Computer Physics Communications 184 (2013) 1172–1182

Contents lists available at SciVerse ScienceDirect

Computer Physics Communications

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

New developments in evolutionary structure prediction algorithm USPEX

Andriy O. Lyakhov [a,b,c,*], Artem R. Oganov [a,b,c,d], Harold T. Stokes [e], Qiang Zhu [a,b,c]

[a] Department of Geosciences, Stony Brook University, Stony Brook 11794-2100, NY, USA
[b] Department of Physics and Astronomy, Stony Brook University, Stony Brook 11794-2100, NY, USA
[c] New York Center for Computational Sciences, Stony Brook University, Stony Brook 11794-2100, NY, USA
[d] Geology Department, Moscow State University, 119992, Moscow, Russia
[e] Department of Physics and Astronomy, Brigham Young University, Provo 84602, UT, USA

ARTICLE INFO

Article history:
Received 4 May 2012
Received in revised form
7 December 2012
Accepted 10 December 2012
Available online 21 December 2012

Keywords:
Crystal structure prediction
Cluster structure prediction
Particle swarm optimization
Evolutionary algorithms
Genetic algorithms
Global optimization
Fingerprint function

ABSTRACT

We present new developments of the evolutionary algorithm USPEX for crystal structure prediction and its adaptation to cluster structure prediction. We show how to generate randomly symmetric structures, and how to introduce 'smart' variation operators, learning about preferable local environments. These and other developments substantially improve the efficiency of the algorithm and allow reliable prediction of structures with up to ~200 atoms in the unit cell. We show that an advanced version of the Particle Swarm Optimization (PSO) can be created on the basis of our method, but PSO is strongly outperformed by USPEX. We also show how ideas from metadynamics can be used in the context of evolutionary structure prediction for escaping from local minima. Our cluster structure prediction algorithm, using the ideas initially developed for crystals, also shows excellent performance and outperforms other state-of-the-art algorithms.

1. Introduction

Evolutionary algorithms are routinely used for global optimization. Recently, such algorithms [1–6] became widely used (e.g., [7–9]) in the field of crystal structure prediction (CSP), where one has to find a crystal structure with the lowest free energy[1] for a given chemical composition at given pressure–temperature conditions (for an overview, see [12,13]). The First Blind Test for inorganic crystal structure prediction [12], conducted in 2010, showed that the evolutionary algorithm USPEX [3–5] is the most efficient and reliable method. In this paper, we present new developments of USPEX, extending its applicability to different types of systems and improving its efficiency for large systems.

The basic idea of the evolutionary approach is to start from a set of structures, called population, and evolve them using selection and specially designed variation operators—i.e. recipes for creating offspring from parent structures. Variation operators, such as heredity (creates a child structure from two or more parents) and mutation (creates a child from a single parent), must retain some essential memory of parent structures in the offspring, while introducing new structural features. The flowchart of a typical evolutionary algorithm is shown in Fig. 1. The initial population is usually created randomly, unless some information about the ground state of the system is known, such as likely candidate structures, or lattice parameters, space group, etc. All structures created by the algorithm have to be locally optimized (i.e. relaxed) at a chosen level of theory—e.g., empirical potentials, density functional theory, hybrid functionals, or Quantum Monte Carlo.

As the system size increases, certain difficulties arise related to the very nature of the task that is being solved [5]. A major problem for most population-based global optimization methods is the risk of being trapped in some local minimum instead of the global one. The number of local minima rises exponentially with the system size and so does (for any algorithm) the risk of trapping, thus one has to develop techniques to avoid this. In evolutionary algorithms, such 'genetic drift' [14] is caused by the fact that the current best solution tends to create children that are similar to it. However, this behaviour is another side of learning, which drives the initial

* Corresponding author at: Department of Geosciences, Stony Brook University, Stony Brook 11794-2100, NY, USA. Tel.: +1 6315766330; fax: +1 631 6328240.
E-mail addresses: andriy.lyakhov@gmail.com, andriy.lyakhov@stonybrook.edu (A.O. Lyakhov).

[1] It is possible to optimize properties other than free energy, and USPEX supports such optimization [10,11]. For clarity, all explanations here assume free energy as fitness (i.e., search space = energy landscape).

0010-4655/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.cpc.2012.12.009

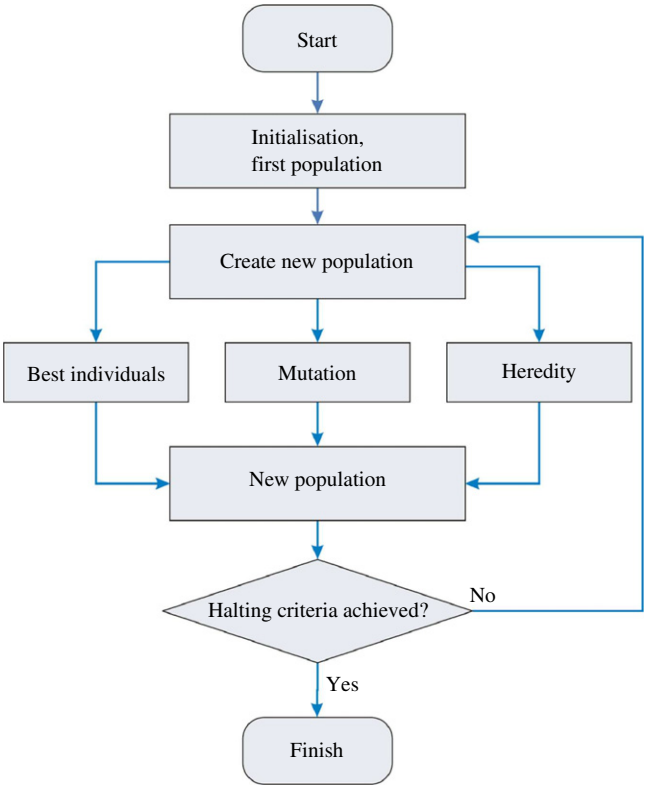

Fig. 1. Flowchart of a typical evolutionary algorithm.

random population to lower-energy structures. Thus we have to solve the drift problem without sacrificing the learning, if possible. A few key ideas of how to avoid sticking to a local minimum were already proposed in [5]. In this paper, we develop them further.

Exponential increase of the combinatorial complexity of the problem leads to the decline in the quality and diversity of random structures with the increasing number of degrees of freedom. To deal with it we proposed a new initialization scheme [5], the idea being to reduce the number of degrees of freedom by randomly applying symmetry to randomly generated atomic coordinates. To that end, we used additional translational symmetries and pseudosymmetries ('subcells' and 'pseudo-subcells') [5], and here we add to this trick the full apparatus of 230 space groups. This has the advantages of being unbiased, providing very diverse structures and covering the search space better. In this procedure, during relaxation the initial symmetry of the structures can increase or decrease (to allow symmetry breaking, we apply slight perturbations to the atomic positions[2]). Variation operators also break symmetry and enable totally new structures with very different symmetries to emerge. It is often beneficial to add random symmetric structures in every generation. Just improving initialization does not fully solve the problems of "curse of dimensionality". One has to improve the variation operators—so that not only good structures are given preference for creating offspring, but also good fragments of structures are identified and nurtured. We apply these ideas both to 3D-crystals and to clusters, and in both areas the method presented here outperforms other approaches.

We have also experimented with the adaptation of ideas from other global optimization techniques. We show that with minor programming effort the USPEX method can be transformed into a new form of the Particle Swarm Optimization (PSO) [15–17] method. We also implemented the ideas of metadynamics [18] into our algorithm in the form of so called 'antiseeds' to avoid the genetic drift. We discuss how these ideas affect the performance of the algorithm.

The remainder of this paper is organized as follows. In Section 2 we discuss the new initialization scheme. 'Smart' evolutionary operators are described in Section 3. In Section 4 we discuss the implementation of the PSO algorithm and the use of ideas of metadynamics in our evolutionary approach. An algorithm for cluster structure prediction is described in Section 5. Section 6 presents numerical results to measure performance of the ideas introduced here versus other methods. For completeness, Section 7 mentions other developments that have been implemented in the USPEX code and described elsewhere. Finally, we summarize the paper with conclusions.

