Accepted Manuscript

SGO: A fast engine for ab initio atomic structure global optimization by differential evolution

Zhanghui Chen, Weile Jia, Xiangwei Jiang, Shu-Shen Li, Lin-Wang Wang

PII: S0010-4655(17)30135-2
DOI: http://dx.doi.org/10.1016/j.cpc.2017.05.005
Reference: COMPHY 6210

To appear in: Computer Physics Communications

Received date: 7 August 2016
Revised date: 25 April 2017
Accepted date: 7 May 2017

Please cite this article as: Z. Chen, et al., SGO: A fast engine for ab initio atomic structure global optimization by differential evolution, Computer Physics Communications (2017), http://dx.doi.org/10.1016/j.cpc.2017.05.005

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.
SGO: a fast engine for ab initio atomic structure global optimization by differential evolution

Zhanghui Chen,1,2,* Weile Jia,1 Xiangwei Jiang,2 Shu-Shen Li,2 and Lin-Wang Wang1,†

1Materials Sciences Division, Lawrence Berkeley National Laboratory,
One Cyclotron Road, Mail Stop 50F,
Berkeley, California 94720, United States
2State Key Laboratory of Superlattices and Microstructures,
Institute of Semiconductors, Chinese Academy of Sciences,
P.O. Box 912, Beijing 100083, People’s Republic of China

(Dated: April 25, 2017)

Abstract

As the high throughput calculations and material genome approaches become more and more popular in material science, the search for optimal ways to predict atomic global minimum structure is a high research priority. This paper presents a fast method for global search of atomic structures at ab initio level. The structures global optimization (SGO) engine consists of a high-efficiency differential evolution algorithm, accelerated local relaxation methods and a plane-wave density functional theory code running on GPU machines. The purpose is to show what can be achieved by combining the superior algorithms at the different levels of the searching scheme. SGO can search the global-minimum configurations of crystals, two-dimensional materials and quantum clusters without prior symmetry restriction in a relatively short time (half or several hours for systems with less than 25 atoms), thus making such a task a routine calculation. Comparisons with other existing methods such as minima hopping and genetic algorithm are provided. One motivation of our study is to investigate the properties of magnetic systems in different phases. The SGO engine is capable of surveying the local minima surrounding the global minimum, which provides the information for the overall energy landscape of a given system. Using this capability we have found several new configurations for testing systems, explored their energy landscape, and demonstrated that the magnetic moment of metal clusters fluctuates strongly in different local minima.
I. INTRODUCTIONS

Global search of atomic structures\textsuperscript{1-6} is very useful in material science, especially for discovering new materials in various kinds of applications such as magnetic storage\textsuperscript{7,8}, energy-storage\textsuperscript{9}, superconductivity\textsuperscript{10,11}, photocatalytic water splitting\textsuperscript{12,13} and high-pressure materials\textsuperscript{14-17}. Not only the ground state structure, some small-energy local minima close to the ground state can also play important roles. For example, the magnetic moment of metal clusters can depend sensitively on the structures. Understanding the relationship between magnetic moment and structure could help us to use such nanoclusters as information storage medium. Due to the complexity, the theoretical prediction of atomic structures and the energy landscape surrounding the ground state can be extremely difficult\textsuperscript{1,18}. The conventional global search methods could need tens of thousands of local relaxations of atomic structure to locate the global minimum\textsuperscript{1,2,19}. Since each local relaxation is performed by density functional theory (DFT) calculations or other ab initio methods, the whole search procedure can be extremely time consuming\textsuperscript{18,20}. A simple search job could cost several days or even weeks. Therefore, it is important to accelerate the search process by adopting highly efficient algorithms, for both the global searching scheme and the local atomic relaxation.

There are some groups that developed efficient global search engine for atomic structures. For example, Oganov et al. developed a package USPEX (Universal Structure Predictor: Evolutionary Xtallography) for crystal structure prediction based on genetic algorithm\textsuperscript{2,21}. Pickard et al. developed ab initio random structure searching (AIRSS) that is capable of discovering the structures of solids, point defects, surfaces, and clusters\textsuperscript{22,23}. Ma et al. set up a crystal searching method through particle-swarm optimization and space group symmetry, obtaining great success in high-pressure materials\textsuperscript{16,17,24}. Catlow et al. used genetic algorithm to generate plausible crystal structures from the knowledge of only the unit cell dimensions and constituent elements\textsuperscript{1,25}. Zunger and Zhang et al. used genetic algorithm and global space-group optimization for inverse design of materials\textsuperscript{26-28}. K. M. Ho and C. Z. Wang developed an adaptive genetic algorithm to search crystal structures and grain boundary\textsuperscript{29,30}. Besides, basin/minima hopping algorithms have been adopted by many groups for atomic clusters searches\textsuperscript{19,31,32}. Our own PDECO (parallel differential evolution cluster optimization) procedure also showed excellent performance in the search of various clusters\textsuperscript{33}. These activities in this area indicate the importance of high-efficiency search
algorithms for crystals, clusters and other atomic structures for emergent applications.

In this paper, we present an fast engine for structures global optimization (SGO)\textsuperscript{34} with acceleration on all levels of the algorithm. The goal is to demonstrate that by combining all the top of line algorithms, the structure search for mid-sized systems (with less than 25 atoms) can be made easy and practical, thus such structure search can be a routine part of our research activities and can play important roles in material genome projects. Our algorithm at the level of global search uses parallel differential evolution (DE) that has been demonstrated to be of high efficiency in searching structures\textsuperscript{33,35,36}. At the middle level, the local relaxation of atomic structures employs curved-line-search (CLS) method\textsuperscript{37} and preconditioned conjugate gradient (PCG) algorithm\textsuperscript{38}, both of which are based on force fitting and have been demonstrated to be 2-6 times faster than conventional conjugate gradient (CG) algorithm\textsuperscript{39,40}. At the level of the energy and force evaluation, we use a DFT plane-wave code (PWmat) run on GPU machines\textsuperscript{41–44}. The integration of three-level acceleration enables a fast global search of atomic structures. Symmetry restriction that is widely used for acceleration in other packages\textsuperscript{2,16,24} is not required here although could be added later if needed. Tests for crystals, two-dimensional (2D) materials and quantum clusters demonstrate the excellent performance of SGO engine. Comparisons with existing packages also show its high efficiency. Moreover, this engine can be used to survey the energy landscape of a given system, which is relatively scarce in previous studies but could be very useful for the understanding of the finite temperature effect or dynamic behaviours of the studied systems.

