Computational Materials Science 149 (2018) 416–423

Contents lists available at ScienceDirect

Computational Materials Science

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

Cluster-based niching differential evolution algorithm for optimizing the stable structures of metallic clusters

Yuan-Hua Yang[a], Xian-Bin Xu[a], Shui-Bing He[a], Jin-Bo Wang[b], Yu-Hua Wen[b],*

[a] *School of Computer, Wuhan University, Wuhan 430072, China*
[b] *Department of Physics, Xiamen University, Xiamen 361005, China*

ARTICLE INFO

Keywords:
Cluster
Niching differential evolution
Cluster pool
Structural optimization

ABSTRACT

In this article, a cluster-based niching differential evolution algorithm, which combines the cluster pool, the niche method, and the differential evolution algorithm, has been employed to optimize the stable structures of iron clusters. The cluster pool is responsible for generation of the niche sub populations, and the differential evolutionary algorithm is used for the evolution of the population. A variety of mutation strategies have been applied in the algorithm instance. Moreover, the crossover operator of plane cut cross and the adjustment strategy make the algorithm more suitable for structural optimization of clusters. Subsequently, the performance of the algorithm has been examined by the effect of cluster pool size on the convergence speed and structural diversity. The accuracy and effectiveness of our algorithm have been verified by analyses of energy and structural evolutions. Finally, structural evolution of iron clusters with 3–80 atoms has been predicted by this algorithm.

1. Introduction

Metallic clusters have attracted great attention due to their potential applications in many fields such as physics, chemistry, biology and so on [1–4]. Among metallic clusters, iron (Fe) clusters are of considerable interest due to their exceptional magnetic properties such as ferromagnetism, high coercive force, low Curie temperature, high magnetic susceptibility. To date, Fe clusters have been extensively used in the aspects of giant magnetoresistance, magnetic recording, magnetic refrigeration, and magnetic probes [5]. As cheap metallic catalysts, Fe clusters have been widely used in Fischer-Tropsch reaction for producing hydrocarbon by using CO and H_2 in coal and natural gas. They can also be used as a cathode catalyst for fuel cell [6]. However, both the magnetic and catalytic properties of Fe clusters are strongly dependent on their structures. Therefore, an investigation on the structural properties of Fe clusters is crucial for understanding their physical and chemical performances.

Theoretically, to predict the structure of clusters is a typical global optimization problem. The optimization goal is to get the lowest-energy structure of clusters [7]. Essentially, exploring the stable structures of Fe clusters is to search the lowest energy of potential energy function. Usually, the potential energy function describes a potential energy surface of multi-dimensional space. The potential energy surface is considerably complex, thus searching the lowest energy on the potential energy surface is rather time-consuming. Furthermore, there are plenty of local minimum corresponding to metastable structures of cluster on the potential energy surface, the number of local minimum grows exponentially with the cluster size [8]. So far, many global optimization methods, such as heuristic algorithms and evolutionary algorithms, have been developed to optimize the structure of clusters. According to the number of individuals in the searching process, the algorithms can be divided into three categories: single individual searching algorithms, single population searching algorithms, and multi populations searching algorithms. Single individual searching algorithms, such as Monte Carlo method [9], Basin Hopping algorithm [10,11], simulated annealing algorithm [12], belong to simple searching algorithms. The search efficiency of these algorithms is poor due to the lack of repeatable search. Single population searching algorithms, such as genetic algorithm [13,14], particle swarm optimization algorithm [15,16], and artificial immune algorithm [17,18], are superior in comparison with the single individual searching algorithms because there exists the information exchange between different individuals in single population searching algorithms. However, they are apt to be trapped into the local optimum, leading to the premature convergence. Multi populations searching algorithms, including common pool [19], topology structure [20], and niche method [21], are able to improve the search capability of global optimization remarkably because they may maintain population diversity effectively and avoid

* Corresponding author.
E-mail addresses: yangyuanhua123@163.com (Y.-H. Yang), xbxu@whu.edu.cn (X.-B. Xu), yhwen@xmu.edu.cn (Y.-H. Wen).

https://doi.org/10.1016/j.commatsci.2018.03.055
Received 18 January 2018; Received in revised form 23 March 2018; Accepted 23 March 2018
Available online 28 March 2018

the premature phenomena in single population searching algorithms.

In this article, we have proposed, for the first time, a cluster-based niching differential evolution algorithm to optimize the structure of Fe clusters by the multi populations searching algorithms with cluster-based niching method. As a first step, the effect of cluster pool size on convergence speed of algorithm and structural diversity has been analyzed. Secondly, the accuracy and effectiveness of the proposed algorithm have been verified by comparison experiments. Finally, we have examined the stable structures of Fe clusters with 3–80 atoms by using the proposed algorithm, and predict the evolutional law of stable structures with increasing cluster size. This article is structured as follows. Section 2 describes the potentials of Fe and the cluster-based niching differential evolution algorithm. Section 3 presents the calculated results and discussion. The main conclusions are summarized in Section 4.

2. Methodology

2.1. Potential description

In theoretical study of clusters, it is considerably important to accurately describe the interatomic interaction. In this work, the Finnis-Sinclair (FS) potentials [22], which are based on the second-moment approximation of the tight-bonding formulation, have been employed to describe the interaction between atoms in Fe clusters. The FS potentials represent many-body interactions, and their parameters are optimized to describe the lattice parameter, cohesive energy, elastic constants, vacancy formation energy, stacking-fault energy, and pressure-volume dependency. They have been confirmed to reproduce very well the basic structural and dynamics properties of Fe [23]. The total energy for a system of N atoms is given as

$$E_{tot} = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} V_{ij}(r_{ij}) - A\sum_{i=1}^{N}\sqrt{\rho_i}, \tag{1}$$

where r_{ij} represents the distance between atoms i and j; ρ_i is the electronic charge density at the site of atom i, it can be expressed by

$$\rho_i = \sum_{j=1, i\neq j}^{N} \phi_{ij}(r_{ij}), \tag{2}$$

in which $\phi_{ij}(r_{ij})$ is a cohesive term related to the sum of squares of overlap integrals for the valance electrons, represents the contribution of electronic charge density for j atom to i atom, can be defined as

$$\Phi(r) = \begin{cases} (r-d)^2 & r \leqslant d \\ 0 & r > d \end{cases}, \tag{3}$$

where d is a cut-off parameters assumed to lie between the second- and third-neighbors, the value of d is$a < d < \sqrt{2}a$, a is a lattice constant. For Fe element, the expression of $\Phi(r)$ can be modified by

$$\Phi(r) = (r-d)^2 + \frac{\beta}{d}(r-d)^3, \tag{4}$$

where the range of these parameters should be set to enable the $\Phi(r)$ reaching the maximum in first- nearest-neighbor. For example, if $d = a$, then $\beta < 4.975$; otherwise, $d = \sqrt{2}a$, then $\beta < 1.7199$.