2. Improved initialization

A fully random initialization of the first generation is a poor choice for large systems, as was shown and discussed in [5]. For large systems it leads to nearly identical 'glassy' structures that have similar (high) energies and low degree of order [19]. From such starting conditions, it is difficult to obtain well ordered crystalline states. We proposed [5] a novel initialization scheme that involved cell splitting—i.e. reduced the number of degrees of freedom (only at the initialization stage) by introducing additional translational symmetry or pseudosymmetry. Here we extend this idea by using 230 space groups, which allows one to produce a diverse initial population of ordered and chemically reasonable structures. The idea to use symmetric structures in the first generation has led to success before [5,17,20,21], and with the development presented here reaches its logical completion.

To generate a random symmetric structure, one of 230 space groups (or one of a user-provided list of groups) compatible with the number of atoms in the unit cell, is selected. A unit cell shape consistent with that symmetry is then produced, and rescaled to have the prespecified initial volume. An atom is randomly placed on a general Wyckoff position and is multiplied by space group symmetry operations. If two or more symmetry-related atoms are closer to each other than a user-defined threshold, we merge them into one atom on a special Wyckoff position by averaging their coordinates. This is equivalent to projecting the initial site onto a symmetry element (point, axis, plane), or, equivalently—into a high-symmetry Wyckoff position and multiplying it afterwards; see Fig. 2. At each step we make sure that the remaining number of atoms is compatible with a chosen space group, which sometimes requires discarding an atom with its images. Adding one by one new atoms on general positions (some of which are then moved into special positions using the above procedure), we ensure that no atoms, including symmetrically unrelated ones, are too close to each other. To have exactly the needed number of atoms in the unit cell, some atoms may have to be placed on special Wyckoff positions with the right multiplicity. Unlike implementations [17,21], this general and unbiased procedure not only fairly samples all possible space groups, but also naturally tends to place atoms on special positions, and ensures reasonable distances between all atoms. If symmetrically related atoms are closer than the user-defined threshold, they are merged into a special position; if symmetrically non-equivalent atoms are too close to each other, the structure is discarded without relaxation.

Relaxation (local optimization) does not break symmetry, but can increase it to a supergroup. To allow the possibility of symmetry breaking, we apply a small random displacement to atomic positions. We should also add that variation operators break symmetry, which removes any bias that may have been caused by the initial choice of symmetries.

[2] It makes most sense to introduce these perturbations after the structure is already relaxed, i.e. close to a local minimum or a saddle point.

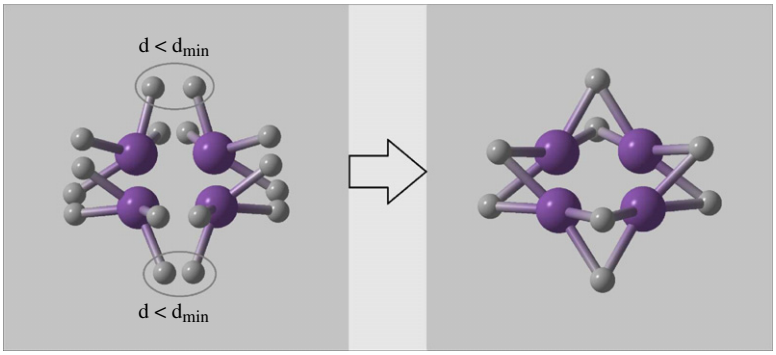

Fig. 2. Example of merging atoms onto special Wyckoff positions.

Since the implementation of symmetric random structures substantially improved the algorithm, we also found it in most cases beneficial to include a few new symmetric random structures in every generation to improve the diversity of the population and provide new structural motifs to be explored by the algorithm.

3. Smart variation operators

Variation operators are the heart of an evolutionary algorithm. They are used to produce new structures out of those that were already sampled and drive the search towards the globally optimal structure. The random nature of variation operators is both a blessing and a curse. On the one hand, randomness is required for diversity of the population and is essential for exploring the whole search space. It also ensures unbiasedness and slows down *genetic drift* [14]. That is why random mutations and random selection of spatially coherent slabs for heredity are preferred. On the other hand, completely random variation operators may be inefficient when applied to large systems [5]. Therefore, we have to design 'smart' operators that retain a degree of randomness, but are directed by the system itself to choices that have a higher probability to improve the fitness. The use of fingerprint theory for improving the conventional variation operators was thoroughly explored in [5]; here we build up on these developments and discuss a novel operator, that we call soft-mode mutation (or 'softmutation').

3.1. Improved heredity using order-fitness correlation

In the heredity operator, spatially coherent slabs are taken from parent structures and then combined into a single child structure [3]. Earlier [5] we proposed to use the concept of local order to improve variation operators. Order parameter quantifies the degree of symmetry of the environment of a given atom, and one can also define the average degree of order of a fragment of structure. Thus, the less distorted and less defective parts of the structure can be potentially chosen for participating in heredity (and the more defective ones—in mutation). If distortions or defects are favourable, preferences should be reversed. To incorporate both situations in a seamless manner, we calculate the correlation between the average atomic order of the structure and its energy. Then, depending on the strength of (anti)correlation, the extent to which the order parameter influences variation operators is defined. When there is a positive (negative) correlation, we make N_s attempts to randomly cut slabs from each parent and among these slabs choose the least (most) ordered one:

$$N_s = \frac{L}{L_{\text{char}} + (L - L_{\text{char}}) \cdot \cos\left(\frac{\pi}{2} r\right)^2} \tag{1}$$

from which N_s is rounded to the nearest integer. Here $r \in [-1; 1]$ is Pearson product-moment correlation coefficient, L is the thickness of the unit cell in the direction in which we cut it for heredity and L_{char} is the characteristic length that can be defined, e.g., as half of the cubic root of the average atomic volume. Therefore N_s changes from 1 in the absence of correlation ($r = 0$) to L/L_{char} for perfect (anti)correlation; see Fig. 3.

3.2. Softmutation

In our previous work [5] we have briefly described softmutation, a special coordinate mutation operator. The idea is to move the atoms along the eigenvectors of the softest modes, since low-frequency eigenmodes correspond to directions of low curvature of the energy surface and likely also low energy barriers [22]. As Roy et al. have shown [23], crossing low energy barriers will, on average, lead into the basin of attraction of lower-energy local minima compared to crossing high energy barriers. The physical idea behind such mutation is to move the system into a new structure across the lowest energy barrier. Such moves enable a very effective local and semilocal exploration of the energy landscape. The power of this operator is illustrated in Fig. 4 (and also Fig. 8 later in the text).

For this operator, the amplitude of displacement is a user-defined parameter, usually set to ∼1.5 times the average bond length. If the displacement brings atoms too close to each other, the amplitude of displacement is decreased until minimum-distance constraints are satisfied. If a structure has already been softmutated, the next lowest-frequency[3] mode is used. From a set of degenerate modes, only one mode is used; the mode degeneracy is conveniently analyzed using fingerprints [19].

It is important to note that it is not necessary to move the structure *exactly* along the mode eigenvector. To overcome the energy barrier and end up in another minimum, it is enough to have an approximate direction and sufficiently large mutation amplitude to arrive upon relaxation in a new low-energy structure. Since one does not need an exact and computationally expensive *ab initio* dynamical matrix [25] to calculate the phonon modes, we proposed [5] to construct this matrix from bond hardness coefficients [26] (and setting atomic masses to unity); at zero wavevector this dynamical matrix is:

$$D_{\alpha\beta}(a, b) = \sum_m \frac{\partial^2}{\partial \alpha_a^0 \partial \beta_b^m} \left(\frac{1}{2} \sum_{i,j,l,n} H_{i,j}^{l,n} \left(r_{i,j}^{l,n} - r_{0\,i,j}^{l,n} \right)^2 \right) \tag{2}$$

[3] More precisely, one arranges modes in order of increasing square of mode eigenfrequency (ω^2), starting from lowest-w^2 and neglecting long-wavelength acoustic modes.

