Computer Physics Communications 183 (2012) 2063–2070

Contents lists available at SciVerse ScienceDirect

Computer Physics Communications

journal homepage: www.elsevier.com/locate/cpc

CALYPSO: A method for crystal structure prediction

Yanchao Wang, Jian Lv, Li Zhu, Yanming Ma *

State Key Laboratory of Superhard Materials, Jilin University, Changchun 130012, China

ARTICLE INFO

Article history:
Received 18 August 2011
Received in revised form
2 May 2012
Accepted 8 May 2012
Available online 15 May 2012

Keywords:
Structure prediction
Particle swarm optimization algorithm
Crystal structure

ABSTRACT

We have developed a software package CALYPSO (*Crystal structure AnaLYsis by Particle Swarm Optimization*) to predict the energetically stable/metastable crystal structures of materials at given chemical compositions and external conditions (e.g., pressure). The CALYPSO method is based on several major techniques (e.g. particle-swarm optimization algorithm, symmetry constraints on structural generation, bond characterization matrix on elimination of similar structures, partial random structures per generation on enhancing structural diversity, and penalty function, etc.) for global structural minimization from scratch. All of these techniques have been demonstrated to be critical to the prediction of global stable structure. We have implemented these techniques into the CALYPSO code. Testing of the code on many known and unknown systems shows high efficiency and the highly successful rate of this CALYPSO method [Y. Wang, J. Lv, L. Zhu, Y. Ma, Phys. Rev. B 82 (2010) 094116] [29]. In this paper, we focus on descriptions of the implementation of CALYPSO code and why it works.

1. Introduction

Understanding the behaviors of materials at the atomic scale is fundamental to modern science and technology. As many properties and phenomena are ultimately controlled by the crystal structures, the prediction of crystal structure is an important task in chemistry and condensed matter physics. However, the structural prediction with the only known information of chemical compositions is extremely difficult as it basically involves in classifying a huge number of energy minima on the lattice energy surface. Owing to the significant progress in both computational power and basic materials theory, it is now possible to predict the crystal structure at zero Kelvin using the quantum mechanical methods. One way to predict structure is by extracting known structures from databases of structures previously found in similar materials [1]. However, this method has a limited success rate and is incapable of generating new crystal structure types. Recently, the more advanced methods including simulated annealing [2,3], minima hopping [4], basin hopping [5], metadynamics [6], the genetic algorithm [7–14], and random sampling method [15] have been developed and applied, which allow a systematic search for the ground state structures based on the chemical composition and the external conditions. The simulated annealing, basin hopping, minima hopping and metadynamics focus on overcoming the energy barriers and are successful in many researches [2–6], particularly, when the starting structure is close to the global minimum. The genetic algorithm starts to use a self-improving method and is thus able to correctly predict many structures [16–19]. The random sampling method, as a simple and efficient method, is also successful in many applications [20–23].

The particle swarm optimization (PSO), first proposed by Kennedy and Eberhart [24,25], is a population-based optimization method. As a stochastic global optimization method, PSO is inspired by the choreography of a bird flock and can be seen as a distributed behavior algorithm that performs multidimensional search. According to PSO, the behavior of each individual is affected by either the best local or the best global individual to help it fly through a hyperspace. Moreover, an individual can learn from its past experiences to adjust its flying speed and direction. Therefore, all the individuals in the swarm can quickly converge to the global position. The PSO algorithm is a highly efficient global optimization method which has been applied successfully to many optimization problems such as network training [26,27] and transactions on power systems [28]. However, the application of PSO to the structural prediction of condensed matters remains a major challenge. Due to the existence of a large number of energy minima on the lattice energy surface, rapid swarm convergence, as one of the main advantages of PSO, can also be problematic. If an early solution is sub-optimal, the swarm can easily stagnate around it without any pressure to continue further exploration, i.e., being premature. We recently have developed a CALYPSO method/code [29] on crystal structure prediction by implementation of the PSO algorithm and many other important techniques, including symmetry constraints on structural generation, the bond characterization matrix on the elimination of similar structures, partial random structures per generation on enhancing structural diversity, and the penalty

* Corresponding author.
E-mail address: mym@jlu.edu.cn (Y. Ma).

