

PCCP

Accepted Manuscript



This is an *Accepted Manuscript*, which has been through the Royal Society of Chemistry peer review process and has been accepted for publication.

Accepted Manuscripts are published online shortly after acceptance, before technical editing, formatting and proof reading. Using this free service, authors can make their results available to the community, in citable form, before we publish the edited article. We will replace this *Accepted Manuscript* with the edited and formatted *Advance Article* as soon as it is available.

You can find more information about *Accepted Manuscripts* in the [Information for Authors](#).

Please note that technical editing may introduce minor changes to the text and/or graphics, which may alter content. The journal's standard [Terms & Conditions](#) and the [Ethical guidelines](#) still apply. In no event shall the Royal Society of Chemistry be held responsible for any errors or omissions in this *Accepted Manuscript* or any consequences arising from the use of any information it contains.



Cite this: DOI: 10.1039/xxxxxxxxxx

ABCluster: The Artificial Bee Colony Algorithm for Cluster Global Optimization[†]

Jun Zhang^{*a} and Michael Dolg^{*a}Received Date
Accepted Date

DOI: 10.1039/xxxxxxxxxx

www.rsc.org/journalname

Global optimization of cluster geometries is of fundamental importance in chemistry and an interesting problem in applied mathematics. In this work, we introduce a relatively new swarm intelligence algorithm, i.e. the artificial bee colony (ABC) algorithm proposed in 2005, to this field. It is inspired by the foraging behavior of a bee colony, and only three parameters are needed to control it. We applied it to several potential functions of quite different nature, i.e., the Coulomb–Born–Mayer, Lennard–Jones, Morse, Z and Gupta potentials. The benchmarks reveal that for long-ranged potentials the ABC algorithm is very efficient in locating the global minimum, while for short-ranged ones it is sometimes trapped into a local minimum funnel on a potential energy surface of large clusters. We have released an efficient, user-friendly, and free program “ABCluster” to realize the ABC algorithm. It is a black-box program for non-experts as well as experts and might become a useful tool for chemists to study clusters.

1 Introduction

A cluster is an aggregate composed of a few to several million identical or different structural units like atoms and molecules. Clusters have attracted intense attention from scientific communities^{1,2} due to their practical and theoretical interest. In practise, clusters are of micro- to mesoscopic size, which leads to many amazing properties. For instance, gold clusters can help building novel materials³ and the icosahedral 13-atom platinum cluster has anomalous large diamagnetic susceptibility⁴. Theoretically, clusters can be used to explore the size-dependence of properties from atomic to bulk level: Atomic clusters have been studied to predict the melting temperatures of neon and argon solids⁵ as well as crystal growth⁶. Besides their wide applications, clusters are also attractive per se. They can have fascinating geometries⁷, sometimes behave like a “superatom”⁸, and be challenging cases for accurate quantum chemical methods⁹.

Among the various aspects, searching the global minimum (GM) of a cluster is an important and fundamental topic, since the GM dominates for low temperature, at which many experiments involving clusters are performed. Unfortunately a GM search is usually a rather difficult task. The potential energy surface (PES) of a cluster containing N structural units has $3N$ degrees of freedom (DOFs). The number of local minima (LMs) increases exponentially with N ^{1,10}, and the complex distribution of these huge

number of LMs makes the PES locally very rugged, making an “ergodic” sampling on the PES of large clusters by computer simulation nearly impossible. In fact, even by *nature* such ergodicity cannot be realized in the lifetime of the universe, according to, e.g. an estimation for a short peptide¹¹. However, nature can indeed fold a protein to its native structure (which often is the GM) in several seconds, which raises Levinthal’s paradox^{12,13}. Anyway, a successful deterministic search for the GM of a large cluster is usually impossible.

Mathematically, searching the GM of a cluster is a typical unconstrained global optimization problem, which is widely encountered in scientific, engineering, economic and social researches. The variables to be optimized are the position vectors or coordinates $\mathbf{X} \equiv (\mathbf{r}_1, \dots, \mathbf{r}_N) \equiv (x_1, y_1, z_1, \dots, x_N, y_N, z_N)$ of the cluster, and the objective function is the potential energy function $U(\mathbf{X})$. Almost all popular algorithms for global optimization are nondeterministic, which means that the chance of the obtained solution being the true GM is beyond a significant probability. In the context of cluster optimization, these algorithms fall into two groups: biased and unbiased ones. In a biased algorithm, the characters of the known GMs are used as much as possible. For example, for intermediate-size Lennard–Jones (LJ) clusters, based on the observation that the decahedral and icosahedral structures are more favored than the face-centered cubic (fcc) ones, a lattice construction scheme (LCS) can locate the putative GMs of LJ_N ($N = 150$ to 1600) very efficiently^{14–16}. However, such strategy is not transferable to other kinds of clusters, since clusters of the same size can have GMs of quite distinct topology for different potentials and sometimes no reference GMs are

^a Theoretical Chemistry, University of Cologne, Greinstr. 4, 50939 Cologne, Germany. Fax: ++49 (0)221 470 6896; Tel: ++49 (0)221 470 6893; E-mail: zhangjun-qcc@gmail.com, m.dolg@uni-koeln.de.

[†] Electronic Supplementary Information (ESI) available: [...TODO...]. See DOI: 10.1039/b000000x/

known. The unbiased algorithm is more general and makes no assumptions on what the GM should look like. Some algorithms perform the search with an individual cluster, e.g. basin hopping (BH)^{17,18}, Monte Carlo minimization (MCM)¹⁹ and simulated annealing (SA)²⁰, although SA is not as efficient as BH and MCM in this field. Other algorithms are the population-based ones. These algorithms manipulate the entire population (a collection of clusters with different geometry) and improve the individuals by some strategies to search the solution space. After a number of iterations, the GM will appear with some probability. One category are the evolutionary algorithms (EA)²¹, such as the genetic algorithm (GA)²² and differential evolution (DE)²³. These algorithms mimic the biology evolution in nature, generating offsprings (clusters with new geometry) by crossover (e.g. “cut-and-splice”²⁴) or mutation of the individuals. Only the ones with high fitness (i.e. sufficiently low energy) can survive and in the best case after several generations (iterations) the offspring with highest fitness (the GM) will dominate the population. They have been widely applied for clusters and the interested readers are referred to several excellent reviews^{1,25,26} and literatures therein. Another category are swarm intelligence algorithms²⁷, including particle swarm optimization (PSO)²⁸, etc. They have been applied to, e.g. prediction of crystal structures²⁹. Very recently, some algorithms taking permutation space into account for the global optimization of multicomponent clusters have been proposed^{30–32}.

All optimization methods mentioned above have been proven to be efficient and successful for specific kinds of clusters. However, there remain still some problems to be solved in this field. First, many algorithms need a lot of input parameters which usually have to be set empirically and require a deeper knowledge of the method. The performance of the algorithms depends on the quality of the parameters significantly. Second, some codes can work well with a few kinds of potentials but not with others. Third, except for a few cases like CALYPSO²⁹, GMIN³³, OGOLEM³⁴, G42+³⁵ and some codes in the ASE project³⁶ there is no black-box program for nonexperts to perform the global optimization of clusters readily.