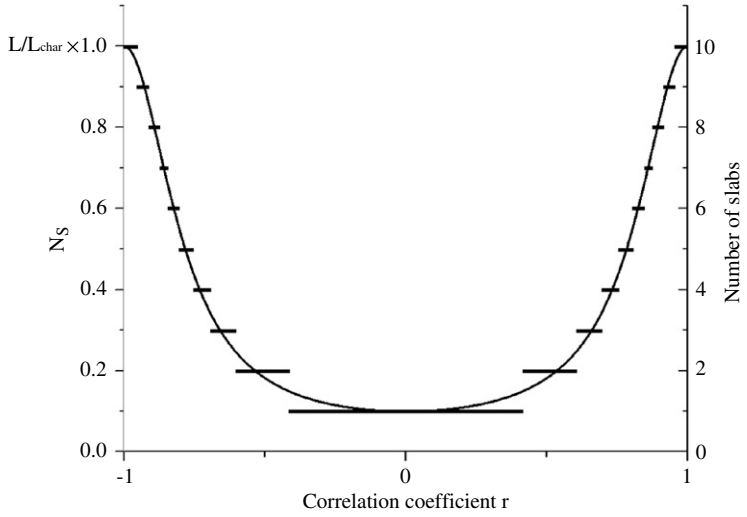

Fig. 3. Number of attempts to cut slabs for heredity as a function of the order-fitness correlation coefficient. Solid line (left axis) shows function (1). Right axis and step function show how function (1) determines the number of slabs for a particular case $L/L_{\text{char}} = 10$.

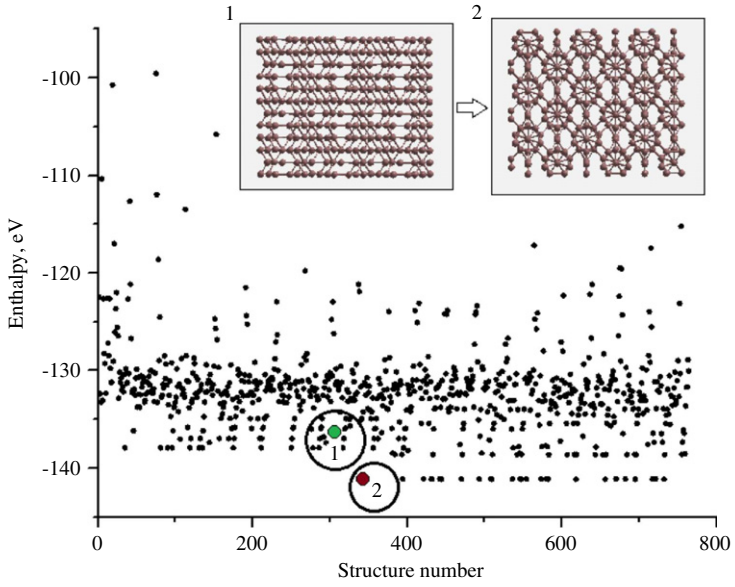

Fig. 4. Fully unconstrained variable-cell structure search for boron at 40 GPa with 28 atoms in the unit cell. Softmutation with subsequent relaxation transforms structure **1** into structure **2**, which is the global minimum ($\gamma - B_{28}$ [8]). One can clearly see that softmutation acts non-trivially and greatly improved the structure. Previous attempts to find $\gamma - B_{28}$ using evolutionary algorithms without fixed cell parameters found this task exceedingly difficult [24].

where coefficients α, β denote coordinates (x, y, z); coefficients a, b, i, j describe the atom in the unit cell; coefficients l, m, n describe the unit cell number; $r_{i,j}^{l,n}$ is the distance between atom i in the unit cell l and atom j in the unit cell n, while $r_{0\,i,j}^{l,n}$ is the corresponding bond distance. $H_{i,j}^{l,n}$ is bond hardness coefficient between the atom i in the unit cell l and atom j in the unit cell n. Bond hardness coefficients are computed from bond lengths and tabulated atomic properties—covalent radii and electronegativities [27]. The approximation (2) corresponds to a central pairwise harmonic potential with *a priori* determined force constants. In essence, we use a simple mechanistic analogy: atoms are treated as point particles and bonds as springs that connect them with a stiffness determined by $H_{i,j}^{l,n}$.

Such 'classical' matrix (1) combines very low computational costs with surprisingly good results. More accurate (and expensive) ways of constructing the dynamical matrix could be used. However, this is hardly needed, given the excellent performance of the approximate treatment. Dynamical matrix can also be used for another coordinate mutation operator where atoms are moved in random directions with amplitudes determined by their thermal ellipsoids, which can also be calculated from $D_{\alpha\beta}(a, b)$. Another possibility is to create linear combinations (with random weights) of low-frequency mode eigenvectors.

3.3. Clustering

It is important to keep the population diverse and enable sampling of all low-energy regions of the landscape (energy funnels). For this, we allow a number of structures, at the same time as diverse and as low-energy as possible, to survive into the next generation. This is achieved by combining fingerprint theory [19] with an appropriate clustering algorithm. After relaxation, the lowest-energy part of the population (fraction determined by the user) is clustered into groups using some value of the distance threshold $d_{\min}$: (1) first group is formed from structures with the cosine distance [19] to the best structure smaller than $d_{\min}$; (2) this process is

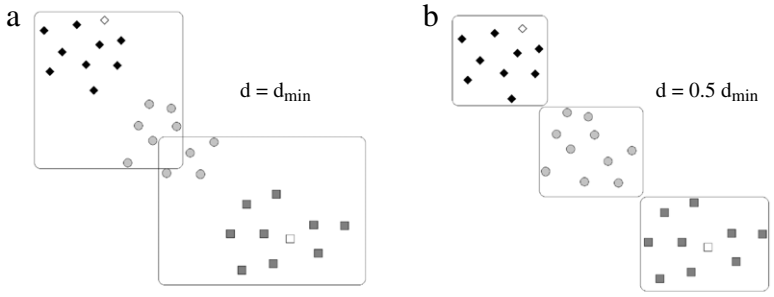

Fig. 5. Illustration of clustering into 3 groups. The open square is the best structure in the population, the open diamond is the 2nd best structure. (a) Initial grouping into two clusters. (b) Correct grouping into 3 clusters.

repeated with all ungrouped structures until all the selected structures belong to some group.

One could use a fixed user-defined $d_{\min}$, but we prefer an automated dynamical criterion. We propose to use a dichotomic algorithm similar to a binary search in a sorted array. If the number of such groups differs from the desired, the threshold is modified by an addition/subtraction of d_i, and reclustered. Initially d_i is equal to $0.5 \cdot d_{\min}$ and on every reclustering step it is reduced by half. After we obtain the desired number of groups, the best structure from every group survives into the next generation; see Fig. 5.

4. Combining ideas of particle swarm optimization (PSO) and metadynamics with USPEX

4.1. PSO

In the field of crystal and cluster structure prediction, several approaches proved to be successful for small systems. Particle Swarm Optimization (PSO), pioneered in this field by Boldyrev [16], is a special class of evolutionary algorithms where a population (swarm) of candidate solutions (called "particles") is moved in the search space according to a few simple formulae. The movements of the particles are guided by their own best known position in the search space as well as the entire swarm's best known position. Initially, the coordinates x and 'velocities' v of the particles are generated randomly. Then at every step, the positions and velocities are updated according to the formulae:

$$\begin{aligned} v_i' &= \omega \cdot v_i + \varphi_p \cdot r_p \cdot (p_i - x_i) + \varphi_g \cdot r_g \cdot (g - x_i) \\ x_i' &= x_i + v_i'. \end{aligned} \tag{3}$$

Here ω, φ_p and φ_g are weight factors that control the behaviour and efficiency of the PSO algorithm; r_p and r_g are random numbers in the [0; 1] range generated separately for every particle at every step; p_i is the best known position of particle i and g is the best known position of the entire swarm.

Such an algorithm, despite its simplicity, can work [16]. Key points to improve with respect to previous implementations [16,17] are (1) metric of the search space (it is not trivial to map crystal structure uniquely onto a coordinate system) and (2) ways to evolve structures in PSO—i.e. variation operators.

