Available online at www.sciencedirect.com
ScienceDirect
Computer Physics Communications 175 (2006) 713–720
Computer Physics Communications
www.elsevier.com/locate/cpc

USPEX—Evolutionary crystal structure prediction

Colin W. Glass [a,*], Artem R. Oganov [a,b], Nikolaus Hansen [c]

[a] *Laboratory of Crystallography, Department of Materials, ETH Zurich, 8093 Zurich, Switzerland*
[b] *Geology Department, Moscow State University, 119899 Moscow, Russia*
[c] *CoLab Computational Laboratory, Institute of Computational Science, ETH Zurich, 8092 Zurich, Switzerland*

Received 2 May 2006; received in revised form 20 July 2006; accepted 31 July 2006

Available online 20 September 2006

Abstract

We approach the problem of computational crystal structure prediction, implementing an evolutionary algorithm—USPEX (Universal Structure Predictor: Evolutionary Xtallography). Starting from chemical composition we have tested USPEX on numerous systems (with up to 80 atoms in the unit cell) for which the stable structure is known and have observed a success rate of nearly 100%, simultaneously finding large sets of competitive metastable structures. Here focus is on implementation and discussion of our method.
© 2006 Elsevier B.V. All rights reserved.

PACS: 61; 61.18.-j; 61.50.Ah

Keywords: Crystal structure prediction; Evolutionary algorithm; *Ab initio*; Free energy

1. Introduction

The crystal structure is a great bearer of information on a given material: knowledge of the crystal structure allows determination of numerous properties. The traditional way of solving crystal structure is based on experiment. Ideally experiment yields the diffraction pattern, which then is inverted to give the structure. For this method, the material needs to be synthesized and studied at relevant conditions. At high pressures and temperatures conditions diffraction patterns often are incomplete and noisy, rendering experimental structure solution difficult or impossible. Furthermore, in materials design searching for desired properties experimentally leads to elaborate trial and error.

A different approach to solve the crystal structure is computational crystal structure prediction, based on optimization. Many optimization methods, like simulated annealing [1,2], metadynamics [3,4], minima hopping [5] and evolutionary algorithms [6–8] have been applied. Although successes were made, many cases were not predicted correctly, leaving this major scientific problem [9] essentially unsolved [10].

Since the stable crystal structure is the structure with the lowest free energy, the task is to minimize the free energy. This is far from trivial for the following reasons:

- The search space is high-dimensional.[1]
- The free energy response surface is extremely rugged, since the free energy is very sensitive to small changes in interatomic distances.[2]
- The representation of structures by unit cells leads to redundancies within the search space.
- *Ab initio* free energy calculations are highly accurate, but computationally expensive. While cheap methods to approximate the free energy exist, they often relate poorly to reality. This can lead to a misguided search.

* Corresponding author.
E-mail address: coglass@mat.ethz.ch (C.W. Glass).

[1] The dimensionality is $6 + 3(N - 1)$, where N is the number of atoms. In detail, the six lattice parameters and three coordinates for each atom except one (due to rigid coordinate system shifts).

[2] No matter what evaluation function is used, the optimal solution has to be optimal regarding the free energy.

0010-4655/$ – see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.cpc.2006.07.020

Seeing these difficulties, carefully choosing a strategy becomes all the more important. The formulation of an appropriate evaluation function is a core issue in any optimization task. *Ab initio* calculations of the free energy are by far the most accurate and universally applicable estimates, while the cost thereof seems affordable. Therefore and to avoid misguided searching, *ab initio* free energy calculation is chosen as evaluation function.[3]

The next step is to find a good way to sample new structures. In crystal structures the main information lies in the relative position of the nearby atoms—spatially selected fractions of one structure can carry a large fraction of the information present in the whole structure. Therefore combining such fractions of 'parent' structures seems like a promising way to sample new structures, allowing to focus the search on an area defined by a set of structures.

