JCTC Journal of Chemical Theory and Computation

Article

Cite This: J. Chem. Theory Comput. 2018, 14, 3100−3112

pubs.acs.org/JCTC

Enhanced Particle Swarm Optimization Algorithm: Efficient Training of ReaxFF Reactive Force Fields

David Furman,*[,†,‡] Benny Carmeli,[‡] Yehuda Zeiri,[§] and Ronnie Kosloff[†]

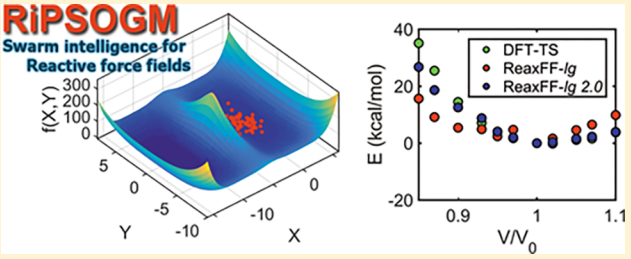
[†]Fritz Haber Research Center for Molecular Dynamics, Institute of Chemistry, Hebrew University of Jerusalem, Jerusalem 91904, Israel

[‡]Division of Chemistry, NRCN, P.O. Box 9001, Beer-Sheva 84190, Israel

[§]Department of Biomedical Engineering, Ben Gurion University, Beer-Sheva 94105, Israel

Supporting Information

ABSTRACT: Particle swarm optimization (PSO) is a powerful metaheuristic population-based global optimization algorithm. However, when it is applied to nonseparable objective functions, its performance on multimodal landscapes is significantly degraded. Here we show that a significant improvement in the search quality and efficiency on multimodal functions can be achieved by enhancing the basic rotation-invariant PSO algorithm with isotropic Gaussian mutation operators. The new algorithm demonstrates superior performance across several nonlinear, multimodal benchmark functions compared with the rotation-invariant PSO algorithm and the well-established simulated annealing and sequential one-parameter parabolic interpolation methods. A search for the optimal set of parameters for the dispersion interaction model in the ReaxFF-*lg* reactive force field was carried out with respect to accurate DFT-TS calculations. The resulting optimized force field accurately describes the equations of state of several high-energy molecular crystals where such interactions are of crucial importance. The improved algorithm also presents better performance compared to a genetic algorithm optimization method in the optimization of the parameters of a ReaxFF-*lg* correction model. The computational framework is implemented in a stand-alone C++ code that allows the straightforward development of ReaxFF reactive force fields.



1. INTRODUCTION

Reactive force fields, such as ReaxFF[1] and COMB,[2] allow unprecedented accuracy in emulating complex condensed-phase chemically reactive systems.[3,4] However, their development requires fitting many (occasionally above 100) empirical parameters to quantum-chemical reference data.[5,6] Correlations between the parameters make the process of finding a set of near-optimal solutions a particularly challenging global optimization problem. Therefore, the development of reactive force fields heavily relies upon a deep understanding of the underlying chemistry. For this reason, it is often considered to be an "art" reserved to few expert developers in academia.[7,8]

Until recently, the development of ReaxFF parameters has been mainly carried out using a sequential one-parameter parabolic interpolation (SOPPI) method.[9] Despite its simplicity, several drawbacks of this algorithm have been identified. The most prominent drawback is that the minimization procedure transforms a global optimization problem into a consecutive set of local one-parameter parabolic minimizations. This means that the algorithm finds a local minimum in parameter space that could be very far from the absolute minimum. Also, since only one parameter is optimized at a time, the procedure has to be repeated for many rounds until convergence is achieved. Lastly, if the process does not start with a very good initial guess, the resulting force field will converge to a poor-quality solution. Therefore, an efficient, automated optimization method is highly desirable and indeed remains an active area of research. Several global optimization approaches were suggested recently for fitting ReaxFF parameters to reference data, based on genetic algorithms (GAs),[10−12] simulated annealing (SA),[13,14] evolution strategies (ESs),[15] parallel parabolic search,[16] and advanced packages with multiple optimization schemes.[17] All of these methods have shown superior performance compared with the standard one-parameter search approach.

Particle swarm optimization (PSO) was proposed by Eberhart and Kennedy[18] in the mid 1990s as a promising global optimization algorithm. Originally intended for simulation of social behavior of bird flocks and fish schools, it was soon discovered to perform optimization in real-valued search spaces. The basic algorithm emulates the ability of a set of agents to work as a group in locating promising positions in a given search area. Since its invention, many studies have used PSO to find near-optimal solutions of complex multidimensional energy landscapes. Notable applications include crystal structure prediction,[19,20] screening of polymers for targeted self-assembly,[21,22] parametrization of Hamiltonian matrix

Received: December 19, 2017
Published: May 4, 2018

DOI: 10.1021/acs.jctc.7b01272
J. Chem. Theory Comput. 2018, 14, 3100−3112

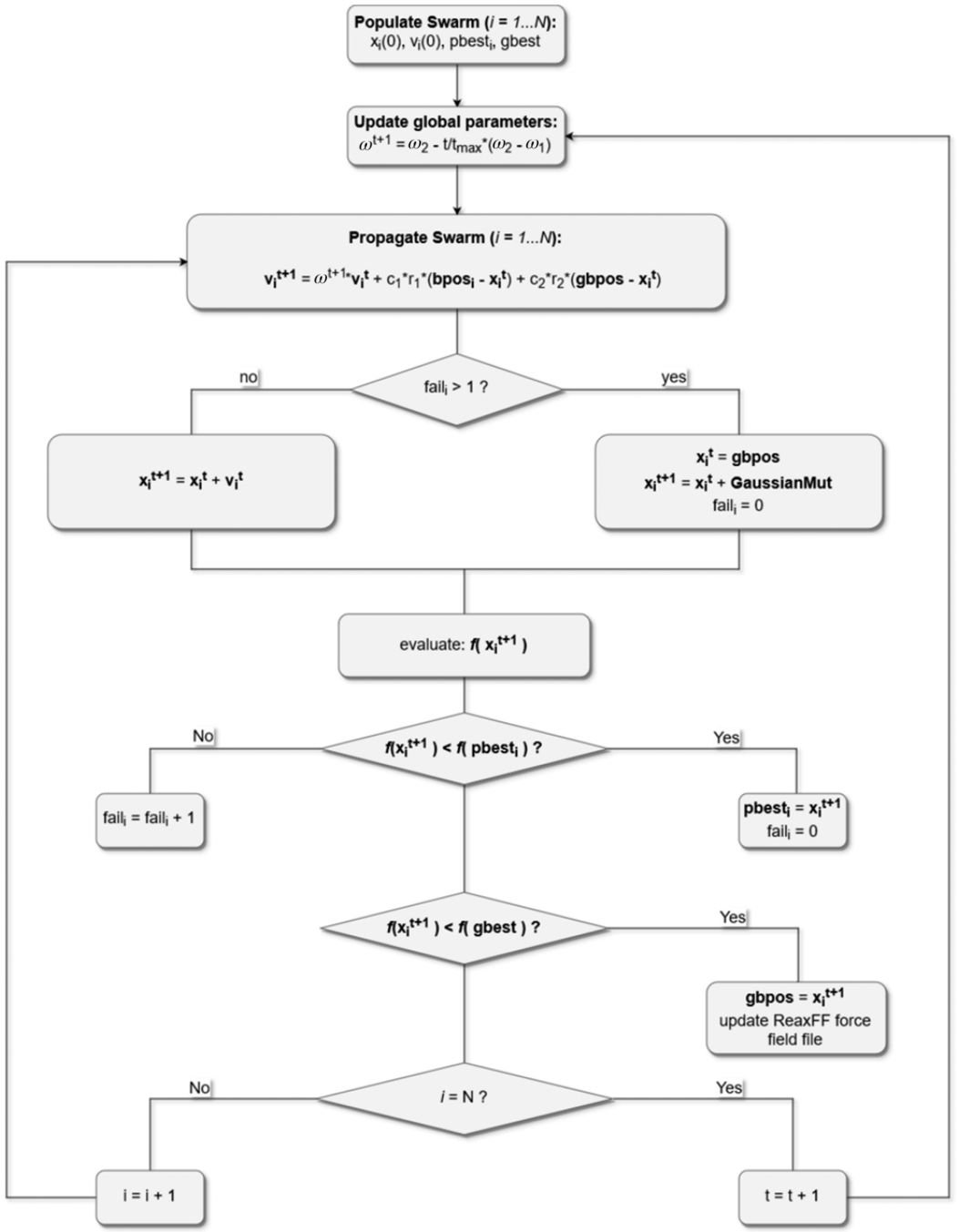


Figure 1. General scheme of the RiPSOGM global optimization algorithm. $c_1 = 2.0$, $c_2 = 2.0$, $\omega_1 = 0.9$, and $\omega_2 = 0.4$ are the recommended values for the PSO parameters.[46] The total number of iterations (t_{max}), the number of agents in the swarm (N), and the scale factor for mutation (γ, eq 2) are set following user input when running the program.

elements in DFTB models,[23] optimization of batch processing units,[24] and estimation of cosmological parameters.[25]

In this study, we propose to use and demonstrate the performance of a modified PSO algorithm augmented with mutation operations. The new algorithm has enhanced search abilities and is less prone to being trapped in local minima compared with previous formulations. The performance of the algorithm is analyzed in terms of solution quality, reproducibility, and cost compared with the standard rotation-invariant PSO, SA, SOPPI, and GA algorithms. Finally, we employ the method to find an optimal set of parameters in the low-gradient (*lg*) dispersion model used in ReaxFF[26] with respect to a new, extensive set of density functional theory with Tkatchenko–Scheffler dispersion correction (DFT-TS) calculations on energetic molecular crystals.