Clearly, the cell parameters and atomic coordinates cannot be used as a correct search space metric. Each structure would then correspond to an infinite number of representations, or 'PSO particles', since the choice of the unit cell and its origin is not unique. Therefore evolving the particles by determining the speed v_i (3) directly from coordinates of the atoms and cell parameters of two structures (as in [17]) will not be productive. Our solution is to use fingerprint distances [19] as the most natural metric for the energy landscape, and variation operators of USPEX for evolving the 'PSO particles' (i.e. structures) as the most efficient unbiased ways to evolve a population of structures. Namely, the particle is either mutated (to imitate a random move), or participates in heredity with its best known position or in the heredity with the best known population position (to imitate PSO moves in the direction of these positions). Instead of applying at each step all moves with some weights (see Eq. (3)), we apply them one at a time with *probabilities* described by formulae:

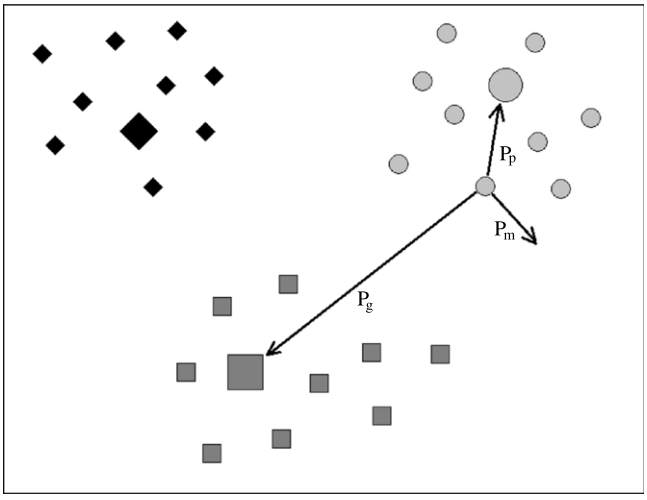

Fig. 6. Illustration of PSO–USPEX hybrid algorithm for the population of three individuals (marked by diamonds, squares and circles) after 10 generations. Best position for every particle is marked by an enlarged symbol. The best structure is the big square. The structure shown by circle can be either mutated, create a child with its historically best position (large circle) or the best position of entire population (large square) using heredity operator with probabilities P_m, P_p and P_g, respectively.

$$\begin{aligned} &P_m = \frac{\omega}{\Sigma}; \qquad P_p = \frac{\varphi_p \cdot r_p \cdot D_p}{\Sigma}; \\ &P_g = \frac{\varphi_g \cdot r_g \cdot D_g}{\Sigma}; \quad \Sigma = \omega + \varphi_p \cdot r_p \cdot D_p + \varphi_g \cdot r_g \cdot D_g, \end{aligned} \tag{4}$$

where D_p is a fingerprint distance between current and best position of a particle, while D_g is a fingerprint distance between the current position of the particle and best known position of the entire population. For more information about the fingerprint metric space and how to measure the degree of similarity between structures, see [19]. As in conventional PSO, the position of a particle is influenced by its historically best known position, and by the globally (across the whole swarm) best known position—but in different ways; see Fig. 6. Our tests, performed on a few diverse systems, show that this approach (which we call "cor-PSO", i.e. corrected PSO) is relatively successful, though it cannot compete with the USPEX algorithm [3,5] for success rate or efficiency.

4.2. Antiseeds

Metadynamics is an interesting method for sampling the energy landscape [18]. Its core idea, a form of conformational flooding [28], is to add Gaussian potentials to the real energy landscape of the system at all sampled points. This is done to discourage the already sampled regions of the search space from being visited again, and to explore other regions of the landscape. This idea, combined with the ideas of USPEX, has led to a very powerful method—evolutionary metadynamics [29], which combines the strengths of both approaches. Here we experiment with another idea inspired by metadynamics: every structure produced by USPEX is stored together with two parameters—the width and the heights of the Gaussian. We call such structures "antiseeds".[4] The Gaussian parameters evolve with time and depend on the diversity of the population at the moment of antiseed creation. To create an additional evolutionary pressure on already sampled structures, the fitness of structure i is then modified using the formula

$$f_i = f_{i0} + \sum_a W_a \exp\left(-\frac{D_{ia}^2}{2\sigma_a^2}\right), \tag{5}$$

where f_{i0} is the initial fitness function (for example, the energy), summation goes over all stored antiseeds that have heights W_a and width σ_a, D_{ia} is the distance between the current structure and antiseed structure in the fingerprint metric space [19]. Parameter σ_a is proportional to the average cosine distance between fingerprints of all pairs of structures in the population (for example, $0.05\langle d\rangle$). Parameter W_a is proportional to the fitness variance (for example, $0.01\sigma(E)$). As we will show below, antiseeds can enhance the success rate of the algorithm, especially when applied to systems with complex multifunnel energy landscapes. Modification of fitness described by Eq. (5) introduces ageing (and, eventually, death) of individuals in the evolutionary process. As an alternative to ageing via antiseeds, one could also introduce the direct ageing of the surviving structures by lowering their probability to be selecting as parents every time they survive into the next generation [30].

5. Cluster structure prediction

Cluster (nanoparticle) structure prediction has similarities to crystal structure prediction, but there are also critical differences. For consistency with most widely used codes, we treat the problem with periodic boundary conditions. This gives reliable results if there is enough 'vacuum' around the nanoparticle to eliminate its interaction with its periodic images. Variation operators do not act on empty space, and one operates with the "small cell"—a minimal rectangular parallelepiped built around the nanoparticle; see Fig. 7. The size of this "small cell" is adjusted after relaxation. The thickness of the vacuum region around the cluster is a user-defined parameter; more vacuum means more accurate results, but (for some approaches, such as plane-wave methods) greater computational costs.

When the cluster is generated or relaxed, we place its centre of mass to the centre of the unit cell and rotate it so that principal moment of inertia axis with the highest moment is pointed in the z-direction. Before performing a 'cut-and-splice' heredity, the nanoparticles are randomly rotated around a random axis that goes through their centre of mass. This idea is similar to random 'shifts' for heredity in crystal structure prediction [3]. As in CSP algorithms, we cut the nanoparticles close to their centres so that not less than 30% of atoms from each parent participate in heredity.

[4] By analogy with "seeds"—input structures aimed at quickly directing search to chemically reasonably structures.

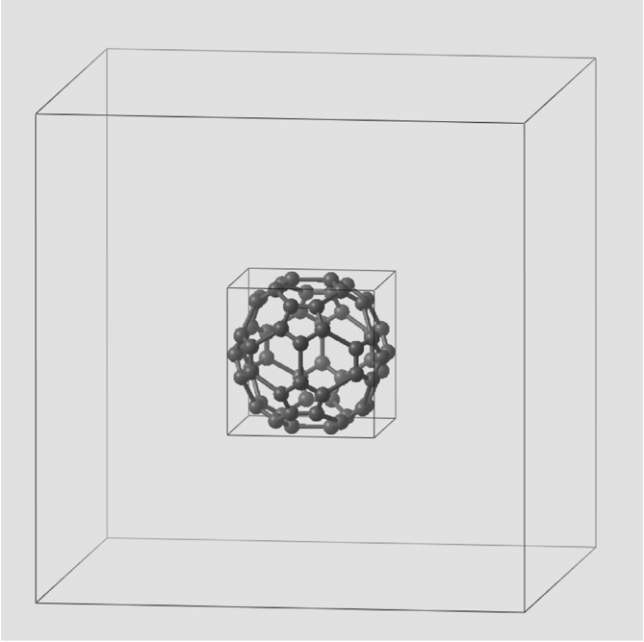

Fig. 7. Cluster in a 'small cell' surrounded by vacuum ('big cell'). Variation operators act within the 'small cell'. During relaxation atoms are allowed to get outside of the 'small cell', in which case the latter is adjusted to include those atoms, and the large cell is adjusted accordingly, to maintain enough vacuum.

Lattice mutation, an operator introduced for crystal structure prediction [3], can be adapted for nanoparticles—if one applies it to the "small cell" described above. This results in deformations of the nanoparticle, which lead to new structures after the relaxation.