Considering the landscape, we reason that the global optimum (stable structure) is surrounded by many very good local optima (metastable structures), leading to a valley in the reduced response surface[4]—reduced to just the locally optimal points—in the vicinity of the global optimum.[5] The complete response surface per contra is very rugged, featuring high peaks and saddle points.[6] Therefore, a set of locally optimized structures is a good source of information on the response surface.

Sampling method and landscape shape strongly suggest an evolutionary algorithm and so, on the basis of these considerations, we decided to implement an evolutionary algorithm featuring local optimization of each candidate structure, spatial recombination and *ab initio* free energy calculation as evaluation function. The first application of our method was given in [20] and a basic discussion of the method with many further applications was presented in [24]. Here focus is on implementation, methodology and discussion thereof.

Section 2 outlines the implementation of USPEX. A short overview of results can be found in Section 3, followed by a discussion of USPEX, including future plans, in Section 4.

2. The algorithm

Quoting from Michalewicz and Fogel [13]:

'In evolutionary algorithms a population of candidate solutions is evolved over successive iterations of random variation and selection. Random variation provides the mechanism for discovering new solutions. Selection determines which solutions to maintain as a basis for further exploration.'

For encoding the solutions, two types of variables are discriminated: Lattice parameters and atomic coordinates. There are 6 lattice parameters, three angles (α, β, γ)—coded as a fraction of 2π—and the lengths of the three lattice vectors.[7] Each atom has three coordinates, coded as a fraction of the corresponding lattice vector.[8]

A complete set of values defines one structure and a locally optimized structure is referred to as an individual. A set of individuals is called a population or, depending on context, a generation.

Quality comparisons between different individuals are based on the corresponding fitness values of these individuals, being the negative of the *ab initio* free energy (see Section 2.1).

A candidate for a new individual is obtained by applying one variation operator (see Section 2.2) to selected individuals. For every operation one or two—depending on the operation—individuals are chosen stochastically from the population. The probability of a given individual being chosen for a given operation is a function of the individual's fitness rank,[9] where a predefined number of worst individuals has a probability of zero. The selected individual is not removed and can thus be selected multiple times. The generated candidates are scaled to a certain unit cell volume, V_{UC} (see Section 2.3), and those not fulfilling the hard constraints (see Section 2.5) are discarded. The rest gets locally optimized (see Section 2.4) and hereby new individuals are created. Each operation is repeated until the user—requested number of new individuals for this operation are produced. The total number of new individuals equals the population size. After the calculation of the fitness value of each new individual, the new population is obtained by taking the best individuals from the combination of offspring and a user-defined number of best individuals from the parental population.[10] Candidates for the initial population are acquired by randomly generating and/or taking structures provided by the user. These undergo the same steps as candidates produced by variation operators (see above) before becoming individuals.

A description in pseudo-code can be found in Algorithm 1. USPEX is implemented in Matlab.

2.1. Evaluation function

USPEX uses the negative of the *ab initio* free energy of the locally optimized structure as fitness value. *Ab initio* free energy is the most accurate and universal measure of quality and requires no prior assumptions on the system. Surrogate evaluation functions are computationally much cheaper, but typically work only for a narrow range of system types and otherwise often fail completely. Furthermore, many types of physical interactions are not captured in a satisfactory manner by any surrogate evaluation function.

[3] USPEX is interfaced with SIESTA [11] and VASP [12] for quantum-mechanical calculations.

[4] The response surface is the surface defined by the fitness values.

[5] Not to be confused with valleys in the complete response surface, accommodating local minima.

[6] This discouraged us from simulated annealing, since escaping local minima gets in general more difficult with increasing height of peaks/saddle points.

[7] For certain operations the lower-triangular matrix form is used.

[8] Since parameters describing crystal structures are continuous numbers, USPEX represents every variable by a floating-point value, resulting in sufficient resolution and intuitive handling.

[9] The user can decide between linear and quadratic dependence.