The paper is organized as follows. Section 2 introduces the search engine with the three-level acceleration algorithms. Section 3 presents the tests on crystals, 2D materials and quantum clusters as well as the searches of energy landscape. The conclusion remarks are drawn in Section 4.

II. METHODS

A. Overview of the SGO engine

Our search engine SGO consists of three levels, as shown in Figure 1. At the top level, we employ an improved parallel DE algorithm for global search. At the middle level, CLS
and PCG algorithms are employed for accelerated local relaxation of each atomic structure into its nearest local minimum. At the bottom level, the energy and atomic force of each structure are calculated self-consistently by the DFT plane-wave code (PWmat) run on GPU machines (Mstation).

![Flow chart of the SGO engine. It consists of three modules: global search (left), local relaxation (middle) and energy/force evaluation (right).](image)

**B. Fast global search**

The global search is implemented by a high-efficiency parallel DE algorithm. DE\textsuperscript{35,36,45–49} belongs to the family of evolutionary algorithms (EAs)\textsuperscript{50–52}. It employs multiple individuals’ cooperation and evolutionary operators to find the global minimum of the objective function. Compared to most other EAs, DE is much simpler and straightforward to implement. The main search engine requires less than 30 lines of C or FORTRAN code\textsuperscript{36,49}. Due to robust control parameters and low space complexity\textsuperscript{36,46,49}, DE is highly efficient. Since it is proposed in 1995\textsuperscript{35}, DE continues to rank top in various competitions and exhibits superior...
performance than all the other EAs in most of optimization problems.\textsuperscript{35,36,45–49}

Assuming an optimization problem is formulated as: $\min_{X} f(X)$, where $X$ is a vector of $1 \times n$ variables (e.g., atomic positions in our case), DE works with a population of $N_p$ candidate solutions, i.e., $X_{i,G}$, $i = 1, \ldots, N_p$, where $i$ is the index of the individual and $G$ is the generation in the evolution. Random initialization, mutation, crossover and selection operators are then employed to evolve the population. These operators are similar as other EAs except the mutation procedure, in which one applies the vector differential between the existing population members to determine both the degree and direction of the perturbation to the individual subject. A conventional \textit{differential mutation} is designed to be a linear combination of a randomly selected \textit{base individual} ($X_{r3,G}$) and a scaled difference between two other \textit{donor individuals} ($X_{r1,G}$, $X_{r2,G}$):

$$V_{i,G} = X_{r3,G} + F \cdot (X_{r1,G} - X_{r2,G})$$

where $i = 1, \ldots, N_p$; $r1$, $r2$, $r3 \in 1, \ldots, N_p$ are randomly selected and satisfy $r1 \neq r2 \neq r3 \neq i$; $V_{i,G}$ is the perturbed individual and the scaling factor $F$ ($F \in [0, 1]$) is a input control

\begin{algorithm}
1: Read the control parameters: scale factor $F$, crossover rate $C_r$, and population size $N_p$.
2: $G \leftarrow 0$
3: Initialization of population: $X_{i,G}(j) = \text{rand}(0, 1) \cdot (b_U(j) - b_L(j)) + b_L(j)$, where $X_{i,G}(j)$ is the $j^{\text{th}}$ element in the $i^{\text{th}}$ individual of the $G^{\text{th}}$ generation; $b_L(j)$ and $b_U(j)$ are the lower and upper bound of the $j^{\text{th}}$ element, respectively.
4: Fitness evaluation: $f_{i,G} = f(X_{i,G})$.
5: while (stopping criterion is not satisfied) do
6: \quad Mutation: generate donor vectors $V_{i,G} = X_{r3,G} + F \cdot (X_{r1,G} - X_{r2,G})$.
7: \quad Crossover: generate trial vectors $U_{i,G}(j) = \begin{cases} V_{i,G}(j) & \text{if } \text{rand}(0,1) \leq C_r \text{ or } j = j_{\text{rand}} \\ X_{i,G}(j) & \text{otherwise} \end{cases}$
8: \quad Fitness evaluation: $f'_{i,G} = f(U_{i,G})$.
9: \quad Selection: $X_{i,G+1} = \begin{cases} U_{i,G} & \text{if } f'_{i,G} \leq f_{i,G} \\ X_{i,G} & \text{otherwise} \end{cases}$
10: $G \leftarrow G + 1$
11: end while
\end{algorithm}
parameter. The $V_{i,G}$ is then passed to the crossover step, which further generates the trial individuals for the next generation. The whole of the conventional DE procedure for general global minimization problems is presented in Algorithm 1 in a pseudo-code format.

Our DE search engine is modified on top of the conventional procedure of Algorithm 1 based on the characteristics of atomic structures optimization, as will be discussed below.

The complete search flow chart of our algorithm (left module of Figure 1) consists of initialization of individuals (i.e., atomic structures) on each computer node followed by local relaxation to their nearest local minimum. Then, improved DE mutation and crossover operators are implemented to produce possible descendants which are further locally relaxed. We then use greedy selection and migration operators\(^{48,49}\) to generate the individuals in the next generation. Such procedure is repeated until the global minimum is found or end condition is reached.

More specifically, the population initialization includes the following operations. First, the supercell box size of each atomic structure is generated by a small random perturbation from the initial values set by the user. The atoms are then randomly arranged inside the supercell. Nevertheless, the atomic coordinates can be constrained depending on the type of systems (atomic clusters, 2D materials, 3D periodic systems, adsorption systems, etc.). For example, atoms will be placed within a slab for 2D materials search. To guarantee the quality of the individual and the diversity of the population, SGO will check distance between atoms in a given structure and check similarity between different structures\(^{33}\). Structures with bad atomic bonding or similar to other structures will be refined or re-generated. Such bonding check and similarity check are performed in both the initialization and the whole subsequent evolution steps when new structures are generated.