The purpose of the present work is dual. First, we introduce a relatively new optimization technique which was proposed in 2005, i.e. the artificial bee colony (ABC) algorithm³⁷, to the field of GM search of clusters. We have modified it for our problem and its performance is examined. Second, we introduce ABCluster (ABC for Clusters), which is an efficient, user-friendly and free software to perform the global optimization of clusters by the ABC algorithm. It will be shown that ABCluster is a powerful tool for the global optimization of various kinds of clusters.

2 Theory

2.1 Potential Energy Functions

As mentioned above, searching the GM of a cluster of size N is actually the unconstrained global optimization of the potential energy function of $3N$ variables:

$$U(\mathbf{X}) = \sum_{i < j}^N u_2(r_{ij}) + \sum_{i < j < k}^N u_3(r_{ij}, r_{ik}, r_{jk}) + \dots + u_N(r_{12}, \dots, r_{N-1, N}) \quad (1)$$

$$\mathbf{X} \equiv (\mathbf{r}_1, \dots, \mathbf{r}_N) \equiv (x_1, y_1, z_1, \dots, x_N, y_N, z_N) \quad (2)$$

Here, $U(\mathbf{X})$ can be a first-principle or phenomenological potential. In this work we only consider the latter. Many phenomenological potential functions contain only the pairwise interaction $u_2(r)$ term in (1). The canonical Lennard-Jones or Morse potential belongs to this class. However, in modern computational chemistry the polarization and delocalization of the structural units of a cluster, i.e. the many-body terms beyond $u_2(r)$ ^{38,39}, must be taken into account, since otherwise the properties of the clusters computed may turn out to be very inaccurate^{5,40}. Therefore, many-body potentials will also be considered. In our program ABCluster, the following potentials will be examined (from here on the cluster of size N interacted by the potential “U” will be denoted by U_N):

1. Coulomb–Born–Mayer potential (CBM_N). Two-body potential. CBM_N are very suitable for a description of ionic clusters. The exponential term (Born–Mayer potential⁴¹) reflects the Pauli repulsion. Here q , B and ρ stands for the charge, repulsion strength, and repulsion range of the particles, respectively.

$$U_{\text{CBM}} = \sum_{i=1}^N \sum_{i < j}^N \left(\frac{e^2}{4\pi\epsilon_0} \frac{q_i q_j}{r_{ij}} + B_{ij} \exp\left(-\frac{r_{ij}}{\rho_{ij}}\right) \right) \quad (3)$$

2. Lennard-Jones potential⁴² (LJ_N). Two-body potential. This classic “12–6” potential is widely used to describe the dispersion interaction in chemistry. It is also a very important benchmark model to test the performance of global optimization algorithms. Here ϵ and σ are the pair well depth and the equilibrium distance, respectively.

$$U_{\text{LJ}} = \sum_{i=1}^N \sum_{i < j}^N 4\epsilon_{ij} \left(\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}}\right)^6 \right) \quad (4)$$

3. Morse potential⁴³ (M_N). Two-body potential. This is a more advanced form than the harmonic potential for the description of the vibration of molecules, since it also considers anharmonic effects. The parameters ϵ , r^0 and β are the pair well depth, equilibrium distance and force range, respectively. A smaller β indicates a longer force range. Usually $n = 2$. A larger n leads to a steeper potential well.

$$U_{\text{M}} = \sum_{i=1}^N \sum_{i < j}^N \epsilon_{ij} \left(\exp\left(-n\beta_{ij}(r_{ij} - r_{ij}^0)\right) - n \exp\left(-\beta_{ij}(r_{ij} - r_{ij}^0)\right) \right) \quad (5)$$

4. Z potential⁴⁴ (Z_N). Two-body potential. This is inspired by an earlier potential proposed by Dzugutov^{45,46} which was designed to study glass formation. Unlike CBM, LJ and M potentials, they contain minima as well as maxima, which make the close-packing energetically unfavourable and lead

to amorphous structures. Here, r_c is a cutoff distance beyond which the interaction disappears (denoted by the Heaviside function $\theta(r_c - r_{ij})$), and k_F is the wave vector at the Fermi level.

$$U_Z = \sum_{i=1}^N \sum_{i<j}^N \left(a \frac{e^{\alpha r_{ij}}}{r_{ij}^3} \cos(2k_F r_{ij}) + b \left(\frac{\sigma}{r_{ij}} \right)^n + V_0 \right) \theta(r_c - r_{ij}) \quad (6)$$

5. Gupta potential⁴⁷ (G_N). Many-body potential. This is derived from a second-moment approximation of tight-binding theory and is a very important potential for modeling metals, consisting of a repulsive short-ranged pairwise term (the exponential part) and an attractive embedded-atom term (the square root part). The parameters in the potential are fitted to bulk properties.

$$U_G = \sum_{i=1}^N \left(\sum_{\substack{j=1 \\ j \neq i}}^N A_{ij} \exp\left(-p_{ij} \left(\frac{r_{ij}}{d_{ij}} - 1\right)\right) - \sqrt{\rho(\mathbf{r}_i)} \right) \quad (7)$$

where:

$$\rho(\mathbf{r}_i) = \sum_{\substack{j=1 \\ j \neq i}}^N \xi_{ij}^2 \exp\left(-2q_{ij} \left(\frac{r_{ij}}{d_{ij}} - 1\right)\right) \quad (8)$$

The functional form of the potential has deep implications on the shape of the GM. An interesting example is that for a Gupta cluster of $N = 13$, whose two different sets of parameters for gold can lead to an icosahedron and a disordered GM, respectively. The form further affects the difficulty to locate the GM. On the PES, the number of LMs does not only increase exponentially with the cluster size N as mentioned above, but it also *increases* as the force range of the potential *decreases*^{48,49}. Thus, the search of GMs is easier for long-ranged potentials like CBM. For short-ranged or multimodal ones, the PES can be of *multi-funnel* topography (see Figure 1), like LJ₇₅⁵⁰ or Z_N⁴⁴, leading to GMs of a less compact or even hollow shape⁴⁴. This is because the finite force range and the presence of the maxima lead to the existence of many barriers, making the PES rather rugged. Thus, the GM search algorithm may be trapped in the deep funnel of a LM by these barriers^{51,52}.

In order to increase the efficiency of the GM search algorithm, two strategies have been applied. First, a good initial guess. A “perfect” GM search algorithm should locate the GM from an arbitrary initial guess. However, due to the high complexity and dimension of the PES mentioned above, this may take an extremely long time for some potentials. Thus, besides the random initial guess, one could also include a “seed guess”, which is formed e.g. by adding atoms to the GM structures of clusters of smaller sizes. In fact, for some disordered metallic clusters, the global optimization cannot succeed without such seeds⁵³. In the present work for most cases we did not make use of seeded guesses in order to investigate the efficiency of the unbiased algorithm. Second, one should “smooth” the PES to reduce its ruggedness¹. In this work we use⁵⁴:

$$\tilde{U}(\mathbf{X}) = \min : \{U(\mathbf{X})\} \quad (9)$$

where “min” means performing a local optimization of U starting from \mathbf{X} . As shown in Figure 1, this removes the barriers along the downhill movement towards a funnel, making the optimization more efficient⁵⁴. This transformation resembles the Lamarckian genetic algorithm used in the molecular docking software AutoDock⁵⁵, and has been widely applied in the global optimization of clusters. Unfortunately, (9) cannot remove the barriers between different funnels, which is the main bottleneck of a global optimization algorithm. In the following, the ABC algorithm will be used to search for the GM of clusters, which is the essential function of our program ABCluster.