[10] This is an elitist environmental selection.

```
set Percentages, V_UC, N_cons
initialize X, fit_X   #X is the population, fit_X contains the fitness values
while not done do
  Y = ∅   #Y is the offspring
  for operator_i do
    Y_OP = ∅   #Y_OP is the subset produced by the current operator
    #The following while loop terminates when the required number of offspring is reached (for the current operator)
    while |Y_OP| ⩽ (|X| × Percentages_i) do
      # The current operator OP_i generates a new candidate y. Parents are selected, where probabilities depend on fitness ranking
      y = OP_i(select(X, fit_X))
      # Candidate y is scaled to the volume V_UC
      y ← scaleV(y, V_UC)
      # If y satisfies the constraints, y gets locally optimized and accepted
      if constraints_check(y) then
        y ← local_opt(y)
        Y_OP ← Y_OP ∪ y
      end if
    end while
    Y ← Y ∪ Y_OP
  end for
  # Fitness values of the offspring are determined
  fit_Y = evaluation(Y)
  # The best individuals from Y ∪ keep(X, fit_X, N_cons) survive
  (X, fit_X) ← select_env(X, Y, fit_X, fit_Y, N_cons)
  # V_UC gets adapted, see Section 2.3
  V_UC ← adaptV(V_UC, X)
end while
```

Algorithm 1. Basic structure of USPEX.

2.2. Variation operators

USPEX features three different variation operators: heredity,[11] mutation and permutation. These operators are described in detail below.

2.2.1. Heredity

Two individuals are selected and used to produce one new candidate. This is achieved by taking a fraction of each individual and combining these. However, the fraction of each individual should contain as much information of the individual as possible. The main information within crystal structures is the relative position of the nearby atoms. Thus, to conserve information, the fraction of an individual is selected by taking a spatially coherent slab. The two slabs, one of each individual, are fitted together and the result thereafter made feasible by adjusting the number of atoms of each type to the requirements.

In more detail this works as follows. One lattice vector is picked randomly, $\vec{a}_{\mathrm{ch}}$. Before the cut is realized, atoms may be shifted along the lattice vectors. For each individual and each vector along which to be shifted, a random number between zero and one is generated and added to the respective coordinates. Since the unit cell is periodically repeated, the atoms ending up with a coordinate value greater than 1 (outside the unit cell) are adapted so as to lie within the unit cell again, by subtracting 1. Original and shifted systems are physically identical. Both for $\vec{a}_{\mathrm{ch}}$ and the other vectors the user can specify in how many percent of the cases this should be done. This operation increases diversity and enhances the power of the algorithm, if set correctly. Typically a setting close to 100% for $\vec{a}_{\mathrm{ch}}$ and close to 5% for the remaining vectors works well.[12] If shifting was performed, all following operations apply to the shifted individual. Now a value x between 0 and 1 is determined randomly. From the first individual, every atom is taken which has a coordinate value on $\vec{a}_{\mathrm{ch}}$ between 0 and x. From the second individual every atom is taken with a respective coordinate value between x and 1.[13]

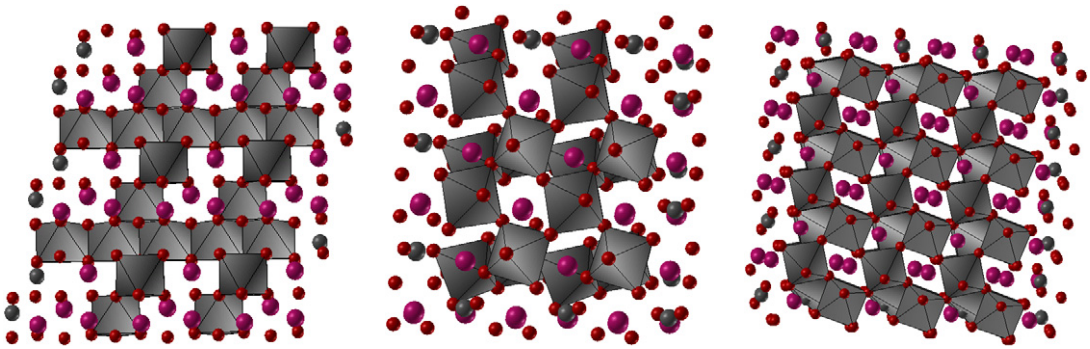
Now the atoms of both individuals are put together. The total number of atoms of each type is counted and compared with the required number. If there are too many atoms of a type, atoms of this type are removed randomly. If there are too few, the following procedure is repeated until the number is correct: An interval ([0,x] or [x,1]) is picked randomly. The probability of an interval being chosen is either equal to the width of this interval or it is inverse proportional to the atom density (of this type) in the interval. Then one atom of this type (having coordinates falling in this interval) is chosen randomly from the individual not having originally provided the atoms within this interval.