For the mutation scheme, the following modified trigonometric differential mutation\(^ {45}\) is used:

\[
V_{i,G+1} = (X_{r1,G} + X_{r2,G} + X_{r3,G})/3 + (p_2 - p_1) (X_{r1,G} - X_{r2,G}) \
+ (p_3 - p_2) (X_{r2,G} - X_{r3,G}) + (p_1 - p_3) (X_{r3,G} - X_{r1,G})
\]  

(2)

where $p_1 = |f(X_{r1,G})|/p'$, $p_2 = |f(X_{r2,G})|/p'$, $p_3 = |f(X_{r3,G})|/p'$ and $p' = |f(X_{r1,G})| + |f(X_{r2,G})| + |f(X_{r3,G})|$. \( i = 1, \cdots , N_p; r_1, r_2, r_3 \in 1, \cdots , N_p \) are randomly selected and satisfy $r_1 \neq r_2 \neq r_3 \neq i$. $f(X_{r_i,G})$ is the fitness function, defined as the normalized total energy of a atomic structure. Compared to Eq. (1) which uses random perturbation, this scheme is relatively greedy with fast convergent speed\(^ {33,45}\). To better maintain the balance
between the local convergent speed and the global search diversity when the population size is small, we combine both Eq. (2) and Eq. (1), performing Eq. (2) with a probability of $M_r$, and Eq. (1) with a probability of $1 - M_r$.

For the crossover scheme, the conventional multi-point crossover is replaced by the cut-and-splice crossover$^{6,54}$ which can take the advantage of the three-dimensional genomic patterns of the parent structures. SGO provides two schemes: plane-cut-splice and sphere-cut-splice. Assuming the two parent structures are noted as $P_1$ and $P_2$, the first scheme chooses a random plane passing through the mass center of each structure. Then it cuts the structures on this plane, and assembles one child from the atoms of $P_1$ lying above the plane, and the atoms of $P_2$ lying below the plane. If the child structure does not contain the correct number of atoms and chemical species, the operator repeats the process by rotating the plane until the generated offspring satisfies such requirements. The second scheme chooses a random sphere around each structure. Parent structures exchange their atoms inside the sphere to produce the offspring. It is found that the sphere scheme has better performance in large systems or core-shell systems$^6$. Figure 2(a) and 2(b) illustrate the plane-cut-splice and sphere-cut-splice crossover, respectively.

![FIG. 2. Schematic diagram for the cut-and-splice crossover. (a) is the plane scheme and (b) is the sphere scheme. $P_1$ and $P_2$ are the two parent structures while $S_1$ is the generated offspring structure.](image-url)
After the individual trial structures are generated and locally relaxed, they need to be selected in order to become the individual of the next generation. Here, SGO uses the same greedy selection operator as Algorithm 1, i.e., the lower-energy one between parents and offsprings survives. For the parallel scheme, SGO employs a subpopulation-type parallelization. The whole population is partitioned into several subpopulations, each of which communicates with others by the migration operator. A individual in one subpopulation have a certain possibility to migrate to its neighboring subpopulation. This has shown to improve the search efficiency. More details about these evolutionary operators can be found in Ref. 6 and 33.

The whole DE procedure in the SGO engine is illustrated in Algorithm 2.

---

**Algorithm 2** DE algorithm in the SGO engine for atomic structure search

1: Read the control parameters: scale factor $F$, mutation rate $M_r$, crossover rate $C_r$, local relaxation method, the number of GPU processors, population size $N_P$, etc.
2: $G \leftarrow 0$
3: Initialization of population $X_{i,G}$ with atomic bonding check and individual similarity check. $X_{i,G}$ includes atomic position $X_{i,G}^p$ and supercell parameters $X_{i,G}^a$.
4: Local relaxation: $(f_{i,G}, X_{i,G}) \leftarrow$ Relax $(X_{i,G})$. $f_{i,G}$ is the fitness value (i.e., energy) of the system $X_{i,G}$ after atomic relaxation.
5: while (stopping criterion is not satisfied) do
6: Mutation: generate donor vectors $V_{i,G}$ by Eq. (1) and (2).
7: Crossover: generate trial vectors $U_{i,G}$ by cut-and-splice crossover (Figure 2).
8: Local relaxation: $(f'_{i,G}, U_{i,G}) \leftarrow$ Relax $(U_{i,G})$.
9: Selection: $X_{i,G+1} = \begin{cases} U_{i,G} & \text{if } f'_{i,G} \leq f_{i,G} \\ X_{i,G} & \text{otherwise} \end{cases}$
10: Migration: randomly select an individual in one subpopulation to migrate to its neighboring subpopulation (if multi-population are used).
11: $G \leftarrow G + 1$
12: end while
C. Accelerated local relaxation

The initial and trial structures passed to the module of local relaxation (middle module of Figure 1) are usually far from their nearest local minima, and the landscape around each minimum could be ill-conditioned. Conventional relaxation methods, such as CG algorithm\textsuperscript{39}, need hundreds of steps to reach convergence\textsuperscript{37,38}. It is thus crucial to speed up this process.

We have proposed two accelerated local relaxation algorithms in Ref. 37 and Ref. 38. One is CLS algorithm\textsuperscript{37}, which is based on on-the-flight force learning and a corresponding curved line minimization algorithm. The idea is to carry out an ab initio line minimization search along a guided curved line, rather than along the conventional straight line from conjugate gradient direction. Assuming we have a fast force field (surrogate function) whose landscape is close to the DFT one, we could use this force field to construct a curved search line from the initial configuration to its local minimum, and use this search line to guide the DFT line minimization. The hope is to use this curved line to navigate the narrow energy valley, allowing the DFT line minimization to strike a much larger step. One issue is how to find such a force field. Our CLS algorithm employs on-the-flight learning to refine a surrogate potential to reproduce the DFT local landscape. More specifically, CLS algorithm firstly calls a non-linear force-fitting procedure to improve the parameters of a classical force field model to reproduce the DFT atomic forces at a given configuration. Such force-fitting could reproduce a local energy landscape similar to the DFT. The fitted force field then provides a guiding curved line from the initial configuration to its minimum. CLS then deploys DFT line minimization along this guided curve to reach the DFT energy minimum along this curve. The algorithm repeats this process (i.e, force fitting, constructing guided curve and ab initio line minimizations) until the DFT force is smaller than a preset criteria. The whole procedure of CLS is shown in Algorithm 3 in a pseudo-code format. The CLS algorithm has demonstrated a speedup of 2 to 6 for the testing systems and the details can be found in Ref. 37.

