despite the fact that we used training sets from the literature without modification, except for one typographical error described in section S1 of the supporting information, we did encounter difficulties reproducing literature values of the Error for optimal parameters published with these training sets. The last three columns of Table 1 compare Errors reported in the literature with the one we computed with ADF2018 α . While the correspondence is acceptable for the Cobalt training set, the Error computed with ADF for the Silica and Disulfide training sets is significantly larger. These two training sets contain several very flexible molecules, which are hard to fully optimize with ReaxFF. To facilitate these geometry optimizations, we improved the robustness of the line-search in the L-BFGS algorithm and we also fixed several bugs in the evaluation of the ReaxFF forces (without changing the energy expression). In addition, the optional Torsion2013 improvement was implemented to facilitate geometry optimization. (See supporting information, section S2 for technical details.) We noticed that the default torsion energy term results in large discontinuities in

the forces during bond breaking and formation, which impaired the geometry optimization. Our Torsion2013 improvement heavily reduces these discontinuities without significantly altering the energy landscape. For example, the last two columns of Table 1 show that the Torsion2013 correction has little impact on the Error value. (The error for Cobalt is not affected because this ReaxFF parameterization does not use torsional terms.) These bug fixes and technical improvements facilitate geometry optimization, which significantly reduce numerical noise in the Error function, which then facilitates the parameter optimization. However, these improvements also have an adverse effect: using published parameters, we can now find lower-energy structures for molecules in the Silica and Disulfide training sets. These lower energy structures also existed prior to our improvements to the code but were just not found by the geometry optimizer. Because these lower energy structures deviate further from the reference data, we observe an increased Error compared to literature values, as can be seen in Table 1.

In case of the Silica training set, there is a second reason for the large deviation between our computed Error and the literature value. In this case, it was not possible to use the optimal parameters from the literature because these often resulted in unphysical geometries for molecules in the training set, also due to the improved geometry optimization. Hence, we used a minimally corrected set of parameters instead, as described in section S1 of the supporting information. For example, some of the original parameters exceeded their allowed interval, sometimes even with a change of sign. This caused unrealistic behavior of the ReaxFF force field, such as molecules collapsing into a single point in space. Such unphysical results only appeared after we introduced a more robust geometry optimization algorithm in our ReaxFF implementation. We assume similar issues were not encountered previously due to weaknesses of geometry optimization algorithms in other ReaxFF implementations.

2.3 Initial guess of the parameter vector

We used different types of initial guesses for the parameters, to test the influence of the quality of the guess on the outcome of the MCFE and CMA-ES optimizers. The ‘best’ type of initial guess is somewhat artificial, in the sense it is the optimal parameter vector previously reported for these training sets (except for modifications described in the supporting information, section S1). This ‘best’ guess may not be optimal in this work for two reasons: (i) we improved the numerical stability of the Error function in several ways and (ii) the literature parameters were in some cases the result of an early-stopping algorithm to avoid overfitting.

A more realistic ‘educated’ initial guess was constructed from a database of ReaxFF force-field parameters maintained by SCM.⁴⁸ We only extracted information from parameter sets that were published before the corresponding training set. For each parameter, the following steps were taken:

1. All unique historical parameter values were looked up for the same type of parameter, associated with the same combination of chemical elements. Only those lying in the allowed interval for the training set, $[p_i^{\min}, p_i^{\max}]$, were retained and used to compute a median value as guess.
2. If in the previous step, no historical parameters were found in the allowed interval, the search for unique historical parameters was extended and they only had to be of the same type but they were allowed it to be associated with other chemical elements. Of all these values, we took the median.
3. If again no historical values could be found in the previous step, we just took the center of the allowed interval as initial guess.

In case of the Cobalt training set, there is only one older parameterization predating the training set²² and the result of the above procedure is that the educated guess for Cobalt coincides with the parameters from Ref. 22.

Finally, we also constructed for each training set, 10 ‘random’ initial parameter vectors. For each parameter, we sampled random values from a uniform distribution over the interval $[p_i^{\min} + (p_i^{\max} - p_i^{\min})/4, p_i^{\max} - (p_i^{\max} - p_i^{\min})/4]$, that is, the central segment of the allowed interval spanning half the width.

2.4 MCFF

The Monte-Carlo force field optimizer²⁵ uses simulated annealing to find optimal parameters.³⁵ While this is in principle a global optimization method, a true global optimization with simulated annealing requires very slow cooling rates, which can be too costly to realize in practice. At every iteration, MCFF makes a small change to the parameter vector and computes the corresponding change in Error function. In this work, the small step consisted of a random change, of 10% of the parameters, sampled from a uniform distribution over the interval $[-(p_i^{\max} - p_i^{\min})/100, (p_i^{\max} - p_i^{\min})/100]$. The step-size rescaling feature of MCFF was disabled as we found it often lead to premature convergence. When the error decreases, the step is always accepted. In case of an increase, it is accepted with a probability $\exp(-\beta_n \Delta \text{Error})$, where β_n is the inverse dimensionless temperature at iteration n .

For each combination of training set and initial guess of the parameters, we performed three MCFF runs with 9k, 3x3k and 45k iterations. In case of 3x3k, 3 MCFF runs of 3000 steps were done in series, where the second and the third run are restarts using the optimal parameters from the previous run as initial guess.¹⁰ The initial inverse temperature was always determined by

$$\beta_0 = \sqrt{\frac{N_{\text{par}}}{2}} \frac{1}{C_1 \text{Error}_0}, \quad (4)$$

where N_{par} is the number of optimized parameters, Error_0 is the Error for the initial parameters. In case of the second and third segment of a 3x3k run, β_0 is derived from the initial Error of the current restart, not the Error of the initial guess. C_1 is approximately the initial magnitude of relative thermal fluctuations of the Error and was set to 1, such that MCFF

can escape local minima easily. If one prefers MCFF to perform a more local search, C_1 should be reduced by one or two orders of magnitude. The final inverse temperature is set to

$$\beta_N = \sqrt{\frac{N_{\text{par}}}{2}} \frac{1}{C_2}, \quad (5)$$

where C_2 is approximately the absolute fluctuation on the error at the final iterations and was set to 5. Such a small value for C_2 will let MCFF converge to the bottom of a (local) minimum in the Error. The above relation between Error fluctuations and inverse temperature would be exact if the Error were a quadratic function of the ReaxFF parameters and the sampling were complete, as shown in section S3 of the supporting information. In practice, these relations are approximate because the Error is a more intricate function and the number of steps is too small for a complete sampling, especially when the Error is nearly flat in some directions.²⁹ To obtain an annealing simulation, β_n is divided by a constant factor at every iteration of the MCFF algorithm:

$$\frac{\beta_n}{\beta_{n+1}} = \sqrt[N]{\frac{C_2}{C_1 \text{Error}_0}} \quad (6)$$

where N is the total number of iterations. The above configuration of the MCFF algorithm can be implemented with the control parameters in Table 2.