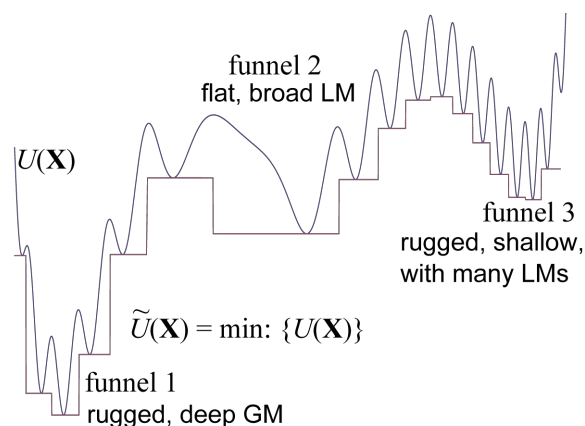


Fig. 1 An illustration of a typical PES topography. The original PES U has 3 funnels with different ruggedness and width, with the true GM located in funnel 1. After smoothing we get \tilde{U} , then the barriers along the downhill movement toward a funnel are removed since the PES becomes staircase-like, see e.g. funnel 3. However, the barriers separating different funnels still exist.

2.2 Artificial Bee Colony Algorithm in ABCluster

The artificial bee colony (ABC) algorithm, which is a swarm intelligence based one, was first proposed by Karaboga in 2005³⁷. It was inspired by the foraging behavior of honey bee colonies. In these colonies, the honey bees want to find the best nectar sources. To efficiently achieve this, the bees are specialized for different tasks. In the model of the ABC algorithm, there are three kinds of bees: employed, onlooker and scout bees. Each bee can find nectar and estimate its “quality”. More importantly, it can share this information to other bees by, e.g., a waggle dance. This communication between the individuals is fundamental to the random and feedback behaviors of the colony. In one search cycle, the employed bees look for new nectar sources based on their own as well as *other* bees’ knowledge. Then, the onlooker bees communicate with the employed bees and look for new nectar sources around the “good” ones. Based on the feedback from employed and onlooker bees, the nectar sources of low quality will be discarded and the scout bees search for the new ones. After several cycles, a “best” nectar source is finally selected.

The ABC algorithm models the above-mentioned behavior of bee colonies. Extensive descriptions of the ABC algorithm can be found in some reviews^{56–58}. In the context of the global optimization of a cluster, a trial solution \mathbf{X} is the nectar source and

the potential energy $U(\mathbf{X})$ is its quality (a lower numeric value indicates a higher quality). The parameters needed for a global optimization with the ABC algorithm are the following: the size of the population of trial solutions SN , the scout limit g_{limit} and the maximum cycle number g_{max} . The cluster is characterized by its size N , the estimated maximum coordinate value L , and of course the potential parameters. Now, the global optimization (foraging) can be performed (see Figure 2 for a flowchart):

1. Initialize the population: $\mathbf{X}_1^1, \dots, \mathbf{X}_{SN}^1$. One can use random initial guesses: each component of \mathbf{X}_i^1 is randomly taken from the range $[0, L]$. For difficult cases a seeded guess is used, which is usually the GM of the cluster of size $N-1$ plus one atom randomly positioned. Then all the structures are locally optimized by the limited-memory-Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) algorithm⁵⁹.

2. Modelling employed bees: in cycle g , for each \mathbf{X}_i^g ($i = 1, \dots, SN$), a new trial solution \mathbf{V}_i according to other \mathbf{X}_j^g 's ($j = 1, \dots, SN, j \neq i$) (information sharing) is generated (employed bee search). Here we use the *trigonometric mutation operator*⁶⁰:

$$\mathbf{V}_i = \frac{1}{3} \left(\mathbf{X}_{k_1}^g + \mathbf{X}_{k_2}^g + \mathbf{X}_{k_3}^g \right) + (p_2 - p_1) \left(\mathbf{X}_{k_1}^g - \mathbf{X}_{k_2}^g \right) + (p_3 - p_2) \left(\mathbf{X}_{k_2}^g - \mathbf{X}_{k_3}^g \right) + (p_1 - p_3) \left(\mathbf{X}_{k_3}^g - \mathbf{X}_{k_1}^g \right) \quad (10)$$

where k_1, k_2 and k_3 are random integers in $\{1, \dots, SN\}$ and $k_1 \neq k_2 \neq k_3 \neq i$, and

$$p_m = \frac{|\tilde{U}(\mathbf{X}_{k_m})|}{|\tilde{U}(\mathbf{X}_{k_1})| + |\tilde{U}(\mathbf{X}_{k_2})| + |\tilde{U}(\mathbf{X}_{k_3})|} \quad (m = 1, 2, 3) \quad (11)$$

\tilde{U} is the transformed potential energy function (9), where the local optimization “min” is realized by the L-BFGS algorithm⁵⁹ mentioned above. The operator (10) can exchange the information within the population very efficiently, e.g. in the DE algorithm^{60,61}. Then this solution is updated with a greedy selection scheme (12):

$$\mathbf{X}_i^{g+1} = \begin{cases} \mathbf{V}_i & \text{if } \tilde{U}(\mathbf{V}_i) < \tilde{U}(\mathbf{X}_i^g) \\ \mathbf{X}_i^g & \text{otherwise} \end{cases} \quad (12)$$

3. Modelling onlooker bees: after the employed bee search, each onlooker bee will select a “good” solution \mathbf{X}_k^g , and a new trial solution \mathbf{V}_k is generated again according to other \mathbf{X}_j^g 's ($j = 1, \dots, SN, j \neq k$). There are two ways of selecting the “good” ones, the *roulette wheel* and *tournament selection*²⁵. In the former way one solution is selected with a probability proportional to its quality, however this strategy has a large bias on the *best solution* $\mathbf{X}_{\text{best}}^g$ (the one with lowest energy in the population), which reduces the efficiency of the algorithm. Thus the tournament selection strategy is used. In this way, N_{players} (in this work, $N_{\text{players}} = 5$) individuals are randomly selected from the population and the best one (with lowest energy) is chosen. To use the information that \mathbf{X}_k^g being “good”, we generate \mathbf{V}_k based on \mathbf{X}_k^g , i.e. (13)

instead of (10):

$$\mathbf{V}_k = \begin{cases} \mathbf{X}_k^g + F \left(\mathbf{X}_{k_1}^g + \mathbf{X}_{k_2}^g - \mathbf{X}_{k_3}^g - \mathbf{X}_{k_4}^g \right) & \text{if } \eta < 0.5 \\ \mathbf{X}_{\text{best}}^g + F \left(\mathbf{X}_{k_1}^g + \mathbf{X}_{k_2}^g - \mathbf{X}_{k_3}^g - \mathbf{X}_{k_4}^g \right) & \text{otherwise} \end{cases} \quad (13)$$

where k_1, k_2, k_3 and k_4 are random integers in $\{1, \dots, SN\}$ and $k_1 \neq k_2 \neq k_3 \neq k_4 \neq k$. F and η are random numbers in $[0, 1)$. The two expressions in (13) are denoted by “ABC/current/2” and “ABC/best/2”, respectively. They mean that the trial solution is generated by improving the current/best solution with 2 differences of vectors. Similar expressions have appeared in the DE algorithm literature⁶² and they are able to improve the performance of the ABC algorithm⁶³. Then, \mathbf{X}_k^g is also updated with the greedy selection scheme (12).