Another DFT atomic relaxation speedup scheme is the PCG method\textsuperscript{38}. It is well known that a good pre-conditioner can accelerate the CG algorithm. The best pre-conditioner is the inverse of Hessian matrix of the energy landscape. However, the accurate calculation of the Hessian matrix is more expense than the atomic relaxation itself. The idea of our PCG method is to construct an approximate Hessian with a surrogate function. This is achieved...
Algorithm 3 Curved line search (CLS) algorithm for atomic relaxation.

1: \( k \leftarrow 0 \)
2: Read the initial atomic configurations \( X_0 \) and evaluate DFT force \( G_0 \).
3: while \( |G_k| > g_{\text{error}} \) do
4: Reproduce DFT force on a surrogate potential by force fitting.
5: Construct a decent line (i.e., the guided curved line) from \( X_k \) to its local minimum on this surrogate potential by CG algorithm.
6: Implement \textit{ab initio} line minimization along this guided curved line by a line search algorithm. The final \textit{ab initio} minimum along this curve is updated as the new \( X_{k+1} \) and \( G_{k+1} \).
7: \( k \leftarrow k + 1 \)
8: end while

by a force field with on-the-flight force fitting. In details, PCG firstly calls force-fitting procedure to improve the parameters of a classical force field model, so as to reproduce the DFT atomic forces. Then, a Hessian matrix is calculated by the second-order finite difference of this force field model. Because there are three zero eigenvalues for Hessian matrix corresponding to three translational modes, we constructed the inverse Hessian matrix \( H^{-1} \) as \( \sum_{j \neq 0} V_j^T V_j / \tau_j \), where \( V_j \) and \( \tau_j \) are the eigenvector and eigenvalue of \( H \), respectively. Finally, this inverse Hessian matrix is used as the pre-conditioner of the common PCG procedure. Such force-fitted pre-conditioner could be updated during the relaxation process after a few steps, depending on how far away the initial structure is from its local minimum. The whole PCG procedure is illustrated in Algorithm 4 in a pseudo-code format. We have tested this algorithm on bulk, slab, sheets, clusters and adsorption systems\textsuperscript{38}, and found it showed a speedup factor of 2 to 5.

In our SGO code, CLS, PCG, conventional CG and Broyden-Fletcher-Goldfarb-Shanno (BFGS)\textsuperscript{55} algorithms are provided to the user as input relaxation options. Note that SGO does not require a high-precision local relaxation, especially at the initial stage of the evolution. Computational parameter such as gross K-points sampling grid, small plane-wave energy cutoff and large force convergence criteria (e.g., 0.04 eV/Å) can be used to save the computational cost.
Algorithm 4 Preconditioned conjugate gradient (PCG) algorithm for atomic relaxation.

1: \( k \leftarrow 0 \)

2: Read the initial atomic configuration \( X_0 \) and evaluate DFT force \( G_0 \).

3: Reproduce DFT force on a surrogate potential by force fitting.

4: Calculate the Hessian matrix \( H \) of this surrogate potential and its inverse matrix \( H^{-1} \).

5: Take the \( H^{-1} \) as the pre-conditioner and implement the conventional PCG procedure as follows:

6: Calculate the search vector \( P_k = H^{-1}G_k \).

7: while \( |G_k| > g_{\text{error}} \) do

8: Update the atomic position \( X_{k+1} = X_k + a_k P_k, a_k \) is determined by line search along \( P_k \).

9: Calculate the new DFT force \( G_{k+1} \).

10: Update the search vector \( P_{k+1} = H^{-1}G_{k+1} + \frac{(H^{-1}G_{k+1})'(G_{k+1}-G_k)}{H^{-1}G_k} P_k \).

11: \( k \leftarrow k + 1 \)

12: Return to step 3 to update \( H^{-1} \) if force decreases slowly.

13: end while

D. DFT plane-wave code on GPU

The computations of ab initio energy and atomic force of each structure (right module of Figure 1) are performed by PWmat\textsuperscript{41–44}, a DFT Plane Wave materials simulation code run on GPU clusters or workstations\textsuperscript{41}. PWmat is developed based on a open source CPU DFT plane-wave pseudopotentials code: PEtot\textsuperscript{56}. Compared to PEtot, significant improvements have been made in terms of stabilities and speed. It has simple input file, and is easy to use. It supports several sets of pseudopotentials in both norm-conserving and ultrasoft format. The current PWmat version (v1.5)\textsuperscript{41} has \( 18 \sim 30 \) times of speedup over the CPU PEtot code\textsuperscript{43,44} if the same number of processors are used.

The speedup is achieved mainly by adapting the self-consistent electronic structure calculation to the heterogeneous computer architecture. One self-consistent step begins with an initial charge density \( \rho_{in} \), then Kohn-Sham equation is solved to get the converged wavefunction \( \Psi_i \) and the output charge density \( \rho_{out} \). \( \rho_{out} \) is then mixed with \( \rho_{in} \) for the next step.