In Eq. (1), $V(r)$ is a repulsive two-body interaction, interpreted in the tight-binding theory as the repulsion between core electrons on neighboring atoms, expressed as

$$V(r) = \begin{cases} (r-c)^2(c_0 + c_1 r + c_2 r^2) & r \leqslant c \\ 0 & r > c \end{cases} \tag{5}$$

where c is a cut-off parameter, just like the parameter d; c_0, c_1, and c_2 are free parameters fitting to experimental data based on specific elements. All parameters of FS potentials for Fe have been listed in Table 1.

2.2. Transformation for potential energy surface

Actually, the potential function of a cluster corresponds to a complicated potential energy surface in hyperspace. The potential energy surface describes the relationship between the cluster energy and the relative position of each atom in the cluster. Therefore, investigating the stable structure of a cluster by minimizing its total energy is to search the global minimum of the potential energy surface. However, it is difficult to find the lowest energy on the potential energy surface directly even for a system consisting a few atoms due to the complexity of potential energy surface. To reduce the search space and improve the searching efficiency, in this work we transform the potential energy surface into many less-intricate basins by employing a local minimization for the structures [24].

Since the potential energy surface is a curve in the multi-dimensional space, it is impossible to directly depict the surface. Here, we display the transformation diagram of the potential energy surface in two-dimensional surface in Fig. 1. We may transform the complicated potential energy surface into less-intricate basins by local minimization procedure. The transformation not only avoids the unstable transition state on potential energy surface, but also decreases the energy barrier. It makes the system freely go through the basin boundaries of potential energy, therefore simplifies the optimization process.

2.3. Cluster-based niching differential evolution algorithm

2.3.1. Structural optimization of a cluster

The structural optimization of a cluster can be described as follows. For a cluster consisting of N atoms, the total energy of all atoms is defined as its potential energy. When the potential energy reaches its minimum, the search target is the atom coordinates in the three-dimensional space.

The objective function of the structural optimization is the potential energy according to Eq. (1). It can be described as

$$\min f = \min E_{tot}(R) \tag{6}$$

where E_{tot} is the potential energy of a system, R represents the atomic distance matrix, and is given by

$$R = \begin{bmatrix} r_{11} & \cdots & r_{1N} \\ \vdots & \ddots & \vdots \\ r_{N1} & \cdots & r_{NN} \end{bmatrix} \tag{7}$$

$$r_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \tag{8}$$

x_i, y_i, and z_i denote the coordinates of atom i in three-dimensional space. Since the interatomic distance is relative, it means $r_{ij} = r_{ji}$. Therefore, the distance matrix R is a symmetric matrix, and the diagonal values are zero because the distance between an atom and itself is zero ($r_{ii} = 0$).To make the general multi-populations differential evolution algorithm more effectively during the structural optimization of cluster, in the base of primal algorithms [25,26], we have proposed a cluster-based niching differential evolution algorithm by combining the cluster pool, the niche method, and the differential evolution algorithm instance, as shown in Fig. 2.

2.3.2. Cluster pool

Essentially, the cluster pool is a collection of many different clusters (individuals) with the same atomic number. The cluster pool is used to keep the individual diversity during structural optimization of clusters. Moreover, it is a great solution for clustering and distinguishing the clusters. The initialization, clustering, and update of the cluster pool play a significant role in the whole algorithm. The three procedures have been described in detail below.

2.3.2.1. Initialization of cluster pool. Consider a cluster consisting of N atoms, the atom coordinates are represented by three dimensional

Table 1
Parameters of FS potentials for Fe [22].

Parameter	d [Å]	A [eV]	β	c [Å]	c_0	c_1	c_2
Fe	3.569745	1.828905	1.8	3.40	1.2371147	−0.3592185	−0.0385607

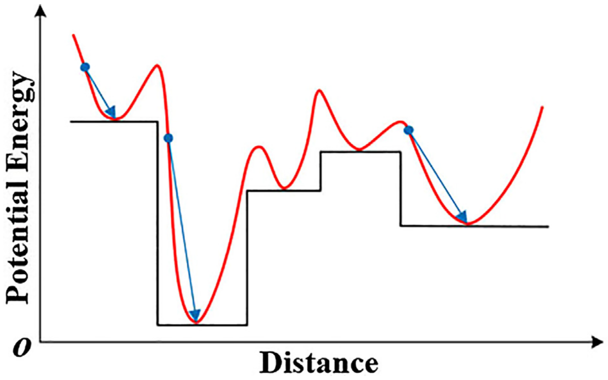

Fig. 1. Transformation diagram of potential energy surface.

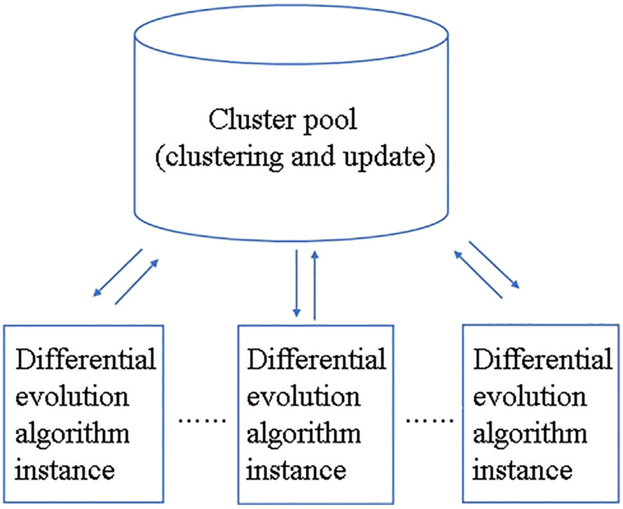
Cluster pool
(clustering and update)
Differential evolution algorithm instance
......
Differential evolution algorithm instance
......
Differential evolution algorithm instance

Fig. 2. Schematic diagram of cluster-based niching differential evolution algorithm.