4. Modelling scout bees: after the onlooker bee search, each \mathbf{X}_i^g ($i = 1, \dots, SN$) is examined. If \mathbf{X}_i^g does not change in the last g_{limit} cycles, it will be replaced by a random trial solution \mathbf{X}_i^{g+1} (scout bee search) no matter whether it is better than \mathbf{X}_i^g or not. Without scout bees the algorithm may converge to a LM too fast and then be trapped in, like a PSO algorithm. In fact, scout bee search is very important for keeping the diversity of the solution population and jumping out the funnel around a LM.

5. If $g \geq g_{\text{max}}$, the algorithm is accomplished, otherwise go to step 2.

One can realize: the random assignment of some solutions introduces *fluctuations* into the system; the employed and onlooker bee searches introduce *multiple interactions*; the onlooker and scout bee searches introduce *positive* and *negative feedback*, respectively. These four characteristics are fundamental for a dynamical system being *self-organizing*⁶⁴. This makes it a very attractive algorithm compared with other ones. This is due to the fact that, rather than assigning a crossover or mutation rate manually like in the DE⁶¹ or PSO²⁹ algorithms, the self-organization can automatically provide the mechanism to keep the diversity of the solutions as well as guarantee the population evolving positively in an efficient way. For example, in a study⁵⁶ of the global optimization of 23 benchmark functions, it was found that ABC performs better than or at least similar to the GA, DE or PSO algorithms. Thus, the ABC algorithm has gained extensive applications in many problems⁵⁸. In chemistry, it has been used in the optimization of force field parameters⁶⁵, the prediction of the protein secondary structure⁶⁶, etc.

Before we discuss the performance of ABCcluster for the global optimization of clusters, we note that some previous work has also tried the ABC algorithm for this purpose^{67,68}. However, those studies only tested medium-sized systems ($N \leq 50$) and a few potentials. They used simple move operators rather than the more sophisticated operators (10) and (13). We will show that our implementation can be effective for larger clusters and a wide variety of potentials.

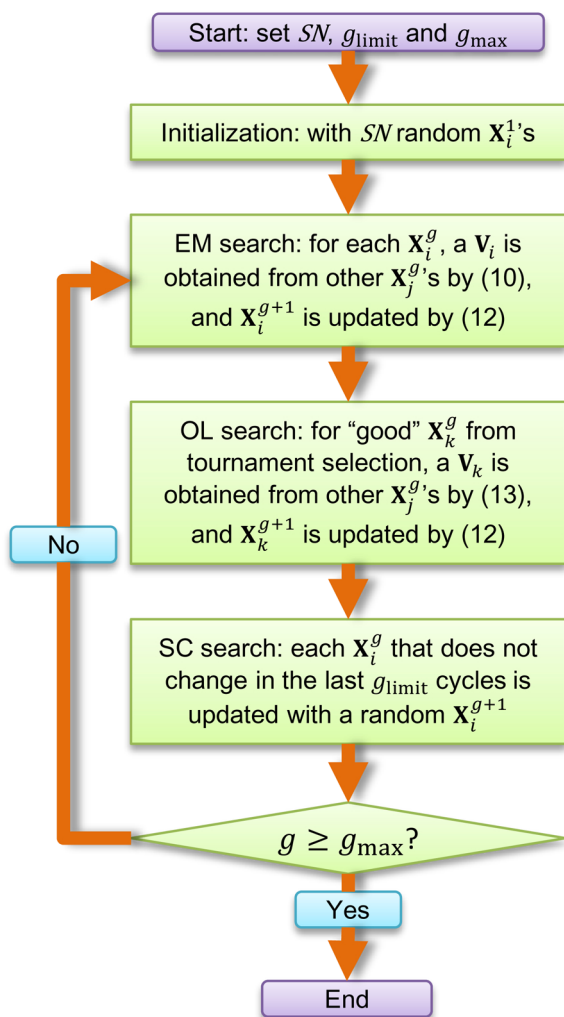


Fig. 2 The ABC algorithm for searching the GM of clusters used in ABCluster. EM: employed bees; OL: onlooker bees; SC: scout bees.

3 Performance Test

We applied the ABC algorithm to search the GMs of clusters with the potentials shown in Table 1. All the tests were accomplished by the program ABCluster, which is developed by the authors. The GMs were rendered by CYLview⁶⁹. Here we point out that according to our exploration, the performance of the ABC algorithm is very stable. Unless using a tuned initial guess, the optimization starting with a random initial guess requires a number of steps of the same order of magnitude to find the GM in different runs. Thus only one representative instance of optimization for each case is given in the following.

3.1 The Coulomb–Born–Mayer Potential

The CBM potential (3) involves the long-ranged Coulomb interaction and thus it is the easiest case for the GM search. We chose the ionic cluster $(\text{MgO})_N$ for benchmarking. In a work by Roberts and Johnston⁷³ their GMs have been searched by the GA algorithm. It was observed that as the formal charge Q on the ions increases the GM changes from cuboidal to cage structures. Here we set Q as 1, 1.5 and 2 on the ions, corresponding to U_{CBMQ1} , $U_{\text{CBMQ1.5}}$

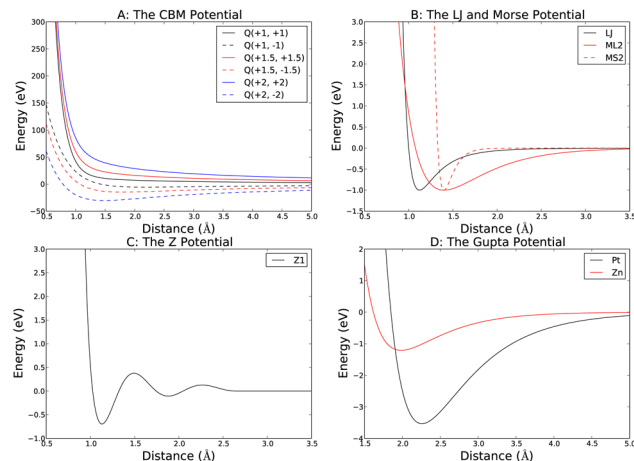


Fig. 3 The distance-dependence of the two-body potentials examined in this work.