2.5 CMA-ES

The Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES) is a stochastic gradient-free optimization algorithm proposed by Hansen *et al.*^{36,49} Starting from a user-provided initial guess, CMA-ES iteratively improves a multivariate normal distribution in the parameter space to find a distribution whose random samples minimize the objective function. In essence, one iteration consists of the following steps and we refer to Ref. 36 for a detailed description:

1. A population of $\lambda = 4 + \lceil 3 \ln N_{\text{par}} \rceil$ random points (trial parameter vectors) is drawn

Table 2: MCFF and CMA-ES settings used in the ReaxFF control file. Control parameters not included in the table are left to their default value.

MCFF		
<code>mcffit</code>	$N = 9000, 3000 \text{ or } 45000$	Number of MCFF iterations.
<code>mcbeta</code>	$\beta_0 = \sqrt{\frac{N_{\text{par}}}{2} \frac{1}{C_1 \text{Error}_0}}$	Initial inverse temperature.
<code>mcbsca</code>	$\frac{\beta_n}{\beta_{n+1}} = \sqrt{\frac{C_2}{C_1 \text{Error}_0}}$	Value by which the inverse temperature is divided at each step.
<code>mcstep</code>	1.0	Controls the (initial) step size. The maximum change of a parameter in one MC step is <code>mcstep/mcrxdd</code> .
<code>mcrxdd</code>	100	Controls the (initial) step size.
<code>mcscps</code>	1.0	Controls step size rescaling. By setting it to 1.0, step size rescaling is disabled.
<code>mcacpf</code>	0.1	Fraction of the parameters that is changed in one MCFF step.
CMA-ES		
<code>mcffit</code>	$N = 20000$	Maximum number of CMA-ES iterations.
<code>ffotol</code>	$\text{To1X} = 10^{-6} \text{ or } 10^{-5}$	CMA-ES convergence criterion. (See text for details.)
<code>mcrxdd</code>	$N_\sigma = 4$	Controls the width of the initial normal distribution in parameter space.

from the normal distribution.

2. The Error is computed for each trial parameter vector.
3. The population is sorted by increasing Error and only the first $\lambda/2$ points are retained and assigned a weight.
4. The mean (center) and covariance of the normal distribution are updated using the weighted points, using heuristic rules explained in Ref. 36.
5. When $\sigma\|\mathbf{p}_g\|$ and $\sigma \max_i \sqrt{C_{ii}}$ drop below a threshold, To1X , convergence is reached and the algorithm stops. In these criteria, σ is a variable step size, C is the current estimate of the covariance matrix and \mathbf{p}_g is an average over previous steps with an exponential window. More details on these quantities can be found in Ref. 36. Alternatively, one may also stop after a maximum number of iterations.

The value of the objective function at each trial point is thus only used to rank the points, which makes CMA-ES invariant to any rank-preserving (strictly increasing) transformation of the objective function. As the evaluation of each trial point is completely independent of the rest of the population, this step of CMA-ES is trivially parallel.

Starting from the initial guess, the covariance matrix C is incrementally improved by the feedback from sampled points and tends to approximate the inverse Hessian matrix, thus capturing the relative sensitivities of the parameters and also the correlations between them. The step length σ (overall width of the sampled distribution) is automatically controlled by the algorithm in every iteration, depending on the directions of previous steps. If subsequent steps tend to move in a similar direction, the step length is increased accordingly. If, instead, subsequent steps tend to be in opposite directions, the algorithm is overshooting an optimum and thus responds by scaling down the step length.

The initial mean of the normal distribution is set to the initial guess (see section 2.3) and the initial covariance is diagonal with each diagonal element set to $((p_i^{\max} - p_i^{\min})/N_\sigma)^2$, where

$N_\sigma = 4$. With this value of N_σ , CMA-ES starts with a relatively broad initial distribution, such that the algorithm explores a large portion of the parameter space before converging. One may turn CMA-ES into a more local optimizer with higher values of N_σ . We used $\text{To1X} = 10^{-6}$ for the Cobalt training set and $\text{To1X} = 10^{-5}$ for the two other training sets. These convergence criteria are very tight and will let CMA-ES continue even if it would seem practically converged. This allows us to test if there is anything to be gained by letting the algorithm continue for many steps. Finally, we also terminate CMA-ES when it reaches 20k steps. Our configuration of the CMA-ES algorithm can be implemented with the control parameters in Table 2.

The CMA-ES algorithm is implemented using the `c-cma-es` library.⁵⁰ The upper and lower bounds on the parameters (p_i^{\min} and p_i^{\max}) are imposed by setting the Error to the largest possible double precision number (approximately 1.8×10^{308}) whenever parameters fall outside their allowed interval.

Because multiple Error evaluations are used in one CMA-ES iteration (λ defined above), one must be careful when comparing the efficiency with MCFF, in terms of number of iterations needed to achieve a low Error. For the Cobalt, Silica and Disulfide training sets, λ is 11, 16 and 17, respectively. In a serial calculation, this would be the relative cost of one CMA-ES iteration compared to MCFF. However, in ADF, multiple Error evaluations can be carried out in parallel and within one Error computation parallelism is further exploited. In practice, this means that CMA-ES can make better usage of parallelism than MCFF.

2.6 Evaluation criteria

While the main performance criterion in this work is obviously the lowest value of the Error reached during an optimization, we also check other aspects of the trajectories through parameter space followed by the optimization algorithms. For each combination of training set, initial guess and optimizer settings, we performed 10 optimizations. Each optimization run is ranked by the lowest Error reached at any point during the optimization. The lowest

Error for the best, second best and worst run are compared to test (i) the sensitivity to the initial guess and (ii) the tendency of the algorithm to get locked into local minima. We also report the number of iterations needed to reach this lowest Error for the best run, which is a good indication for the computational cost of the optimization. To assess how much the parameters have changed, we also compute the distance (Euclidean norm) d_1 between the initial guess and the parameter vector of the lowest Error (for the best run) in reduced parameter units:

$$\tilde{p}_i = \frac{p_i}{p_i^{\max} - p_i^{\min}} \quad (7)$$

Finally, we also compute the distance d_2 between the optimal parameters of the best and second best run, to investigate parameter degeneracies. It was previously observed that the Error is not sensitive to certain linear combinations of parameters, making the optimal parameters degenerate.²⁹

3 Results and discussion

In total 36 sets, with 10 parameter optimizations each, were carried out in this study. There is one set for every combination of (i) three different training sets (Cobalt, Silica and Disulfide), (ii) two optimizers (MCFF and CMA-ES) with three different cooling rates in MCFF and (iii) three different qualities of the initial guess (best, educated and random). A summary of the numerical results for each set, in line with the evaluation criteria discussed above, is given in Table 3. In addition, Figure 1 shows the decrease of the error as function of optimizer iteration for three representative sets out of the 36. Similar plots for all 36 sets are included in section S4 of the supporting information.

The most important result in Table 3 is that, with any of the 36 sets of optimization runs, there is a significant spread on the lowest Error, i.e. the difference between best and worst run. The variation over 10 runs in one set is in the first place due to the initialization of the random seed, which affects results because the optimizers are stochastic. Obviously, when

Table 3: Overview of all parameter optimization results. Each row summarizes 10 optimization runs carried out with identical settings. (See text for details.) d_1 is the distance from the initial guess to the optimal parameter vector for the best run. d_2 is the distance between the optimal parameter vectors in the best and second best runs. Distances are in dimensionless parameter units, see Eq. (7).