0010-4655/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.cpc.2012.05.008

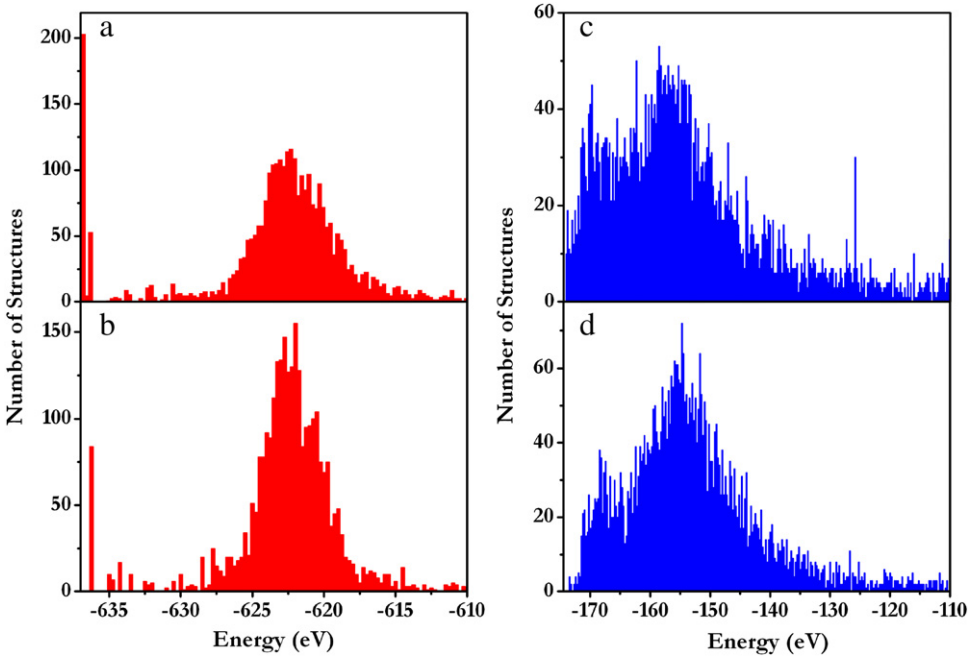

Fig. 1. (Color online) The energy distributions of randomly generated structures containing 16 TiO_2 units in the simulation cell and 6 units of binary Lennard-Jones crystal A_2B in the simulation cell after local optimization. (a) and (b) indicate the energy distribution of TiO_2 structures generated with and without the symmetric constraints, respectively. (c) and (d) indicate the energy distribution of A_2B structures generated with and without the symmetric constraints, respectively.

function, etc. We found that these latter techniques are critical to avoid the premature nature of the PSO algorithm and to significantly accelerate the structural convergence.

The description of the CALYPSO method and its first applications to the prediction of crystal structures can be found in Ref. [29]. This paper is organized as follows. The detailed descriptions of implementation of CALYPSO code and the principles on illustrating why the method works are presented in Section 2. In Section 3, various parameters in CALYPSO code are optimized for TiO_2 as a benchmark. The input and output files are provided in Section 4. A short overview of the applications obtained from our method can be found in Section 5, followed by the conclusion in Section 6.

2. Implementation and discussions

As depicted in the pseudo-code of Algorithm 1, the CALYPSO method comprises four main steps: (i) generation of random structures with the constraint of symmetry; (ii) local structural optimization; (iii) post-processing for the identification of unique local minima by bond characterization matrix; (iv) generation of new structures by PSO for iterations.

2.1. Symmetry constraints on structural generation

There are two types of variables to define a crystal structure: lattice parameters (three angles and the lengths of the three lattice vectors) and atomic coordinates (three coordinates coded as a fraction of the lattice vector for each atom). The first step of the CALYPSO method is to generate random structures constrained within 230 space groups. Once a particular space group is selected, the lattice parameters are generated within the chosen symmetry according to the confined volume and the corresponding atomic coordinates are obtained by a combination of a set of symmetrically related coordinates (Wyckoff Positions) in accordance to the number of atoms in the simulation cell. For example, if the confined volume is 64 $Å^3$ per cell and there are 12 atoms in the simulation cell for the group 223 (Pm–3n), the lengths of three lattice vectors should be 4 Å and the lattice angles are fixed to 90°, while the atomic positions can be obtained by different combination of Wyckoff Positions (e.g., $6b + 6c$, $6b + 6d$, and $12f$, etc). Moreover, a backup database of space groups of all generated structures is built and used to compare with the space groups of newly generated structures. The appearance of identical space group is forbidden with a certain probability (80%). This allows the initial samplings to cover different regions of the search space, which is crucial for the diversity of population. The generation of random structures ensures unbiased sampling of the energy landscape. The explicit application of symmetric constraints leads to significantly reduced search space and optimization variables, and thus fastens global structural convergence.

In order to examine the efficiency of symmetric constraints as implemented in CALYPSO code, the system of $TiO_2$with 16 TiO_2 units (48 atoms) per simulation cell was used as a test case. 3250 structures at ambient pressure were randomly generated and then structurally optimized using the GULP code [30] with a combination of Buckingham and Lennard-Jones potentials [10,31]. Fig. 1(a) and (b) show the energy distributions of these generated structures with and without symmetry constraints, respectively. It is found that the rutile structure, i.e., the global stable structure cannot be generated if without symmetry constraints. However, once the symmetry is implemented in the generation of random structures, 203 (∼6.2% in total) rutile structures were successfully produced. In order to further compare the structural search efficiency of generation of random structures with or without the symmetry constraints, the binary Lennard-Jones crystal A_2B (18 atoms per simulation cell) was used as another test case. 5000 structures were randomly generated and then structurally optimized using the GULP code [30] with Lennard-Jones potentials ($\sigma_{AA} = \varepsilon_{AA} = 1.0$, $\sigma_{BB} = 0.88$, $\varepsilon_{BB} = 0.5$, $\sigma_{AB} = 0.932$ and $\varepsilon_{AB} = 1.5$) [32]. The energy distributions of the structures generated with and without symmetry constraints are shown in Fig. 1(c) and (d), respectively. It is obvious that the energies of these structures generated with symmetry constraints distribute lower energy regions. We also have examined the structural search efficiency of CALYPSO runs with or without the symmetry constraints on structural generation as shown in Table 1. Obviously,

Table 1
The structural search efficiency of CALYPSO calculations with or without the symmetric constraints on structural generation for the system of TiO_2.We have performed ten different CALYPSO runs and the total generations for these ten runs needed to find the global stable rutile structure are listed. As an illustration, we choose here the population size as 20. Notably, we generally use larger population sizes for larger systems; there much fewer generations are needed to find the stable structure. Other typical CALYPSO run parameters of $V_{\max}$ and the percentage of PSO generated structures are chosen as 0.1 and 0.6, respectively.