Table 1 The parameters of the potentials examined in this work^a

Potentials	Parameters
Common ⁷⁰	$B(\text{Mg}-\text{O}) = 821.6$; $B(\text{Mg}-\text{Mg}) = 0$; $B(\text{O}-\text{O}) = 22764$ $\rho(\text{Mg}-\text{O}) = 0.3242$; $\rho(\text{Mg}-\text{Mg}) = 0$; $\rho(\text{O}-\text{O}) = 0.1490$
U_{CBMQ1}	$q_{\text{Mg}} = +1.0$; $q_{\text{O}} = -1.0$
$U_{\text{CBMQ1.5}}$	$q_{\text{Mg}} = +1.5$; $q_{\text{O}} = -1.5$
U_{CBMQ2}	$q_{\text{Mg}} = +2.0$; $q_{\text{O}} = -2.0$
U_{LJ}	$\epsilon = 1.0$; $\sigma = 1.4$
U_{ML2}	$\epsilon = 1.0$; $n = 2$; $r^0 = 1.4$; $\beta = 2.14285714$
U_{MS2}	$\epsilon = 1.0$; $n = 2$; $r^0 = 1.4$; $\beta = 10.0$
U_{Z1} ⁴⁴	$a = 1.58$, $\alpha = -0.22$, $k_{\text{F}} = 4.120$, $b = 4.2 \times 10^{+8}$ $\sigma = 0.331$, $n = 18.0$, $V_0 = 0.04682632$, $r_c = 2.64909$
Gupta	
U_{Pt} ⁷¹	$A = 0.2975$, $\xi = 2.695$, $d = 2.7747$, $p = 10.612$, $q = 4.004$
U_{Zn} ⁷²	$A = 0.1477$, $\xi = 0.8900$, $d = 2.50$, $p = 9.689$, $q = 4.602$

^a units of the parameters are given below. The CBM potential: B (eV), ρ (\AA), q (arbitrary). The LJ potential: ϵ (eV), σ (\AA). The Morse potential: ϵ (eV), n (arbitrary), r^0 (\AA), β (\AA^{-1}). The Z potential: all units are arbitrary, thus for convenience we assign its energy as eV. The Gupta potential: A and ξ (eV), d (\AA), p and q (arbitrary).

and U_{CBMQ2} , respectively, in Table 1 (see also Figure 3A). The results are given in Table 2 and Figure 4. It is found that $N = 25$ is a difficult case, since it required much more steps to locate the GM's than $N = 22$ did. In fact, it is also a difficult case for the GA⁷³. The reason is that there exists a LM (Energy: -193.8975 eV) which is energetically very close to the GM. Nevertheless, for U_{CBMQ1} and U_{CBMQ2} our GMs completely agree with Roberts and Johnston's results⁷³. Moreover, we found that for small clusters ($N \leq 20$) the GMs with $Q = 1.5$ are tube structures, being different from those with $Q = 1$. Thus, for ionic clusters with components carefully tuned, one may get many fascinating structures. Since the ABC algorithm is very efficient for locating GMs of ionic clusters, it has great potential in aiding the design of nanocrystals, as long as the potential energy function is accurate.

3.2 The Lennard-Jones, Morse and Z Potential

These potentials are all short-ranged. The former two are unimodal while the last one is multimodal. The potential parameters are given in Table 1, and the results are shown in Table 3, Table 4 and Table 5. Some GMs are shown in Figure 5.

Table 2 Benchmark for the CBM potential

ABC algorithm parameters: $SN = 100$, $g_{\text{limit}} = 5$, $g_{\text{max}} = 600$				
Initial guess: random				
Potential	(MgO) $_N$	Steps	Energy (eV)	Reference ⁷³ (eV) ^a
U_{CBMQ1}	16	1	-123.1609	-123.1643
	20	2	-154.6450	-154.6494
	22	2	-170.1957	-170.2005
	25	327	-194.1070	-194.1125
$U_{\text{CBMQ1.5}}$	16	1	-314.6231	N/A
	20	10	-394.7516	N/A
	22	1	-434.6638	N/A
	25	49	-495.0921	N/A
U_{CBMQ2}	16	1	-624.8734	-624.8921
	20	5	-783.3009	-783.3243
	22	22	-861.6615	-861.7859
	25	521	-980.5489	-980.5783

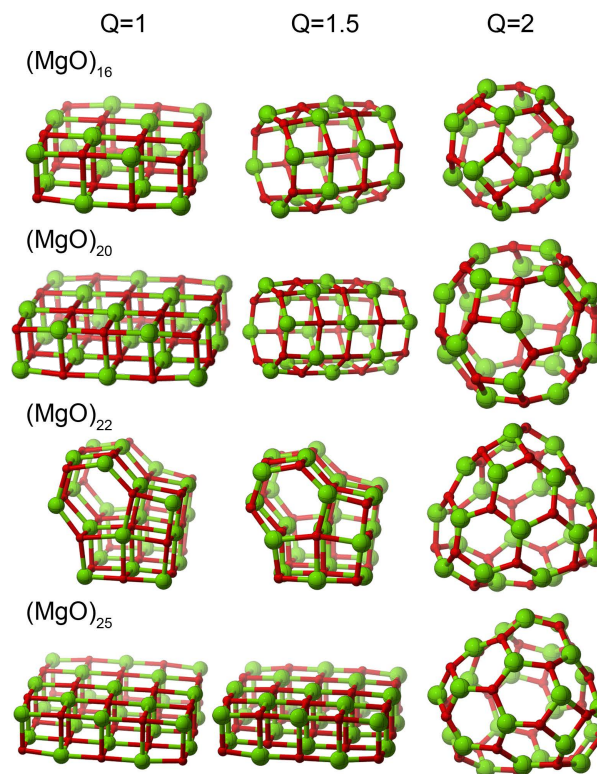
^a The reference energies are calculate by " $E_b \times N_{\text{ions}}$ " from Table 2 of Ref. 73. The small difference between the energy and the reference is due to the different precision of " $\frac{e^2}{4\pi\epsilon_0}$ " in (3).

From Table 3, the difficulty in locating the GM of LJ $_N$ increases with the size N . The ABC algorithm finds the true GM in g_{max} steps for all cases except for LJ $_{75}$. This is a very difficult case¹⁷ since its PES has two funnels: a broad and deep one corresponding to an icosahedral LM (with energy -396.2822 as we found), and a very narrow one leading to the Marks decahedral GM⁵⁰. Thus the ABC algorithm is easily trapped in the icosahedral funnel. This is of course a drawback of the algorithm, but it also suggests that the trapped region is of high entropy. Thus, at finite temperature the LM in this region may dominate. In Figure 5, the GM of LJ $_{80}$ is given as an example.

Table 3 Benchmark for the LJ potential. The difficult cases are marked with a "*"

ABC algorithm parameters: $SN = 100$, $g_{\text{limit}} = 5$, $g_{\text{max}} = 1000$				
Initial guess: random				
Potential	LJ $_N$	Steps	Energy (eV)	Reference ¹⁷ (eV)
U_{LJ}	13	1	-44.3268	-44.3268
	38	20	-173.9284	-173.9284
	40	2	-185.2498	-185.2498
	69	61	-359.8826	-359.8826
	74	19	-390.9085	-390.9085
	75(*)	86	-396.2822	-397.4923
	80	53	-428.0836	-428.0836
	100	43	-557.0398	-557.0398
	115	619	-655.7563	-655.7563
	150	224	-893.3103	-893.3103

The results for the Morse potential are also interesting. From Figure 3B we know that U_{ML2} is longer-ranged than U_{MS2} , which implies that the global optimization is easier for the former one. Table 4 reveals that it is in fact much easier: even for the largest cluster M_{80} with U_{ML2} , the GM is located at the first step! For M_N with U_{MS2} it may need several thousand steps, and for M_{55} and M_{80} the ABC algorithm can only locate a LM which is very close to the GM in energy in 15000 steps, suggesting that the LM is in a deep and broad funnel. Thus the impact of the force range on the search of the GM is emphasized here again. The GMs for the potentials of different force range can be quite different. Figure 5 suggests that the GM of M_{40} with U_{ML2} is icosahedral based while that with U_{MS2} is a fcc structure.