Training set	Algorithm	#iter max [k]	Guess	Best run		2 nd -best run	Worst run	d_1	d_2
				#iter to lowest [k]	Lowest Error	Lowest Error	Lowest Error		
Cobalt	MC FF	9	best	8.2	1358	1367	1420	0.18	0.43
			edu	9.0	1468	1724	2392	0.57	0.39
			rand	9.0	1486	1563	4189	0.39	0.64
		3x3	best	2.8	1360	1364	1435	0.15	0.10
			edu	8.9	1465	1591	2322	0.63	0.42
			rand	8.9	1615	1663	4055	0.41	0.70
		45	best	35	1358	1365	1392	0.48	0.25
			edu	45	1337	1354	1740	0.81	0.79
			rand	36	1369	1691	3786	0.86	1.40
	CMA-ES	20	best	4.5	1180	1199	2437	1.45	0.43
			edu	8.1	1157	1172	2467	1.47	0.99
			rand	2.1	1150	1168	3013	1.01	0.70
Silica	MC FF	9	best	9.0	3690	3692	5069	0.88	0.81
			edu	9.0	4405	4424	6350	0.85	1.06
			rand	7.8	28789	40679	442882	0.90	1.75
		3x3	best	9.0	3718	3762	5518	0.77	0.92
			edu	9.0	4395	4545	5896	0.96	1.01
			rand	8.9	9407	9712	241855	1.24	2.06
		45	best	44	3608	3634	5913	1.34	1.95
			edu	45	3897	3995	5894	1.74	2.07
			rand	44	4966	6377	71165	1.92	2.82
	CMA-ES	20	best	6.0	3791	3890	10217	2.14	2.21
			edu	7.2	3742	3870	10539	2.08	2.40
			rand	13	3727	4097	7888	2.68	2.75
Disulfide	MC FF	9	best	8.2	11899	11981	14305	1.80	2.49
			edu	8.4	14852	15721	19655	2.93	3.06
			rand	8.7	11960	14332	25634	2.75	2.77
		3x3	best	8.9	10914	11248	13389	1.44	1.79
			edu	5.9	13754	14595	20816	2.65	2.79
			rand	6.0	13886	14311	39381	2.71	3.10
		45	best	34	10605	11719	15341	2.83	3.18
			edu	44	9608	11828	19898	3.27	3.33
			rand	44	8507	9684	15274	2.90	2.69
	CMA-ES	20	best	18	8994	10337	13257	2.34	2.89
			edu	15	8693	9128	15386	3.12	2.79
			rand	15	6716	9386	14665	3.10	2.57

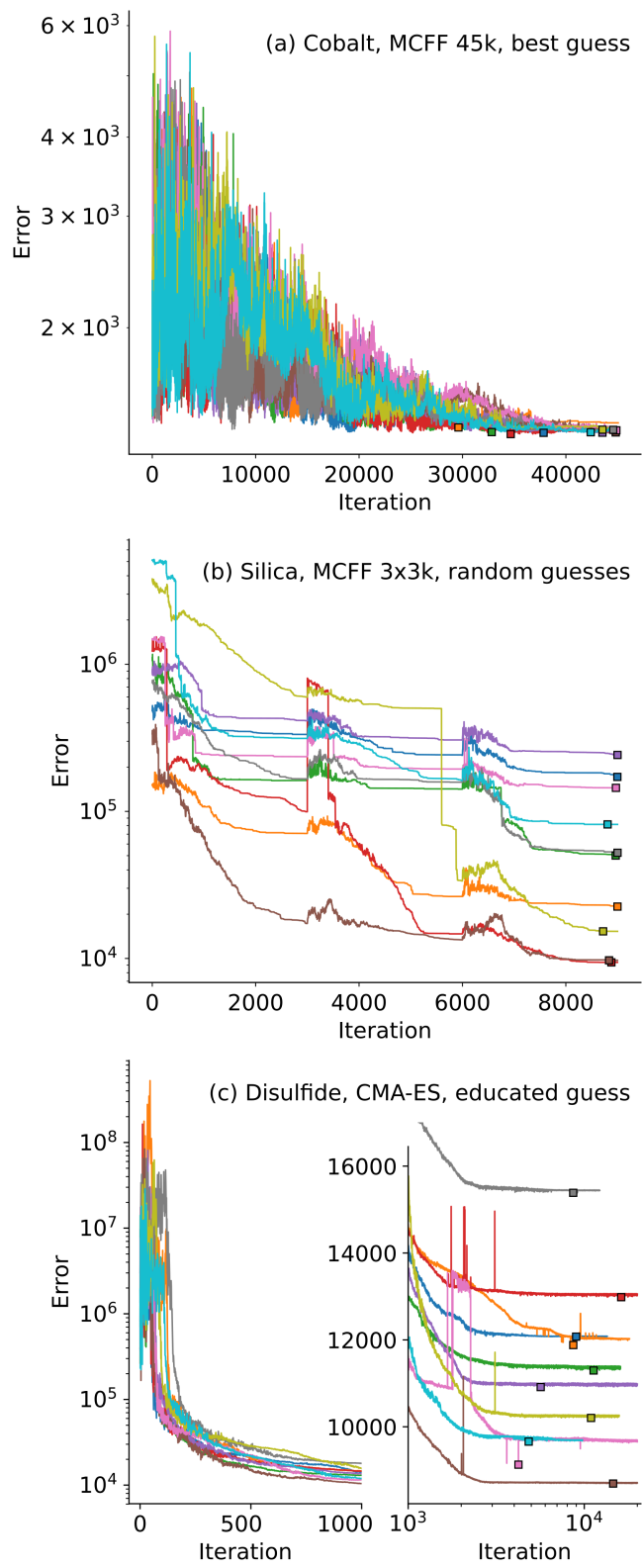


Figure 1: Representative plots of the Error as function of MCFF (a, b) and CMA-ES (c) iteration. Each plot displays the progress of the Error for 10 optimizations, with square boxes showing the lowest error achieved in each run. (Training set and algorithm settings indicated in each panel.) Differences within one plot are due to the stochastic behavior of the optimization algorithms and in panel (b) also due to the different random initial guesses.

random initial guesses are used within one set, the differences in initial guess also affect the outcome. (For the best and educated guess, the same guess is used 10 times.) The lowest Error in the worst run is in most cases too high to be useful, e.g. compared to the Error values from the literature in Table 1. Therefore, we recommend to repeat the optimization several times with different random seeds to obtain a sufficiently low Error.