	Number of atoms in the system			
	12	24	36	48
	Generations	Generations	Generations	Generations
Symmetry constraints	12	15	85	110
No symmetry constraints	12	25	138	254

the application of symmetry constraints technique can greatly improve the search efficiency, especially for larger systems. It is found that an average of 11 generations are necessary to find the global stable structure if with the symmetry constraints on structural generation, however if without, 25.4 generations are needed. These tests clearly illustrate the importance of the symmetry constraints in the generation of random structures for structure prediction.

2.2. Structural optimization

CALYPSO code currently can use ab initio packages (e.g., VASP [33,34], SIESTA [35] and CASTEP [36,37]) and force-field program (e.g., GULP [30]) to perform the structural optimization. Other external programs can also be interfaced on user's request. The use of locally structural optimization techniques (e.g., line minimization, steepest descents, conjugate gradient algorithm or Broyden–Fletcher–Goldfarb–Shanno algorithm) leads the lattice energy to the local minimum. Here, we use free energy (at $T = 0$ K, free energy reduces to enthalpy) as fitness function throughout the simulation. Note that local optimization increases the cost of each individual, but reduces effectively the noise of the energy landscape, enhances comparability between different structures, and provides locally optimal structures for further use. Thus, local optimization is crucial for the structure prediction.

2.3. Elimination of similar structures by using the bond characterization matrix

Our goal is to eliminate the similar structures in the structure generations to enhance the search efficiency of CALYSPO. In our earlier implementation [29], we used the geometrical structure parameter, which is solely based on the bond length, to identify structural similarity. Here, we have developed a more efficient technique named the bond characterization matrix, which is on the basis of all the bond information. In this method, we employ a set of modified bond-orientational order metrics (Q_l) introduced by Steinhardt et al. [38] to quantify the bond angles and an exponential function to quantify the bond length. When the distance between two atoms is less than the cutoff (r_{cut}), bond information, e.g. bond vector ($\overrightarrow{r_{ij}}$), bond angles (θ_{ij}, ϕ_{ij}) and bond-types (δ_{AB}), are evaluated, where $\overrightarrow{r_{ij}}$ is a vector pointing from ith atom to jth atom, while θ_{ij}, ϕ_{ij} are the related polar and azimuthal angles of $\overrightarrow{r_{ij}}$, respectively, and $A(B)$ is the type of ith(jth) atom. In this work, bond characterization matrix is calculated according to the "bond-types", where each vector $\overrightarrow{r_{ij}}$ can be represented by spherical harmonics $Y_{lm}(\theta_{ij}, \phi_{ij})$. Subsequently, for each bond type δ_{AB}, a weighted average is performed,

$$\overline{Q}_{\text{lm}}^{\delta_{AB}} = \frac{1}{N_{\delta_{AB}}} \sum_{i\in A,\ j\in B} \mathrm{e}^{-\alpha(r_{ij}-b_{AB})} Y_{\text{lm}}(\theta_{ij}, \varphi_{ij}) \tag{1}$$

where $N_{\delta_{AB}}$ is the number of bonds formed by type A and B atoms, b_{AB} is the shortest length for each bond type and α is an adjusted parameter driving $e^{-\alpha(r_{cut}-b_{AB})} \to 0$. In order to avoid the dependence on the choice of reference frame, the average $\overline{Q}_{lm}^{\delta_{AB}}$ is used to calculate the rotationally invariant combinations,

$$Q_l^{\delta_{AB}} = \sqrt{\frac{4\pi}{2l+1} \sum_{m=-l}^{l} |\overline{Q}_{lm}^{\delta_{AB}}|^2}. \tag{2}$$

Only even-l spherical harmonics, which are invariant with respect to the direction of the bonds, are used in Eq. (2), and each structure can be characterized by the bond characterization matrix. The similarity between two structures is thus given by the Euclidean distance of their bond characterization matrix.

$$D_{uv} = \left[\sum_{\delta_{AB}} \sum_{l} (Q_l^{\delta_{AB},u} - Q_l^{\delta_{AB},v})^2 \right]^{1/2} \tag{3}$$

where u and v are individual structures.

As an illustrative case, the histograms of Q_l versus l for graphite and diamond are shown in Fig. 2(a) and (b), respectively. Significant differences for Q_l between these two structures are evidenced, which illustrate the efficiency of the bond characterization matrix method to distinguish between different structures. To further demonstrate the robustness of the method, the Euclidean distances between graphite/diamond and its random distortions are calculated as shown in Fig. 2(c)/(d). It is clearly seen that the calculated Euclidean distances monotonously increase with the magnitude of distortions. These tests highlight the capability of this bond characterization matrix method in the characterization of the structural similarities.

We have implemented this bond characterization matrix technique into CALYPSO code to eliminate similar structures. Table 2 shows the influence of this technique on the search efficiency of CALYPSO calculations for the system of TiO_2. It is clearly seen that much fewer optimization steps are needed to find the stable structure when this technique is included in the CALYPSO runs. This is understandable since the use of the bond characterization matrix technique can effectively avoid the presence of very similar or identical structures and thus is able to accelerate the global structure convergence.