**Fig. 4** The GMs found by the ABC algorithm for the potentials U_{CBMQ1} , $U_{\text{CBMQ1.5}}$ and U_{CBMQ2} .

The finite force range and multimodality of the Z potential (see Figure 3C) leads to a multi-funnel and very rugged topography of its PES, and the GMs tend to be of quite unexpected shape. Indeed, Table 5 reveals that it is the most difficult case for GM search. Taking Z_{50} in Figure 5 as an example, the algorithm is easily trapped into a (relatively) compact LM funnel, while the GM is hollow!

3.3 The Gupta Potential

The Gupta potential is of many-body type and has been widely applied in the study of metals and alloys. Here we chose the Gupta potentials for platinum and zinc as testing cases. The reason for choosing Pt and Zn is that they tend to form regular and disordered GMs, respectively, representing two different kinds of potentials. Moreover, due to their functional forms, the PES could be very flat, leading to slow convergence. Thus, we used a small g_{limit} to accelerate the search. Also, for these potentials, especially for zinc whose GMs are disordered, the ABC algorithm may require quite a lot of steps to locate the GM starting from a random guess. Therefore for large clusters we use a seeded initial guess. For example, for Pt $_{40}$, we use an initial guess obtained by adding 2 atoms to the GM of Pt $_{38}$. The results are given in Table 6.

For platinum clusters, all the searches succeeded as long as good initial guesses were provided. For zinc clusters, however, it becomes more difficult. Even using a seeded guess cannot avoid the failures (not shown). Figure 3D reveals that the Gupta poten-

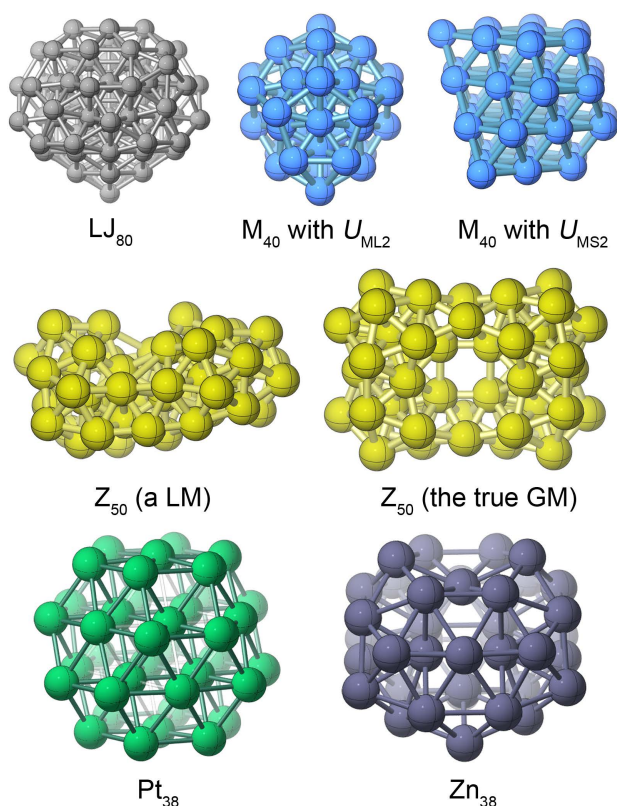


Fig. 5 Some GMs obtained by the ABC algorithm. The true GM of Z_{50} is obtained from Ref. 44.

tial for zinc is much shorter-ranged than that for platinum, which leads to a rugged PES and a disordered GM. This explains why the ABC algorithm has a poor performance for zinc clusters. In Figure 5, we can see that Pt_{38} has a fcc structure, but Zn_{38} is rather disordered.

3.4 Selection of the Algorithm Parameters

We show the minimal energy evolution during the optimization for some cases in Figure 6. We can see that for two-body potentials the optimization shows several plateaus. A plateau indicates that the optimization stagnates, being trapped in a LM funnel. For instance, the largest plateau of M_{60} with U_{MS2} (see Figure 6B) implies that this is a very broad and deep funnel.

For many-body potentials, the optimization exhibits strong oscillations. This is because the PES is so flat that the energies of the population are very close, thus even the “best” solutions are often updated by the scout bees. Nevertheless, the optimization shows an overall downhill movement, although it can be very slow. For Pt_{38} in Figure 6C, the optimization found a LM with energy -199.9001 at the 2042th step, however it took more than 20000 steps to locate the GM with energy -199.9418 !

This gives us some guidance on how to set the three algorithm parameters SN , g_{limit} and g_{max} in practice. According to our tests, a SN between 50 to 100 is sufficient. The g_{limit} can be set as 5 for two-body potentials, and 2 for many-body potentials. The g_{max} should be big for short-ranged potentials or large clusters to enable the algorithm to escape from LM funnels. In practice, it

Table 4 Benchmark for the Morse potential. The difficult cases are marked with a “*”

ABC algorithm parameters: $SN = 100$, $g_{limit} = 5$, $g_{max} = 1000$				
Initial guess: random				
Potential	M_N	Steps	Energy (eV)	Reference ⁷⁴ (eV)
U_{ML2}	13	1	-51.7370	-51.7370
	20	1	-97.4174	-97.4174
	30	1	-177.5786	-177.5786
	35	1	-221.7715	-221.7715
	40	1	-268.3948	-268.3948
	45	1	-318.6607	-318.6607
	50	1	-366.6356	-366.6356
	55	1	-417.9186	-417.9186
	60	1	-470.4485	-470.4485
80	1	-690.5779	-690.5779	
ABC algorithm parameters: $SN = 100$, $g_{limit} = 5$, $g_{max} = 15000$				
Initial guess: random				
U_{MS2}	13	1	-37.2589	-37.2589
	20	24	-64.7919	-64.7919
	30	3209	-106.8358	-106.8358
	35	1207	-129.7374	-129.7374
	40	707	-152.3337	-152.3337
	45	9436	-174.5116	-174.5116
	50	13324	-198.4556	-198.4556
	55 (*)	961	-219.6755	-220.6462
	60	4681	-244.5791	-244.5791
	80 (*)	4303	-340.8026	-340.8114

Table 5 Benchmark for the Z potential. The difficult cases are marked with a “*”

ABC algorithm parameters: $SN = 100$, $g_{limit} = 5$, $g_{max} = 20000$				
Initial guess: random				
Potential	Z_N	Steps	Energy (eV)	Reference ⁴⁴ (eV)
U_{Z1}	13	27	-30.5188	-30.5188
	20	38	-50.0067	-50.0067
	24	919	-60.9371	-60.9371
	30	641	-77.6403	-77.6403
	40 (*)	4502	-104.7577	-105.5638
	50 (*)	18156	-131.1707	-134.4295

is better to run a global optimization several times starting from different initial guesses, increasing the probability of locating the true GM.