A second important observation is that for the Cobalt and Disulfide training sets, we often see Error values significantly below those reported in the literature, e.g. those obtained with genetic algorithms (GA). This is somewhat surprising because GAs are often portrayed as effective global optimizers and expected to find (to good approximation) the global minimum of the Error. It seems that we can still outperform GAs for two of the three training sets with CMA-ES (and for the Cobalt set also with MCFF). While both algorithms are generally not considered to be exhaustive global optimizers, they are still remarkably effective. This positive result may also be due to several improvements and bug fixes in ReaxFF forces and geometry optimization in the frame of this work. These improvements result in a smoother (and thus easier to minimize) Error function. In addition, these improvements may also directly result in a lower (or higher) Error for the same set of parameters, which may also cause differences with literature values of the Error. Finally, as mentioned in the introduction, the low Error from the best run may imply some overfitting, which can be detected by also computing the Error on a validation set.

When comparing MCFF to CMA-ES, the Error as function of iteration is markedly different, as shown in Figure 1. MCFF slowly cools down the parameters and the lowest Error is usually encountered close to the end. Longer MCFF runs (45k steps) lead to lower Errors for the best runs in Table 3, as one would expect from any simulated annealing method. Only for the Silica training set, with a random initial guess, the restarts (3x3k steps) are advantageous over one slower annealing of 9k steps. This suggests that restarting MCFF can, in some cases, indeed reduce the risk of converging to a higher local minimum in the Error landscape.¹⁰ Because these restarts do not dramatically deteriorate the lowest

Errors for the other training sets, it can be safely used by default. There are often still significant improvements in the last restart, see e.g. Figure 1b, such that more restarts could lead to even lower Error values. CMA-ES first explores a wide region of the parameter space, resulting in very high initial Errors. Afterwards, it exhibits a very rapid decrease of the Error in the first 500 iterations, after which the Error levels off. Even when taking the difference in cost of a single MCFF (1 Error evaluation) versus CMA-ES (10 to 20 Error evaluations) in account, the initial decrease of the Error is more efficient in the case of CMA-ES. 83 of the 90 CMA-ES runs reach within 4000 steps an Error below 1.01 times the lowest Error over the entire run. Nevertheless, in a few runs, even when the Error has reached a plateau after 4000 CMA-ES steps, it may still suddenly decrease in later steps. Therefore, when using CMA-ES, it is useful to implement a pruning scheme,³⁷ where one first performs a series of short CMA-ES optimization with different initial guesses or random seeds and then continues only the most promising ones with additional CMA-ES iterations. The number of steps needed in the short runs depends on the complexity of the training set and the number of free ReaxFF parameters. For the training sets considered in this work, a few thousand steps would have been sufficient.

CMA-ES clearly outperforms MCFF in case of the Cobalt and Disulfide trainings set, reaching significantly lower Error values. CMA-ES is also performing well for the Silica training set, except that in one case MCFF reaches even a lower Error (best guess, 45k steps) When also considering the worst out of 10 runs in each set, CMA-ES gives more consistent results, independent of the initial guess, while MCFF may converge at very high Error values when using a random initial guess, e.g. in case of the Silica trainig set with a random initial guess. The superior efficiency of CMA-ES, especially in the first 500 iterations can be easily understood. The covariance matrix in CMA-ES models the shape of the Error function, allowing for more intelligent choices of new trial parameter vectors. In contrast, MCFF uses no such information when stepping through the parameter space.

The quality of the initial guess may have some positive influence on the lowest Error,

but the effect is marginal and in many cases reversed. A logical example can be found when optimizing parameters with MCFE over 9k steps for the Cobalt set: the lowest Error with best, educated and random guess are 1358, 1468 and 1486, respectively. An interesting counter example is the optimization of parameters for the Disulfide set with CMA-ES, where the lowest Error with best, educated and random guess are 8994, 8693 and 6716, respectively. In addition, in all sets of 10 runs with random initial guesses, there is no significant rank correlation between Error for the initial guess and the lowest Error. The limited impact of the initial guess is consistent with our settings of MCFE and CMA-ES, i.e. they will explore a significant part of the parameter space in the first iterations in an attempt to avoid convergence to local minima with a high Error. This also implies that the parameter trajectories quickly depart from the initial guess. One may lower the initial fluctuations in MCFE (lower C_1 in Eq. 4) or the initial width in CMA-ES (higher N_σ) to let these algorithms stay closer to the initial guess. Obviously, this also increases the risk of getting trapped in local minima with a high Error.

Finally, we observed that two solutions with nearly identical Errors, obtained with the same initial guess and the same algorithm, can have significant differences in the parameters. An illustrative example is the optimization of parameters for the Silica training set with CMA-ES using the educated initial guess. In this case, the best and second-best runs have similar Errors: 3742 and 3870, respectively. Yet, the distance between these two solutions ($d_2 = 2.40$) is of the same order as the distance from the initial guess to the optimal parameters of the best run ($d_1 = 2.04$). This is a general pattern: comparable (low) Errors can be obtained with very different parameter vectors. This is most likely due to the presence of several local minima with a similar depth in the Error function. To shed some light on origin of distinct solutions with nearly the same Error, Figure 2 shows the Error as function of a linear interpolation (in 1000 steps) between the solution from the best and second-best run (CMA-ES and educated guess), for the three training sets. These scans show that the Error is not a convex function and may thus have several local minima. Our findings do

not exclude the possibility that the Error is insensitive to certain linear combinations of parameters, which could also result in multiple solutions with a similar Error.²⁹

The curves in Figure 2 also exhibit a significant degree of noise for the Disulfide and to larger extent for the Silica training set, which was also observed in previous works.^{25,44} It may even be surprising that there is no visible noise in case of the Cobalt training set, probably because the Cobalt force field only uses two-body terms, thereby eliminating some sources of noise. To illustrate the severity of the noise, the Error of neighboring points in Figure 2(b) can differ by 1000 units, while the parameters change by less than 0.5%. Such levels of noise alone can create many local minima, most of which are irrelevant. For this reason, noise-resistant (and thus certainly gradient free) parameter optimization techniques are needed. Even though MCFE and CMA-ES can still function with some noise, it does degrade the performance of any optimization algorithm. It appears that the jumps in Figure 2 are caused by 8 of 309 items in the Silica training set and 11 of 4875 items in the Disulfide training set. After removing these items, one obtains the Error plotted in red, which is much smoother. We repeated some of the CMA-ES runs after removing the problematic items from these training sets. While the lowest Error for the worst run decreases notably, the results for the best run do not improve significantly. This means that the problematic items in the training set mainly increase the risk that CMA-ES ends at a higher local minimum. It is therefore desirable to design training sets that result in minimal noise in the Error.