2. THEORETICAL BACKGROUND

2.1. Particle Swarm Optimization. PSO is a stochastic population-based algorithm for optimization of nonlinear, nondifferentiable functions. It is a global search method that does not exploit the use of local minimizers. Its major improvements over pure random search (i.e., "blind search") algorithms are threefold: (a) it accepts both promising and inferior solutions, thus greatly enhancing the search efficiency

DOI: 10.1021/acs.jctc.7b01272
J. Chem. Theory Comput. 2018, 14, 3100–3112

with less risk of premature convergence; (b) it allows a population of possible solutions to exist concurrently, thus greatly enhancing the global exploration characteristics; and (c) unlike SA, the PSO algorithm shares information among the search agents and thus is a self-learning algorithm. In standard PSO variants,[27,28] the position of search agents (i.e., solutions) depends on an inertial term together with an interplay between the agent personal experience and global information about the landscape searched that is shared among all agents (eq 1):

$$\mathbf{v}_{k+1}^{i} = \omega \mathbf{v}_{k}^{i} + c_1 r_{1k}^{i} \cdot (\mathbf{p}_{k}^{i} - \mathbf{x}_{k}^{i}) + c_2 r_{2k}^{i} \cdot (\mathbf{p}_{k}^{g} - \mathbf{x}_{k}^{i}) \quad (1a)$$

$$\mathbf{x}_{k+1}^{i} = \mathbf{x}_{k}^{i} + \mathbf{v}_{k+1}^{i} \quad (1b)$$

In eq 1, $\mathbf{v}_k^i$ and $\mathbf{x}_k^i$ are the velocity and position of agent i at iteration k, respectively, $\mathbf{p}_k^i$ is the personal best position of agent i, and $\mathbf{p}_k^g$ is the global best position of the entire swarm. The positions and velocities are d-dimensional, where d is the number of search variables in the objective function undergoing optimization. The coefficients c_1 and c_2 are constant weights of the personal and global terms, and ω is an inertia factor that acts to damp the momentum of the agents to prevent infinite growth of the velocities as the optimization progresses. The parameters r_{1k}^i and r_{2k}^i represent independent random scalars sampled from a uniform distribution in the range $[0, 1]$.

The use of random scalars (instead of vectors) leads to an algorithm that is invariant to rotations of the search space[29] (rotation-invariant PSO, RiPSO). Rotational invariance is closely related to decomposability and separability because in most cases a separable function becomes nonseparable under rotation. Therefore, a search algorithm can either exploit separability if a priori knowledge on promising search directions is available or be rotation-invariant. If such knowledge does exist and can be used (e.g., by calculating the correlations between search parameters), the algorithm will provide consistent performance across both separable and nonseparable functions.[29,30]

However, such information is seldom known for real-world optimization problems, such as complex molecular force fields. Furthermore, it is known that the RiPSO algorithm lacks diversity, resulting in swarm member trajectories that collapse into a low-dimensional subspace (i.e., line searches), leading to unfavorable performance.[29,31] Several efforts in recent years concentrated on enhancing the algorithm search diversity for efficient optimization on multimodal functions without sacrificing its rotational invariance. Wilke et al.[29] replaced the random scalars with random rotation matrices in the velocity update equation to increase the diversity of the search. This resulted in improved performance on several multimodal benchmark functions, although the rotation invariance was held in a stochastic sense only. The algorithm performed poorly on nonseparable functions where a high degree of correlation among the parameters existed. In the SPSO 2011 algorithm,[32] a geometric update rule exploiting uniform hypersphere sampling of positions was suggested as a novel rotation-invariant formulation. Nevertheless, it performed poorly on complex and multimodal functions and could not escape suboptimal regions.[33] The underlying reason for the apparent "exploration−invariance" trade-off is rooted in the conjecture that as long as no information about the correlations between variables exists (i.e., information about promising search directions), the algorithm should be as invariant as possible without bias to any preferred direction. However, if such information does exist, then surely enough the algorithm would exploit it and can no longer be rotationally invariant. In fact, the same trade-off holds true for other evolutionary algorithms such as standard GA.[34,35] Nevertheless, when no such information exists, the problem of premature convergence and lack of diversity of the standard RiPSO algorithm can be successfully overcome, as will be shown in the next section.

2.2. Enhanced Particle Swarm Optimization with Isotropic Gaussian Mutation. We suggest a simple and straightforward extension of the standard RiPSO algorithm to enhance its global search abilities by preventing its degeneration to line searches. In fact, any isotropic perturbation can prevent the degeneration of the trajectories into low-dimensional subspaces and is a rotation-invariant operator.[36] Taking inspiration from mutation operators used in GAs, we augment the PSO algorithm with an isotropic Gaussian mutation (GM) operator. As long as an agent moves toward better regions compared with its previous-best position, the mutation operator is disabled in order not to interfere with swarm dynamics. However, if it did not improve its fitness during a few consecutive steps, mutation takes place. The mutation operator acts to replace the current agent with a mutated version of the best agent in the swarm as follows:

$$\mathbf{x}_{k+1}^{i} = \mathbf{p}_{k}^{g} + \gamma G(0, 1)\hat{\mathbf{t}}_{k+1}^{i} \quad (2)$$

In eq 2, the agent's new position, $\mathbf{x}_{k+1}^i$, is set as an isotropic randomized value centered on the position of the best particle, $\mathbf{p}_k^g$. The step length is sampled from a Gaussian distribution with a mean of zero and a standard deviation of unity ($G(0, 1)$). The scale parameter γ controls the width of the distribution and should be related to the domain boundaries. The unit vector $\hat{\mathbf{t}}_{k+1}^i$ ensures that the distribution of displacements is isotropic by uniformly picking points on the surface of a d-dimensional hypersphere.[37] Thus, the algorithm is not affected by rotations of the search landscape. A general scheme of the algorithm (denoted as RiPSOGM) is presented in Figure 1. The algorithm starts by initializing the positions ($\mathbf{x}_i(0)$) and velocities ($\mathbf{v}_i(0)$) for all agents in the swarm. The previous best position of each agent ($\mathbf{p}_k^i$) is set to its initial position, and the global best position ($\mathbf{p}_k^g$) and global best objective function value (gbest) are determined. Next, the inertia weight ω is computed as a linearly decreasing function of the iteration number, t, where t_{max} is the maximum number of iterations and $\omega_1 = 0.4$ and $\omega_2 = 0.9$ are constants. Next, propagation of the swarm begins with evaluation of new velocities for the agents using eq 1a. At this point, the algorithm departs from standard PSO and checks the number of failures of the current member (fail_i). If $\text{fail}_i > 1$, a mutation takes place, namely, the position is replaced with a mutated version of the best member in the swarm up to that point. Otherwise, the standard update equation for the position (eq 1b) is used. Next, the objective function value is evaluated at the new position, and the best personal/previous position, global best positions, and corresponding objective function values are updated. The objective function value is evaluated for each agent using the function being optimized calculated at the current position of the agent. The algorithm iterates through all members for a predefined number of iterations or a threshold objective function value.

It is worth mentioning an unfortunate misconception present in many studies[38−43] implicitly suggesting that when employed for the optimization of multimodal functions, the exponentially decreasing tails of Gaussians are ineffective in acting as mutation operators because they are too local. Instead, it is

DOI: 10.1021/acs.jctc.7b01272
J. Chem. Theory Comput. 2018, 14, 3100−3112

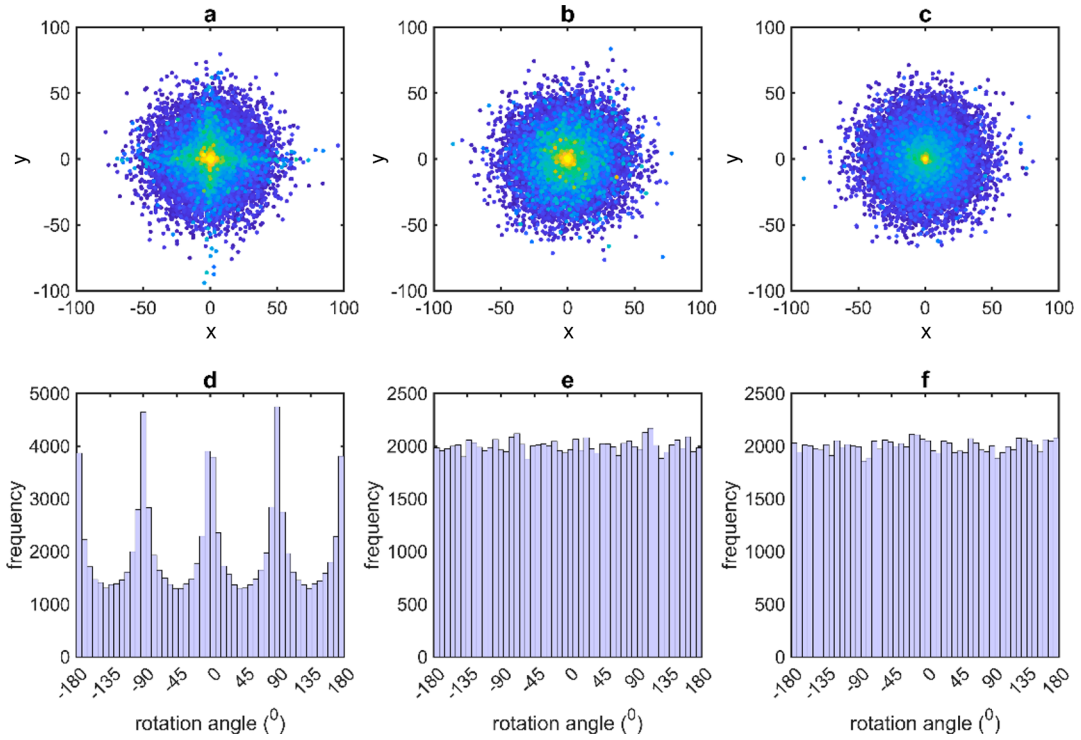


Figure 2. Top panels: search trajectories of 10^4 agents on the 2D distance from the origin function for (a) RbPSO, (b) RiPSO, and (c) RiPSOGM. The total number of function evaluations was 3×10^5. The initial agent positions were radially distributed according to a Gaussian distribution (mean $\mu = 0.0$ and standard deviation $\sigma = 20.0$). Agents are colored according to the iteration number (from dark blue to yellow). Bottom panels: respective angular orientations for (d) RbPSO, (e) RiPSO, and (f) RiPSOGM with respect to the x axis during search on the 2D distance from the origin function.

argued that heavy-tailed distributions, such as those of the Cauchy and Lévy distributions, are more appropriate since long jumps ("Lévy flights"[44]) can occasionally lead to better solutions within the attraction basin of a better optimum.