Heredity of lattice parameters is achieved by taking the weighted average of the a_{ij} matrices (in the lower triangular form) of both individuals, where the weight is chosen randomly. An example for heredity can be found in Fig. 1.

[11] Often referred to as two-parent crossover.

[12] Heredity without shifting basically introduces a bias, insofar as that a given substructure has a 'preferred' position within the unit cell, which is of course unphysical.

[13] This is equivalent to a cutting plane parallel to the plane spanned by the two lattice vectors other than $\vec{a}_{\mathrm{ch}}$.



(a) (b) (c)

Fig. 1. Example of heredity. (a) Parent 1, (b) parent 2, (c) offspring. All structures are locally optimized.

2.2.2. Mutation

One individual is selected and used to produce one new candidate. The lattice vectors $\vec{a}$ are transformed to new vectors $\vec{a}'$ by applying a strain matrix:

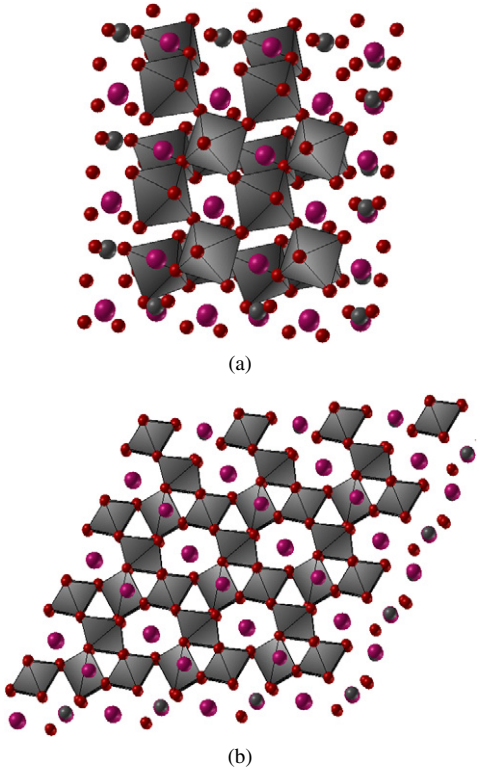
$$\vec{a}' = [I + \epsilon_{ij}]\vec{a}, \tag{1}$$

where I is the unit matrix and ϵ_{ij} is the symmetric strain matrix, such that

$$[I + \epsilon_{ij}] = \begin{bmatrix} 1+\epsilon_{11} & \frac{\epsilon_{12}}{2} & \frac{\epsilon_{13}}{2} \\ \frac{\epsilon_{12}}{2} & 1+\epsilon_{22} & \frac{\epsilon_{23}}{2} \\ \frac{\epsilon_{13}}{2} & \frac{\epsilon_{23}}{2} & 1+\epsilon_{33} \end{bmatrix} \tag{2}$$

and strains are zero mean Gaussian random variables, $\epsilon_{ij} \sim N(0, \sigma^2_{\text{lattice}})$. The new lattice is scaled to the volume V_{UC}. Mutation of atomic positions is achieved by adding zero-mean Gaussian random variables, $N(0, \sigma^2_{\text{atoms}})$.