To understand the origin of the noise in Figure 2, we investigated every term in the Error function separately (see Eq. (3)) along the linear scan. For a given amount of noise in a ReaxFF prediction, Δx_i , around a mean value $\langle x_i \rangle$, the noise in the Error becomes (to first order):

$$\Delta E \approx 2 \frac{|\langle x_i \rangle - x_{i,\text{ref}}| \Delta x_i}{\sigma_i^2} \quad (8)$$

When we observe significant noise in the Error, this can be either due to a large first or second factor. This is consistent with earlier work of Larsson et al,⁴⁴ where it was observed that the Error function becomes smoother near the optimal parameters, which can be explained

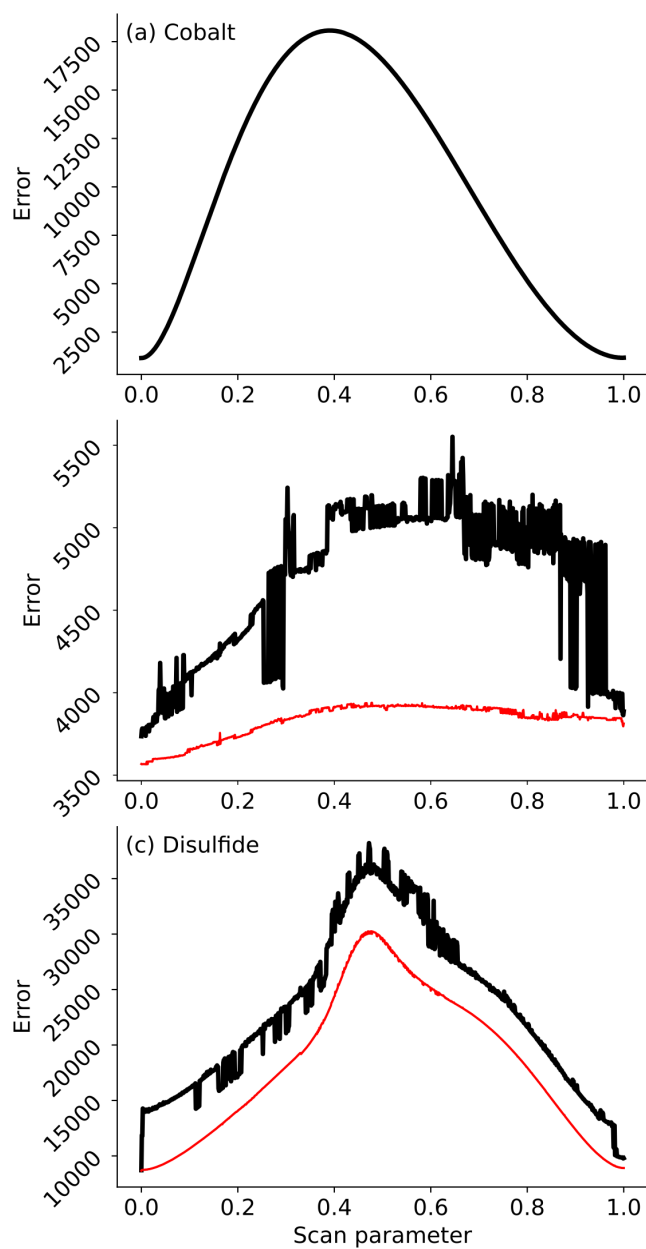


Figure 2: The Error as function of a linear interpolation between two parameter vectors, for each training set. The end points are the solutions of the best and second best run when optimizing the parameters with CMA-ES using the educated initial guess. The black curve is the total Error. In case of the Silica and Disulfide training sets, the red curve is the Error without those contributions that cause large discontinuous jumps in the scan.

by a decrease of the first factor.

For the Silica training set, apparently only 8 out of the 309 Error terms are responsible for the largest jumps in Figure 2. Several smaller discontinuities are also present due to other items in the training set, which we will not discuss here. They appear for exactly the same reasons as the larger jumps. Most of the problematic terms in the Error are related to molecules for which at least two (but often many more) metastable conformations or configurations exist. Two examples are shown in Figure 3. The 12-membered silica ring in Figure 3a can have many slightly different conformations due to the high flexibility of the Si-O-Si angles. Figure 3b represents the products of a silica condensation reaction. The water molecule is weakly bound to the condensed silica cluster, with two possible relative positions of water. In both cases, which of the possible configurations is found after optimization, is very sensitive to the force-field parameters and the corresponding energies differ by 20 kcal/mol, resulting in sudden changes of the Error by approximately 300 units. One could avoid these problems by (a) using more rigid molecules and (b) by including reaction products as separate molecules in the training set instead of combining them into a single complex. Another problematic case is the energy of a slightly expanded unit cell of quartz. The training set specifies that this geometry should only be optimized for five steps, instead of the usual 3000. The energy after five steps is very sensitive to the exact parameters, even though the geometry differences are very small. In this case, the geometry is far from converged after five steps and it is still on a very steep slope of the potential energy surface, such that small changes in geometry cause large differences in energy. The exact geometry after five steps can be very sensitive to algorithmic details of the geometry optimizer, which is usually irrelevant when assessing the performance of a force field. In this case, allowing for more geometry iterations should be helpful but it also increases the computational cost of the Error.

For the Disulfide training set, 11 out of 4875 Error terms are responsible for all the visible noise in Figure 2. These 11 terms measure errors on dihedral angles (torsions about

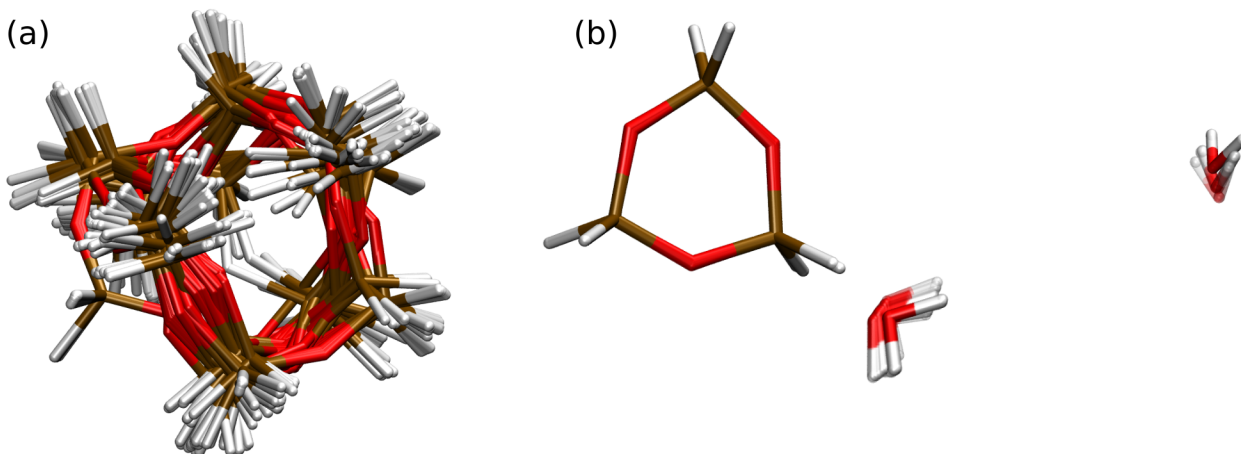


Figure 3: Two of the problematic molecules in the Silica training set responsible for discontinuous jumps in Figure 2b: (a) a 12-membered silica ring and (b) the product of a ring-closing condensation reaction of a linear silica trimer. For each molecule, 100 optimized geometries are shown in overlay, obtained with different ReaxFF parameters along the scan. In part (b), the water molecule, which is a product from this condensation reaction, is part of this geometry, for which ReaxFF predicts roughly two stable positions relative to the three-ring.