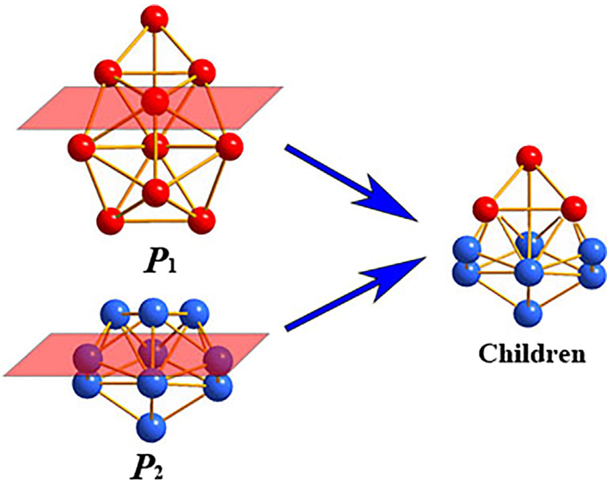

Fig. 3. Generation of plane-cut-splice crossover.

vectors x, y, z. Herein, an individual (cluster) can be denoted by a vector of 1 × 3N. The position of an individual is described as

$$X^0_{j,p} = [x_{1,j,p},x_{2,j,p},...,x_{3N,j,p}], \tag{9}$$

where $X^0_{j,p}$ represents an individual of generation 0, $j = 1,...,N_p$ is the index of the individual in sub-population p, N_p is the maximum number of sub-population p, $p = 1,...,P$ is the index of sub-population, P represents the number of sub-populations.

In the initialization of cluster pool, two methods are usually adopted to generate the individuals. One method is a random initialization method in which all atomic coordinates are produced by initializing the values of x, y, z randomly. To avoid the considerable variation of the generated individuals, we set the values of x, y, z in the range $[0,\mathrm{r}^0{\cdot}N^{\frac{1}{3}}]$. The individuals are randomly generated by

$$x_{i,j,p} = rand\,[0,1) \times r^0 \times N^{1/3} \tag{10}$$

where r^0 is the first-neighbor distance, N denotes the cluster size (the number of atoms). The cluster volume scales correctly with the first-neighbor distance and cluster size. The other method is a growing initialization method to generate the individuals. In the growing method, a new individual is generated based on the lowest-energy structure of the cluster with $N - 1$ atoms plus one additional atom. The additional atom position is randomly chosen from the range $[0,\mathrm{r}^0_{\mathrm{ij}}{\cdot}N^{\frac{1}{3}}]$. This initialization method makes use of the obtained stable structures of small-sized cluster to generate the initial structures of large-sized cluster. It can offer a reasonable structure with relatively low energy and reduce the searching space. However, the obtained structures are very similar if all individuals are generated by the growing method, leading to the premature convergence of the algorithm. To ensure the diversity of the population and prevent the optimization from prematurity, only some individuals are generated by the growing method, while most of the individuals are still generated by the random initialization.

2.3.2.2. Clustering of cluster pool. The clustering is an important step in the cluster pool. According to structural characteristic of clusters, they are classified based on their structural differences despite the difference of atomic position or potential energy. The structure difference can be denoted by the distance of clusters (individuals) on the potential energy surface, and the clustering method puts the adjacent clusters together on the potential energy surface. All individuals can be divided into some niches according to the structure difference, and the individuals in the same niche have similar structures. Here, we adopt the K-means method [27] to generate the niches. The selection of starting center points and the number K of clustering have remarkable effects on the constringency speed of the algorithm and the performance of clustering. We choose the low-energy individuals in the cluster pool for the selection of starting center points, and select the large clustering number K at the beginning. As the algorithm evolves, we choose the small clustering number K because some high-energy individuals have been eliminated and the clusters are concentrated in a smaller range.

2.3.2.3. Update of cluster pool. The update of cluster pool has been carried out to eliminate the worst individual. First, a group of new individuals will be produced when each evolution has finished. Secondly, the new individuals will be classified to the corresponding classes according to their structure differences. Finally, the individuals in the class are listed according to their potential energy and the highest-energy (worst) individual will be eliminated. By eliminating the

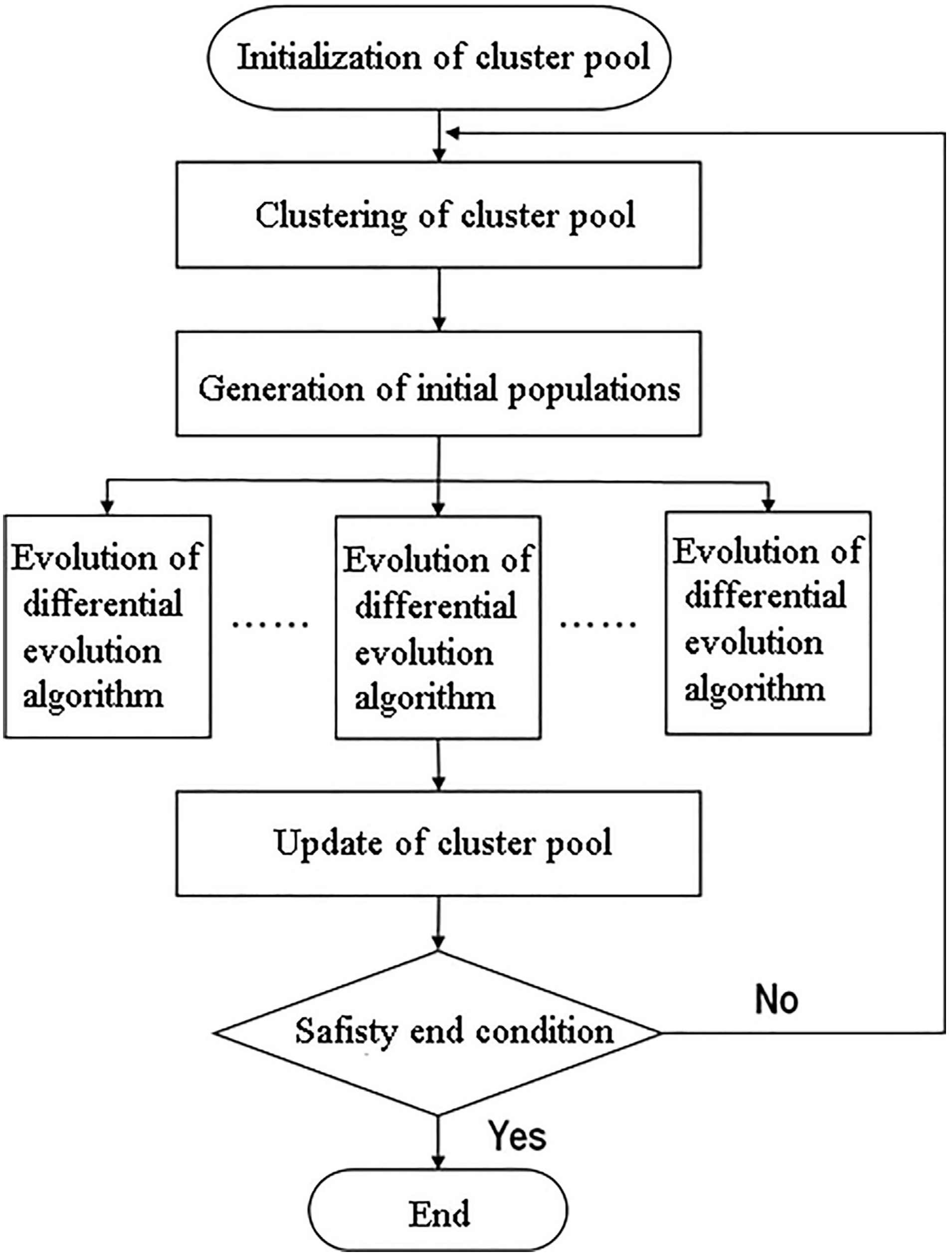
Initialization of cluster pool
Clustering of cluster pool
Generation of initial populations
Evolution of differential evolution algorithm
......
Evolution of differential evolution algorithm
......
Evolution of differential evolution algorithm
Update of cluster pool
Safisty end condition
No
Yes
End