4 Conclusion

In this work, we have successfully introduced a new unbiased, swarm intelligence based algorithm, i.e. the artificial bee colony (ABC) algorithm, to the field of global minimum search of clusters. Compared with other algorithms, it requires only three parameters: SN , g_{limit} and g_{max} , which are very easy to tune in practice. From the benchmarks, we see that the ABC algorithm exhibits excellent performance for long-ranged potentials. This makes it useful in searching the stable structures for ionic or dispersion bound clusters. For glassy and metallic clusters described by short-ranged or many-body potentials, the performance is not so encouraging since the algorithm is often trapped into some LM funnels. Of course, in principle if we set a very large g_{max} , we can always find the true GM for those difficult cases. Since the ABC algorithm is relatively new, it still has potential to be improved, e.g. combined with the basin-sampling strategy to overcome the difficulty of trapping in a LM funnel⁷⁵. Our laboratory is currently exploring this possibility.

The algorithm is very promising in quickly providing initial ge-

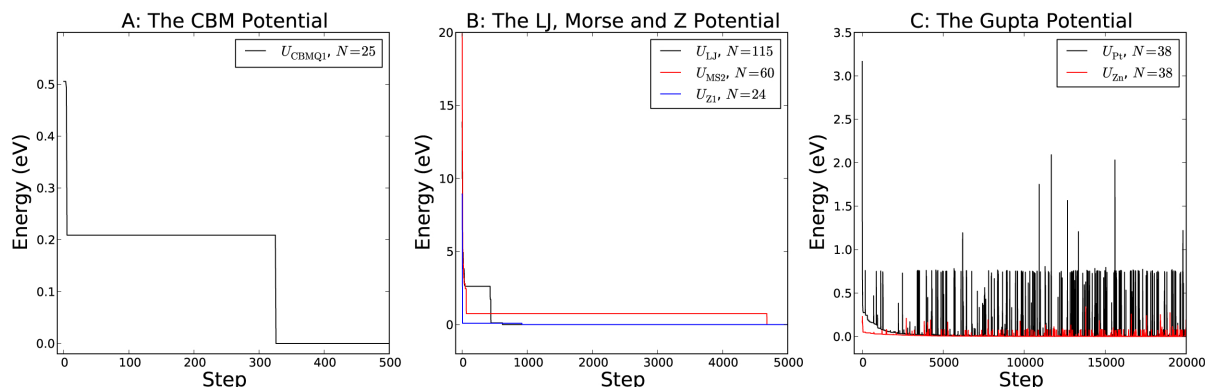


Fig. 6 The the minimal energy evolution during the optimization for some cases. On the Y axis is the difference between the minimal energy at that step and then energy of GM (i.e. $E(\text{Step}) - E_{\text{GM}}$).

Table 6 Benchmark for the Gupta potential. The difficult cases are marked with a “*”

ABC algorithm parameters: $SN = 50$, $g_{\text{limit}} = 2$, $g_{\text{max}} = 30000$				
Initial guess: random				
Potential	GN	Steps	Energy (eV)	Reference ^{53,61} (eV)
U_{Pt}	13	348	-64.9163	-64.9163
	20	23567	-101.9734	-101.9734
	30	24081	-156.2636	-156.2636
	38	22947	-199.9418	-199.9418
	40 (*)	23153	-210.5891	-210.6938
50 (*)	29985	-265.1495	-265.5774	
Initial guess: seeded				
U_{Pt}	40	8978	-210.6938	-210.6938
	50	14829	-265.5773	-265.5774
ABC algorithm parameters: $SN = 50$, $g_{\text{limit}} = 2$, $g_{\text{max}} = 30000$				
Initial guess: random				
U_{Zn}	13	4567	-17.0321	-17.0321
	20 (*)	251	-26.3868	-26.3954
	30	9847	-39.7682	-39.7682
	38	19304	-50.4929	-50.4999
	40 (*)	28378	-53.1587	-53.1821
	50 (*)	25441	-66.5748	-66.6019

ometries of small and medium-sized clusters for chemists’ further exploration. However, one should also realize that the phenomenological potentials (3)–(7) are often obtained by fitting bulk properties and thus may be inaccurate for clusters of small size. For instance, one can obtain GMs of quite different shape for lead clusters when different potentials are used⁷⁶. Thus, in a study the potential function must be carefully constructed and the obtained GMs should be interpreted with caution.

The ABC algorithm described in this work has been integrated into the program ABCluster, enabling both non-experts and experts to apply this algorithm readily. More potentials, or generally any “objective functions” can be integrated into the program, not limited to the ones discussed in this paper. Therefore our program is a useful tool in studying various kinds of clusters for researchers from chemistry and perhaps other fields. It is efficient, user-friendly, and free. The program ABCluster will be available on our group site.

References

- 1 D. J. Wales and H. A. Scheraga, *Science*, 1999, **285**, 1368–1372.

- 2 *Atomic and Molecular Clusters*, ed. R. L. Johnston, Taylor and Francis, London, UK, 2002, pp. 1–11.
- 3 H. Häkkinen, *Nat. Chem.*, 2012, **4**, 443–455.
- 4 E. Roduner, C. Jensen, J. van Slageren, R. A. Rakoczy, O. Larlus and M. Hunger, *Angew. Chem., Int. Ed.*, 2014, **53**, 4318–4321.
- 5 E. Pahl, F. Calvo, L. Koči and P. Schwerdtfeger, *Angew. Chem., Int. Ed.*, 2008, **47**, 8207–8210.
- 6 J. W. Mullin, *Crystallization*, Reed Educational and Professional Publishing Ltd, Oxford, 2001, pp. 216–288.
- 7 H. Terrones and M. Terrones, *New J. Phys.*, 2003, **5**, 126.
- 8 N. Watari and S. Ohnishi, *J. Chem. Phys.*, 1997, **106**, 7531–7540.
- 9 J. Zhang, N. Heinz and M. Dolg, *Inorg. Chem.*, 2014, **53**, 7700–7708.
- 10 T. Erber and G. M. Hockney, *Phys. Rev. Lett.*, 1995, **74**, 1482–1482.
- 11 R. Zwanzig, A. Szabo and B. Bagchi, *Proc. Natl. Acad. Sci. U.S.A.*, 1992, **89**, 20–22.
- 12 C. Levinthal, *J. Chim. Phys.*, 1968, **65**, 44–45.
- 13 C. Levinthal, *Mössbauer Spectroscopy in Biological Systems: Proceedings of a meeting held at Allerton House, Monticello, Illinois*, University of Illinois Press, 1969, pp. 22–24.
- 14 D. Romero, C. Barrón and S. Gómez, *Comput. Phys. Commun.*, 1999, **123**, 87–96.
- 15 Y. Xiang, H. Jiang, W. Cai and X. Shao, *J. Phys. Chem. A*, 2004, **108**, 3586–3592.
- 16 Y. Xiang, L. Cheng, W. Cai and X. Shao, *J. Phys. Chem. A*, 2004, **108**, 9516–9520.
- 17 D. J. Wales and J. P. K. Doye, *J. Phys. Chem. A*, 1997, **101**, 5111–5116.
- 18 M. T. Oakley, R. L. Johnston and D. J. Wales, *Phys. Chem. Chem. Phys.*, 2013, **15**, 3965–3976.
- 19 Z. Li and H. A. Scheraga, *J. Mol. Struct. (Theochem)*, 1988, **179**, 333–352.
- 20 L. Wille, *Chem. Phys. Lett.*, 1987, **133**, 405–410.
- 21 A. Eiben and J. Smith, *Introduction to Evolutionary Computing*, Springer, Berlin, 2003.