In this procedure, the most time consuming computation is to iteratively solve Kohn-Sham equation by the all-band conjugate-gradient (CG) method. This method begins with a initial
set of wavefunctions $\{\Psi_i\}$ and its Hamiltonian matrix $\langle \Psi_i | \hat{H} | \Psi_j \rangle$. The corresponding eigen values $\varepsilon_i$ and eigen vectors (i.e., wavefunctions) $\Psi_i$ are then calculated by subspace diagonalization. The residual $P_i = \hat{H} | \Psi_i \rangle - \varepsilon_i | \Psi_i \rangle$ is calculated and transformed to a preconditioned CG vector, which is further orthogonalized by wavefunctions. Such CG vector is used to improve the wavefunction by a line minimization procedure. Updated wavefunction is then orthogonalized and used to update eigen values. This CG steps will be repeated several times until the residual $P_i$ reaches the convergence condition. After convergence, a new set

**Algorithm 5** All-band conjugate-gradient (CG) method for solving Kohn-Sham equation $\hat{H} | \Psi_i \rangle = \varepsilon_i | \Psi_i \rangle$ in GPU. The time consuming steps indicated by bold numbers are implemented by CUBLAS library, wavefunction data compression or hybrid parallelization of GPU and CPU. Others are implemented by hand-written CUDA kernel subroutines.

```
1: $k \leftarrow 0$
2: Calculate $\hat{H} | \Psi_i \rangle$ and $H(i,j) = \langle \Psi_i | \hat{H} | \Psi_j \rangle$.
3: Calculate a updated set of $\Psi_i$ and eigen value $\varepsilon_i$ by $H$ matrix subspace diagonalization.
4: while (~ convergence) do
5:     $k \leftarrow k + 1$
6:     Residual: $P_i(k) = \hat{H} | \Psi_i \rangle - \varepsilon_i | \Psi_i \rangle$.
7:     if $k = 1$ then
8:         Preconditioned: $P_i(k) = A \cdot P_i(k)$. $A$ is the preconditioned matrix.
9:     else
10:        Preconditioned-CG: $P_i(k) = A \cdot (P_i(k) - \alpha_i \cdot P_i(k - 1))$. $\alpha_i$ is CG coefficient.
11:    end if
12:    Projection: $P_i(k) = P_i(k) - \sum_{j=1,i} | \Psi_j \rangle \langle \Psi_j | P_i \rangle$.
13:    Update wavefunction by line minimization: $| \Psi_i \rangle = | \Psi_i \rangle \cos \theta_i + P_i(k) \sin \theta_i$.
14:    Orthogonalization: $| \Psi_i \rangle = | \Psi_i \rangle - \sum_{j=1,i-1} | \Psi_j \rangle \langle \Psi_j | \Psi_i \rangle$.
15:    Update $\hat{H} | \Psi_i \rangle$ and approximate eigen values $\varepsilon_i = \langle \Psi_i | \hat{H} | \Psi_i \rangle$.
16: end while
17: Calculate $H(i,j) = \langle \Psi_i | \hat{H} | \Psi_j \rangle$.
18: Perform $H$ matrix subspace diagonalization to output the new set of $\Psi_i$ and $\varepsilon_i$.
```
of wavefunctions are finally calculated and output by subspace diagonalization. Flow chart of this all-band CG method is detailed in Algorithm 5. In the PWmat, we have moved the entire all-band CG method into GPU by using the CUBLAS library, hybrid parallelization, and wavefunction data compression to reduce MPI communication\textsuperscript{43,44}. Moreover, the hybrid parallelization scheme is also used in other routines for self-consistent charge density calculations to fully utilize the GPU computing power.

The PWmat and SGO codes can be run on common GPU machines (e.g., Titan of Oak Ridge Leadership Computing Facility) or on custom-made GPU workstation: Mstation (Material Station)\textsuperscript{41,43,44}. One Mstation has 4 NVIDIA Titan X GPU processors and 128 GB RAM, supporting the DFT calculations for systems as large as 1,000 atoms. It has a great performance for single precision operations\textsuperscript{41}. PWmat used a hybrid approach to take the advantage of the single precision speed while guarantee the accuracy by calculating some critical steps using double precision (e.g., the calculation of $\hat{H}\Psi_i$)\textsuperscript{41}.

Note that the local relaxation of all the individual structures in one generation can be implemented in parallel or in serial, depending on the available computational resources. For an example with 8-individuals population, the 8 atomic structures can be relaxed at the same time by 8 chips of GPU, each of which runs one local relaxation job. This can be achieved by two Mstations. They can also be relaxed one by one serially using only 1 chip of GPU resource. In general, the population with 8 individuals could be sufficient for the search of systems with 25 atoms or less, due to the improvement of global search ability. A few multiple independent runs of the whole SGO searching procedure could be needed to ensure the discovery of the true global minimum to compensate the stochastic feature of the differential evolution algorithms.

III. RESULTS

Due to acceleration at all levels of algorithms, the SGO engine is expected to be fast. We first tested SGO on some benchmarks of classical force field and compared the results with other existing packages. Then we tested it on the search of ab initio global minimum of various systems (crystals, 2D materials and clusters), and also presented the results on the ab initio energy landscapes. The ab initio tests were run on the custom-made Mstation machine as discussed above.
A. Tests on benchmarks of classical force field

To demonstrate the global search performance, we first tested SGO on several Lennard-Jones (LJ) clusters and Gupta-potential Cobalt clusters, which have been used as benchmarks in previous studies. Our results are compared with four other methods reported in previous literatures, i.e., Basin Hopping (BH)\textsuperscript{19,57}, Minima Hopping (MH)\textsuperscript{58}, Particle Swarm Optimization (PSO)\textsuperscript{59} and Genetic algorithm (GA)\textsuperscript{58}, as shown in Table I. SGO uses 8 individuals for all the searches and the maximal number of generations is set to 300. All the methods yield the same global minimum configuration for each cluster. We can see that the performance of SGO on small LJ clusters (LJ\textsubscript{26} and LJ\textsubscript{55}) are comparable to other methods, in terms of the successful hit rate ($R_{\text{hit}}$) in multiple runs of the whole procedure and the average number ($N_{\text{LM}}$) of local minimizations needed by one hit of the global minima.