Fig. 4. Flow chart of the cluster-based niching differential evolution algorithm.

worst individual, the overall number of individuals keeps unchangeable in the cluster pool but the quality of individuals in the pool is improved.

2.3.3. *Differential evolution algorithm*

The differential evolution algorithm consists of five main parts including generation of initial populations, mutation, crossover, adjustment and selection. They are described in detail as follows.

2.3.3.1. Generation of initial populations. The initial populations for evolution are from the niches, and are produced in the clustering procedure of cluster pool. Two methods are used for the generation of initial populations. One is that all initial populations for evolution are from the same niches. The aim of this method is to search the lowest-energy individual in the niche. The chance which each individual is selected in the niche is associated with its potential energy. The low-energy individual will have a preferential chance to be selected for evolution. This section can be called as roulette or championship. The other method is that some initial populations are from the same niche while the remaining populations are from other niches. This method is to search the lowest-energy individual in different energy region or in a new energy region since different niche represents different energy region.

2.3.3.2. Mutation. Different mutation strategies are adopted for different differential evolution algorithms. Three common mutation strategies, random mutation (*rand/1*), best mutation (*Best/1*), and random to best mutation (*Rand to Best/1*) are descripted as follows.

Random mutation (*Rand/1*):

$$V_{j,p}^{t} = X_{r3,p}^{t} + F(X_{r1,p}^{t} - X_{r2,p}^{t}) \tag{11}$$

best mutation (*Best/1*):

$$V_{j,p}^{t} = X_{best,p}^{t} + F(X_{r1,p}^{t} - X_{r2,p}^{t}) \tag{12}$$

random to best mutation (*Rand to Best/1*):

$$V_{j,p}^{t} = X_{r3,p}^{t} + F(X_{best,p}^{t} - X_{r3,p}^{t}) + F(X_{r1,p}^{t} - X_{r2,p}^{t}), \tag{13}$$

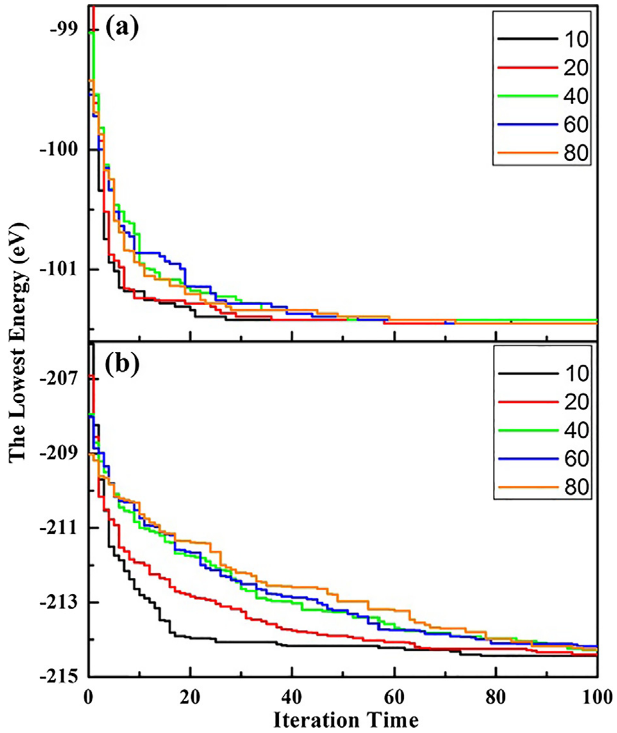

Fig. 5. Evolutionary curves of Fe clusters with (a) 30 and (b) 60 atoms for different cluster pool sizes.

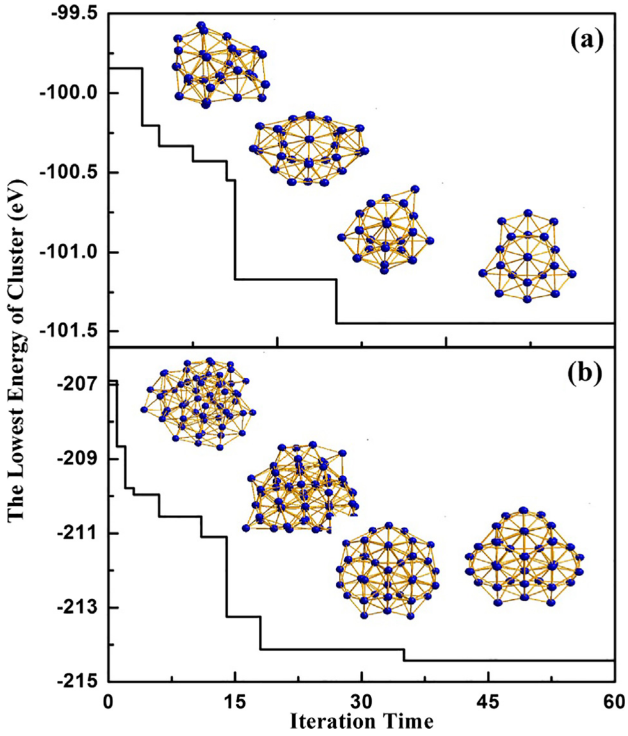

Fig. 6. Evolution curve of clusters with (a) 30 and (b) 60 atoms.

where $j = 1,\ldots,N_p$; $r1,r2,r3 \in [1,\ldots,N_p]r1,r2,r3 \in [1,\ldots,N_p]$, and $r1 \neq r2 \neq r3 \neq j$; F is a scaling factor and $F \in [0,2]$. The three mutation strategies are employed in different evolution algorithms because of theirself advantages. The random mutation (*Rand/1*) improves the global searching capability of different evolution algorithm but its convergence speed is relatively slow. The best mutation (*Best/1*) has the advantages of strong local searching capacity and fast convergence speed while it increases the risk of trapping in local optimum. The third mutation (*Rand to Best/1*) balances the global searching and local searching capacity by combining the different strengths of the previous two mutations.