In fact, a careful analysis reveals that heavy tails are not the reason for the alleged superior performance. Rather, it is the separability of some benchmark functions that makes them an "easy target" for coordinate-wise line-search methods. In this respect, it should be noted that sampling a univariate distribution independently in each dimension (to compute the new position) is equivalent to sampling the corresponding multivariate heavy-tailed distribution of positions. Multivariate Cauchy and Lévy distributions are extremely anisotropic in the sense that large steps occur mostly along coordinate axes.[45] Thus, it is not surprising that they work so well on separable functions: the majority of the new sampling points will always fall along the coordinate axes, so the optimum is easily found, just as it is for the rotationally biased version of PSO[30] and as would be the case for any coordinate-wise search algorithm. When used on nonseparable functions (i.e., cases where correlation among function variables exists) or when the search direction is chosen isotropically, the superior performance is lost since the promising directions are no longer parallel to the search/sampling coordinates. This point was nicely demonstrated for the ES algorithm[45] and will also be demonstrated here in section 3.2.

3. RESULTS

3.1. Tests for Rotation Invariance. To test whether a rotational bias inherently exists in an algorithm, we use the radially symmetric two-dimensional (2D) *distance from the origin* as the first objective function (eq 3):

$$f_1(x, y) = \sqrt{x^2 + y^2} \tag{3}$$

Since the objective function is radially symmetric, there should be no preference in the search direction for a rotationally invariant algorithm. Thus, any deviation of agent positions from radial symmetry in the search trajectory would indicate a rotational bias in the underlying velocity update formula.

Figure 2 compares the agent search trajectories during optimization using the rotationally biased PSO algorithm (RbPSO), the rotationally invariant PSO algorithm (RiPSO), and the enhanced RiPSO with isotropic Gaussian mutation (RiPSOGM). As can be inferred from the resulting search trajectories (Figure 2a–c) and distributions of the corresponding velocity vector orientations (Figure 2d–f), the newly proposed RiPSOGM algorithm is free of any rotational bias, just like the standard RiPSO algorithm, whereas the RbPSO algorithm that uses random vectors instead of scalars in its velocity update formula (eq 1) presents a strong bias toward the axes.

To demonstrate that such an algorithmic bias could drastically degrade the search performance on nonseparable functions, we choose an extremely radially asymmetric 2D ellipse function (eq 4),

$$f_2(x, y) = \sqrt{\left(\frac{x}{0.000076}\right)^2 + \left(\frac{y}{0.000001}\right)^2} \tag{4}$$

and impose correlation between its two variables, x and y, by multiplying them by a Euclidean rotation matrix operator that

DOI: 10.1021/acs.jctc.7b01272
J. Chem. Theory Comput. 2018, 14, 3100–3112

pertains to an angle θ in the range $[0, \pi]$. This series of functions will serve as our objective function for optimization. The results are presented in Figure 3.

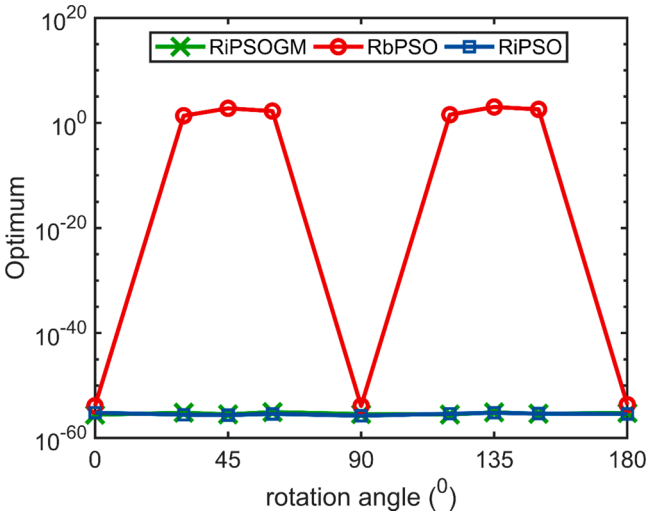


Figure 3. Optimization performance (best function value reached) on the 2D ellipse function as a function of rotation angle of the search landscape. The swarm consisted of 20 agents, and the total number of function evaluations was 10^5. Each point represents the average of 30 independent runs.

As expected, the RiPSO and RiPSOGM algorithms are insensitive to rotations of the coordinates, while RbPSO presents a performance degradation of roughly 60 orders of magnitude. The enhancement in exploration abilities stems from the fact that in each iteration, poor-performing agents are randomly reallocated around the global best particle. This operation ensures their escape to better positions compared with their original position.

3.2. Performance on Benchmark Functions. To assess the performance of the RiPSOGM algorithm in complex cases, four highly nonlinear benchmark functions were chosen as test cases for global optimization, namely, the Rastrigin, Ackley, Rosenbrock, and Schwefel benchmark functions commonly used for testing global optimization algorithms.[47,48] Their functional forms are given in the Supporting Information (Figures S1−S4). In addition, the performances of the RiPSO, SOPPI, and SA algorithms are also included for comparison. SOPPI is a simple local search method based on the assumption that a parabolic relation exists between the total error and the value of a single parameter.[9] SOPPI is a relatively poor approach compared with more advanced local search methods, but it is still being used to train ReaxFF force fields because of its simplicity. The optimal value for a parameter is obtained by calculating the total error at three different values of a single parameter while all of the others are held fixed. This defines a parabola and its corresponding minimum. The two new values are obtained by adding and subtracting a step size that is equal to 10% of the allowed range for the specified parameter. In this regard, the method can also be regarded as extrapolation since values that fall outside the three chosen points are not discarded. In addition, a test is performed to confirm that the extremum is indeed a minimum. In case the determined optimal value falls outside the given range or the parabola is concave, the value that corresponds to the smallest error is chosen. SA is a popular, frequently used global search method that does not use local minimizers. The algorithm we used is based on the Metropolis Monte Carlo method of Kirkpatrick et al.[49] to generate a random trial point. The distance of the trial point from the current point is given by a probability distribution that depends on the square root of the temperature, i.e., Boltzmann annealing.[50] If the new point is better than the current point, it is accepted with probability 1, and otherwise it is replaced by a new trial with an acceptance function based on the Boltzmann distribution. The algorithm then systematically lowers the temperature with an exponential rate of cooling, i.e., $T_{i+1} = 0.95 \cdot T_i$, where the initial temperature is $T_0 = 600$ K. Such a method was recently used to optimize the ReaxFF parameters of various $MgSO_4$ hydrates.[14] It is noteworthy that the different algorithms have different degrees of local search characteristics: SOPPI is a local method, PSO is a global search algorithm with some local search abilities, and SA and GA are global search methods with no local search ability.

The benchmark functions simulate the conditions typically encountered in the training of complex molecular models by having many degrees of freedom (30-dimensional functions); some present significant dependence among the variables, and some have many local minima. In its two-dimensional form, the Rastrigin function has several local minima. Finding the minimum of this function is a fairly difficult task because of its large search space and large number of local minima. In the two-dimensional Ackley function, there are many local minima on a relatively flat outer region, but a large conic "tube" is present at the center, descending toward the global minimum. The function poses a challenge for many optimization algorithms, particularly hill-climbing algorithms, which can get trapped in one of the many local minima. The Rosenbrock function is nonseparable, bimodal (for more than three dimensions) function that in two dimensions has a narrow, parabolic-shaped flat valley that is easy to find; however, convergence to the global minimum is rather difficult.[51] Lastly, the Schwefel function is multimodal with a weak global structure. Hence, there is no overall guiding slope toward the global minimum. The global minimum is located at the edge of the search space and poses a real challenge for many global optimization algorithms.

Figure 4 presents the optimization performance of the proposed RiPSOGM algorithm compared with the standard RiPSO algorithm, the well-known SOPPI algorithm, used for training ReaxFF reactive force fields,[1,5,9] and a standard SA algorithm as commonly used in the literature to optimize empirical force fields.[14,52] In all cases, the number of dimensions was set to 30, and the optimal scale factor γ was chosen following a preliminary analysis (Figure S5). In the case of the 30D Rastrigin function, SOPPI is practically trapped in a local minimum after a few function evaluations and obviously cannot escape. RiPSO does a much better job, but it struggles to find better solutions immediately after a few steps of optimization, and convergence toward better solutions is extremely slow. SA shows performance intermediate between those of SOPPI and RiPSO. Although it keeps improving its solution, the convergence is slow. On the other hand, the RiPSOGM algorithm constantly improves its position until it saturates. In the case of the 30D Ackley function, all of the algorithms but RiPSOGM converge prematurely because of the large number of local minima present. Surprisingly, SOPPI reaches a slightly better solution than the RiPSO and SA algorithms. The relatively poor overall performance of the SA