2.4. Generation of new structures by PSO

Within the PSO scheme, a structure (an individual) in the searching space is regarded as a particle. A set of individual structures is called a population. The lattice parameters (unit cell) of new structures are the same as the corresponding structures of the previous generation. While the atomic positions are updated using the evolutionary Eq. (4). Note that all the new structures produced by PSO (or randomly generated) are tested against the constraint of minimal inter-atomic distances [9].

$$x_{i,j}^{t+1} = x_{i,j}^{t} + v_{i,j}^{t+1}. \tag{4}$$

The initial $v_{i,j}$ was generated randomly. According to Eq. (5), the new velocity ($v_{i,j}^{t+1}$) of each individual i at the jth dimension (X Y Z), is calculated based on the velocity of previous generation ($v_{i,j}^{t}$),

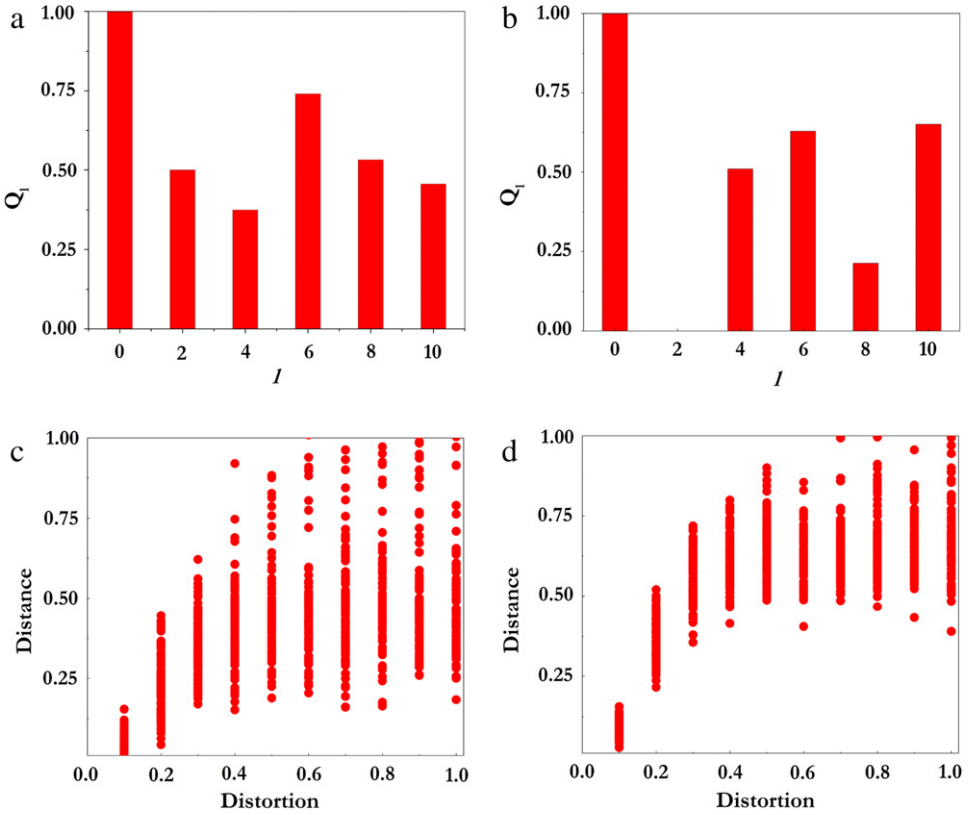

Fig. 2. (Color online) (a) and (b) Q_l histograms for graphite and diamond structures, respectively. (c) and (d) distance against distortion for graphite and diamond structures, respectively. The unit of distortion magnitude is in bond length.

Table 2
The structural search efficiency of CALYPSO calculations with or without the elimination of similar structures for the system of TiO_2. We have performed ten different CALYPSO runs and the total generations for these ten runs needed to find the global stable rutile structure are listed. As an illustration, we choose here the population size as 20. Notably, we generally use larger population sizes for larger systems; there much fewer generations are needed to find the stable structure. Other typical CALYPSO run parameters of $V_{\max}$ and the percentage of PSO generated structures are chosen as 0.1 and 0.6, respectively.

	Number of atoms in the system			
	12	24	36	48
	Generations	Generations	Generations	Generations
To eliminate similar structures	11	14	21	78
To preserve similar structures	10	17	47	118

its previous location ($x^t_{i,j}$) before structural optimization, current location ($pbest^t_{i,j}$) after structural optimization, and the population global location ($gbest^t_{i,j}$) with the best fitness value for the entire population. It is obvious that the velocity of PSO is different from the physical velocity. The velocity of PSO is generated by the atomic coordinates and other dimensionless parameters, so it has the same unit with the atomic position. It is noted that the velocity plays an important role in the determination of the speed and direction of structural movement.

$$v^{t+1}_{i,j} = \omega v^t_{i,j} + c_1 r_1 (pbest^t_{i,j} - x^t_{i,j}) + c_2 r_2 (gbest^t_{i,j} - x^t_{i,j}) \quad (5)$$

where $j \in \{X, Y, Z\}$, $^{\omega}$ denotes the inertia weight, c_1 and c_2 are self-confidence factor and swarm confidence factor. High settings of ω such as 0.9 facilitate global search, and lower settings facilitate rapid local search. In our methodology, ω is dynamically varied and decreases linearly from 0.9 to 0.4 during the iteration according to Eq. (6).

$$\omega = \omega_{\max} - \frac{\omega_{\max} - \omega_{\min}}{iter_{\max}} \times iter \quad (6)$$

where $\omega_{\max}$ and $\omega_{\min}$ equals 0.9 and 0.4, respectively. Accordingly, in our implementation, c_1 and c_2 are kept as a constant 2. r_1 and r_2 are two separately generated random numbers in the range 0 to 1. As shown in Eq. (5), it is quite obvious that the movement of particles in the search space is dynamically influenced by their individual past experience ($pbest^t_{i,j}$, $v^t_{i,j}$) and successful experiences attained by the whole swarm ($gbest^t$). Thus the velocity makes the particles move towards the global minimum and accelerates the convergence speed. The settings of other parameters will be presented in Section 3.