Due to diversification within heredity (see Section 4) and local optimization, mutation of the atomic positions is not important and can be omitted. Mutation of the lattice should be present for optimal performance, both to prevent a possibly premature convergence towards a certain lattice and for efficient exploration of the immediate neighborhood of good individuals. Furthermore, the implementation using distortion by a strain matrix facilitates physically sensible setting of the step size parameter σ_{lattice}, since the required strain for structural transition can be approximated theoretically. An example for mutation can be found in Fig. 2.



(a) (b)

Fig. 2. Example of a mutation. (a) Initial structure, (b) mutated structure. All structures are locally optimized.

2.2.3. Permutation

One individual is selected and used to produce one new candidate. Two atoms of different types are exchanged (as done in [7]), a variable number of times. Permutation facilitates finding the correct atomic ordering. Obviously, permutation is possible only for systems with different types of atoms. An example for permutation can be found in Fig. 3.

2.3. Volume scaling

Every produced candidate is scaled to a certain unit cell volume, V_{UC}, prior to testing against hard constraints and to local optimization.[14] V_{UC} can be adapted during the run. This enhances the performance for systems where the initial value is not sufficiently accurate. For each new generation, the new V_{UC} is a weighted (weight W_{adapt}) average between the old V_{UC} and the average volume of the N_{adapt} best individuals of the previous generation. Thus depending on W_{adapt}, the method adapts more or less fast to the currently most successful volume(s).

2.4. Local optimization

The approach of locally optimizing every candidate has been used with great success, e.g., for the traveling salesman problem [14] and the problem of cluster optimization [15]. Local optimization increases the cost of each individual, but reduces the search space to the local optima, enhances comparability between different structures and provides locally optimal structures for further usage. Many methods used for crystal struc-

[14] After local optimization the volume of the individual may differ from V_{UC}.

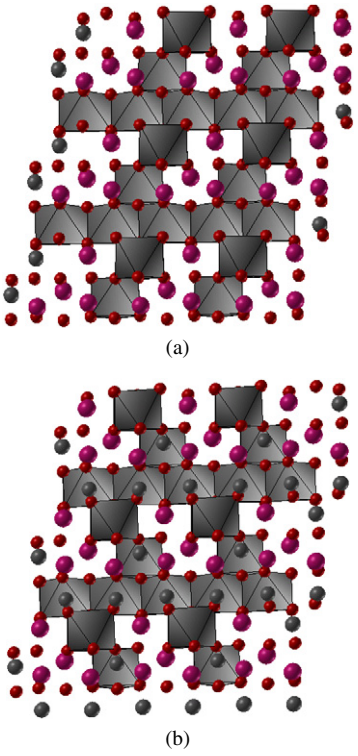


Fig. 3. Example of a permutation. (a) Initial structure, (b) permutated structure. All structures are locally optimized.

ture prediction generate candidate structures and after the run, locally optimize only the most promising ones with accurate methods [2,7,8]. We use the *ab initio* free energy as fitness throughout the simulation. The free energy of structures may vary drastically with respect to small changes of interatomic distances. Such changes can be interpreted as noise. While evaluating the influence thereof, we have found that the correlation between locally optimized and not optimized fitness value ranking is weak.[15] Therefore, to achieve representative and comparable free energy values in terms of structure, we deem local optimization of every candidate to be necessary.

For local optimization, a variety of approaches implemented in standard codes (e.g., [11,12]) can be used, for instance, steepest descent conjugate gradient methods. Both the atomic coordinates and the lattice parameters are locally optimized.

2.5. Hard constraints

Every candidate produced by the variation operators (or randomly generated in the first generation) is tested against three hard constraints:

- Atom-dependent minimal interatomic distances.
- Minimum/maximum values of the angles α, β and γ.
- Minimum lattice vector length.