DOI: 10.1021/acs.jctc.7b01272
J. Chem. Theory Comput. 2018, 14, 3100−3112

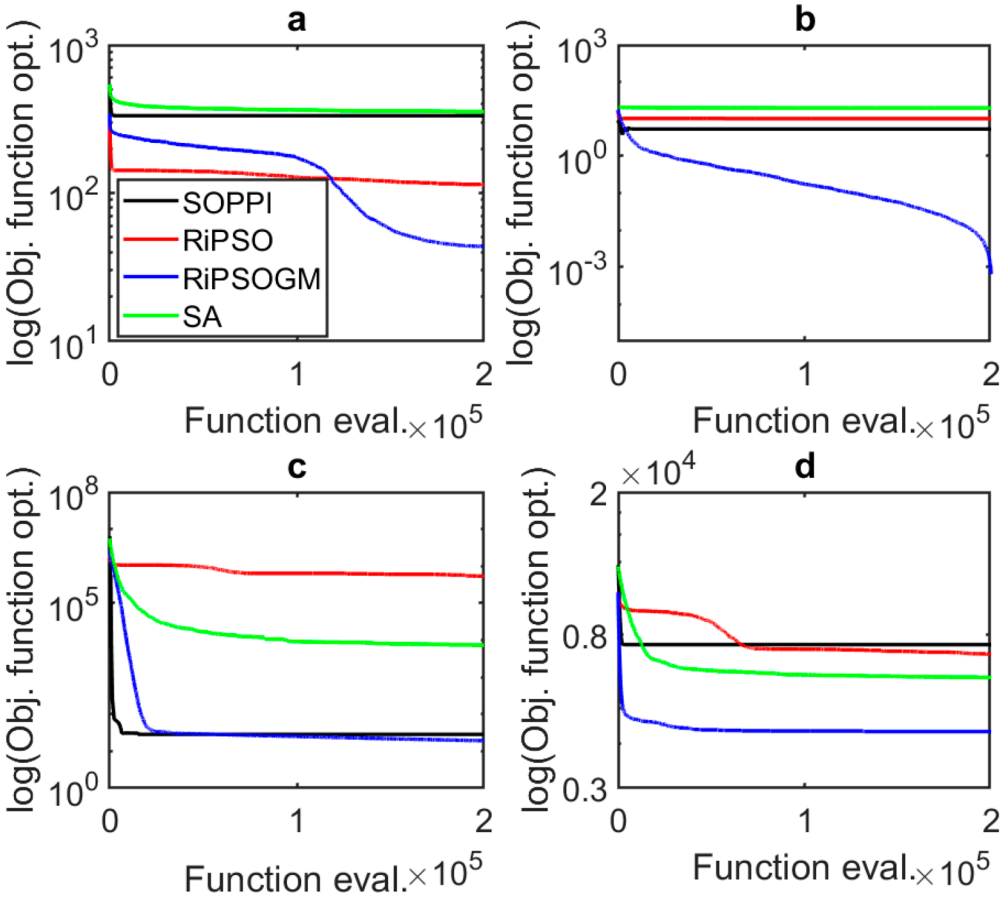


Figure 4. Global optimization performance comparison among the RiPSOGM (blue), RiPSO (red), SA (green), and SOPPI (black) algorithms on the 30-dimenstional (a) Rastrigin, (b) Ackley, (c) Rosenbrock, and (d) Schwefel functions. Search was performed with 20 agents in each case, for a total of 10^4 algorithm iterations, corresponding to 2×10^5 function evaluations. The SA algorithm followed standard Boltzmann annealing[50] with an initial temperature (T_0) of 600 K and an exponential cooling schedule, $T_{i+1} = 0.95 \cdot T_i$. The initial guess was randomly sampled from a uniform distribution within half the search domain boundaries. Each search was repeated 30 times to enhance the statistical significance of the results.

Table 1. Optimization Performance Results Obtained Using the RiPSOGM, RiPSO, SA and SOPPI Algorithms in 30 Independent Searches[a]

	RiPSOGM	RiPSO	SA	SOPPI
Rastrigin	43.58 ± 10.22	113.58 ± 25.67	193.87 ± 20.84	332.90 ± 45.16
Ackley	$(6.42 \pm 1.3) \cdot 10^{-4}$	10.25 ± 0.97	19.90 ± 0.32	5.35 ± 6.19
Rosenbrock	13.96 ± 1.93	133.13 ± 58.03	31.39 ± 4.22	28.42 ± 0.0
Schwefel	4313.11 ± 821.33	7058.02 ± 610.12	5879.32 ± 820.02	7525.50 ± 615.89

[a]The number of dimensions is $d = 30$, and the total number of function evaluations is 2×10^5. The global minimum point for each function is 0.0 at $[0]^D$, except for the Rosenbrock function, where it is at $[1]^D$. The search domains for the Rastrigin, Ackley, Rosenbrock, and Schwefel functions are $[-5.12, 5.12]^D$, $[-32.0, 32.0]^D$, $[-10.0, 10.0]^D$, and $[-500.0, 500.0]^D$, respectively. Initialization in all cases was performed across half of the search domain to avoid possible origin bias. The optimum values found are given as mean ± standard deviation.

algorithm compared with PSO variants is evident and is the result of poor global exploration of the landscape. Ensemble approaches to SA in which concurrent solutions with information sharing are simulated are a promising approach to increase the performance.[53,54] In contrast, RiPSOGM constantly improves its search performance and finally succeeds in finding a nearly optimum value. In the case of the bimodal, nonseparable 30D Rosenbrock function, RiPSO shows the worst performance, which is indicative of the lack of enough diversity in the search. SA significantly outperforms RiPSO, but the convergence slows down after approximately 5000 function evaluations. Both RiPSOGM and SOPPI achieve dramatically better solutions than RiPSO and SA. While SOPPI cannot improve its solution after a few iterations, RiPSOGM continues to slowly reach better solutions. The special difficulty of this function arises from the need to search in an increasingly small region to find the global optimum point, but this bears a high risk of premature convergence since with each iteration the swarm loses its momentum. Overall, RiPSOGM achieves the best performance on this function. Finally, in the case of the Schwefel function, SOPPI arrives at the worst solution, quite expectedly. RiPSO with its lack of diversity reaches a slightly better value. SA performs better than the previous methods but significantly worse than the RiPSOGM algorithm.

In terms of numerical cost, PSO-based methods are relatively expensive because a set of parallel searches is performed (in this

DOI: 10.1021/acs.jctc.7b01272
J. Chem. Theory Comput. 2018, 14, 3100−3112

case 20 search agents). Nevertheless, the PSO algorithm can be readily parallelized with a simple master−slave architecture. Thus, on the basis of the required number of function evaluations to reach an optimal threshold value, it remains highly competitive to the other methods, especially on low-dimensional functions where fewer agents can be effectively used. SOPPI is about 6-fold cheaper compared with the PSO algorithms used here, since only three function evaluations are required at each iteration. Although it shows good performance on unimodal and bimodal functions, SOPPI has a strong tendency to be trapped in local minima due to its local nature. The standard SA algorithm is a single-candidate search method and thus is the cheapest among the above methods. However, this comes at the expense of poor global search performance. The results of the above comparison are summarized in Table 1.

Lastly, we demonstrate that choosing agent step lengths from multivariate Lévy and Cauchy distributions (without a priori knowledge of promising search directions) as opposed to an isotropic Gaussian distribution is ineffective in escaping poor regions. As explained in section 2.2, it is expected that when the search direction is chosen isotropically (i.e., without an inherent bias toward the coordinate axes), the Lévy and Cauchy distributions should be as good as a Gaussian distribution. For this purpose, we repeated the RiPSOGM optimization sessions on the Rastrigin benchmark function. The Rastrigin function is characterized by many local minima, making it ideal to assess the ability of the algorithm to escape from local minima and to see whether heavy tails have any advantage. We found that the resulting performance was indistinguishable from the Gaussian case (see Figure S5), in good agreement with the results of other studies with different optimization algorithms.[45,55,56] One obvious difference between the three distributions is that numerical generation of Gaussian random numbers is much faster than that of Lévy random numbers,[57] making them highly preferable.

3.3. Application: Accurate Dispersion Parameters in the ReaxFF Reactive Force Field. To demonstrate the performance of the RiPSOGM algorithm in a "real-life" scenario, we choose to compare its performance to two popular methods for optimization of ReaxFF reactive force fields. The first method is the conventional SOPPI algorithm of van Duin,[9] and the second method is the advanced GA-based global optimization framework GARFfield.[12] Since parabolic interpolation is a very efficient local search method, we also use it to refine the final optimized parameters obtained using RiPSOGM to arrive at a hybrid algorithm (RiPSOGM +SOPPI).

In the following, we search for the optimal parameters for the low-gradient (*lg*) dispersion correction model of Liu and Goddard[58,59] used in ReaxFF-*lg* force fields[26] to properly describe organic crystals. We begin with a force field that was specifically developed to describe reactive events in energetic materials[60] augmented with the *lg* correction model. We refer the reader to previous references for a complete description of the full ReaxFF potential energy function.[1,26] The correction model includes a set of empirical parameters for each element that represent the coefficients of the multipole expansion.[61] The correction accounts for London dispersion energy, which is essential to accurately describe the crystal structures of organic solids, by introducing an attraction term with a force that smoothly goes to zero at valence distances:

$$E_{lg}(r_{ij}) = -\sum_{i<j}^{N} \frac{C_{ij}^{lg}}{r_{ij}^{6} + dR_{e,ij}^{6}} \quad (5)$$

In eq 5, r_{ij} is the distance between atoms i and j, $R_{e,ij}$ is the equilibrium vdW distance between atoms i and j, C_{ij}^{lg} is the dispersion energy correction parameter, and d is a scaling factor representing how fast the overlap (Pauli repulsion) effects dampen the London forces, which was kept fixed at $d = 1$ as in the study by Liu et al. The algorithms search for the optimal set of dispersion coefficients C_{ij}^{lg} for pairs of C, H, N, and O atoms and equilibrium vdW radii $R_{e,ij}$ for the elements. The search domain boundaries for the parameters are given in Tables 2 and 3.