2.3.3.3. Crossover. Crossover is a key step in the different evolution algorithm instance because it decides the performance of the different evolution algorithm. Here, the classical plane-cut-splice crossover is employed due to its strong searching capacity. Assuming two parent individuals are noted as $P_1 \in X_{i,p}$ and $P_2 \in V_{i,p}$, the plane-cut-splice crossover operator first shifts the mass centers of P_1 and P_2 to the origin, and then randomly chooses a plane to pass through the origin. The plane splits each parent into two parts (S_1 and S_2). If S_1 of each parent has the same number of atoms (for example, the number of atoms for S_1 is L, then that for S_2 is N-L), we exchange the parts S_1 (one from P_1, the other from P_2) to form two child clusters, denoted as $U_{i,g}$. If S_1 of each parent has different number of atoms, the plane passed through the origin will be rotated until it splits the parents to two parts with the same number of atoms for each parent. The procedure of plane-cut-splice crossover is shown in Fig. 3. The crossover probability in the algorithm is denoted as C_r. The energies of new generated individuals are locally minimized before they enter the population.

2.3.3.4. Adjustment and selection. The adjustment operator is used to modify the broken structures produced in crossover. Since some atoms far from the mass center may evaporate in plane-cut-splice crossover, the atomic interaction is not enough to pull the evaporated atoms back to the cluster system. The adjustment strategy is used to adjust the positions of evaporated atoms and keeps each individual to be searched by neighborhood searching approach. Furthermore, a greedy selection is employed to determine whether the new generated individuals survive to the next generation or not. Here, the potential energy of cluster is adopted as the fitness function. The lower-energy individual is more likely to survive to the next generation. The greedy selection is described as

$$X_{i,g+1} = \begin{cases} U_{i,g} \text{if } E(U_{i,g}) \leqslant E(X_{i,g}) \\ X_{i,g} \text{ otherwise} \end{cases} \tag{14}$$

where $X_{i,g}$ represents the parent individual, $U_{i,g}$ is the new generated individual in crossover operator. If the energy of $U_{i,g}$ is lower than that of $X_{i,g}$, $U_{i,g}$ will replace the parent$X_{i,g}$, as a new solution surviving to next generation. Otherwise, $X_{i,g}$ is kept to the next generation.

The complete cluster-based niching differential evolution algorithm is composed of cluster pool and the differential evolution algorithm instance. The procedure has been shown in Fig. 4. The main steps of the algorithm are given as follows.

Step 1: Initialization of cluster pool. The initial individuals in the cluster pool are generated by a random initialization or a growing method. Each initial individual is locally optimized in the cluster pool.

Step 2: Clustering of cluster pool. By adopting the K-means clustering method, all individuals in the pool are classified based on their structure differences. Each category represents a niche. The low-energy individuals in the pool are selected as starting center points.

Step 3: Generation of initial populations for evolution. Each differential evolution algorithm example has to generate initial populations before evolution. Two methods are applied to obtain the initial populations. One is that all initial populations are from the same niche, the other is that some initial populations are from the same niche while the other populations are from other niches.

Step 4: Evolution. The evolution of differential evolution algorithm consists of mutation, crossover and selection. The new

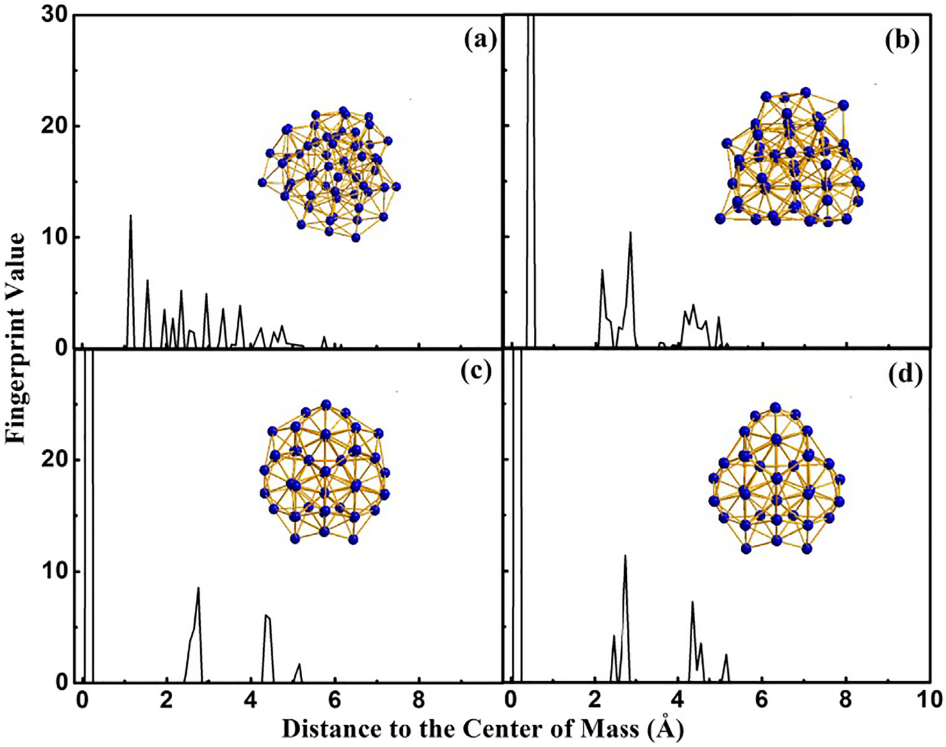

Fig. 7. Fingerprint vector change for structures of Fe_{60} cluster.

Table 2
Global minima energy and space groups (SG) of Fe clusters.