<table>
<thead>
<tr>
<th>Cluster</th>
<th>Energy</th>
<th>BH\textsuperscript{a}</th>
<th>MH\textsuperscript{b}</th>
<th>PSO\textsuperscript{c}</th>
<th>GA\textsuperscript{b}</th>
<th>SGO (DE)\textsuperscript{d}</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$R_{\text{hit}}$</td>
<td>$N_{\text{LM}}$</td>
<td>$R_{\text{hit}}$</td>
<td>$N_{\text{LM}}$</td>
<td>$R_{\text{hit}}$</td>
</tr>
<tr>
<td>LJ\textsubscript{26}</td>
<td>-108.315616</td>
<td>100%</td>
<td>96</td>
<td>100%</td>
<td>73</td>
<td>100%</td>
</tr>
<tr>
<td>LJ\textsubscript{55}</td>
<td>-279.248470</td>
<td>100%</td>
<td>126</td>
<td>100%</td>
<td>159</td>
<td>100%</td>
</tr>
<tr>
<td>LJ\textsubscript{100}</td>
<td>-557.039820</td>
<td>84.0%</td>
<td>5960</td>
<td>82.0%</td>
<td>5376</td>
<td>87.5%</td>
</tr>
<tr>
<td>LJ\textsubscript{150}</td>
<td>-893.310258</td>
<td>90.0%</td>
<td>9490</td>
<td>85.0%</td>
<td>7980</td>
<td>68.8%</td>
</tr>
<tr>
<td>Co\textsubscript{165}</td>
<td>-643.6312979</td>
<td>75.0%</td>
<td>43667</td>
<td>75.0%</td>
<td>43667</td>
<td>50.1%</td>
</tr>
<tr>
<td>Co\textsubscript{185}</td>
<td>-725.0514853</td>
<td>75.0%</td>
<td>106416</td>
<td>75.0%</td>
<td>106416</td>
<td>81.3%</td>
</tr>
</tbody>
</table>

\textsuperscript{a} The data derived from BH for LJ an Co clusters are taken from Ref. 19 and Ref. 57, respectively.
\textsuperscript{b} The data derived from MH and GA are taken from Ref. 58.
\textsuperscript{c} The data derived from PSO are taken from Ref. 59.
\textsuperscript{d} The failed hit means SGO has not found the global minimum within 300 generations, i.e., 2400 local optimizations, while the upper limit for PSO is $1.1\times10^4$ local optimizations, $8\times10^4$ and $2\times10^5$ for BH in Co\textsubscript{165} and Co\textsubscript{185}, and too long time (no values given) for other methods.
There is no significant difference between different methods. But SGO exhibits much better performance than all of BH, MH, PSO and GA on large LJ clusters (LJ$_{100}$ and LJ$_{150}$). It only needs about 1/9 of local minimizations of MH, PSO and GA to reach the LJ$_{100}$ minimum, without losing the hit rate at the same time. Its $N_{LM}$ on LJ$_{150}$ is also several times smaller than the values of other methods, although the hit rate is slightly decreased. For Cobalt clusters, the performance is further enhanced compared with BH. Its $N_{LM}$ is only about 1/63 and 1/129 of BH's values on Co$_{165}$ and Co$_{185}$, respectively. The hit rate is slightly decreased on Co$_{165}$ while increased on Co$_{185}$. This rate is not better because SGO uses a small population (8 individuals) and a small number of generations (300). It can be increased when more searches (more individuals and more generations) are performed. But the better way is to restart the SGO search from scratch, with multiple relatively short runs to compensate such stochastic feature. The settings with a small population are also very useful in ab initio search where the calculation is extremely expensive and long-time search with a large population is usually unaffordable. We see that $N_{LM}$ in SGO increases very smoothly with the cluster size while $N_{LM}$ in BH, MH, PSO and GA increases hugely. It is very difficult to quantitatively prove what causes the differences. We speculate that these methods have not reach their optimal performance in large clusters while SGO is optimized for complex and large systems.

We have to note that the data of BH, MH, PSO and GA in Table I were taken from literatures published several years ago. These methods could have been improved recently and the direct comparisons between different methods could be difficult and sometime unfair. We thus do not claim that the SGO with DE algorithm is much better than all the other methods. These comparisons do show that SGO (DE) is a very competitive method and has a good global search performance when only a small number of local minimizations are used.

B. Search of ab initio global minimum

1. Three-dimensional crystals

We now tested SGO with ab initio calculations on some well-known crystals with one (Si and Mg), two (SiO$_2$ and ZnO) and three elements (CaCO$_3$ and BiFeO$_3$). One of the purpose
is to demonstrate how fast the SGO engine could be reached using ab initio calculations for realistic systems. Table II lists the settings and the search performance. During these global searches, the number of atoms for each element in the supercell are fixed.

We can see that SGO can find the known ground-state structures of Si and Mg in one or two generations with a small population (8 individuals), costing 15.1 and 8.64 local relaxations on average, respectively. For the other cases with two or three elements, SGO are still able to locate the targeted global minimum in less than 15 generations. When the number of atoms in the supercell is no more than 10, these systems need small numbers of local relaxations (less than 100) and less than 2 hours running time on two Mstations.

Similar global searches for some of these crystals have been performed by other methods. For example, PSO\textsuperscript{24} used 25.6 local relaxations on average to locate the diamond-structure Si while GA\textsuperscript{28} used 60 local relaxations (both using a 8-atom supercell just like in our test). PSO used 5 generations with a 20-individuals population for the search of SiO\textsubscript{2}, using in total 100 local relaxations. It used 2, 6 and 13 generations with a 30-individuals population, i.e., 60, 180 and 390 local relaxations for Mg, ZnO and CaCO\textsubscript{3}, respectively. It should be noted that the data were published in 2007 for GA\textsuperscript{28} and in 2010 for PSO\textsuperscript{24}. One might have obtained better results by using other parameter settings (e.g., a smaller population) or by improving algorithm. Unfortunately, we can only provide the literature results for these methods. This is not a guarantee for a systemic and completely fair comparison. A good comparison between different methods is a major undertaking and is beyond the scope of the current paper.

Note that we do not use any symmetry constraint during the initialization and local relaxation. This type of free search is much more difficult than the one using specific symmetry, but it allows the search for all the possible configurations in the whole energy landscape. Figure 3 shows the evolution of the best ZnO individual in the population. SGO not only finds the targeted zincblende structure (space group F\textbar 43m) but also the wurtzite structure (space group P6\textbar 3mc). With the exchange-correlation PBE potential, zincblende structure has slightly lower energy than wurtzite structure.
TABLE II. Global search of the ground-state structure of various systems. The target structure is indicated by space group for crystals and 2D materials or point group for clusters. \( N \) (atoms) is the number of atoms in the searching supercell. \( N \) (Gen) is the number of generations on average needed to locate the global minimum in multiple independent runs. \( N \) (LM) is the corresponding average number of local minimization. \( T \) (hours) is the average computational time for running on 2 Mstations. The time will be double if tests are run on only 1 Mstation. The population size is set to 8 for all the tests.