Table 2. Ranges Assigned for Dispersion Coefficient Parameters C_{ij}^{lg} between Atomic Pairs[a]

parameter	address	search range	optimized value
C_{CC}^{lg}	2×1×33	0.001−10.0	3.588
C_{HH}^{lg}	2×2×33	0.001−10.0	6.988
C_{OO}^{lg}	2×3×33	0.001−2000.0	516.83
C_{NN}^{lg}	2×4×33	0.001−2000.0	89.697
C_{OH}^{lg}	4×2×7	0.001−10.0	9.736
C_{NH}^{lg}	4×3×7	0.001−2000.0	3.256
C_{CO}^{lg}	4×4×7	0.001−2000.0	145.833
C_{CN}^{lg}	4×5×7	0.001−2000.0	24.171
C_{NO}^{lg}	4×6×7	0.001−2000.0	187.943

[a] C_{ij}^{lg} values are given in units of kcal·mol^{-1}·Å^{-1}. The address of each parameter relates to its section ID, type ID, and parameter number in the ReaxFF force field file. The optimized values are the best values obtained out of three independent optimization sessions using RiPSOGM and local refinement with SOPPI.

Table 3. Ranges Assigned for Equilibrium vdW Radius Parameters $R_{e,ij}$ between Elements[a]

parameter	address	search range	optimized value
$R_{e,CC}$	2×1×34	1.2−2.0	1.944
$R_{e,HH}$	2×2×34	1.2−2.0	1.243
$R_{e,OO}$	2×3×34	1.2−2.0	1.844
$R_{e,NN}$	2×4×34	1.2−2.0	1.881

[a] $R_{e,ij}$ values are given in Å. The address of each parameter relates to its section ID, type ID, and parameter number in the ReaxFF force field file. The optimized values are the best values obtained out of three independent optimization sessions using RiPSOGM and local refinement with SOPPI.

In contrast to Liu et al.,[26] who used experimental crystal densities as the reference data, we use the results of accurate DFT-TS calculations performed with the PBE exchange−correlation functional[62] including the Tkatchenko−Scheffler dispersion correction[63] and ultrasoft pseudopotentials (see the Supporting Information for details) to compute the equations of state of several energetic and nonenergetic organic crystals. This combination of theory levels was shown to have excellent performance on organic materials with respect to computational demand[64] and preserves the philosophy of ReaxFF as being developed without experimental input. Specifically, using DFT-TS we calculated the equations of state of the hydrocarbon crystals benzene, biphenyl, hexamine, and anthracene and the energetic crystals 1,3,5,7-tetranitro-1,3,5,7-tetrazocane (HMX), 1,3,5-triamino-2,4,6-trinitrobenzene (TATB), pentaerythritol tetranitrate (PETN), 2,2-dinitroe-

DOI: 10.1021/acs.jctc.7b01272
J. Chem. Theory Comput. 2018, 14, 3100−3112

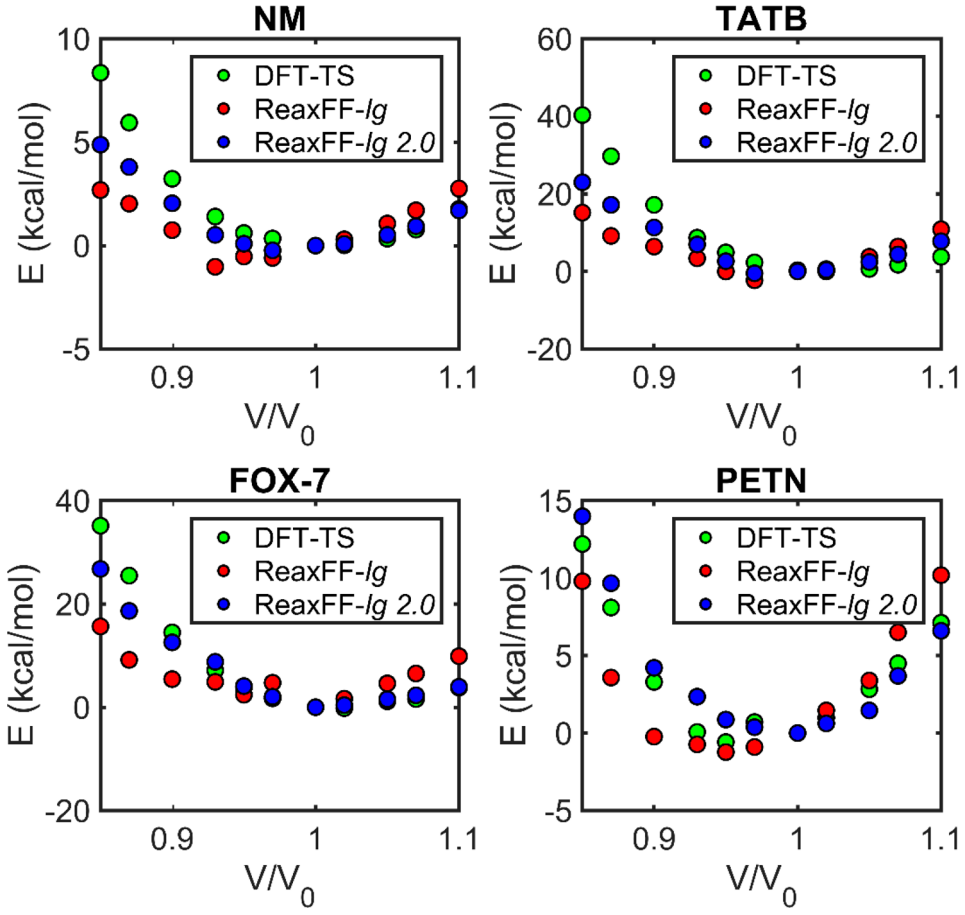


Figure 5. Calculated equations of state for energetic molecular crystals. The DFT-TS calculations (green) include the Tkatchenko−Scheffler dispersion correction. Force field values were calculated with ReaxFF-*l*g 2.0 obtained following a RiPSOGM optimization session (blue) and with the original ReaxFF-*l*g values of Liu et al.[26] (red).

thene-1,1-diamine (FOX-7), and nitromethane (NM). In addition, we augmented the training set with the full dissociation and expansion curves of various vdW-bonded dimers from the S66 database,[65] including $AcNH_2$, NH_3, CH_3NH_2, CH_3NH_2···pyridine, and HN_3 dimers. It is important to mention that since the *l*g model has little effect at the distances relevant to valence interactions,[26] the optimization can be conducted without refitting the already optimized terms that correspond to valence interactions in the ReaxFF force field. The objective function in the training of parameters is the following sum of squares (eq 6):

$$\text{ReaxFF error} = \sum_{i=1}^{n}\left[\frac{(x_{i,\text{DFT}} - x_{i,\text{ReaxFF}})}{\sigma_i}\right]^2 \quad (6)$$

where $x_{i,\text{DFT}}$ is the reference value (in kcal/mol), $x_{i,\text{ReaxFF}}$ is the computed value (in kcal/mol), and σ_i is the weight specified for each optimization target in the training set. Structures near their equilibrium values were given lower weights (0.1−0.5) while high-energy structures were given higher weights (1.0−2.0) to allow a better reproduction of important ambient conditions as opposed to high-energy phases. We used 20 agents for the optimization session with RiPSOGM without any tuning of algorithm parameters besides choosing an appropriate scale factor through preliminary simulations. Obviously, tuning of the inertia factor and the weights of the personal and social random components (ω, c_1, and c_2, respectively, in eq 1) could further improve its performance. The initial force field parameters were chosen randomly inside the prescribed ranges (Tables 2 and 3). In addition, in preliminary calculations we found that reducing the number of energy minimizations for each structure in the training set does not have any significant effect on the algorithm's search performance. Thus, we used only one or two energy minimizations on each structure. This highly accelerated the global search process without wasting time on unphysical high-energy structures that result from momentarily bad force field parameters.

Figure 5 presents the equations of state for NM, TATB, FOX-7, and PETN. We compare our results (named here as ReaxFF-*l*g 2.0) to the original ReaxFF-*l*g model of Liu et al.[26] and to our reference DFT-TS calculations. As can be seen from Figure 5, the newly optimized force field achieves very similar or better agreement with DFT predictions. For NM, better agreement is clearly noticeable for high compressions compared with the original force field. Nevertheless, a slight underestimation of the energy at high compression is preserved compared with the DFT data. This may be the result of inconsistencies in the description of hydrogen bonds between methyl ($-CH_3$) and nitro ($-NO_2$) groups in the crystal as the pressure increases. In the case of TATB, the resulting force field nicely matches the equation of state in expanded crystals but shows a similar underestimation at high compressions. Nevertheless, it performs slightly better than the original ReaxFF-*l*g force field. The performance of our force field in the

DOI: 10.1021/acs.jctc.7b01272
J. Chem. Theory Comput. 2018, 14, 3100−3112

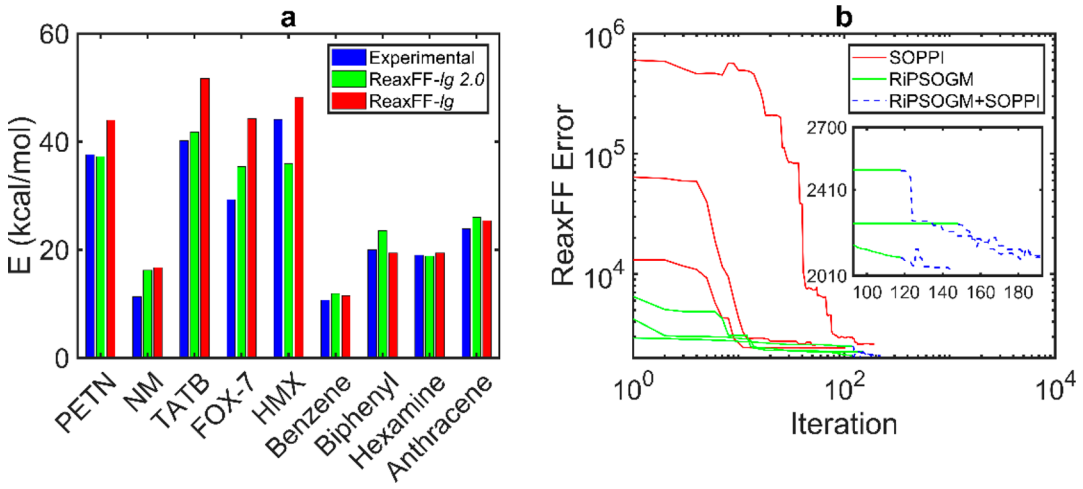


Figure 6. (a) Calculated heats of sublimation as predicted by ReaxFF-*lg* 2.0 obtained after an optimization session with RiPSOGM (green) and using the original ReaxFF-*lg* force field of Liu et al.[26] (red). Experimental heats of sublimation (blue) are also shown for PETN,[67] NM,[68] TATB,[69] FOX-7,[70] HMX,[71] benzene,[72] biphenyl,[72] hexamine,[73] and anthracene.[72] (b) Comparison between sequential one-parameter parabolic interpolation (SOPPI) (red), rotation-invariant particle swarm optimization with Gaussian mutation (RiPSOGM) (green), and local optimization of the final parameters achieved in each RiPSOGM session using SOPPI (i.e., the hybrid RiPSOGM+SOPPI framework) (dashed blue). Each curve is an independent optimization session beginning with random parameters. The inset magnifies the region near the last few iterations of the RiPSOGM algorithm.