N	Energy (eV)	SG	N	Energy (eV)	SG	N	Energy (eV)	SG
3	−5.3985	D_{3h}	29	−97.9671	D_{3h}	55	−194.6868	C_s
4	−8.7233	T_d	30	−101.4513	C_{2v}	56	−198.7520	C_{1v}
5	−11.8598	D_{3h}	31	−104.9369	C_s	57	−202.9373	T_d
6	−14.9990	O_h	32	−108.8106	C_{2v}	58	−206.8059	C_{3v}
7	−18.5493	D_{5h}	33	−112.2849	C_s	59	−210.6376	C_{2v}
8	−21.5183	C_s	34	−116.0135	D_{5h}	60	−214.4278	C_{3v}
9	−25.0658	C_{2v}	35	−119.5971	C_2	61	−218.1707	T_d
10	−28.5357	C_{3v}	36	−123.1219	C_s	62	−221.5913	C_s
11	−31.9849	C_{2v}	37	−126.8454	C_s	63	−225.3541	C_s
12	−35.8818	C_{5v}	38	−130.9645	D_{6h}	64	−229.0662	C_{3v}
13	−40.2985	I_h	39	−134.8397	C_{6v}	65	−232.5331	C_{2v}
14	−43.1772	C_{3v}	40	−138.5112	D_{6h}	66	−236.2673	C_s
15	−46.6375	C_{2v}	41	−141.8420	C_s	67	−240.0078	C_1
16	−50.0240	C_s	42	−145.5473	C_s	68	−243.8329	C_s
17	−53.4553	C_s	43	−149.4047	C_s	69	−247.5954	C_s
18	−57.2367	C_s	44	−153.1767	C_{2v}	70	−251.3606	C_s
19	−61.5615	D_{5h}	45	−156.6997	C_s	71	−254.9198	C_s
20	−64.8385	C_{2v}	46	−160.3679	C_s	72	−259.0745	C_s
21	−68.1871	C_1	47	−164.1621	C_s	73	−262.9092	C_1
22	−71.8589	C_s	48	−168.1839	C_{3v}	74	−266.6987	C_s
23	−76.0972	D_{3h}	49	−171.8014	C_s	75	−270.7753	C_s
24	−79.3414	C_{2v}	50	−175.4720	C_1	76	−274.9962	D_{6h}
25	−82.9402	C_{2v}	51	−179.2199	C_s	77	−278.9273	C_{6v}
26	−87.0660	T_d	52	−183.1521	C_s	78	−282.8276	D_{6h}
27	−90.5422	C_{2v}	53	−187.0497	C_s	79	−286.4929	C_s
28	−93.9368	C_s	54	−190.9403	D_{3h}	80	−290.2863	C_2

individuals, which are generated from mutation and crossover operators, have to be optimized to their local minima. By the greedy selection, the new generated individuals will be determined to save in the cluster pool or not.

Step 5: Update of cluster pool. The elimination of the worst individual is carried out for updating the cluster pool after the generation of new individuals. Based on the structural differences, the new individuals are classified to the corresponding category and the highest-energy individual will be eliminated.

Step 6: Termination decision. After finishing the above steps, the algorithm checks if the result satisfies the end condition. Here, the end condition is the maximum iterations. If all differential evolution algorithm examples arrive at the maximum iterations, the program will terminate, otherwise the program will turn back to Step 2 and continue.

3. Results and discussion

3.1. Effect of cluster pool size on algorithm

In the cluster-based niching differential evolution algorithm, the choice of cluster pool size has a significant influence on algorithm performance because it greatly affects the structural diversity of cluster and the convergence of the algorithm. To analyze the effect of cluster pool size on the algorithm convergence, we first compare the convergence speed of the algorithm at different cluster pool size (10, 20, 40, 60, and 80 are chosen for comparison). For each cluster pool size, ten times independent experiments have been performed for Fe clusters with 30 and 60 atoms. For accurately analyzing the effect of cluster pool size, we adopt the random initialization method to generate the initial individuals rather than the growing method. The reason is that the algorithm may be close to the optimal solution with fewer iteration step in the growing method though it can improve the quality of initial individuals, which is not convenient for observing and comparing the effect of cluster pool size.

Fig. 5 illustrates the evolutionary processes of energy in the algorithm for each cluster pool size. One can find that for both Fe_{30} and Fe_{60} clusters, the convergence speed is gradually decreased with the cluster pool size. With the increasing number of individuals in the cluster pool (*i.e.*, increasing cluster pool size), the individuals with poor quality also rise, leading to the result that they have a high probability to be selected for differential evolution algorithm examples. Accordingly, the convergence speed is also decreased. In contrast, the convergence speed is fast for small cluster pool size but the structural diversity is decreased. Meanwhile, the algorithm is prone to premature convergence