<table>
<thead>
<tr>
<th>Systems</th>
<th>Structures</th>
<th>( N ) (atoms)</th>
<th>( N ) (Gen)</th>
<th>( N ) (LM)</th>
<th>( T ) (hours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Crystals</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Si</td>
<td>Fd( \bar{3} )m</td>
<td>8</td>
<td>1.89</td>
<td>15.1</td>
<td>0.081</td>
</tr>
<tr>
<td>Mg</td>
<td>P6( \bar{3} )/mmc</td>
<td>8</td>
<td>1.08</td>
<td>8.64</td>
<td>0.034</td>
</tr>
<tr>
<td>SiO(_2)</td>
<td>P(_3)1(_2)</td>
<td>9</td>
<td>5.22</td>
<td>41.8</td>
<td>0.37</td>
</tr>
<tr>
<td>ZnO</td>
<td>F( \bar{4} )3m</td>
<td>8</td>
<td>3.25</td>
<td>26.0</td>
<td>0.36</td>
</tr>
<tr>
<td>BiFeO(_3)</td>
<td>R( \bar{3} )c</td>
<td>10</td>
<td>7.43</td>
<td>59.4</td>
<td>0.83</td>
</tr>
<tr>
<td>CaCO(_3)</td>
<td>R( \bar{3} )c</td>
<td>10</td>
<td>12.4</td>
<td>99.2</td>
<td>1.53</td>
</tr>
<tr>
<td>2D materials</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Graphene</td>
<td>P(_6)/mmm</td>
<td>8</td>
<td>8.75</td>
<td>70.0</td>
<td>0.16</td>
</tr>
<tr>
<td>MoS(_2)</td>
<td>P( \bar{6} )m2</td>
<td>12</td>
<td>29.4</td>
<td>235.2</td>
<td>1.36</td>
</tr>
<tr>
<td>Atomic clusters</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pt(_{10})</td>
<td>T(_d)</td>
<td>10</td>
<td>22.7</td>
<td>181.6</td>
<td>2.18</td>
</tr>
<tr>
<td>Pt(_{23})</td>
<td>C(_1)</td>
<td>23</td>
<td>78.0</td>
<td>624.0</td>
<td>10.4</td>
</tr>
</tbody>
</table>

2. Two-dimensional materials

We also tested two known 2D materials: graphene and MoS\(_2\). The known ground state of 2D carbon materials is a monolayer of six-member rings (i.e., graphene) while the ground state of 2D MoS\(_2\) is a three-layer hexagonal structure with two S layers at both sides of Mo layer. A 8-atom supercell is used for the carbon case and a 12-atom supercell is used for the MoS\(_2\) case. SGO spends 8.75 and 29.4 generations in average to locate these two configurations, respectively. The number of generation required is a little larger than the one for 3D crystals with the same number of atoms and elements in the supercell, mostly because there is one dimension without periodic restriction. SGO does not constrain atoms on a fixed plane, but allows them to move freely within a preset slab region. This free dimension will produce a larger number of possible configurations, hence a larger number of
FIG. 3. Evolution of the best individual of the population in one search of ZnO. Etot indicates the total energy of one 8-atom supercell structure compared to the global-minimum structure (which defines the referenced zero energy) in the unit of eV. SGO finds the wurtzite structure in the 9th generation and the targeted zincblende structure in the 10th generation.

local relaxations.

Figure 4 shows the evolution of the best individual of the population in one search of graphene. This search uses 8 generations to reach the global minimum. The lowest-energy configuration in the first generation is a monolayer consisting of five-member rings and seven-member rings. The discovery of this interesting configuration owns to the search without symmetry constraint. It costs another 7 generations for SGO to evolve to the targeted graphene monolayer consisting of six-member rings.

3. Atomic clusters

The energy landscape of atomic clusters is more complex than those of crystals and 2D materials because the lack of periodic boundary constraint makes many local minima possible. There are a large number of free clusters with locally stable configurations. We tested SGO on two Pt clusters: Pt\textsubscript{10} and Pt\textsubscript{23}. The known global minima of Pt\textsubscript{10} and Pt\textsubscript{23} in the literature have the point symmetry of T\textsubscript{d} and C\textsubscript{4v}, respectively\textsuperscript{60}. In our searches, SGO find the T\textsubscript{d}-symmetry Pt\textsubscript{10} in 37th generations and the C\textsubscript{4v}-symmetry Pt\textsubscript{23} in 11th generations. For Pt\textsubscript{23}, SGO also finds a few structures with lower energy than the known C\textsubscript{4v} structure in the literature\textsuperscript{60}. Figure 5 shows the corresponding evolution process of Pt\textsubscript{23}.
FIG. 4. Evolution of the best individual of the population in one search of graphene. Etot indicates the total energy of one 8-atom supercell structure compared to the global-minimum structure in the unit of eV. The configuration found in the 1st generation is a monolayer consisting of five-member rings and seven-member rings while the ground state found in the 8th generation is the six-member rings’ graphene structure.

clusters. We can see that the final minimum is about 0.35 eV lower in energy than the known C_{4v} structure. This new minimum configuration has no symmetry operation.

Compared to the searches of other systems, Pt clusters cost more computer time. There are several reasons. First, the big vacuum around clusters and the large number of valance electrons in pseudopotential make the DFT calculation more expensive. Second, the complex energy landscape results in the need of more steps for each local relaxation, as well as the need of more generations for global search (e.g., 78 generations). Nevertheless, the several hours’ computer times (Table II) are still acceptable for routine calculations.