case of FOX-7 is in significantly better agreement with the DFT-TS predictions. Whereas ReaxFF-*lg* exhibits a rather flat energy surface, ReaxFF-*lg* 2.0 preserves the shape of the energy curve at both low and high compression values. Similarly to FOX-7, the calculated equation of state for PETN is in much better agreement with DFT-TS compared with the original force field. A slight overestimation of the energy at high compression can be seen, but overall a satisfactory description is obtained.

A further comparison between the optimized force field obtained using the RiPSOGM scheme and that of Liu et al. can be made by calculating the heats of sublimation (Figure 6a). Very good agreement is obtained for the experimental heats of sublimation of the nonenergetic organic crystals benzene, biphenyl, hexamine, and anthracene. In the case of the energetic crystals PETN, NM, and TATB, overall good agreement is reached compared with the original ReaxFF-*lg* force field. In the case of FOX-7 an overestimation of the sublimation energy is obtained, whereas in the case of HMX an underestimation of the heat of sublimation is noticeable. However, in almost all cases ReaxFF-*lg* overestimates the heat of sublimation of both nonenergetic and energetic crystals.

We turn now to a comparison of our proposed algorithm with the SOPPI and GARFfield methods. First, to appreciate the dramatic enhancement of the global optimization process offered by RiPSOGM compared with local search methods, we compare three independent optimization sessions of the ReaxFF training set to the SOPPI algorithm. Figure 6b presents the results of the optimization starting with random force field parameters for the dispersion correction model. A striking difference can be noticed already at the first step of optimization: whereas SOPPI begins almost 3 orders of magnitude higher in error scale, RiPSOGM succeeds to sample the multidimensional landscape more thoroughly and to locate potentially better regions. This favorable behavior is the result of using a population of agents instead of a greedy single-individual mode in SOPPI or SA algorithms[66] and thus shows less sensitivity to the initial guess of parameters. Furthermore, on average RiPSOGM reaches lower values of error compared to SOPPI per given iteration. This behavior is manifested in the ability to direct agent movement to promising regions by exploiting information transfer in the swarm.

The final average ReaxFF errors reached after 100 iterations for the SOPPI and RiPSOGM algorithms are 2526.33 ± 84.29 and 2284 ± 204.27, respectively. The local nature of SOPPI does not permit it to converge to a global optimum; instead, it searches only for local solutions in the parameter space. We opted to see whether this could be turned into an advantage by enhancing the local search ability of our RiPSOGM algorithm. We submitted each of the three RiPSOGM runs to a subsequent local search process with SOPPI to see whether we could refine the final positions found by the swarm. Indeed, we succeeded in lowering the average error down to 2071 ± 19.63 in less than 100 additional local search steps, as can be seen from the inset in Figure 6b. This implies that a hybrid method composed of intermittent search patterns where local optimization is performed between successive PSO steps can potentially improve the convergence toward the optimum either by smoothing the search landscape for the global optimizer[74] in intermediate steps in a similar manner to the basin-hopping method[74] or by merely fine-tuning the convergence toward the global optimum. Thus, we have added an option to include a SOPPI local search session during or after a RiPSOGM run.

Lastly, we compare our method to the recently introduced[75] GA-based optimization framework for reactive force fields, GARFfield.[12] It uses an evolutionary scheme to seed a population of solutions (chromosomes) made up of random parameters (genes). The fitness of each individual chromosome is evaluated and used to decide whether it should be replaced or used to evolve new and better offspring. The replacement strategy used is the efficient steady-state replacement method (SSGA), which replaces 10% of the population.[12,76] The evolution of new offspring in GA is achieved by using tournament selection, two-point crossover (at rate of 0.85), and mutation (at rate of 0.1) operators. In addition, GARFfield

DOI: 10.1021/acs.jctc.7b01272
J. Chem. Theory Comput. 2018, 14, 3100−3112

includes two options to potentially improve the convergence. The first option that was used is a hill-climbing move with a frequency of 10 iterations. This enables upward moves in the fitness landscape after no significant improvement is seen and helps the search escape local minimum traps. The second option is switching from the evolutionary GA steps to a Polak−Riebiere version of the conjugate gradient (CG) minimizer. We switched the final 100 iterations (iterations 400 to 500) to be pure CG. During evolution, the population fitness should improve toward a higher-quality set of force field parameters.

Figure 7 presents the optimization sessions conducted by the two population-based methods on a compact set of previous

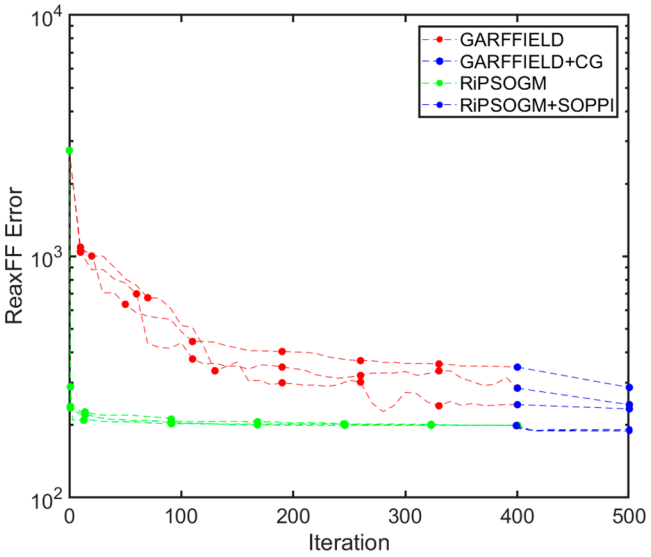


Figure 7. Comparison of global optimization performances of the genetic algorithm code GARFfield[12] with and without a conjugate gradient (CG) minimization, the RiPSOGM method, and the RiPSOGM+SOPPI hybrid framework. GARFfield uses tournament selection, a two-point crossover operator with rate of 0.85, and a mutation operator with a rate of 0.1. Also, hill climbing is performed every 10 iterations. The population size in both methods was 20 agents. For each algorithm, three independent runs were performed. The *y* axis is a log scale, and the dots are guides to the eye.

training data. The training set contains full dissociation and compression curves for vdW-bonded dimers in the S66 database,[65] including $AcNH_2$, NH_3, CH_3NH_2, CH_3NH_2··· pyridine, and also HN_3 dimers. In both cases, we use exactly the same initial force field[26] to evaluate the initial fitness. This results in an initial fitness of 2.75×10^3. After roughly 100 iterations, GARFfield improves the average fitness by a factor of about 2.5, and then a slower improvement can be observed until the final results are obtained. In comparison, RiPSOGM using 20 members succeeds to dramatically lower the average fitness in the first few (about 20) iterations and then shows very slow improvement and stays almost stagnant throughout the remaining search process. This behavior stems from the randomization process, where each of the optimization parameters gets a completely independent random value within its domain boundaries. Then the member with the globally best fitness is chosen as the new force field for the next step in the search process. In contrast, GARFfield generates a population of 20 genes with mutation and crossover operations from the initial force field. Turning on local search (CG) after 400 iterations for the additional 100 iterations in GARFfield leads to further lowering of the fitness to 254.2. The SOPPI in RiPSOGM offers a similar improvement and arrives at the final fitness value of 191.8. Thus, the better performance of RiPSOGM over GARFfield is afforded by (a) the randomization step already at the initial stage and the selection of the globally best member as the new force field for the search process, (b) the efficient information-sharing mechanism in the swarm, and (c) the local-search-like characteristics in each swarm member. Overall, the RiPSOGM framework, which is rotation-invariant, shows better performance in terms of solution quality and computational cost and is less divergent throughout the optimization path.

4. CONCLUSIONS AND OUTLOOK

An improved algorithm for global optimization of nonlinear, multimodal functions of continuous variables is proposed. The algorithm is based on a rotation-invariant formulation of the well-known particle swarm optimization (PSO) algorithm. Our method departs from standard RiPSO by introducing isotropic mutation operators that act to diversify the swarm. The operation replaces poorly performing search agents with a Gaussian mutation of the globally best swarm members. It is termed here rotation-invariant particle swarm optimization with Gaussian mutation (RiPSOGM).

The mutation acts to preserve good-performing members but eliminates members in poor regions that cannot improve their solutions, which helps to diversify the swarm trajectories. The use of isotropic Gaussian distributions to generate random distances from the current global best member preserves the rotation invariance of the algorithm. This is highly beneficial since rotation-invariant algorithms preserve their performance on nonseparable functions, i.e., in cases where the optimization parameters are correlated.[36]

RiPSOGM presents superior performance in comparison with several other optimization algorithms, including standard RiPSO, simulated annealing (SA), and sequential one-parameter parabolic interpolation (SOPPI) on several high-dimensional, nonlinear benchmark functions. In addition, on a ReaxFF model training set it achieves slightly better solutions in fewer iterations in comparison with the GA-based package GARFfield.