Candidates violating at least one of these constraints are disqualified. Hard constraints fulfill two purposes. First the minimal interatomic distances must be sufficient to ensure stability of *ab initio* calculations (e.g., to ensure that there is no pathological overlap of pseudopotential core regions.). Second, they reduce the search space and allow for inclusion of system-specific knowledge (e.g., if one knows a certain interatomic distance to be large in reality, one can set that distance to a large value). Further or other constraints are possible, however ensuring stability is vital and the set of constraints mentioned above works very well in ruling out both infeasible and redundant regions. For large systems the interatomic distance constraint is only applied after optimizing the structure using a potential. This is important, since with increasing number of atoms, the probability of generating candidates that satisfy this constraint decreases fast.

2.6. Input–output

The minimal input is:

- Number of atoms of each type.
- Initial guess of the unit cell volume, V_{UC}.
- Hard constraints.
- Parameters of the algorithm.

The method is therefore completely independent from experimental data.

If lattice parameters are available from experimental data, they can be incorporated and kept constant during a run. This reduces the search space significantly.[16] Furthermore, crystal structures can be supplied before the run, replacing randomly generated individuals in the first generation. Every locally optimized structure is stored with the corresponding free energy. USPEX is linked to the STM3 visualization code [16], thus even large scale visualization of results is straightforward.

2.7. K-point adaptation

Depending on the lattice dimensions of an individual, the grid (k_1, k_2, k_3) is calculated and adapted. This greatly enhances accuracy and speed of the calculations, rendering the method much faster overall. The number k_i for a given reciprocal space dimension is calculated as follows:

$$k_i = \frac{1}{l_i \times k_{\mathrm{resol}}}, \tag{3}$$

where l_i is the length of a lattice vector and k_{resol} is the reciprocal-space resolution specified by the user. The resulting k_i is rounded to the next higher integer.

[15] Taking the ratio of the sum of rank changes between locally not optimized to optimized ranking and the expected sum of rank changes between random to locally optimized ranking, we have obtained the value 0.73.

[16] All atomic coordinates are strongly coupled with the lattice parameters. This strongly suggests that the six lattice dimensions have a larger impact on the difficulty of the task than two extra atoms (also six dimensions) do.

Table 1
Standard parameter values for a 20-atom system

Parameter	Value	Symbol
Population size	30–60	
Percentage heredity	85	P_{her}
Percentage mutation	10	P_{mut}
Percentage permutation	5	P_{perm}
Percentage shifting along $\vec{a}_{ch}$	100	
Percentage shifting along remaining vectors	5	
Percentage of individuals with selection probability zero	40	
Average number of two-atom-exchanges during permutation	2–3	N_{perm}
Standard deviation of lattice strain matrix	0.7	$\sigma_{lattice}$
Standard deviation of atomic position shifts	0.0	σ_{atoms}
Resolution for k_i determination (smaller values required for metals)	0.12 Å^{-1}	k_{resol}
Weight for V_{UC} adaptation from generation to generation	0.5	W_{adapt}
Number of (best) individuals to be averaged over for V_{UC} adaptation	4	N_{adapt}
Number of (best) individuals of parental population to be considered for environmental selection	1–2	N_{cons}

2.8. Parameters

For a system with 20 atoms in the unit cell with unknown lattice, no starting structures and a reasonable guess at the unit cell volume, a reasonable setting for the parameters can be found in Table 1. Hard constraints are system specific. For the angles $60° \leqslant \alpha, \beta, \gamma \leqslant 120°$ makes sense since for any structure there exists a unit cell with these constraints. The minimal lattice vector length should not be larger than the diameter of the largest atom.

A given parameter setting results in a certain behavior of the algorithm. Depending on the system chemistry and the presence/absence of input (lattice parameters, starting structures) the desired behavior will change. Therefore there is no universal optimal set of parameters.