C. Search of ab initio energy landscape

Due to the fast search speed, SGO can also be used to explore the energy landscape of a given system within an affordable time, especially exploring the landscape around a global minimum. We tested this function with the systems of SiO_2 crystal, carbon monolayer and Pt_{23} atomic clusters. We adopted 9-atoms supercell for SiO_2 and 8-atoms supercell for carbon. Pt_{23} cluster is contained in a 24Å×24Å×24Å supercell. About 500 local minimizations in total are searched for SiO_2 and carbon systems, and about 3000 for Pt clusters. Figure
FIG. 5. Evolution of the best individual of the population in one search of Pt\textsubscript{23} clusters. \textit{Etot} indicates the total energy of one structure compared to the global-minimum structure in the unit of eV. The configuration found in the 11\textsuperscript{th} generation is a known structure with C\textsubscript{4v} point symmetry and the one in the 78\textsuperscript{th} generation is the new minimum.

6 shows the found landscapes, in terms of local-minima energy spectrum (i.e., each line represents one stable local minimum).

For SiO\textsubscript{2}, SGO finds a few stable crystals in the energy region of [0, 1.0] eV above the ground state. The space group of the ground-state structure is P3\textsubscript{1}21 (Figure 6d). The configuration 0.11 eV above the ground state has a higher symmetry with space group P6\textsubscript{2}22 (Figure 6c) while the one with 0.15 eV energy has space group P3\textsubscript{2}21 (Figure 6b), very similar to the ground-state structure. Both P6\textsubscript{2}22 and P3\textsubscript{2}21 structures have been confirmed in other studies\textsuperscript{61,62}. Note that the relative energy between different configurations could be changed at different external pressure.

For 2D carbon monolayer, there is no other stable 2D configuration above the well-known graphene (Figure 6h) in the energy region of [0 2.0] eV. This to some extent indicates that there is a large energy barrier from graphene to other structures. Thus, graphene is very stable, even at high temperature or strong external interaction. The configuration (Figure 6g) with an energy 2.2 eV above the graphene consists of five-member and seven-member carbon rings. It has the symmetry C\textsubscript{mmm} and is also very stable. There are two stable configurations with energies about 0.7 eV higher than the five-seven membered rings’ structure. The lower-energy one (Figure 6f, space group P\textsubscript{mmm}) is made up of five-member,
six-member and eight-member rings, while the higher-energy one (space group Cmmm) is made up of four-member, six-member and eight-member rings. Their energy difference is only 0.036 eV. Above these configurations, there are a few other stable isomers with different rings structures, such as four-five-ten membered rings.

For Pt$_{23}$ cluster, the energy spectrum is much denser than the ones of SiO$_2$ crystal and carbon monolayer. There are a lot of local minima configurations in the energy region of [0 2.0] eV. Especially, a few of them have lower energy than the putative global minimum with C$_{4v}$ symmetry in literatures$^{60}$ (Figure 6(j)). Most of these lower-energy configurations have lower-symmetry geometry. This dense local minima spectrum indicates there could be a few configurations co-existing in a given experiment. These configurations could have very different properties. For example, the total magnetic moment for the configurations in Figure 6(j), (k) and (l) is 2.7, 1.1, and 1.3 $\mu$B, respectively. Seen in Figure 7(a-c), the spin density of the configuration of Figure 6(j) distributes uniformly around 19 atoms except the 4 atoms in the bottom edge, while the spin of the other two configurations only surrounds the center atom and 4 edge atoms. Such significantly different magnetic properties also exist in other cluster size. We have further calculated the Pt$_n$ clusters with $n$ from 4 to 24.

FIG. 6. The energy landscape of (a-d) SiO$_2$ crystal, (e-h) 2D carbon monolayer and (i-l) Pt$_{23}$ atomic cluster. Each blue line in (a), (e) and (i) indicates one configuration. (b-d), (f-g) and (j-l) are several representative configurations. Etot is the total energy of one configuration compared to the global-minimum configuration in the unit of eV.
FIG. 7. (a-c) Spin density of Pt$_{23}$ clusters of Figure 6 (j-l). (d) Relationship between magnetic moment per atom (in unit of $\mu$B) and the Pt cluster size. Each circle represents one local minimum configuration found in each cluster size. Each asterisk in the solid line is the mean magnetic moment of all the found configurations in one size.

The magnetic moment (per atom) of these clusters and their fluctuations for different local minima are represented in Figure 7(d). We can see that their magnetic moment fluctuates strongly in different local minima of the same size. Their values can vary from 0.0 to as large as 1.5 $\mu$B per atom. The mean magnetic moment of all found minima in each size decreases with the increasing size and will converge to zero in the bulk. This sensitivity of the magnetic moment on the local minima structure can be used as a probe to measure the cluster structure information, and the understanding of the magnetic properties depending on the structure and size can help us to use metal nanoclusters as information storage medium in the future.

Note that the energy landscapes of Figure 6 are given by a small number of searches with a fixed number of atoms in the supercell. There could be more configurations when we use a different number of atoms or use more searches.
IV. CONCLUSIONS

A fast engine SGO for ab initio atomic structure global optimization is presented. We speeded up the engine by a high-efficiency DE algorithm, accelerated local atomic relaxation methods and a fast DFT code run on GPU machines. We tested SGO on well-known crystals, 2D materials and atomic clusters. SGO can find the targeted minimum of the systems with less than 25 atoms from a few minutes to a few hours running on GPU workstations. Comparisons with other existing packages support its high performance. SGO is also able to explore the energy landscape of a given system within an affordable computer time. We have studied the energy landscape of SiO$_2$ crystal, carbon monolayer and Pt clusters, and investigated the magnetic properties of Pt clusters with the finding of the strong magnetic moment fluctuations on different local minima structures. Our work demonstrates that by combing the best of line algorithms and computer codes, it is now possible to make atomic structure search for med-sized systems a routine computational task.

ACKNOWLEDGMENTS

The algorithm development is supported by the U.S. Department of Energy, Office of Science, Office of Basic Energy Sciences, Materials Sciences and Engineering Division, of the under Contract No. DE-AC02-05-CH11231 within the Non-Equilibrium Magnetic Materials program (MSMAG). The code development is supported by the National Natural Science Foundation of China under Grants No. 11574304.

* zhanghuichen88@gmail.com
† lwang@lbl.gov

3 D. J. Wales, Science 293, 2067 (2001).
34 https://www.sgo.ac.cn.