Following the application of RiPSOGM to optimize the low-gradient (*lg*) dispersion correction model, we accurately described the equations of state and heats of formation of several families of organic materials, including energetic crystals.

The new algorithm is implemented in a stand-alone C++ package with a straightforward interface to the serial version of the ReaxFF molecular dynamics package. The package allows efficient training of reactive force fields in a highly automated manner. The package is freely available upon request from the authors. Further improvements of the framework will include parallelization using the MPI architecture, allocating each swarm member to a CPU core. Such efforts have already shown excellent parallel performance[77−79] and are expected to dramatically reduce the computational cost of training reactive force fields with large parameter spaces.

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge on the ACS Publications website at DOI: 10.1021/acs.jctc.7b01272.

DOI: 10.1021/acs.jctc.7b01272
J. Chem. Theory Comput. 2018, 14, 3100−3112

Description of benchmark functions; influence of scale and mutation frequency parameters on optimization; comparison of Gaussian, Lévy, and Cauchy mutation on optimization performance; and DFT calculation details (PDF)

AUTHOR INFORMATION

Corresponding Author

*E-mail: furman.david@mail.huji.ac.il.

ORCID

David Furman: 0000-0002-6645-4211

Yehuda Zeiri: 0000-0002-5488-5823

Funding

This material is based upon work partially supported by the U.S. Department of Homeland Security, Science and Technology Directorate, Office of University Programs, under Grant 2013-ST-061-ED0001.

Notes

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Department of Homeland Security.

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

D.F. thanks Adri van Duin for providing his serial ReaxFF code, Sergey Zybin for providing the low-gradient implementation of ReaxFF, and Andres Jaramillo-Botero for helpful discussions on the use of GARFfield.

REFERENCES

(1) 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.

(2) 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.

(3) Furman, D.; Kosloff, R.; Zeiri, Y. Mechanism of Intact Adsorbed Molecules Ejection Using High Intensity Laser Pulses. *J. Phys. Chem. C* **2016**, *120*, 11306−11312.

(4) Oxley, J. C.; Furman, D.; Brown, A. C.; Dubnikova, F.; Smith, J. L.; Kosloff, R.; Zeiri, Y. Thermal Decomposition of Erythritol Tetranitrate: A Joint Experimental and Computational Study. *J. Phys. Chem. C* **2017**, *121*, 16145−16157.

(5) Furman, D.; Dubnikova, F.; van Duin, A. C. T.; Zeiri, Y.; Kosloff, R. Reactive Force Field for Liquid Hydrazoic Acid with Applications to Detonation Chemistry. *J. Phys. Chem. C* **2016**, *120*, 4744−4752.

(6) Furman, D.; Kosloff, R.; Zeiri, Y. Effects of Nanoscale Heterogeneities on the Reactivity of Shocked Erythritol Tetranitrate. *J. Phys. Chem. C* **2016**, *120*, 28886−28893.

(7) Hartke, B.; Grimme, S. Reactive Force Fields Made Simple. *Phys. Chem. Chem. Phys.* **2015**, *17*, 16715−16718.

(8) Martinez, J. A.; Yilmaz, D. E.; Liang, T.; Sinnott, S. B.; Phillpot, S. R. Fitting Empirical Potentials: Challenges and Methodologies. *Curr. Opin. Solid State Mater. Sci.* **2013**, *17*, 263−270.

(9) 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.

(10) 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.

(11) Dittner, M.; Müller, J.; Aktulga, H. M.; Hartke, B. Efficient Global Optimization of Reactive Force-Field Parameters. *J. Comput. Chem.* **2015**, *36*, 1550−1561.

(12) 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.

(13) 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.

(14) Iype, E.; Hütter, M.; Jansen, A. P. J.; Nedea, S. V.; Rindt, C. C. M. Parameterization of a Reactive Force Field Using a Monte Carlo Algorithm. *J. Comput. Chem.* **2013**, *34*, 1143−1154.

(15) 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.

(16) Deetz, J. D.; Faller, R. Parallel Optimization of a Reactive Force Field for Polycondensation of Alkoxysilanes. *J. Phys. Chem. B* **2014**, *118*, 10966−10978.

(17) Martinez, J. A.; Chernatynskiy, A.; Yilmaz, D. E.; Liang, T.; Sinnott, S. B.; Phillpot, S. R. Potential Optimization Software for Materials (Posmat). *Comput. Phys. Commun.* **2016**, *203*, 201−211.

(18) Eberhart, R.; Kennedy, J. A New Optimizer Using Particle Swarm Theory. In *Proceedings of the Sixth International Symposium on Micro Machine and Human Science, 1995 (MHS '95)*; IEEE: Piscataway, NJ, 1995; pp 39−43.

(19) Yu, X.; Li, L.; Xu, X.-W.; Tang, C.-C. Prediction of Two-Dimensional Boron Sheets by Particle Swarm Optimization Algorithm. *J. Phys. Chem. C* **2012**, *116*, 20075−20079.

(20) Wang, Y.; Lv, J.; Zhu, L.; Ma, Y. Crystal Structure Prediction via Particle-Swarm Optimization. *Phys. Rev. B: Condens. Matter Mater. Phys.* **2010**, *82*, 094116.

(21) Paradiso, S. P.; Delaney, K. T.; Fredrickson, G. H. Swarm Intelligence Platform for Multiblock Polymer Inverse Formulation Design. *ACS Macro Lett.* **2016**, *5*, 972−976.

(22) Khadilkar, M. R.; Paradiso, S.; Delaney, K. T.; Fredrickson, G. H. Inverse Design of Bulk Morphologies in Multiblock Polymers Using Particle Swarm Optimization. *Macromolecules* **2017**, *50*, 6702−6709.

(23) Chou, C.-P.; Nishimura, Y.; Fan, C.-C.; Mazur, G.; Irle, S.; Witek, H. A. Automatized Parameterization of Dftb Using Particle Swarm Optimization. *J. Chem. Theory Comput.* **2016**, *12*, 53−64.

(24) Tang, L.; Yan, P. Particle Swarm Optimization Algorithm for a Batching Problem in the Process Industry. *Ind. Eng. Chem. Res.* **2009**, *48*, 9186−9194.

(25) Prasad, J.; Souradeep, T. Cosmological Parameter Estimation Using Particle Swarm Optimization. *Phys. Rev. D* **2012**, *85*, 123008.

(26) Liu, L.; Liu, Y.; Zybin, S. V.; Sun, H.; Goddard, W. A. ReaxFF-Lg: 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.

(27) Bratton, D.; Kennedy, J. Defining a Standard for Particle Swarm Optimization. In *Proceedings of the 2007 IEEE Swarm Intelligence Symposium (SIS2007)*; IEEE: Piscataway, NJ, 2007; pp 120−127.

(28) Ab Wahab, M. N.; Nefti-Meziani, S.; Atyabi, A. A Comprehensive Review of Swarm Optimization Algorithms. *PLoS One* **2015**, *10*, e0122827.

(29) Wilke, D. N.; Kok, S.; Groenwold, A. A. Comparison of Linear and Classical Velocity Update Rules in Particle Swarm Optimization: Notes on Scale and Frame Invariance. *Int. J. Numer. Meth. Eng.* **2007**, *70*, 985−1008.

(30) Spears, W. M.; Green, D. T.; Spears, D. F. Biases in Particle Swarm Optimization. *Int. J. Swarm Intell. Res.* **2010**, *1*, 34−57.