2.5. Penalty function

According to Bell–Evans–Polanyi principle [15,39], the low energy basins in potential energy surfaces are expected to occur near other low energy basins. Thus, in order to improve the efficiency of the procedure, a certain number of high-energy structures are rejected, and the remaining low energy structures, which are on the most promising areas of the configuration space, are selected to produce the next generation by PSO. Fig. 3(a) and (b) show the evolution of lattice energy distributions with and without the inclusion of penalty function during the simulation (shown here for TiO_2 with 48 atoms in the simulation cell). Obviously, most of structures are in the low-energy region ($\leq$620.0 eV) when

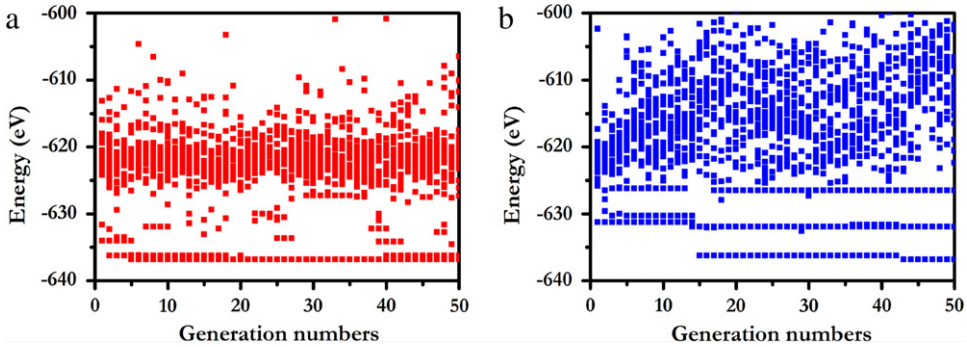

Fig. 3. (Color online) (a) and (b) represent the evolution of lattice energy distributions during structural iterations with and without the inclusion of penalty function, respectively.

the penalty function technique is included and it significantly accelerates the structural convergence to the global minimum as demonstrated in the CALYPSO runs (Fig. 3).

2.6. Structural diversity

Structural diversity plays an important role in the prediction of crystal structures by using the population-based methods, such as the genetic algorithm and our developed CALYPSO method. During the structural evolution, if the systems lose the structural diversity, it happens quite often that the systems stagnate, particularly for a large system. We here have designed a critical technique to enhance the structural diversity by including certain percentage of random structures in each generation, which has been implemented in CALYPSO code. Again, we use TiO_2 with 16 formula units per simulation cell as a test example. The history of CALYPSO runs with and without including the randomly generated structures is shown in Fig. 4. It is seen that the inclusion of a certain number of structures whose symmetries must be distinguished from any of previously generated ones is indeed crucial to converge the system to the global minimum. This all comes to the true fact that the inclusion of random structures allows the generation of diverse structures Table 3. Note that it might come up with the question of whether the global stable structure is in fact generated by those random structures. We have performed a certain number of tests and found out that only a few stable structures are generated randomly, especially for smaller systems. For most cases, the structural evolution of CALYPSO runs derives the global stable structures.

2.7. Convergence

The CALYPSO simulation is stopped when the halting criterion is reached. In accordance with our experience, the stable crystal structure can usually be found at $\sim$10 generations for systems $\leq$10 atoms per simulation cell. In practice, the halting criterion in CALYPSO is by default set to 10 further generations if the simulation can not find other better structures.

3. Optimization of parameters

In order to provide a reasonable default setting for various parameters in our CALYPSO code, a test was performed on a TiO_2 system with 16 formula units per simulation cell by using the GULP code for the structural optimization and total energy calculations. An earlier study [25] has demonstrated that $c_1 = c_2 = 2$ and the linear decrease of ω from 0.9 to 0.4 during the iteration usually give the best overall performance for PSO simulations. Thus, we adopt these parameters and other parameters such as the population size (N_{POP}), the proportion of the structures generated by PSO (P_{PSO})

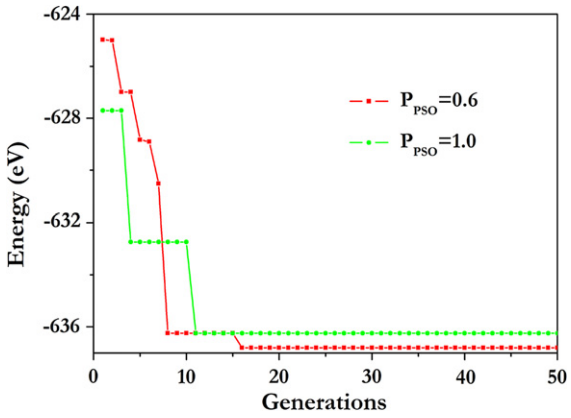

Fig. 4. (Color online) The history of CALYPSO search performed on TiO_2 with 48 atoms per cell. The red line represents the CALYPSO runs when a certain number of the low energy structures (0.6 of total) are selected to produce the next generation by PSO, while the rest of the structures are generated randomly. The green line represents when all the structures are used to generate the next generation by PSO. Note that the stable structure is produced by PSO in these calculations.

and the max magnitudes of the velocity (V_{max}) are determined by using the benchmark of TiO_2. We repeat 5 successful CALYPSO calculations, i.e., the correct finding of rutile structure, to derive the proper parameters. The results and suggested parameter values can be found in Table 4.

4. Input and output files

4.1. Input file

The main input file named as input.dat, contains all the necessary parameters for the simulation. There are several examples for the input.dat file in the Examples directory of the CALYPSO package.