Another change required for cluster structure prediction is a different formula for the structure fingerprint [19]:

$$F_{A_iB}(R) = \sum_{B_j} \frac{\delta(R - R_{ij})}{4\pi\, R_{ij}^2 N_B \Delta} \tag{6}$$

here F_{A_iB} describes the fingerprint function of the individual atom A_i relative to all atoms of the type B surrounding it. The sum runs over all atoms of the type B, N_B is the number of atoms of the type B in the cluster and R_{ij} is interatomic distance between atoms A_i and B_j. $\delta(R - R_{ij})$ is a Gaussian-smeared delta-function, absorbing numerical errors and making $F(R)$ a smooth function, which is then discretized over bins of width Δ. Compared to crystal fingerprint [19], there is no cut-off radius, the volume normalization disappears and the unity is not subtracted—all these changes being due to the finite size of clusters, in contrast to the infinite ideal crystal.

One can calculate the order parameter from fingerprint function (6) and use order-fitness correlation to improve the heredity operator in a fashion similar to the algorithm described in Section 3. Since clusters are oriented according to their moments of inertia, instead of searching for most (least) ordered cuts in both parent clusters we randomly select one of the two clusters, and for that cluster evaluate N_s different cuts. The most (least) ordered cut is chosen and the second nanoparticle is cut with the same plane, to obtain the second part of the child cluster. The number of evaluated cuts is calculated using a modified formula (1):

$$N_s = \frac{(\alpha\pi^2 N)^{1/3}}{1 + \left((\alpha\pi^2 N)^{1/3} - 1\right) \cdot \cos\left(\frac{\pi}{2} r\right)^2} \tag{7}$$

where N is the number of atoms in the nanoparticle and α is a user defined parameter (usual values 1–6).

Similarly to the technique described in Section 2, we have developed a random symmetric initialization scheme also for

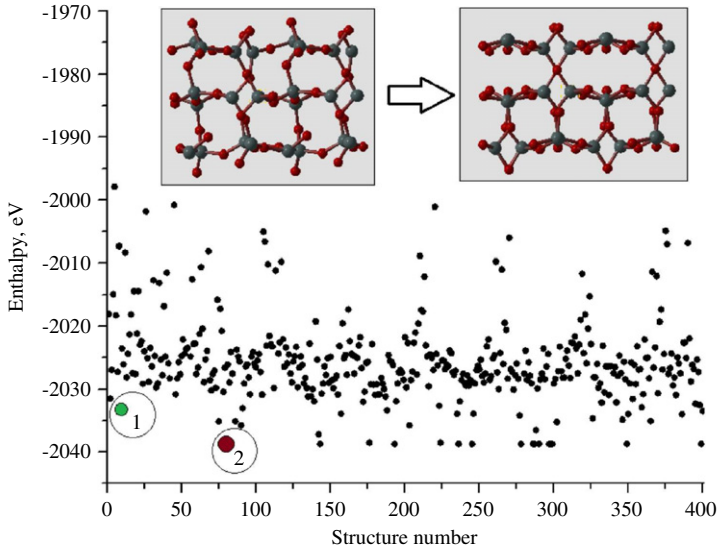

Fig. 8. Calculation for coesite at fixed unit cell. Coesite (structure **2**) was obtained from structure **1** by softmutation.

Table 1
Results for different crystalline systems.

System (number of atoms/cell)	Antiseeds	Symmetric initialization	Success rate (%)	Average number of structures until global minimum is found	Dispersion
$SrTiO_3$ (50)	−	−	94	524	297
TiO_2(48)	−	−	100	41	40
SiO_2-coesite (48)	+	+	100	591	537
$Mg_{24}Al_{16}Si_{24}O_{96}$ (160)	+	+	100	296	327

nanoparticles. The user gives a list of possible point groups[5] (like, for example, C_2, D_{6h}, etc.) and nanoparticles are generated by randomly placing atoms inside the ellipsoid inscribed in the "small cell", and then replicating them using the point group symmetry operators. We use the same procedure for merging atoms from general positions into special ones; see Fig. 2. It is very important to note though, that such initialization scheme does not bias the algorithm to favour only the symmetric structures. As we will show in Section 6, the search for the ground state of the Lennard-Jones cluster with 44 atoms, which has no symmetry at all, is not slowed down by symmetric initialization (the slight decrease of the success rate is reversed with the use of antiseeds). The reason for this is that the initial symmetry is broken by variation operators already in the second generation. The use of symmetric initialization creates a diverse and high-quality initial population, which can evolve even to an asymmetric state. This is very different from symmetry-enhanced random sampling [21], where the use of symmetry would completely prohibit any chances of finding an asymmetric ground state.

We would like to note that there are already efficient evolutionary algorithms for cluster structure prediction (see, for example, [31,32] and references therein). Here we describe and demonstrate a different implementation in the USPEX code, which expands the capabilities of this approach. The numerical results below show that for systems tested with our code it outperforms other *general* search algorithms and allows efficient search for systems with over 100 atoms.

[5] For random symmetric initialization, if nothing is known about the point group, for generating each new structure we randomly draw the point group from the list of all 32 crystallographic point groups and the most common non-crystallographic groups—pentagonal, decagonal and icosahedral.

6. Numerical results

6.1. Crystal structure prediction

The improved version of the USPEX algorithm was tested on a range of different systems; see Tables 1 and 2.

$SrTiO_3$ with 50 atoms in the cell was the central test case used by Lonie and Zurek for their code [6], and the success rate was 7%–12%, which was claimed to be good for a system of this size. To study the performance of USPEX we performed 35 calculations with the same interatomic potential model [33] and limiting the number of generations to 40 and the size of the generation to just 25 structures (to replicate the 1000-structures searches from [6]). The success rate was 94%, which increased to 100% when we extended the calculations to 70 generations. The average number of structures required to obtain the global minimum was just 524. The ground state (with this potential model) has 5 atoms in the unit cell. In this calculation we did not use antiseeds or space group initialization. We used softmutation and the initial generation was produced by the cell splitting technique with the split-factors 2 and 4.

The other system tested in [6] was titania (TiO_2) with 48 atoms in the cell. We used the same potential [34] as used in [6]. The ground state is the rutile structure, which contains 6 atoms in the unit cell. For this system we performed 73 calculations and obtained a 100% success rate. The average number of structures required to obtain the ground state is 41. In this calculation we did not use antiseeds or space group initialization. Cell splitting (split-factors 2 and 4) for the first generation and softmutation were used.

We also tested USPEX on coesite—a polymorph of SiO_2 with 48 atoms in the conventional cell. In this calculation we used the relatively crude fully ionic potential model with shell model and parameters from [35–37]. As a constraint, we used the

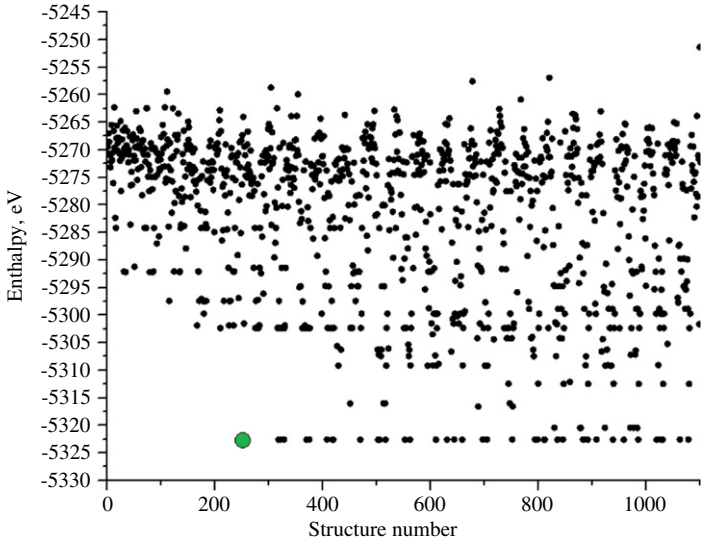

Fig. 9. Example of the calculation for pyrope ($Mg_3Al_2(SiO_4)_3$) with 160 atoms/cell in a fixed cubic cell. One can clearly see the rapid improvement of the population quality in the first generations. Also, this figure shows that even after the ground state was found, the algorithm continued to improve the general quality of the population and found new very low-energy structures.