DOI: 10.1021/acs.jctc.7b01272
J. Chem. Theory Comput. 2018, 14, 3100−3112

(31) Wilke, D. N.; Kok, S.; Groenwold, A. A. Comparison of Linear and Classical Velocity Update Rules in Particle Swarm Optimization: Notes on Diversity. *Int. J. Numer. Meth. Eng.* **2007**, *70*, 962−984.
(32) Bonyadi, M. R.; Michalewicz, Z. SPSO 2011: Analysis of Stability; Local Convergence; and Rotation Sensitivity. In *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*; ACM: New York, 2014; pp 9−16.
(33) Zambrano-Bigiarini, M.; Clerc, M.; Rojas, R. In Standard Particle Swarm Optimisation 2011 at CEC-2013: A Baseline for Future PSO Improvements. In *Proceedings of the 2013 Congress on Evolutionary Computation (CEC)*; IEEE: Piscataway, NJ, 2013; pp 2337−2344.
(34) Ras, M. N.; Wilke, D. N.; Groenwold, A. A.; Kok, S. On Rotationally Invariant Continuous-Parameter Genetic Algorithms. *Adv. Eng. Softw.* **2014**, *78*, 52−59.
(35) Salomon, R. Re-Evaluating Genetic Algorithm Performance under Coordinate Rotation of Benchmark Functions. A Survey of Some Theoretical and Practical Aspects of Genetic Algorithms. *BioSystems* **1996**, *39*, 263−278.
(36) Hansen, N.; Ros, R.; Mauny, N.; Schoenauer, M.; Auger, A. Impacts of Invariance in Search: When Cma-Es and Pso Face Ill-Conditioned and Non-Separable Problems. *Appl. Soft. Comput.* **2011**, *11*, 5755−5769.
(37) Hicks, J. S.; Wheeling, R. F. An Efficient Method for Generating Uniformly Distributed Points on the Surface of an *N*-Dimensional Sphere. *Commun. ACM* **1959**, *2*, 17−19.
(38) Yang, X.-S., Firefly Algorithm, Lévy Flights and Global Optimization. In *Research and Development in Intelligent Systems XXVI*; Bramer, M., Ellis, R., Petridis, M., Eds.; Springer: London, 2010; pp 209−218.
(39) Arora, S.; Singh, S. Butterfly Algorithm with Lévy Flights for Global Optimization, In *Proceedings of the 2015 International Conference on Signal Processing, Computing and Control (ISPCC)*; IEEE: Piscataway, NJ, 2015; pp 220−224.
(40) Yang, X.-S.; Deb, S. Cuckoo Search via Lévy Flights. In *Proceedings of the 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*; IEEE: Piscataway, NJ, 2009; pp 210−214.
(41) Yan, B.; Zhao, Z.; Zhou, Y.; Yuan, W.; Li, J.; Wu, J.; Cheng, D. A Particle Swarm Optimization Algorithm with Random Learning Mechanism and Lévy Flight for Optimization of Atomic Clusters. *Comput. Phys. Commun.* **2017**, *219*, 79−86.
(42) Haklı, H.; Uğuz, H. A Novel Particle Swarm Optimization Algorithm with Lévy Flight. *Appl. Soft. Comput.* **2014**, *23*, 333−345.
(43) Jensi, R.; Jiji, G. W. An Enhanced Particle Swarm Optimization with Lévy Flight for Global Optimization. *Appl. Soft. Comput.* **2016**, *43*, 248−261.
(44) Viswanathan, G. M.; Buldyrev, S. V.; Havlin, S.; da Luz, M. G. E.; Raposo, E. P.; Stanley, H. E. Optimizing the Success of Random Searches. *Nature* **1999**, *401*, 911−914.
(45) Hansen, N.; Gemperle, F.; Auger, A.; Koumoutsakos, P. When Do Heavy-Tail Distributions Help? In *Parallel Problem Solving from Nature—PPSN IX*; Runarsson, T. P., Beyer, H. G., Burke, E., Merelo-Guervós, J. J., Whitley, L. D., Yao, X., Eds.; Lecture Notes in Computer Science, Vol. *4193*; Springer: Berlin, 2006; pp 62−71.
(46) Shi, Y.; Eberhart, R. C. Empirical Study of Particle Swarm Optimization. In *Proceedings of the 1999 Congress on Evolutionary Computation (CEC 99)*; IEEE: Piscataway, NJ, 1999; pp 1945−1950.
(47) Finck, S.; Hansen, N.; Ros, R.; Auger, A. Documentation for COCO (COmparing Continuous Optimisers). http://coco.lri.fr/COCOdoc/ (accessed Dec 19, 2017).
(48) Jamil, M.; Yang, X.-S. A Literature Survey of Benchmark Functions for Global Optimisation Problems. *Int. J. Math. Model Num. Opt.* **2013**, *4*, 150−194.
(49) Kirkpatrick, S.; Gelatt, C. D.; Vecchi, M. P. Optimization by Simulated Annealing. *Science* **1983**, *220*, 671−680.
(50) Ingber, L. Simulated Annealing: Practice Versus Theory. *Math. Comput. Model.* **1993**, *18*, 29−57.
(51) Lu, Y. Z.; Chen, Y. W.; Chen, M. R.; Chen, P.; Zeng, G. Q. *Extremal Optimization: Fundamentals, Algorithms, and Applications*; CRC Press: Boca Raton, FL, 2016.
(52) SAKAE, Y.; OKAMOTO, Y. Protein Force-Field Parameters Optimized with the Protein Data Bank Ii: Comparisons of Force Fields by Folding Simulations of Short Peptides. *J. Theor. Comput. Chem.* **2004**, *03*, 359−378.
(53) Salamon, P.; Sibani, P.; Frost, R. *Facts, Conjectures, and Improvements for Simulated Annealing*; SIAM Monographs on Mathematical Modeling and Computation; Society for Industrial and Applied Mathematics: Philadelphia, PA, 1987.
(54) Ruppeiner, G.; Pedersen, J. M.; Salamon, P. Ensemble Approach to Simulated Annealing. *J. Phys. I* **1991**, *1*, 455−470.
(55) Rudolph, G. Local Convergence Rates of Simple Evolutionary Algorithms with Cauchy Mutations. *IEEE Transactions on Evolutionary Computation* **1997**, *1*, 249−258.
(56) Obuchowicz, A. Multidimensional Mutations in Evolutionary Algorithms Based on Real-Valued Representation. *Int. J. Syst. Sci.* **2003**, *34*, 469−483.
(57) Leccardi, M. Comparison of Three Algorithms for Lévy Noise Generation. Presented at the Fifth EUROMECH Nonlinear Dynamics Conference (ENOC'05), Eindhoven, The Netherlands, 2005.
(58) Kim, H.; Choi, J.-M.; Goddard, W. A., III. Universal Correction of Density Functional Theory To Include London Dispersion (up to Lr, Element 103). *J. Phys. Chem. Lett.* **2012**, *3*, 360−363.
(59) Liu, Y.; Goddard, W. A., III. First-Principles-Based Dispersion Augmented Density Functional Theory: From Molecules to Crystals. *J. Phys. Chem. Lett.* **2010**, *1*, 2550−2555.
(60) Zhang, L.; Zybin, S. V.; van Duin, A. C. T.; Dasgupta, S.; Goddard, W. A.; Kober, E. M. Carbon Cluster Formation during Thermal Decomposition of Octahydro-1,3,5,7-tetranitro-1,3,5,7-tetrazocine and 1,3,5-Triamino-2,4,6-trinitrobenzene High Explosives from ReaxFF Reactive Molecular Dynamics Simulations. *J. Phys. Chem. A* **2009**, *113*, 10619−10640.
(61) London, F. The General Theory of Molecular Forces. *Trans. Faraday Soc.* **1937**, *33*, 8b−26.
(62) Perdew, J. P.; Burke, K.; Ernzerhof, M. Generalized Gradient Approximation Made Simple. *Phys. Rev. Lett.* **1996**, *77*, 3865−3868.
(63) Tkatchenko, A.; Scheffler, M. Accurate Molecular van der Waals Interactions from Ground-State Electron Density and Free-Atom Reference Data. *Phys. Rev. Lett.* **2009**, *102*, 073005.
(64) Kronik, L.; Tkatchenko, A. Understanding Molecular Crystals with Dispersion-Inclusive Density Functional Theory: Pairwise Corrections and Beyond. *Acc. Chem. Res.* **2014**, *47*, 3208−3216.
(65) Řezáč, J.; Riley, K. E.; Hobza, P. S66: A Well-Balanced Database of Benchmark Interaction Energies Relevant to Biomolecular Structures. *J. Chem. Theory Comput.* **2011**, *7*, 2427−2438.
(66) Grosso, A.; Locatelli, M.; Schoen, F. An Experimental Analysis of a Population Based Approach for Global Optimization. *Comput. Optim. Appl.* **2007**, *38*, 351−370.
(67) Lau, K. H.; Hildenbrand, D. L.; Crouch-Baker, S.; Sanjurjo, A. Sublimation Pressure and Vapor Molecular Weight of Pentaerythritol Tetranitrate. *J. Chem. Eng. Data* **2004**, *49*, 544−546.
(68) Sorescu, D. C.; Rice, B. M.; Thompson, D. L. Theoretical Studies of Solid Nitromethane. *J. Phys. Chem. B* **2000**, *104*, 8406−8419.
(69) Rosen, J. M.; Dickinson, C. Vapor Pressures and Heats of Sublimation of Some High-Melting Organic Explosives. *J. Chem. Eng. Data* **1969**, *14*, 120−124.
(70) Muravyev, N. V.; Pivkina, A. N.; Kiselev, V. G. Comment on "Studies on Thermodynamic Properties of Fox-7 and Its Five Closed-Loop Derivatives. *J. Chem. Eng. Data* **2017**, *62*, 575−576.
(71) Lyman, J. L.; Liau, Y.-C.; Brand, H. V. Thermochemical Functions for Gas-Phase, 1,3,5,7-Tetranitro-1,3,5,7-Tetraazacyclooctane (Hmx), Its Condensed Phases, and Its Larger Reaction Products. *Combust. Flame* **2002**, *130*, 185−203.
(72) Roux, M. V.; Temprado, M.; Chickos, J. S.; Nagano, Y. Critically Evaluated Thermochemical Properties of Polycyclic Aromatic Hydrocarbons. *J. Phys. Chem. Ref. Data* **2008**, *37*, 1855−1996.
(73) Verevkin, S. P. Relationships among Strain Energies of Mono- and Poly-Cyclic Cyclohexanoid Molecules and Strain of Their Component Rings. *J. Chem. Thermodyn.* **2002**, *34*, 263−275.

DOI: 10.1021/acs.jctc.7b01272
J. Chem. Theory Comput. 2018, 14, 3100−3112

(74) Wales, D. J.; Doye, J. P. K. Global Optimization by Basin-Hopping and the Lowest Energy Structures of Lennard-Jones Clusters Containing up to 110 Atoms. *J. Phys. Chem. A* **1997**, *101*, 5111−5116.
(75) Levine, D. *Users Guide to the Pgapack Parallel Genetic Algorithm Library*; Report ANL-95/18; Argonne National Laboratory: Argonne, IL, 1996.
(76) Whitley, D. A Genetic Algorithm Tutorial. *Stat. Comput.* **1994**, *4*, 65−85.
(77) Farmahini-Farahani, A.; Vakili, S.; Fakhraie, S. M.; Safari, S.; Lucas, C. Parallel Scalable Hardware Implementation of Asynchronous Discrete Particle Swarm Optimization. *Eng. Appl. Artif. Intel.* **2010**, *23*, 177−187.
(78) Schutte, J. F.; Reinbolt, J. A.; Fregly, B. J.; Haftka, R. T.; George, A. D. Parallel Global Optimization with the Particle Swarm Algorithm. *Int. J. Numer. Meth. Eng.* **2004**, *61*, 2296−2315.
(79) Koh, B.-I.; George, A. D.; Haftka, R. T.; Fregly, B. J. Parallel Asynchronous Particle Swarm Optimization. *Int. J. Numer. Meth. Eng.* **2006**, *67*, 578−595.

DOI: 10.1021/acs.jctc.7b01272
J. Chem. Theory Comput. 2018, 14, 3100−3112