We here take SiC as an example:

```
System Name = SiC
Number Of Species = 2
Name Of Atoms = C Si
Number Of Atoms = 1 1
Number Of Formula = 2 2
Atomic Number = 6 14
Max Step = 50
Volume = 20.0
@Distance Of Ion
1.2 1.5
1.5 1.9
@End
PsoRatio = 0.6
```

Table 3
The structural search efficiency of CALYPSO calculations with or without partial random structures per generation for the system of TiO_2. We have performed ten different CALYPSO runs and the total generations for these ten runs needed to find the global stable rutile structure are listed. As an illustration, we choose here the population size as 20. Notably, we generally use larger population sizes for larger systems; there much fewer generations are needed to find the stable structure. Another typical CALYPSO run parameter of V_{max} is chosen as 0.1.

P_{PSO}	Number of atoms for TiO_2			
	12	24	36	48
	Generations	Generations	Generations	Generations
0.6	12	15	85	110
1.0	11	23	124	229/9[a]

[a] It fails to find the global stable structure in 100 generations one time out of ten.

Icode = 1
Kgrid = 0.12 0.08
Command = vasp
PopSize = 20
PickUp = *F*
PickStep = 0

Here follows a description of the variables defined in the input file (input.dat), including the data types and default values.

SystemName (string): A string of one or several words contains a descriptive name of the system (max. 40 characters).

Default value: CALYPSO

Number Of Species (integer): Number of different atomic species.

Default value: No default.

Name Of Atoms (string): Element symbols of the different chemical species.

Default value: No default.

Atomic Number (integer): Atomic Number of each chemical species.

Default value: No default.

Number Of Atoms (integer): Number of atoms for each chemical species in one formula unit.

Default value: No default.

Number Of Formula (integer): The desired range of formula units per simulation cell. The first and second numbers are the lower and upper limits per simulation cell in the formula units.

Default value: 1 4

Volume (real): The volume per formula unit. Units are in Å3. The volume can be estimated by the atomic volume of given elements. If it is set to zero, the program will automatically generate the estimated volume by the radius of ions.

Default value: 0

@Distance Of Ion and @End (real): Minimal distances between different chemical species. Units are in angstrom. The determination of this parameter is in accordance with "Number Of Species". For example, if the Number Of Species = 2, $a2 \times 2$ matrix is used to indicate the minimal distances between different chemical species.

@Distance Of Ion
d11 d12
d21 d22
@End

Default value: 0.7 Å

Icode(integer): It determines which local optimization package should be interfaced within the simulation.

1: VASP
2: SIESTA
3: GULP
4: CASTEP

Default value: 1

PsoRatio (real): The proportion of the structures generated by PSO, and the other structures will be generated randomly.

Default value: 0.6

PopSize (integer): The population size. Normally, it will have a larger value for larger systems.

Default value: 30

Kgrid (real): The precision of the *K*-point sampling for local optimization (VASP or SIESTA). The Brillouin zone sampling uses a grid of spacing $2\pi \times$ Kgrid Å-1. The first value controls the precision of the first two local optimizations, and the second value with denser *K*-points controls the last optimization. The smaller value normally gives finer optimization results.

Default value: 0.12 0.06

Command (string): The command to perform local optimization on your computer.

Default value: submit.sh.

Max Step (integer): The maximum number of PSO iterations. It should have a larger value for a larger system.

Default value: 50

Pick Up (logical): If True, a previous calculation will be continued.

Default value: false

Pick Step (integer): At which step will the previous calculation be picked up.

Default value: There is no default. If Pick Up = True, you must supply this variable.

4.2. Output files

The main outputs of CALYPSO are in the "results" folder:

CALYPSO.log: It includes the information of the structures (the space group, the volume, the number of atoms, et al.).

similar.dat: It includes the bond characterization matrixes of predicted structures.

pso_ini_*: It includes the information of the initial structures of the *-th iteration step.

pso_opt_*: It includes the enthalpy and structural information after local optimization of the *-th iteration.

pso_sor_*: The enthalpy sorted in ascending order of the *-th iteration step.

5. Applications

We have earlier illustrated that the CALYPSO method can be used to predict various structures on elemental, binary and ternary compounds with various chemical bonding environments (e.g., metallic, ionic, and covalent bonding) [29,40,41]. Here, we discuss some other applications on the discovery of hitherto unknown structures. All the *ab initio* structure relaxations were performed using density functional theory within the projector augmented wave method, as implemented in the VASP code [33,34]. An overview of systems with unknown structures for which we have performed calculations and discovered new structures can be found in Table 5.

Lithium (Li) is a "simple" metal at ambient pressure, but exhibits complex phase transitions under compression. Experimentally, it has been demonstrated that Li takes the phase transition sequence of bcc → fcc→ hR1 → cI16, above which new phases

Table 4
The test of variable parameters in CALYPSO.

Test results						Suggested values
P_{PSO}	0.5	0.6	0.7	0.8	0.9	0.7–0.8
Generations	61/5	81/5	37/5	39/5	86/5	
V_{max}	0.05	0.1	0.2	0.3	0.4	0.1–0.2
Generations	32/5	31/5	27/5	37/5	31/5	
N_{POP}	10	20	30		40	30
Structures	1480/5	2000/5	840/51		600/5	

Table 5
Systems with unknown structures, for which we have done calculations and revealed new structures.