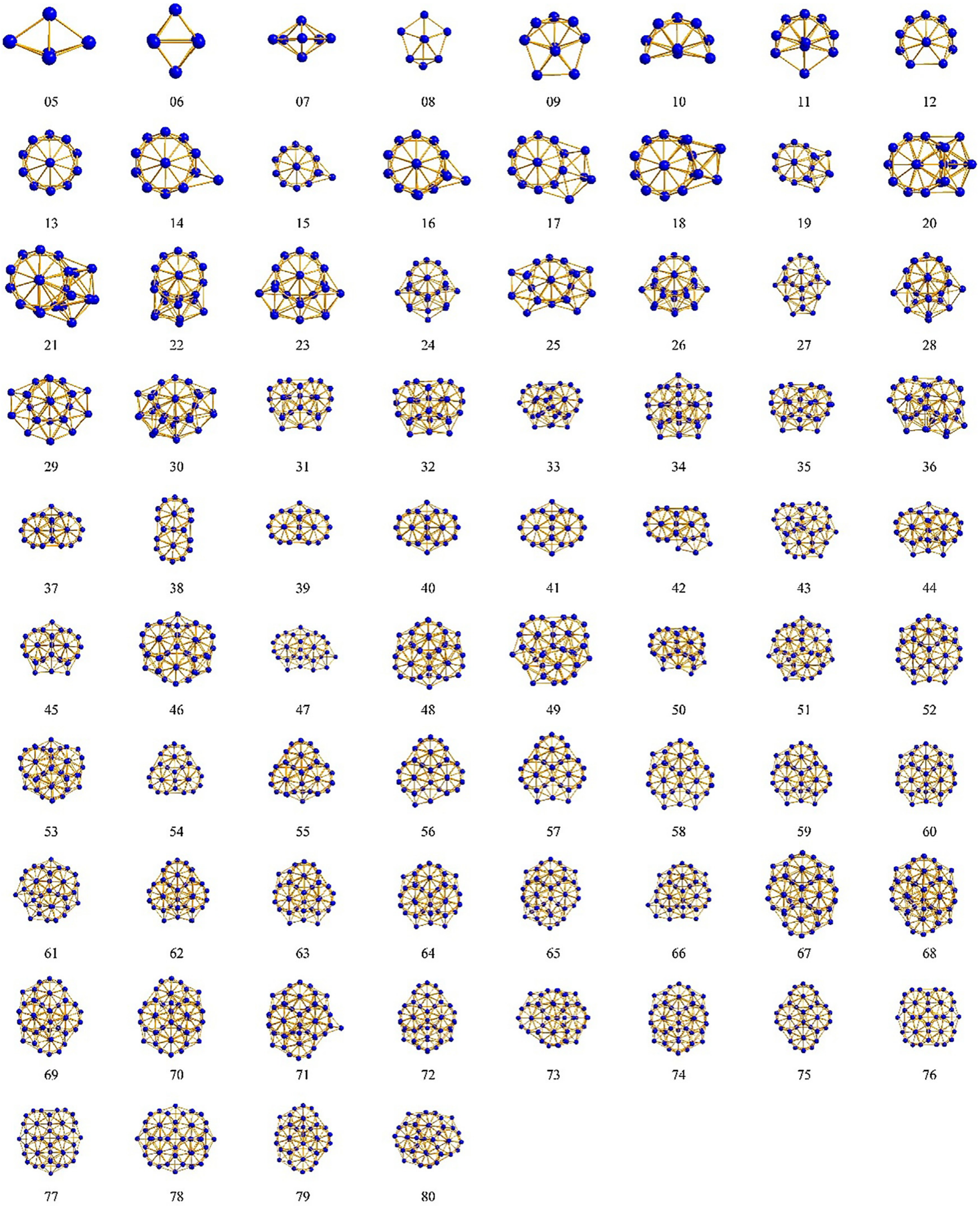

Fig. 8. The lowest-energy structures of Fe clusters with $N = 5–80$.

because the structure is easily trapped into local optima for small cluster pool size. Therefore, the best cluster pool size should be determined by balancing the algorithm convergence and the structural diversity.

3.2. Performance analysis of algorithm

To evaluate the performance of the cluster-based niching differential evolution algorithm, we have calculated the lowest energy of clusters (individuals) in the cluster pool and illustrated the change of the lowest-energy structures with the number of iterations in Fig. 6. Here, Fe_{30} and Fe_{60} clusters are selected as representatives. It can be seen from this figure that the energies of individuals are considerably high in early iterations, and the structures of clusters are confusing without any ordering and symmetry. The potential energy of individuals (cluster) is quickly decreased with the increase of iteration times. Moreover, the energy curve is not continuously dropping but jumps to next lowest energy point (stair-step-like shape), indicating the considerable difference of energy in different metastable structures. Also, it means that the algorithm adopts the jumping search between metastable structures rather than the continuous search along the potential energy surface of structures. Furthermore, the algorithm is gradually close to the optimal solution with increasing iteration. Also, the structures in Fig. 6 tend to be order and symmetrical with iteration time.

To further examine the change of structures with energy curves, the fingerprint vector of cluster [28] has been calculated during evolution. As an example, the result of Fe_{60} cluster is shown in Fig. 7. The four structures in Fig. 7 correspond to different stages of evolution curve. One can find that the fingerprint vector diagram has many peaks at the beginning of evolution. The peaks are not high, and the atoms are randomly arranged and disordered (see Fig. 7a). With the increasing of iterations, some peaks disappear, and some great peaks may split into two or more peaks (see Fig. 7b and c). However, the fingerprint vector diagram has fewer and sharp peaks in final phase of the evolution, and these peaks mainly concentrate on three areas, as indicated in Fig. 7d. Therefore, the energy of cluster is frequently decreased and the structure becomes more ordered with the increase of iteration.

It can be seen from the aforementioned analyses that our algorithm may effectively optimize the random mixed structures to become ordered and symmetrical. By examining the change of structure, one can find that the more symmetrical and orderly the structure of the cluster is, the lower its potential energy is, resultantly the more stable the structure is. Moreover, the fingerprint vector diagram has many peaks for disordered structures while their peak values are small. In contrast, the peaks for ordered structure are few but considerably sharp.

3.3. Optimized results of Fe clusters

In this work, structures of Fe clusters have been optimized by using our proposed cluster-based niching differential evolution algorithm. With the framework of the FS many-body potentials, the energies and space groups of the lowest-energy structures of Fe clusters with 3–80 atoms have been summarized in Table 2. One can find that most of the lowest-energy structures have a good symmetry. As seen from the table, the global minima of Fe clusters obtained by our algorithm are in complete agreement with the putative global minima available in Cambridge Cluster Database (CCD) [29] except for 71 atoms. For 71-atom Fe cluster, the minimal energy by our algorithm is -254.9198 eV, slightly higher than that of the CCD (-255.18697 eV). Nevertheless, this comparison still indicates the accuracy and effectiveness of our algorithm. Due to the complexity of searching the global minima on multi-dimensional potential energy surface and the limitation of computational resources, the atomic number of Fe cluster is less than 80 in this work.

The optimized stable structures of Fe clusters corresponding to Table 2 are further illustrated in Fig. 8. Evidently, it can be seen that Fe clusters possess extremely high symmetry at small size ($N < 13$). Especially, a complete icosahedron has been formed for the cluster with $N = 13$. Moreover, the number of icosahedrons in stable structures gradually rises with the atomic number. For example, the cluster presents two icosahedral rings for $N = 38$, and there are three icosahedral rings in the cluster for $N = 54$. With the atomic number increasing to 75, four icosahedral rings gradually replace the three icosahedral rings. According to this growth regularity, the lowest-energy structures of Fe clusters will contain more icosahedral rings if the number of atoms continues to increase. Therefore, it can be predicted that there are icosahedral rings in the stable structures of Fe clusters, and the number of the icosahedral rings will rise with the increase of atomic number.

4. Conclusions

In summary, we have proposed a cluster-based niching differential evolution algorithm to search the globally stable structures of Fe clusters with the atomic number from 3 to 80. For exploring the structural characteristic of clusters, the cluster pool, the niche method, and the differential evolution algorithm have been combined. First, a local minimization method is adopted by transforming the potential energy surface into many less-intricate basins to simplify the optimization process. Secondly, K-means clustering method is employed to classify the structures into some niches. Moreover, the complete cluster-based niching differential evolution algorithm have been described, and the effect of cluster pool size on algorithm performance is examined. Additionally, the effectiveness of our algorithm has been analyzed by ten times independent experiments for structural optimization of Fe cluster. Finally, the stable structures of Fe clusters with 3–80 atoms have been explored by using the structural clustering based niching differential evolution algorithm. The growth trend of stable structures indicates that the lowest-energy structure of Fe cluster contains many icosahedra, and the number of the icosahedral rings rises with the atomic number. It can be expectable that the structural clustering based niching differential evolution algorithm should be extended to predict the stable structures of other metallic and alloy clusters.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grants 11474234 and 51271156).