C-O and C-S bonds) in the four molecules shown in 4. The geometry optimization of the four molecules is usually not complete after 3000 steps. In case of convergence failure, the last geometry is used to evaluate the Error, because it's expected to be the best available approximation of a converged result. However, due to the incomplete convergence, the internal coordinates contain a virtually random component. Even though we have improved the geometry optimizer and fixed bugs in the computation of the forces to reduce geometry convergence issues, it appears extremely challenging to eliminate this problem completely. This is partially caused by small discontinuities in the ReaxFF forces, when any pair of atoms passes the bond-order cutoff. While the discontinuities are small, they become significant when the geometry optimizer is nearly converged. Even with perfectly smooth forces, similar problems may still arise. For example, if the curvature of the PES is very ill-conditioned for a set of trial parameters, the optimal geometry becomes ill-defined. Another problematic situation appears when a molecule has two or more conformations with comparable energies. With small changes in force field parameters, the geometry optimizer may converge to one

or the other conformer, resulting in a discontinuous Error. In short, it is in principle still valuable to improve the ReaxFF geometry convergence in future versions of ADF, but this will never completely eliminate noise from the Error function. For an effective parameter optimization with ReaxFF, every geometry optimization in the training set must be carefully designed to avoid convergence issues. Alternatively, one could also avoid geometry optimization altogether, but that would significantly raise the challenge of designing good training sets.

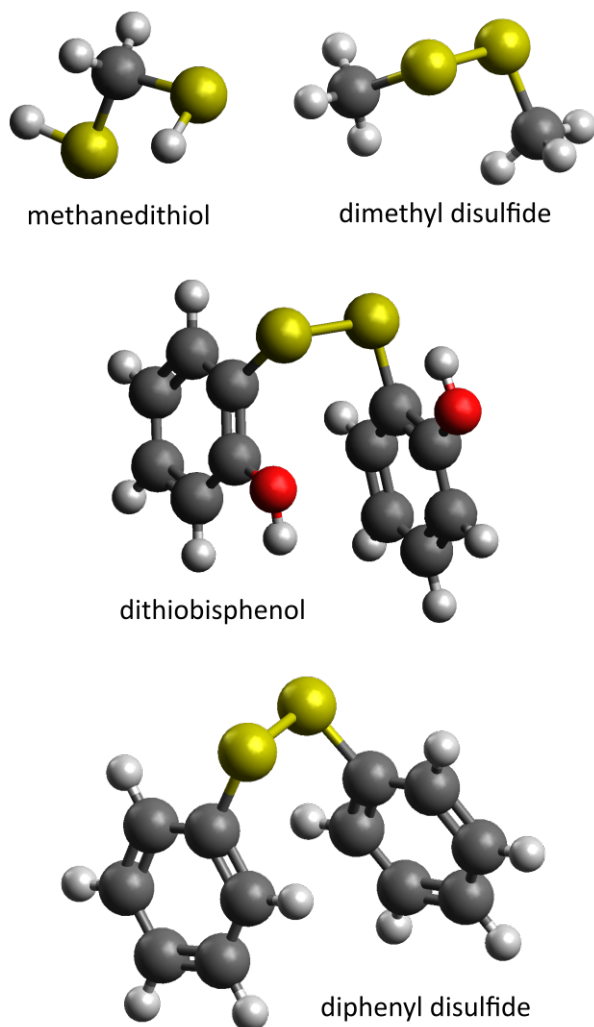


Figure 4: Four molecules in the Disulfide training set, whose Errors on the dihedral angles are responsible for the noise in Figure 2c.

The relation between geometry convergence and noise in the Error provides a second

explanation for the observation of Larsson *et al.* that the Error becomes smoother near the optimal parameters. The geometries provided in the training set are normally the optimal ones that ReaxFF should reproduce. Hence, with good ReaxFF parameters, fewer geometry optimization steps are needed, such that there is a lower risk for geometry convergence failures and corresponding discontinuities in the Error function.

4 Conclusions and outlook

We have systematically compared the performance of the MCFF and CMA-ES optimizers for three ReaxFF training sets, which have been used to assess genetic algorithms in the literature. Based on our results, we can make the following recommendations for an efficient ReaxFF parameter optimization with the methods considered in this work:

1. CMA-ES is generally preferred over MCFF for its robustness w.r.t the initial guess and the low Error values that can be reached. In some cases, most notably with the Silica training set and random initial guesses, MCFF may converge at very high Error values.
2. Good or clever initial guesses for the parameters have a limited impact on the performance of CMA-ES with the settings used in this work. (See Table 2.) Instead of trying to carefully construct one initial guess, we recommend to repeat the optimization with many random guesses.
3. Because CMA-ES nearly converges in the first few thousand iterations, a pruning scheme with multiple runs is more efficient than running all of them until convergence. This can be implemented by first performing a series of optimizations for a fixed number of iterations, with different guesses or random seeds. Only those few runs that have reached a relatively low Error within a fixed number of iterations, are then worth continuing until CMA-ES reaches convergence.

For two of the three training sets considered in this work, the optimal parameters have lower Error values than those reported in the literature with genetic algorithms. While the optimizers tested in this work should not be considered as exhaustive global optimizers, their ability to find low minima of the Error is very encouraging. In terms of computational cost, CMA-ES is also attractive compared to a GA. With the above recommendations, optimal parameters can be found with a few 10k Error evaluations, which is an order of magnitude less than a GA.²⁷

Our benchmarks resulted, for each of the training sets considered, in clearly different ReaxFF parameters (most likely different minima in the Error function) but with a comparably low Error value. Intuitively, one may simply prefer the lowest minimum. However, the availability of different solutions of comparable quality opens up new opportunities for future work. For example, one of these solutions may suffer less from overfitting, which can be checked by computing the Error on a validation set. In that case, the lowest Error on the validation set could be a good selection criterion. In addition, these multiple solutions may be used to estimate the reliability of a ReaxFF production run. If different sets of near-optimal parameters give inconsistent outcomes, the simulations are not reliable. To overcome this issue, one may identify properties of molecules that are predicted differently by the near-optimal parameter sets and add them to the training set. Including such reference data will narrow down the region in the parameter space where the Error is low, potentially also reducing overfitting artifacts.

Our assessment also revealed the importance of a robust geometry optimization for the calibration of ReaxFF parameters. When geometries converge poorly, the corresponding ReaxFF predictions contain some numerical noise, impairing the calibration of parameters. In the course of this work, we have refined the geometry optimization algorithm used in the ADF2018 implementation of ReaxFF and we eliminated several bugs in the computation of the forces, resulting in a smoother Error function. While such improvements are clearly beneficial, discontinuities in the Error may still appear when optimized geometries have a high

sensitivity to the force field parameters, e.g. in case of very flexible systems in the training set with different possible conformations. For an effective ReaxFF parameter calibration, one must carefully analyze each geometry optimization to avoid convergence issues or multiple (meta)stable configurations.

Acknowledgement

This project received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreements No. 764810 (S4CE) and 739746 (NET) and under the Marie Skłodowska-Curie grant agreement No. 641887 (DEFNET). T.V. acknowledges the Research Board of Ghent University (BOF) for its financial support. The computational resources and services used in this work were partially provided by the VSC (Flemish Supercomputer Center), funded by the FWO (Research Foundation Flanders).