Systems	Pressure (GPa)	Structures	Generations	N_{pop}
Li	80	*Aba*2-40[a]	3	30
	200	*Cmca*-56[a]	4	30
Mg	500	fcc[b]	4	20
	800	sh[b]	5	30
Bi_2Te_3	12	β-Bi_2Te_3[c]	1	40
	14	γ-Bi_2Te_3[c]	5	30
	20	C2/m(bcc-like)[c]	2	40
BC_3	0	*Pmma*[d]	4	30
BC_7	0	*P*-4*m*2[e]	6	20

[a] Ref. [42].
[b] Ref. [43].
[c] Ref. [44].
[d] Ref. [45].
[e] Ref. [46].

are observed but remain unsolved [47]. We thus have extensively explored the high-pressure phases of Li through CALYPSO code. We successfully predicted all the experimental structures at certain pressure ranges by the CALYPSO method [29]. In particular, two new orthorhombic *Aba*2-40 (40 atoms/cell) and *Cmca*-56 (56 atoms/cell) structures of Li [42] were predicted at 80 and 200 GPa. These two complex structures (Aba2-40 and Cmca-56) are successfully predicted only at the third and fourth generation with a population size N_{pop} of 30, respectively. Note that *Aba*2-40 (oC40) structure has been later verified by an independent experiment [48].

Being a known best thermoelectric material and a topological insulator at ambient condition, bismuth telluride experiences phase transitions into several superconducting states under pressure. However, the high-pressure structures have remained unsolved since 1972. We have recently predicted two low-pressure phases of bismuth telluride through CALYPSO calculations as seven-fold (β-Bi_2Te_3) and eight-fold (γ-Bi_2Te_3) monoclinic structures at 12 GPa and 14 GPa, respectively [44]. These two structures were identified at the first and fifth generation with a population size of 30 and 40. These structures also have been subsequently verified by our experiment through Reitveld refinement [44]. Other compounds (Mg [43],BC_3 [45] and BC_7 [46]) with unknown structures also are discovered at high pressure by CALYPSO simulations Table 5. All the structures rapidly converge to the global minimum with fewer than 150 local optimizations. These results demonstrated that our method is a powerful and efficient tool on crystal structure determination.

The reason why our method is so successful can be traced to several powerful techniques. Firstly, PSO is a highly efficient global optimization algorithm, which has been applied successfully into many multi-objective optimization problems. Secondly, symmetry constraints on structural generation make the initial sampling to cover different regions of the search space, which is crucial for the efficiency of global minimization. Thirdly, the elimination of similar/identical structures using the bond characterization matrix technique and rejection of high-energy structures for each generation is able to accelerate the global structural convergence. Fourth, the inclusion of a certain number of structures whose symmetries are distinguished from previous ones can maintain population diversity and is critical to the prediction of global stable structures. Finally, the local optimization effectively reduces the noise of the landscape and may also be one of the key issues for our method's success.

6. Conclusions

In this paper, we outline descriptions of the implementation of CALYPSO code, which can be used to predict crystal structures of materials at given chemical compositions and external conditions. Our CALYPSO method has incorporated several major techniques (e.g. the PSO algorithm, symmetry constraints on structural generation, bond characterization matrix on the elimination of similar structures, partial random structures per generation on enhancing structural diversity, and the penalty function, etc.), which have been demonstrated to be crucial to the prediction of globally stable structures. Suggested values for various parameters in CALYPSO have been presented by performing a benchmark on the TiO_2 system. The high success rate and high efficiency of the structural searches of the CALYPSO methodology have demonstrated its reliability and promise as a major tool in crystal structure determination.

Algorithm 1. The pseudo-code of the implementation of CALYPSO.

```
Number of particles, N; swarm, S; volume, V; Percentage of
PSO generated structures, P_PSO.
  Initialization of S (Generation of random structures
   with constraint of symmetry)
  Evaluation of S (Local optimization) and definition of the
   pbest and gbest
  List of the bond characterization matrixes (BCM)
  While not done do
    S_PSO = S* P_PSO and S_random = S* (1 − P_PSO)
    While i <= S_PSO do
      S(i) (Generation of new structures by PSO)
      If S(i)∉BCM then
        i = i + 1
        To update the list of BCM
      End if
    End while
    While i <= S_PSO + S_random
      S(i) Generation of random structures with
       constraints of symmetry
      If S(i)∉BCM then
        i = i + 1
        To update the list of BCM
      End if
    End while
    To Evaluate S (local optimization) and update the gbest
    To update the list of BCM
  End while
```

Program availability.

CALYPSO is available via http://nlshm-lab.jlu.edu.cn/~calypso.html The software is free of charge for non-profit organizations, and delivered with the Fortran source code. The details of installation instructions, the user's manual in PDF format and examples are included in the package.

Acknowledgment

The authors acknowledge the funding support from the National Natural Science Foundation of China under grant Nos. 11025418 and 91022029.