Table 2
Parameters of evolutionary runs. In our tests we did not attempt to optimize these parameters and chose parameters based on intuition and experience.

	$SrTiO_3$	TiO_2	SiO_2	$Mg_{24}Al_{16}Si_{24}O_{96}$
Population size	25	10	40	40
Max. number of generations	40	40	60	30
Number of calculations	35	73	68	64
% of structures generated by:				
Heredity	40%	50%	30%	30%
Softmutation	20%	20%	30%	30%
Permutation	20%	10%	10%	10%
Lattice mutation	20%	20%	0%	0%
Symmetric random	0%	0%	30%	30%

experimental unit cell parameters. The success rate is 100% and the structure is often obtained within a few generations; see Fig. 8.

For the garnet pyrope ($Mg_{24}Al_{16}Si_{24}O_{96}$) with 160 atoms in the cubic unit cell, previous version of our algorithm could not find the ground state, but the new algorithm is able to do so easily. We performed 64 fixed-cell calculations (30 structures/generation, with a maximum of 100 generations) and the success rate is 100%. An example of the calculations, in which ground state was found in the 6th generation, is shown in Fig. 9. For this calculation we used all our methodological developments—symmetric initialization, antiseeds, cell splitting and softmutation.

To our knowledge, USPEX is the first and so far the only general global optimization code which shows high success rates for crystalline systems of such size. A plethora of quantitative improvements of the algorithm described in [5] and in this work have resulted in a qualitative breakthrough, which allows one to confidently predict structures containing more than 100 atoms in the unit cell. The vast difference in performance between different evolutionary algorithms emphasizes the importance of careful design of the algorithm and ideas behind it.

6.2. Testing particle swarm optimization (PSO)

To compare the particle swarm optimization regime (cor-PSO) with conventional USPEX algorithm we did calculations for $MgAl_2O_4$ with 28 atoms/cell and pressure of 100 GPa and TiO_2 with 48 atoms/cell. For $MgAl_2O_4$, we use Lewis–Catlow potentials [38], and 20 global optimizations were run for up to 50 generations. For TiO_2, comparison with earlier PSO (CALYPSO code) [17] is available [39]. We performed 100 global optimizations with each method (up to 400 structure relaxations per run), and here, in contrast with results presented in Table 1, no cell-splitting was used. The results are presented in Table 3. One can see that PSO is generally less reliable than our conventional algorithm.

For $MgAl_2O_4$, PSO without symmetric initialization has low success rate, but the number of structures required to find the global minimum in successful calculations was very low. It seems that PSO algorithm explores the landscape fast in the area where initial structures were generated, however has difficulties in exploring the whole search space. Our results show that the addition of new random symmetric structures to the 'swarm' at every generation substantially improves the efficiency of PSO and is thus a crucial element of the algorithm of Wang [17]. This is also confirmed in their latest paper [39].

For crystalline TiO_2, the performance of both methods was similar. Statistics for this system is also available for conventional PSO algorithm [39], in which the ground state is found after (on average) 500 structures without the use of symmetry and after 156–400 structures (with the success rate 90%–100%), depending on the algorithm parameters, when symmetric initialization was used. This example highlights our improvement of the conventional PSO methodology.

Latest versions of CALYPSO algorithm [39,48] aimed at correcting the search space metric problem; but, to define search space metrics, incorrectly [19] chose to use Steinhardt's orientational order parameter [39] and a parameter [48] analogous to our local order parameter [19], instead of using proper fingerprints [19].

6.3. Tests for cluster structure prediction

For our cluster structure prediction algorithm, we used the standard testing system—Lennard-Jones clusters, which have been well studied in the past two decades [31,40–51], and allow comparison with other algorithms, and the calculations are fast enough for collecting good statistics, summarized in Table 4. The results are compared with the *state-of-the-art* minima hopping (MH) algorithm [45,46], evolutionary algorithms (EA) from [46,47] and PSO algorithm [48]. Heredity improved by order was tested for two cases that required substantial number of structures to obtain the ground state–LJ_{44} and LJ_{75}.

Table 3
Comparison of the PSO (CALYPSO code [48]) and cor-PSO calculations with our evolutionary algorithm USPEX, with and without symmetric initialization.

	Success rate (%)	Average number of structures until global minimum is found	Dispersion	Average energy of the best structure in the calculation
$MgAl_2O_4$				
USPEX sym.	100	1677	1162	−655.0622
USPEX no sym.	100	1403	1011	−655.0622
Cor-PSO sym.	70	1317	783	−654.6132
Cor-PSO no sym.	20	231	110	−653.7715
TiO_2				
USPEX sym.	100	77	76	−636.8056
USPEX no sym.	100	80	69	−636.8056
Cor-PSO sym.	100	79	78	−636.8056
Cor-PSO no sym.	100	87	78	−636.8056
CALYPSO sym.	90–100	156–400	N/a	N/a
CALYPSO no sym.	100	500	N/a	−636.8056

Table 4
Statistics for Lennard-Jones clusters with different algorithms. Best algorithms are highlighted in bold.

	Symmetric initialization	Antiseeds	Order	Success rate (%)	Average number of structures until global minimum is found	Dispersion	Number of calculations
LJ_{38} (PSO [48])	+	−	−	100	605	N/a	100
LJ_{38}(USPEX)	+	+	−	**100**	**35**	**58**	**183**
LJ_{38}(USPEX)	−	−	−	67	2291	1443	100
LJ_{38}(USPEX)	−	+	−	98	3080	2119	100
LJ_{38} (EA [46])[b]	−	−	−	N/a	1265	N/a	100
LJ_{38} (MH [46])[b]	−	−	−	100	1190	N/a	100
LJ_{38} (EA [47])[b]	−	−	−	N/a	∼2000[a]	N/a	N/a
LJ_{38} (PSO [48])	−	−	−	100	1649	N/a	20
LJ_{44}(USPEX)	−	−	−	100	1510	1079	35
LJ_{44}(USPEX)	−	+	+	**100**	**859**	**524**	**37**
LJ_{44}(USPEX)	+	+	+	100	1129	765	41
LJ_{44}(USPEX)	+	−	−	86	1551	1020	35
LJ_{44}(USPEX)	+	+	−	94	1423	867	35
LJ_{55} (PSO [48])	+	−	−	100	159	N/a	100
LJ_{55}(USPEX)	+	−	−	**100**	**11**	**30**	**60**
LJ_{55}(USPEX)	−	−	−	100	717	407	103
LJ_{55} (EA [46])[b]	−	−	−	100	100	N/a	100
LJ_{55} (MH [46])[b]	−	−	−	100	190	N/a	100
LJ_{75} (PSO [48])	+	−	−	98	2858	N/a	50
LJ_{75}(USPEX)	+	+	+	**100**	**2145**	**2024**	**53**
LJ_{75}(USPEX)	+	+	−	100	5419	4513	47

[a] Depending on algorithm parameters, 2000–12 000 structures were required to reach the global minimum in [47].
[b] Results for algorithms [46,47] are given for optimized algorithm parameters. We did not attempt to optimize USPEX parameters and chose them based on intuition and experience. We could achieve even greater advantage over other approaches with parameter optimization, but did not do this because in realistic calculations (using quantum-mechanical energy evaluations) one cannot afford parameter tuning.

Lennard-Jones cluster with 55 atoms (LJ_{55}) is a good example of a moderately large system with a simple landscape: there is only one deep funnel. Any reasonable global optimization algorithm will sooner or later find the ground state. The stable structure has a very high (icosahedral) symmetry and therefore symmetric initialization greatly improves the efficiency of the algorithm. 103 calculations were done without symmetric initialization and 60 with symmetric initialization (in both cases, we used 40 structures per population, with a maximum of 40 generations). While both sets of calculations have a 100% success rate, the ground state is found on average after relaxing 717 structures without symmetric initialization (best calculation—94 relaxations) and after just 11 structures with symmetric initialization. In most cases with symmetric initialization, the ground state was found in the first generation, in spite of the fact that many point groups (43) were used for initialization (though not all of these are compatible with the required number of atoms). The fact that another method with symmetric initialization [48] requires substantially higher number of structures to obtain the ground state shows the advantage of our initialization procedure (see also a result for LJ_{38} cluster).