Supporting Information Available

The changes to the literature values of the Silica parameters and the allowed ranges for these parameters during the calibration, a detailed description of the Torsion2013 correct, a derivation of initial and final inverse temperatures for MCFF and plots of Error versus iteration for all parameter calibration reported in table 3 are documented in the Supporting Information. Also a ZIP file is provided with the (updated) training sets, initial and the optimal parameters from each calibration run. This material is available free of charge via the Internet at <http://pubs.acs.org/>.

References

- (1) Daw, M. S.; Baskes, M. I. Semiempirical, quantum mechanical calculation of hydrogen embrittlement in metals. *Phys. Rev. Lett.* **1983**, *50*, 1285–1288.

- (2) Tersoff, J. New empirical approach for the structure and energy of covalent systems. *Phys. Rev. B* **1988**, *37*, 6991–7000.
- (3) Brenner, D. W.; Shenderova, O. A.; Harrison, J. A.; Stuart, S. J.; Ni, B.; Sinnott, S. B. A second-generation reactive empirical bond order (REBO) potential energy expression for hydrocarbons. *J. Phys. Condens. Mat.* **2002**, *14*, 783–802.
- (4) Phillpot, S. R.; Antony, A. C.; Shi, L.; Fullarton, M. L.; Liang, T.; Sinnott, S. B.; Zhang, Y.; Biner, S. B. Charge Optimized Many Body (COMB) potentials for simulation of nuclear fuel and clad. *Comput. Mater. Sci.* **2018**, *148*, 231–241.
- (5) van Duin, A. C. T.; Dasgupta, S.; Lorant, F.; Goddard, W. a. ReaxFF: A Reactive Force Field for Hydrocarbons. *J. Phys. Chem. A* **2001**, *105*, 9396–9409.
- (6) Liang, T.; Shin, Y. K.; Cheng, Y.-T.; Yilmaz, D. E.; Vishnu, K. G.; Verners, O.; Zou, C.; Phillpot, S. R.; Sinnott, S. B.; van Duin, A. C. Reactive Potentials for Advanced Atomistic Simulations. *Annu. Rev. Mater. Res.* **2013**, *43*, 109–129.
- (7) Han, Y.; Jiang, D.; Zhang, J.; Li, W.; Gan, Z.; Gu, J. Development, applications and challenges of ReaxFF reactive force field in molecular simulations. *Front. Chem. Sci. Eng.* **2016**, *10*, 16–38.
- (8) Senftle, T. P.; Hong, S.; Islam, M. M.; Kylasa, S. B.; Zheng, Y.; Shin, Y. K.; Junkermeier, C.; Engel-Herbert, R.; Janik, M. J.; Aktulga, H. M.; Verstraelen, T.; Grama, A.; van Duin, A. C. T. The ReaxFF reactive force-field: development, applications and future directions. *npj Comput. Mater.* **2016**, *2*, 15011.
- (9) Qian, H.-J.; van Duin, A. C. T.; Morokuma, K.; Irle, S. Reactive Molecular Dynamics Simulation of Fullerene Combustion Synthesis: ReaxFF vs DFTB Potentials. *J. Chem. Theory Comput.* **2011**, *7*, 2040–2048.

- (10) Boes, J. R.; Groenenboom, M. C.; Keith, J. A.; Kitchin, J. R. Neural network and ReaxFF comparison for Au properties. *Int. J. Quant. Chem.* **2016**, *116*, 979–987.
- (11) Chenoweth, K.; van Duin, A.; Goddard, W. ReaxFF Reactive Force Field for Molecular Dynamics Simulations of Hydrocarbon Oxidation. *J. Phys. Chem. A* **2008**, *112*, 1040–1053.
- (12) Ashraf, C.; van Duin, A. C. Extension of the ReaxFF Combustion Force Field toward Syngas Combustion and Initial Oxidation Kinetics. *J. Phys. Chem. A* **2017**, *121*, 1051–1068.
- (13) Berman, D.; Deshmukh, S. A.; Sankaranarayanan, S. K. R. S.; Erdemir, A.; Sumant, A. V. Extraordinary Macroscale Wear Resistance of One Atom Thick Graphene Layer. *Adv. Funct. Mater.* **2014**, *24*, 6640–6646.
- (14) Liu, L.; Liu, Y.; Zybin, S. V.; Sun, H.; Goddard, W. A. ReaxFF-ig: Correction of the ReaxFF Reactive Force Field for London Dispersion, with Applications to the Equations of State for Energetic Materials. *J. Phys. Chem. A* **2011**, *115*, 11016–11022.
- (15) Islam, M. M.; Kolesov, G.; Verstraelen, T.; Kaxiras, E.; van Duin, A. C. T. eReaxFF: A Pseudoclassical Treatment of Explicit Electrons within Reactive Force Field Simulations. *J. Chem. Theory Comput.* **2016**, *12*, 3463–3472.
- (16) Budzien, J.; Thompson, A. P.; Zybin, S. V. Reactive Molecular Dynamics Simulations of Shock Through a Single Crystal of Pentaerythritol Tetranitrate. *J. Phys. Chem. B* **2009**, *113*, 13142–13151.
- (17) Gale, J. D.; Raiteri, P.; van Duin, A. C. T. A reactive force field for aqueous-calcium carbonate systems. *Phys. Chem. Chem. Phys.* **2011**, *13*, 16666.
- (18) Nakano, A.; Kalia, R. K.; Nomura, K.-i.; Sharma, A.; Vashishta, P.; Shimojo, F.; van Duin, A. C.; Goddard, W. A.; Biswas, R.; Srivastava, D. A divide-and-conquer/cellular-

- decomposition framework for million-to-billion atom simulations of chemical reactions. *Comput. Mat. Sci.* **2007**, *38*, 642–652.
- (19) Aktulga, H.; Fogarty, J.; Pandit, S.; Grama, A. Parallel reactive molecular dynamics: Numerical methods and algorithmic techniques. *Parallel Comput.* **2012**, *38*, 245–259.
- (20) Kamat, A. M.; van Duin, A. C. T.; Yakovlev, A. Molecular Dynamics Simulations of Laser-Induced Incandescence of Soot Using an Extended ReaxFF Reactive Force Field. *J. Phys. Chem. A* **2010**, *114*, 12561–12572.
- (21) van Duin, A. C. T.; Strachan, A.; Stewman, S.; Zhang, Q.; Xu, X.; Goddard, W. A. ReaxFF_SiO Reactive Force Field for Silicon and Silicon Oxide Systems. *J. Phys. Chem. A* **2003**, *107*, 3803–3811.
- (22) Nielson, K. D.; Duin, A. C. T. V.; Oxgaard, J.; Deng, W.-Q.; Iii, W. A. G.; van Duin, A.; Goddard, W. Development of the ReaxFF Reactive Force Field for Describing Transition Metal Catalyzed Reactions, with Application to the Initial Stages of the Catalytic Formation of Carbon Nanotubes. *J. Phys. Chem. A* **2005**, *109*, 493–499.
- (23) Fogarty, J. C.; Aktulga, H. M.; Grama, A. Y.; van Duin, A. C. T.; Pandit, S. A. A reactive molecular dynamics simulation of the silica-water interface. *J. Chem. Phys.* **2010**, *132*, 174704.
- (24) LaBrosse, M. R.; Johnson, J. K.; van Duin, A. C. T. Development of a Transferable Reactive Force Field for Cobalt. *J. Phys. Chem. A* **2010**, *114*, 5855–5861.
- (25) Iype, E.; Hütter, M.; Jansen, A. P. J.; Nedeia, S. V.; Rindt, C. C. M. Parameterization of a reactive force field using a Monte Carlo algorithm. *J. Comput. Chem.* **2013**, *34*, 1143–1154.
- (26) van Duin, A. C. T.; Zou, C.; Joshi, K.; Bryantsev, V.; Goddard, W. A. *Comput. Catal.*; **2013**; pp 223–243.