- 22 J. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, University of Michigan Press, Oxford, England, 1975.
- 23 R. Storn and K. Price, *Differential Evolution – A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces*, Technical report, International Computer Science Institute, Berkeley, 1995.
- 24 D. M. Deaven and K. M. Ho, *Phys. Rev. Lett.*, 1995, **75**, 288–291.
- 25 R. L. Johnston, *Dalton Trans.*, 2003, 4193–4207.
- 26 S. Heiles and R. L. Johnston, *Int. J. Quantum Chem.*, 2013, **113**, 2091–2109.
- 27 R. Eberhart, Y. Shi and J. Kennedy, *Swarm Intelligence*, Morgan Kaufmann, California, 2001.
- 28 J. Kennedy and R. Eberhart, *Proc. IEEE Int. Conf.*, 1995, **4**, 1942–1948.
- 29 Y. Wang, J. Lv, L. Zhu and Y. Ma, *Phys. Rev. B*, 2010, **82**, 094116.
- 30 D. Schebarchov and D. J. Wales, *The Journal of Chemical Physics*, 2013, **139**, –.
- 31 D. Schebarchov and D. J. Wales, *Phys. Rev. Lett.*, 2014, **113**, 156102.
- 32 D. Schebarchov and D. J. Wales, *Phys. Chem. Chem. Phys.*, 2015, –.
- 33 <http://www-wales.ch.cam.ac.uk/software.html>. Access time: Jun. 26, 2015.
- 34 <https://www.ogolem.org>. Access time: Jun. 26, 2015.
- 35 http://www.fkf.mpg.de/452065/employee_page?c=3732&employee_id=31475. Access time: Jun. 26, 2015.
- 36 See, e.g. <https://wiki.fysik.dtu.dk/ase/tutorials/minimahopping/minimahopping.html>. Access time: Jun. 26, 2015.
- 37 D. Karaboga, *An Idea Based on Honey Bee Swarm for Numerical Optimization*, Technical Report TR06, Erciyes University, 2005.
- 38 M. J. Elrod and R. J. Saykally, *Chem. Rev.*, 1994, **94**, 1975–1997.
- 39 J. Zhang and M. Dolg, *J. Chem. Theory Comput.*, 2013, **9**, 2992–3003.
- 40 J. A. Barker and M. L. Klein, *Phys. Rev. B*, 1973, **7**, 4707–4712.
- 41 M. Born and J. E. Mayer, *Z. Physik*, 1932, **75**, 1–18.
- 42 J. Lennard-Jones, *Proc. R. Soc. Lond. A.*, 1924, **106**, 463–477.
- 43 P. Morse, *Phys. Rev.*, 1929, **34**, 57–64.
- 44 J. P. K. Doye, D. J. Wales, F. H. M. Zetterling and M. Dzugutov, *J. Chem. Phys.*, 2003, **118**, 2792–2799.
- 45 M. Dzugutov, *Phys. Rev. A*, 1992, **46**, R2984–R2987.
- 46 M. Dzugutov, *J. Non-Cryst. Solids*, 1993, **156-158**, 173–176.
- 47 R. P. Gupta, *Phys. Rev. B*, 1981, **23**, 6265–6270.
- 48 J. P. K. Doye and D. J. Wales, *J. Phys. B*, 1996, **29**, 4859.
- 49 J. P. K. Doye and D. J. Wales, *Phys. Rev. B*, 1999, **59**, 2292–2300.
- 50 J. P. K. Doye, M. A. Miller and D. J. Wales, *J. Chem. Phys.*, 1999, **111**, 8417–8428.
- 51 J. P. K. Doye and D. J. Wales, *J. Chem. Phys.*, 1996, **105**, 8428–8445.
- 52 M. A. Miller, J. P. K. Doye and D. J. Wales, *Phys. Rev. E*, 1999, **60**, 3701–3718.
- 53 J. P. K. Doye, *Phys. Rev. B*, 2003, **68**, 195418.
- 54 J. P. K. Doye and D. J. Wales, *Phys. Rev. Lett.*, 1998, **80**, 1357–1360.
- 55 G. M. Morris, D. S. Goodsell, R. S. Halliday, R. Huey, W. E. Hart, R. K. Belew and A. J. Olson, *J. Comput. Chem.*, 1998, **19**, 1639–1662.
- 56 D. Karaboga and B. Akay, *Appl. Math. Comput.*, 2009, **214**, 108 – 132.
- 57 B. Verma and D. Kumar, *Int. J. Eng. Tech.*, 2013, **2**, 175–186.
- 58 D. Karaboga, B. Gorkemli, C. Ozturk and N. Karaboga, *Artif. Intell. Rev.*, 2014, **42**, 21–57.
- 59 D. Liu and J. Nocedal, *Math. Program.*, 1989, **45**, 503–528.
- 60 H.-Y. Fan and J. Lampinen, *J. Global Optim.*, 2003, **27**, 105–129.
- 61 Z. Chen, X. Jiang, J. Li, S. Li and L. Wang, *J. Comput. Chem.*, 2013, **34**, 1046–1059.
- 62 S. Das and P. N. Suganthan, *IEEE Trans. Evol. Comput.*, 2011, **15**, 4–31.
- 63 W. Gao and S. Liu, *Inform. Process. Lett.*, 2011, **111**, 871 – 882.
- 64 E. Bonabeau, M. Dorigo and G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems*, Oxford University Press, New York, NY, 1999.
- 65 C. Schiffmann and D. Sebastiani, *J. Chem. Theory Comput.*, 2011, **7**, 1307–1315.
- 66 B. Li, Y. Li and L. Gong, *Eng. Appl. Artif. Intell.*, 2014, **27**, 70 – 79.
- 67 C. Wehmeyer, G. F. von Rudorff, S. Wolf, G. Kabbe, D. Schärf, T. D. Kühne and D. Sebastiani, *J. Chem. Phys.*, 2012, **137**, 194110.
- 68 G. F. von Rudorff, C. Wehmeyer and D. Sebastiani, *Comput. Phys. Comm.*, 2014, **185**, 1639–1646.
- 69 *CYLview, 1.0b*; Legault, C. Y., *Université de Sherbrooke*, 2009 (<http://www.cylview.org>).
- 70 G. V. Lewis and C. R. A. Catlow, *J. Phys. C*, 1985, **18**, 1149.
- 71 L. O. Paz-Borbón, R. L. Johnston, G. Barcaro and A. Fortunelli, *J. Phys. Chem. C*, 2007, **111**, 2936–2941.
- 72 F. Cleri and V. Rosato, *Phys. Rev. B*, 1993, **48**, 22–33.
- 73 C. Roberts and R. L. Johnston, *Phys. Chem. Chem. Phys.*, 2001, **3**, 5024–5034.
- 74 J. P. K. Doye and D. J. Wales, *J. Chem. Soc., Faraday Trans.*, 1997, **93**, 4233–4243.
- 75 D. J. Wales, *Chem. Phys. Lett.*, 2013, **584**, 1–9.
- 76 J. P. Doye, *Comput. Mater. Sci.*, 2006, **35**, 227 – 231.