References

[1] J. Feng, W. Grochala, T. Jaronacute, R. Hoffmann, A. Bergara, N.W. Ashcroft, Phys. Rev. Lett. 96 (2006) 017006.
[2] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, Science 220 (1983) 671.
[3] J. Pannetier, J. Bassas-Alsina, J. Rodriguez-Carvajal, V. Caignaert, Nature 346 (1990) 343–345.
[4] M. Amsler, S. Goedecker, J. Chem. Phys. 133 (2010) 224104.
[5] J. David, J.P.K. Doye, J. Phys. Chem. A 101 (1997) 5111–5116.
[6] A. Laio, A. Rodriguez-Fortea, F.L. Gervasio, M. Ceccarelli, M. Parrinello, J. Phys. Chem. B 109 (2005) 6714–6721.
[7] S.M. Woodley, P.D. Battle, J.D. Gale, C.R.A. Catlow, Phys. Chem. Chem. Phys. 6 (2004) 1815–1822.
[8] N. Abraham, M. Probert, Phys. Rev. B 73 (2006) 224104.
[9] C.W. Glass, A.R. Oganov, N. Hansen, Comput. Phys. Commun. 175 (2006) 713–720.
[10] D.C. Lonie, E. Zurek, Comput. Phys. Commun. 182 (2011) 372–387.
[11] G. Trimarchi, A. Zunger, Phys. Rev. B 75 (2007) 104113.
[12] Y. Yao, J.S. Tse, K. Tanaka, Phys. Rev. B 77 (2008) 052103.
[13] D. Deaven, K. Ho, Phys. Rev. Lett. 75 (1995) 288–291.
[14] J.A. Niesse, H.R. Mayne, J. Chem. Phys. 105 (1996) 4700.
[15] C.J. Pickard, R. Needs, J. Phys.: Condens. Matter 23 (2011) 053201.
[16] Y. Ma, M. Eremets, A.R. Oganov, Y. Xie, I. Trojan, S. Medvedev, A.O. Lyakhov, M. Valle, V. Prakapenka, Nature 458 (2009) 182–185.
[17] Y. Ma, A.R. Oganov, Z. Li, Y. Xie, J. Kotakoski, Phys. Rev. Lett. 102 (2009) 65501.
[18] Y. Ma, Y. Wang, A.R. Oganov, Phys. Rev. B 79 (2009) 054101.
[19] Q. Li, Y. Ma, A.R. Oganov, H. Wang, Y. Xu, T. Cui, H.K. Mao, G. Zou, Phys. Rev. Lett. 102 (2009) 175506.
[20] C.J. Pickard, R. Needs, Phys. Rev. Lett. 97 (2006) 45504.
[21] C.J. Pickard, R. Needs, Nature Mater. 7 (2008) 775–779.
[22] C.J. Pickard, R. Needs, Phys. Rev. Lett. 102 (2009) 146401.
[23] C.J. Pickard, R.J. Needs, Nature Phys. 3 (2007) 473–476.
[24] J. Kennedy, R.C. Eberhart, A Discrete Binary Version of the Particle Swarm Algorithm, vol. 4105, IEEE, 1997, pp. 4104–4108.
[25] R.C. Eberhart, Y. Shi, Particle Swarm Optimization: Developments, Applications and Resources, IEEE, Piscataway, NJ, USA, 2001, pp. 81–86.
[26] R. Mendes, P. Cortez, M. Rocha, J. Neves, Particle swarms for feedforward neural network training, in: Neural Networks, IJCNN '02, in: Proceedings of the 2002 International Joint Conference on, IEEE, 2002, 2002, pp. 1895–1899.
[27] M. Meissner, M. Schmuker, G. Schneider, BMC Bioinformatics 7 (2006) 125.
[28] H. Yoshida, K. Kawata, Y. Fukuyama, S. Takayama, Y. Nakanishi, Power Systems, IEEE Transactions on 15 (2000) 1232–1239.
[29] Y. Wang, J. Lv, L. Zhu, Y. Ma, Phys. Rev. B 82 (2010) 094116.
[30] J.D. Gale, J. Chem. Soc., Faraday Trans. 93 (1997) 629–637.
[31] S. Woodley, C. Catlow, Comput. Mater. Sci. 45 (2009) 84–95.
[32] J.R. Fernandez, P. Harrowell, J. Chem. Phys. 120 (2004) 9222.
[33] G. Kresse, J. Furthmuler, Comput. Mater. Sci. 6 (1996) 15–50.
[34] G. Kresse, J. Furthmuler, Phys. Rev. B 54 (1996) 11169.
[35] J.M. Soler, E. Artacho, J.D. Gale, A. Garcia, J. Junquera, P. Ordejon, D. Sachez-Portal, J. Phys.: Condens. Matter 14 (2002) 2745.
[36] S.J. Clark, M.D. Segall, C.J. Pickard, P.J. Hasnip, M.I.J. Probert, K. Refson, M.C. Payne, Z. Kristallogr. 220 (2005) 567–570.
[37] M. Segall, P.J.D. Lindan, M. Probert, C. Pickard, P. Hasnip, S. Clark, M. Payne, J. Phys.: Condens. Matter 14 (2002) 2717.
[38] P.J. Steinhardt, D.R. Nelson, M. Ronchetti, Phys. Rev. B 28 (1983) 784.
[39] F. Jensen, Introduction to Computational Chemistry, Wiley, 2007.
[40] X. Luo, J. Yang, H. Liu, X. Wu, Y. Wang, Y. Ma, S.H. Wei, X. Gong, H. Xiang, J. Am. Chem. Soc. 133 (2011) 16285–16290.
[41] H. Xiang, B. Huang, Z. Li, S. Wei, J. Yang, X. Gong, Phys. Rev. X 2 (2012) 011003.
[42] J. Lv, Y. Wang, L. Zhu, Y. Ma, Phys. Rev. Lett. 106 (2011) 15503.
[43] P. Li, G. Gao, Y. Wang, Y. Ma, J. Phys. Chem. C 114 (2010) 21745–21749.
[44] L. Zhu, H. Wang, Y. Wang, J. Lv, Y. Ma, Q. Cui, G. Zou, Phys. Rev. Lett. 106 (2011) 145501.
[45] H. Liu, Q. Li, L. Zhu, Y. Ma, Phys. Lett. A 375 (2011) 771–774.
[46] H. Liu, Q. Li, L. Zhu, Y. Ma, Solid State Commun. 151 (2011) 716–719.
[47] T. Matsuoka, K. Shimizu, Nature 458 (2009) 186–189.
[48] C.L. Guillaume, E. Gregoryanz, O. Degtyareva, M.I. McMahon, M. Hanfland, S. Evans, M. Guthrie, S.V. Sinogeikin, H. Mao, Nature Phys. 7 (2011) 211–214.