References

[1] A.W. Castleman, P. Jena, Clusters: A bridge across the disciplines of environment, materials science, and biology, Proc. Natl. Acad. Sci. USA 103 (2006) 10554–10559.

[2] J. Bansmann, S.H. Baker, C. Binns, J.A. Blackman, J.P. Bucher, J. Dorantes-Davila, V. Dupuis, L. Favre, D. Kechrakos, A. Kleibert, Magnetic and structural properties of isolated and assembled clusters, Surface Sci. Rep. 56 (2005) 189–275.

[3] N. Lopez, T.V.W. Janssens, B.S. Clausen, Y. Xu, M. Mavrikakis, T. Bligaard, J.K. Nørskov, On the origin of the catalytic activity of gold nanoparticles for low-temperature CO oxidation, J. Catal. 223 (2004) 232–235.

[4] Y. Zhang, Y.H. Wen, Z.Z. Zhu, S.G. Sun, Structure and stability of Fe nanocrystals: An atomistic study, J. Phys. Chem. C 114 (2010) 18841–18846.

[5] D.L. Huber, Synthesis, properties, and applications of iron nanoparticles, Small (Weinheim an der Bergstrasse, Germany) 1 (2005) 482–501.

[6] D.H. Deng, L. Yu, X.Q. Chen, G.X. Wang, L. Jin, X.L. Pan, J. Deng, G.Q. Sun, X.H. Bao, Iron encapsulated within pod-like carbon nanotubes for oxygen reduction reaction, Angew. Chem. Int. Ed. 52 (2013) 371–375.

[7] J.A. Elliott, Y. Shibuta, D.J. Wales, Global minima of transition metal clusters described by Finnis-Sinclair potentials: a comparison with semi-empirical molecular orbital theory, Philos. Mag. 89 (2009) 3311–3332.

[8] F. Baletto, R. Ferrando, Structural properties of nanoclusters: energetic, thermodynamic, and kinetic effects, Rev. Mod. Phys. 77 (2005) 371–423.

[9] Z. Li, H.A. Scheraga, Monte Carlo-minimization approach to the multiple-minima problem in protein folding, Proc. Natl. Acad. Sci. USA 84 (1987) 6611–6615.

[10] J.P.K. Doye, D.J. Wales, Global minima for transition metal clusters described by Sutton-Chen potentials, New J. Chem. 22 (1998) 733–744.

[11] R.H. Leary, Global optimization on funneling landscapes, J. Global Optim. 18 (2000) 367–383.

[12] W. Cai, X. Shao, A fast annealing evolutionary algorithm for global optimization, J. Comput. Chem. 23 (2002) 427–435.

[13] B. Hartke, Global geometry optimization of clusters using genetic algorithms, J. Phys. Chem. 97 (1993) 9973–9976.

[14] Y. Zeiri, Prediction of the lowest energy structure of clusters using a genetic algorithm, Phys. Rev. E 51 (1995) R2769.

[15] S.T. Call, D.Y. Zubarev, A.I. Boldyrev, Global minimum structure searches via particle swarm optimization, J. Comput. Chem. 28 (2007) 1177–1186.

[16] X. Luo, J. Yang, H. Liu, X. Wu, Y. Wang, Y. Ma, S.H. Wei, X. Gong, H. Xiang, Predicting two-dimensional boron–carbon compounds by the global optimization method, J. Am. Chem. Soc. 133 (2011) 16285–16290.

[17] X. Shao, L. Cheng, W. Cai, An adaptive immune optimization algorithm for energy minimization problems, J. Chem. Phys. 120 (2004) 11401–11406.

[18] L. Cheng, W. Cai, X. Shao, A connectivity table for cluster similarity checking in the evolutionary optimization method, Chem. Phys. Lett. 389 (2004) 309–314.

[19] A. Shayeghi, D. Gotz, J.B. Davis, R. Schafer, R.L. Johnston, Pool-BCGA: a parallelised generation-free genetic algorithm for the ab initio global optimisation of nanoalloy clusters, Phys. Chem. Chem. Phys. 17 (2015) 2104–2112.

[20] Z. Chen, X. Jiang, J. Li, S. Li, L. Wang, PDECO: parallel differential evolution for clusters optimization, J. Chem. 34 (2013) 1046–1059.

[21] W.G. Sheng, A. Tucker, X.H. Liu, A niching genetic k-means algorithm and its applications to gene expression data, Soft Comput. 14 (2010) 9–19.

[22] M.W. Finnis, J.E. Sinclair, A simple empirical N-body potential for transition metals, Philos. Mag. A 50 (1984) 45–55.

[23] C. Engin, L. Sandoval, H.M. Urbassek, Characterization of Fe potentials with respect to the stability of the bcc and fcc phase, Modell. Simul. Mater. Sci. Eng. 16 (2008) 035005.

[24] K. Yun, Y.H. Cho, P.R. Cha, J. Lee, H.S. Nam, J.S. Oh, J.H. Choi, S.C. Lee, Monte Carlo simulations of the structure of Pt-based bimetallic nanoparticles, Acta Materialia 60 (2012) 4908–4916.

[25] R. Mukherjee, G.R. Patra, R. Kundu, S. Das, Cluster-based differential evolution with Crowding Archive for niching in dynamic environments, Inform. Sci. 267 (2014) 58–82.

[26] M.Y. Cheng, D.H. Tran, M.T. Cao, Chaotic initialized multiple objective differential evolution with adaptive mutation strategy for construction project time-cost-quality trade-off, J. Civ. Eng. Manage. 22 (2016) 210–223.

[27] Y. Shu, J. Li, S. He, H. Tang, N. Wang, L. Shen, H.Y. Du, Clustering of hyperspectral image based on spatial-spectral Chinese restaurant process mixture model, Spectrosc. Spect. Anal. 36 (2016) 1158–1162.

[28] A.R. Oganov, M. Valle, How to quantify energy landscapes of solids, J. Chem. Phys. 130 (2009) 104504.

[29] http://www-wales.ch.cam.ac.uk/~wales/CCD/FS_Fe/iron_fs.html.