LJ_{44} is an example of a cluster with ground state that has no symmetry. To find whether symmetric initialization worsens performance of the algorithm in cases of asymmetric ground states, we did 35 calculations for this system (100 generations, 40 structures per generation) with and without symmetric initialization. The average number of structures required to obtain the global minimum was 1551 and 1510, respectively, if order-enhanced heredity is not used, and 859 and 1129, respectively, with such heredity. Symmetric initialization in this system slightly increases the chances of sticking to low-energy local minima, i.e. success rate is slightly decreased. However, the use of antiseeds and order helps to escape them. All types of calculations showed an overall similarly excellent performance.

LJ_{38} is a smaller system with a much more complex landscape, and is known to pose problems for global optimization algorithms [44,45]. It has two funnels with minima that are energetically very close to each other energetically ($\Delta E = 0.018$ Lennard-Jones units per atom) and the global minimum corresponds to the narrower funnel. Therefore, it is easy to get trapped in a local minimum. The use of antiseeds (see Fig. 10 and Table 4) easily solves this problem and brings the success rate to almost 100%. If symmetric initialization is used, the problem becomes trivial and 100% success rate is observed.

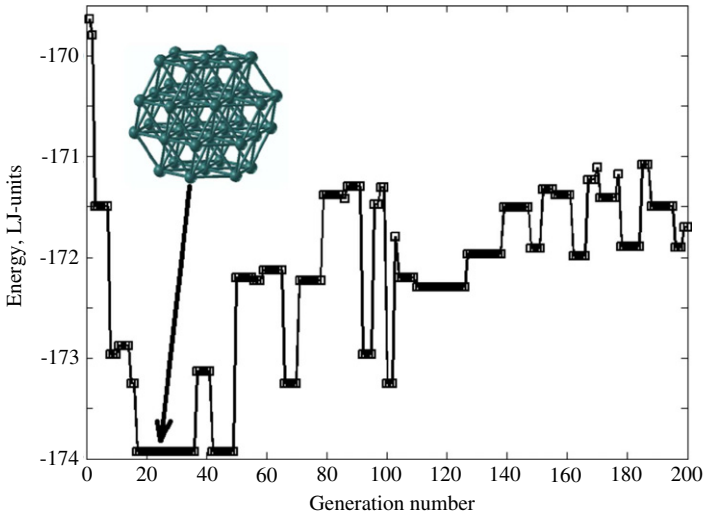

Fig. 10. Example of a calculation on LJ_{38} with the use of antiseeds. The energy of the best structure in every generation is plotted. One can clearly see that the algorithm does not stick for a long time to any of the candidate minima and quickly finds the ground state.

The first set of calculations was done with antiseeds and without symmetric initialization. 100 calculations were performed, each with 200 generations and 40 structures per generation. Success rate was 98%, the average number of structures required to find the global minimum was 3080. The structure was guessed in the initial generation only once. The fastest answer not in the first generation was obtained after 118 structure relaxations, in the third generation. Symmetric initialization drastically improves performance and allows the correct ground state to be found within just a few generations (after 35 relaxations on average). In most cases, the ground state was found within three generations. Calculations without antiseeds and symmetric initialization turned out to have the success rate of just 67%. It is interesting to note that without antiseeds the speed of finding the new structure is higher, while success rate is much lower; see Table 1. When the system is stuck in the wrong funnel, antiseeds will require some time to push it out of that funnel and thus increase the average time required to obtain the solution. At the same time, symmetric initialization makes it more likely to sample all good funnels, which greatly speeds up the calculation. Thus, the combination of improved initialization and antiseed technique has a dramatic effect on improving the success rate and the speed of finding the best structure (up to an order of magnitude). It also allows us to study larger and more complex systems. Only in the rare cases, when the ground state has no symmetry, as in the case of LJ_{44} nanoparticle, these modifications do not substantially improve (but do not worsen either) the performance of the algorithm.

LJ_{75} is a relatively large and exceptionally complex multifunnel system. Symmetric initialization, antiseeds and heredity with order parameter prove to be extremely useful in this case, allowing us to once again outperform the best general global optimization algorithms [46,48]. We did 45 calculations for LJ_{75} with up to 300 generations, and 50 structures per generation. The success rate is 100%, the average number of structures required to find the global minimum was 2145. Without the use of order, USPEX has to sample on average 5419 structures to obtain the ground state.

Twelve tests performed for LJ_{147} were also all successful (99 structure relaxation before finding the ground state, on average); however, we have not endeavoured to obtain statistics for that large system with a relatively simple energy landscape. Both LJ_{55} and LJ_{147} have Mackay icosahedra as ground states.

We would like to note, that the comparison was made with general cluster structure prediction algorithms. There are algorithms specifically optimized for Lennard-Jones clusters (for example [50,51], which utilize the 'directed mutation' operator), but in accordance with 'no free lunch' theorem [52], these are likely to work poorly for other types of clusters.

7. Other USPEX developments

Some other USPEX developments, not included in this article but described elsewhere, are:

- A constrained evolutionary algorithm for molecular crystal structure prediction [53]. It allows the search for optimal structure with rigid and/or flexible building blocks (molecules), and proved extremely useful in a number of complex problems [53,54].
- Evolutionary metadynamics [29]. This is a novel method for crystal structure prediction, a fusion of metadynamics [18] and evolutionary approach, which already resulted in predictions of new synthesizable low-enthalpy carbon allotropes [55].
- Variable-cell nudged elastic band method (vc-NEB) [56]. It allows one to find idealized phase transition pathways when both the initial and final states are known.
- Transition path sampling (TPS) algorithm [57], which realistically models nucleation events necessary for physically meaningful transformation kinetics, and which was recently used to establish [58] the structure of a new metastable phase of carbon.
- Variable-composition crystal structure prediction algorithm, developed in 2008 and briefly described in [12,59]. It allows one to simultaneously search for all stable compounds and their structures, provided just the names of the chemical elements. This method has resulted in startling predictions of thermodynamic stability of such unusual compounds as Na_3Cl, Na_2Cl, Na_3Cl_2, $NaCl_3$, $NaCl_7$ [60] and Mg_3O_2 [61].
- An algorithm for predicting surface reconstructions, which will be released in the near future.

8. Conclusions

We presented new developments in the evolutionary algorithm USPEX [3,5], which improve its efficiency and enable studies of much larger systems. The symmetry-driven initialization provides a substantial improvement. Clustering, new variation operator—softmutation, and improvement of the heredity operator via fitness-order correlation allow us to sample the search

space better and drive the search towards more chemically reasonable structures. We also discussed the changes to the algorithm required for nanoparticle structure prediction, which is now an integral part of the USPEX code. Problems of the standard PSO algorithm [17] with the definition of the space metric and moves were discussed and resolved. The corrected PSO algorithm showed reasonable performance, which was, however, not competitive with our evolutionary algorithm USPEX. Starting with the ideas of metadynamics, we have implemented the concept of ageing and death of structures in this evolutionary algorithm through the antiseed technique. Indeed, ageing and death can help evolution to broader sample the search space and solve the problem of genetic drift. The performance of the improved algorithm was demonstrated on a range of different systems with up to 160 atoms in the unit cell.

Acknowledgements

We gratefully acknowledge funding from DARPA (grant N66001-10-1-4037), National Science Foundation (EAR-1114313), Rosnauka and CRDF-Global. Calculations were performed on the Blue Gene supercomputer at the New York Center for Computational Sciences, on Skif MSU supercomputer (Moscow State University), and at the Joint Supercomputer Centre of the Russian Academy of Sciences. USPEX code, with options for global optimization of the thermodynamic potential (energy, enthalpy, free energy), density, hardness, and other properties, is available at: http://han.ess.sunysb.edu/USPEX.