- (27) Müller, J.; Hartke, B. Reax FF Reactive Force Field for Disulfide Mechanochemistry, Fitted to Multireference ab Initio Data. *J. Chem. Theory Comput.* **2016**, *12*, 3913–3925.
- (28) van Duin, A. C. T.; Baas, J. M. A.; van de Graaf, B. Delft molecular mechanics: a new approach to hydrocarbon force fields. Inclusion of a geometry-dependent charge calculation. *J. Chem. Soc., Faraday Trans.* **1994**, *90*, 2881–2895.
- (29) Pahari, P.; Chaturvedi, S. Determination of best-fit potential parameters for a reactive force field using a genetic algorithm. *J. Mol. Mod.* **2012**, *18*, 1049–1061.
- (30) Jaramillo-Botero, A.; Naserifar, S.; Goddard, W. A. General Multiobjective Force Field Optimization Framework, with Application to Reactive Force Fields for Silicon Carbide. *J. Chem. Theory Comput.* **2014**, *10*, 1426–1439.
- (31) Dittner, M.; Müller, J.; Aktulga, H. M.; Hartke, B. Efficient global optimization of reactive force-field parameters. *J. Comput. Chem.* **2015**, *36*, 1550–1561.
- (32) Larentzos, J. P.; Rice, B. M.; Byrd, E. F. C.; Weingarten, N. S.; Lill, J. V. Parameterizing Complex Reactive Force Fields Using Multiple Objective Evolutionary Strategies (MOES). Part 1: ReaxFF Models for Cyclotrimethylene Trinitramine (RDX) and 1,1-Diamino-2,2-dinitroethene (FOX-7). *J. Chem. Theory Comput.* **2015**, *11*, 381–391.
- (33) Deetz, J. D.; Faller, R. Parallel Optimization of a Reactive Force Field for Polycondensation of Alkoxysilanes. *J. Phys. Chem. B* **2014**, *118*, 10966–10978.
- (34) Hu, X.; Schuster, J.; Schulz, S. E. Multiparameter and Parallel Optimization of ReaxFF Reactive Force Field for Modeling the Atomic Layer Deposition of Copper. *J. Phys. Chem. C* **2017**, *121*, 28077–28089.
- (35) Kirkpatrick, S.; Gelatt, C. D.; Vecchi, M. P. Optimization by Simulated Annealing. *Science* **1983**, *220*, 671–680.

- (36) Hansen, N. In *Towards a New Evolutionary Computation. Advances on Estimation of Distribution Algorithms*; Lozano, J., Larranaga, P., Inza, I., Bengoetxea, E., Eds.; Studies in Fuzziness and Soft Computing 192; Springer-Verlag Berlin Heidelberg, 2006; pp 75–102.
- (37) Trnka, T.; Tvaroška, I.; Koča, J. Automated Training of ReaxFF Reactive Force Fields for Energetics of Enzymatic Reactions. *J. Chem. Theory Comput.* **2018**, *14*, 291–302.
- (38) Barnes, B. C.; Gelb, L. D. Meta-Optimization of Evolutionary Strategies for Empirical Potential Development: Application to Aqueous Silicate Systems. *J. Chem. Theory Comput.* **2007**, *3*, 1749–1764.
- (39) Tafipolsky, M.; Schmid, R. Systematic First Principles Parameterization of Force Fields for MetalOrganic Frameworks using a Genetic Algorithm Approach. *J. Phys. Chem. B* **2009**, *113*, 1341–1352.
- (40) Ivanov, M. V.; Talipov, M. R.; Timerghazin, Q. K. Genetic Algorithm Optimization of Point Charges in Force Field Development: Challenges and Insights. *J. Phys. Chem. A* **2015**, *119*, 1422–1434.
- (41) France-Lanord, A.; Soukiassian, P.; Glattli, C.; Wimmer, E. Ab initio parameterization of a charge optimized many-body forcefield for SiSiO₂: Validation and thermal transport in nanostructures. *J. Chem. Phys.* **2016**, *144*, 104705.
- (42) Rodríguez-Fernández, R.; Pereira, F. B.; Marques, J. M.; Martínez-Núñez, E.; Vázquez, S. A. GAFit: A general-purpose, user-friendly program for fitting potential energy surfaces. *Comput. Phys. Commun.* **2017**, *217*, 89–98.
- (43) Zahariev, F.; De Silva, N.; Gordon, M. S.; Windus, T. L.; Dick-Perez, M. ParFit: A Python-Based Object-Oriented Program for Fitting Molecular Mechanics Parameters to ab Initio Data. *J. Chem. Inf. Model.* **2017**, *57*, 391–396.

- (44) Larsson, H. R.; Van Duin, A. C. T.; Hartke, B. Global optimization of parameters in the reactive force field ReaxFF for SiOH. *J. Comput. Chem.* **2013**, *34*, 2178–2189.
- (45) Barcaro, G.; Monti, S.; Sementa, L.; Carravetta, V. Parametrization of a Reactive Force Field (ReaxFF) for Molecular Dynamics Simulations of Si Nanoparticles. *J. Chem. Theory Comput.* **2017**, *13*, 3854–3861.
- (46) Moqadam, M.; Riccardi, E.; Trinh, T. T.; Åstrand, P.-O.; van Erp, T. S. A test on reactive force fields for the study of silica dimerization reactions. *J. Chem. Phys.* **2015**, *143*, 184113.
- (47) Hubin, P. O.; Jacquemin, D.; Leherte, L.; Vercauteren, D. P. Parameterization of the ReaxFF reactive force field for a proline-catalyzed aldol reaction. *J. Comput. Chem.* **2016**, *37*, 2564–2572.
- (48) Database of ReaxFF parameters managed by SCM. https://www.scm.com/doc/ReaxFF/Included_Forcefields.html, Accessed May 24, 2018.
- (49) Hansen, N.; Kern, S. Evaluating the CMA Evolution Strategy on Multimodal Test Functions. *Parallel Problem Solving from Nature - PPSN VIII*. 2004; pp 282–291.
- (50) Hansen, N. CMA-ES written in ANSI C. <https://github.com/CMA-ES/c-cmaes>, Accessed February 9, 2018.

Graphical TOC Entry