References

[1] T.S. Bush, C.R.A. Catlow, P.D. Battle, J. Mater. Chem. 5 (1995) 1269–1272.
[2] S.M. Woodley, Struct. Bonding 110 (2004) 95–132.
[3] A.R. Oganov, C.W. Glass, J. Chem. Phys. 124 (2006) 244704.
[4] C.W. Glass, A.R. Oganov, N. Hansen, Comput. Phys. Comm. 175 (2006) 713–720.
[5] A.O. Lyakhov, A.R. Oganov, M. Valle, Comput. Phys. Comm. 181 (2010) 1623.
[6] D. Lonie, E. Zurek, Comput. Phys. Comm. 182 (2011) 2305–2306.
[7] A.R. Oganov, C.W. Glass, S. Ono, Earth Planet. Sci. Lett. 241 (2006) 95–103.
[8] A.R. Oganov, J. Chen, C. Gatti, Y.-Z. Ma, Y. Ma, C.W. Glass, Z. Liu, T. Yu, O.O. Kurakevych, V.L. Solozhenko, Nature 457 (2009) 863–867.
[9] Y. Ma, M.I. Eremets, A.R. Oganov, Y. Xie, I. Trojan, S. Medvedev, A.O. Lyakhov, M. Valle, V. Prakapenka, Nature 458 (2009) 182–185.
[10] A.O. Lyakhov, A.R. Oganov, Phys. Rev. B 84 (2011) 092103.
[11] Q. Zhu, A.R. Oganov, M.A. Salvado, P. Pertierra, A.O. Lyakhov, Phys. Rev. B 83 (2011) 193410.
[12] A.R. Oganov (Ed.), Modern Methods of Crystal Structure Prediction, Wiley-VCH, Berlin, 2010.
[13] A.R. Oganov, A.O. Lyakhov, M. Valle, Acc. Chem. Res. 44 (2011) 227–237.
[14] D.E. Goldberg, P. Segrest, Proceedings of the Second International Conference on Genetic Algorithms and their Applications, Lawrence Erlbaum, New Jersey, 1987, pp. 1–8.
[15] J. Kennedy, R. Eberhart, Proceedings of IEEE International Conference on Neural Networks, Vol. IV, 1995, pp. 1942–1948.
[16] S.T. Call, D.Yu. Zubarev, A.I. Boldyrev, J. Comput. Chem. 28 (2007) 1177–1186.
[17] Y. Wang, J. Lv, L. Zhu, Y. Ma, Phys. Rev. B 82 (2010) 094116.
[18] R. Martonak, A. Laio, M. Parrinello, Phys. Rev. Lett. 90 (2003) 075503.
[19] A.R. Oganov, M. Valle, J. Chem. Phys. 130 (2009) 104504.
[20] M.D. Wolf, U.J. Landman, J. Phys. Chem. A 102 (1998) 6129.
[21] C.J. Pickard, R.J. Needs, J. Phys.: Condens. Matter 23 (2011) 053201.
[22] M. Sicher, S. Mohr, S. Goedecker, J. Chem. Phys. 134 (2011) 044106.
[23] S. Roy, S. Goedecker, V. Hellmann, Phys. Rev. E 77 (2008) 056707.
[24] M. Ji, C. Wang, K.M. Ho, Phys. Chem. Chem. Phys. 12 (2010) 11617.
[25] M. Born, K. Huang, Dynamical Theory of Crystal Lattices, Oxford University Press, Oxford, 1954.
[26] K. Li, X. Wang, F. Zhang, D. Xue, Phys. Rev. Lett. 100 (2008) 235504.
[27] A.O. Lyakhov, A.R. Oganov, Phys. Rev. B 84 (2011) 092103.
[28] H. Grubmüller, Phys. Rev. E 52 (1995) 2893.
[29] Q. Zhu, A.R. Oganov, A.O. Lyakhov, Cryst. Eng. Comm. 14 (2012) 3596–3601.
[30] A. Ghosh, S. Tsutsui, H. Tanaka, Proceedings of the 1998 IEEE International Conference on Evolutionary Computation, ICEC'98, Mai, Seoul, 1998, pp. 666–671.
[31] D.M. Deaven, N. Tit, J.R. Morris, K.M. Ho, Chem. Phys. Lett. 256 (1996) 195.
[32] R.L. Johnston, Dalton Trans. 22 (2003) 4193–4207.
[33] N.A. Benedek, A.L.-S. Chua, C. Elsaesser, A.P. Sutton, M.W. Finnis, Phys. Rev. B 78 (2008) 064110.
[34] S.M. Woodley, C.R.A. Catlow, Comput. Mater. Sci. 45 (2009) 84.
[35] C.R.A. Catlow, Proc. R. Soc. Lond. Ser. A 353 (1977) 533–561.
[36] M.J. Sanders, M. Leslie, C.R.A. Catlow, J. Chem. Soc. Chem. Commun. 19 (1984) 1271.
[37] A. Gavezzotti, Acc. Chem. Res. 27 (1995) 309–314.
[38] G.V. Lewis, C.R.A. Catlow, J. Phys. C: Solid State Phys. 18 (1985) 1149–1161.
[39] Y. Wang, J. Lv, L. Zhu, Y. Ma, Comput. Phys. Comm. 183 (2012) 2063–2070.
[40] J.A. Northby, J. Chem. Phys. 87 (1987) 6166.
[41] T. Coleman, D. Shalloway, J. Global Optim. 4 (1994) 171.
[42] J.P.K. Doye, D.J. Wales, R.S. Berry, J. Chem. Phys. 103 (1995) 4234.
[43] D.J. Wales, J.P.K. Doye, J. Phys. Chem. A 101 (1997) 5111–5116.
[44] J.P.K. Doye, D.J. Wales, Phys. Rev. Lett. 80 (1998) 1357.
[45] S. Goedecker, J. Chem. Phys. 120 (2004) 9911.
[46] S.E. Schonborn, S. Goedecker, S. Roy, A.R. Oganov, J. Chem. Phys. 130 (2009) 144108.
[47] V.A. Froltsov, K. Reuter, Chem. Phys. Lett. 473 (2009) 363–366.
[48] J. Lv, Y. Wang, L. Zhu, Y. Ma, J. Chem. Phys. 137 (2012) 084104.
[49] The list of global minima of Lennard-Jones clusters is available through the Cambridge cluster database. http://physchem.ox.ac.uk/~doye/jon/structures/LJ.html.
[50] B. Hartke, J. Comput. Chem. 20 (1999) 1752–1759.
[51] W. Pullan, J. Comput. Chem. 26 (2005) 899–906.
[52] D.H. Wolpert, W.G. Macready, IEEE Trans. Evol. Comput. 1 (1997) 67–82.
[53] Q. Zhu, A.R. Oganov, C.W. Glass, H.T. Stokes, Acta Crystallogr. B 68 (2012) 215–226.
[54] X.-F. Zhou, A.R. Oganov, G.R. Qian, Q. Zhu, Phys. Rev. Lett. 109 (2012) 245503.
[55] Q. Zhu, Q. Zeng, A.R. Oganov, Phys. Rev. B 85 (2012) 201407.
[56] G.-R. Qian, X. Dong, X.-F. Zhou, Y. Tian, A.R. Oganov, H.-T. Wang, Variable cell nudged elastic band method for studying solid-solid structural phase transitions (submitted for publication).
[57] C. Dellago, P.G. Bolhuis, F.S. Csajka, D. Chandler, J. Chem. Phys. 108 (1998) 1964–1977.
[58] S.E. Boulfelfel, A.R. Oganov, S. Leoni, Sci. Rep. 2 (2012) 471.
[59] A.R. Oganov, Y. Ma, A.O. Lyakhov, M. Valle, C. Gatti, Rev. Mineral. Geochem. 71 (2010) 271.
[60] W.W. Zhang, A.R. Oganov, A.F. Goncharov, Q. Zhu, S.E. Boulfelfel, A.O. Lyakhov, M. Somayazulu, V.B. Prakapenka, Unexpected stable stoichiometries of sodium chlorides (submitted for publication). Available at: http://arxiv.org/abs/1211.3644.
[61] Q. Zhu, A.R. Oganov, A.O. Lyakhov, Unexpected stoichiometries in Mg-O system at high pressure (submitted for publication). Available at: http://arxiv.org/abs/1211.6521.