If, for example, a set of good starting structures is available, the proposed parameter values change significantly. Most importantly N_{cons}, P_{mut} and P_{perm} increase, while N_{perm}, P_{her} and $\sigma_{lattice}$ decrease. Thus the search would be more localized and by keeping more individuals, be more restrained to the currently best region—both enhancing exploitation of the information present in the starting structures.

2.9. Parallelization

The computationally expensive part of the algorithm is the local optimization. Locally optimizing different candidates within one generation is independent and can thus be processed in parallel. However only calculations within the same population can be parallelized.[17]

3. Results

The method has been successfully tested on various systems with known structure. An overview of the systems can be found in Table 2. For all these systems calculations were performed with minimal input (see Section 2.6) or providing the lattice parameters where this is specified. For non-molecular systems with up to 80 atoms/cell, we have observed a success rate of close to 100%. Usually the correct prediction was achieved in the first run. With up to approximately one dozen atoms/cell, the global minimum can be found with reasonable effort by random search. Molecular systems are generally harder to predict.

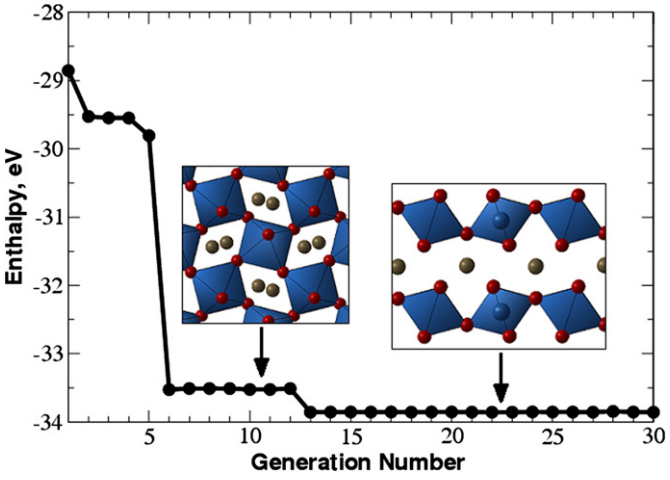


Fig. 4. $MgSiO_3$ at 120 GPa. Enthalpy of the best individual versus generation. Population size: 30.

Furthermore USPEX yields numerous metastable structures, some of which highly competitive, and is extremely efficient: e.g. for structure prediction of $MgSiO_3$ at 120 GPa (20 atoms/cell) with minimal input, only between 150 and 400[18] individuals were calculated before the structure of post-perovskite was found.[19] An example, where both perovskite and post-perovskite were identified, can be found in Fig. 4.

[17] Since in order to generate a new population, all fitness values of the old population need to be known (see Section 2).

[18] Exact timing differed between runs, depending on parameter setting and random factors.

[19] High-pressure behavior of $MgSiO_3$ was thoroughly studied by standard computational methods over the last 20–30 years, but the post-perovskite phase was found [17,18] only after an analogous phase was identified for Fe_2O_3 [19]. This discovery has significantly changed models of the Earth's internal structure and evolution. USPEX finds this structure straightforwardly.

Table 2
Systems with known structures on which calculations were performed—USPEX always found the stable structure

System	Conditions
C	0, 100, 300, 500, 1000, 2000 GPa
C	exp. cell of diamond
Xe	200, 1000 GPa
Si	0, 10, 14, 20 GPa
N	100 GPa
Fe	350 GPa
TiO_2	exp. cell of anatase
SiO_2	0 GPa
Al_2O_3	300 GPa
$MgSiO_3$	80, 120, 1000 GPa
$MgSiO_3$	exp. cell of post-perovskite
Si_2N_2O	0 GPa
$SrSiN_2$	0 GPa
$(NH_2)_2CO$	exp. cell of tetragonal urea
Li	100 GPa

Table 3
Systems with unknown structures, for which we have done calculations and discovered new structures

System	Conditions
O	exp. cell of ϵ—and ξ—phase
O	25, 50, 130, 250 GPa
N	250 GPa
F	50, 100 GPa
$MgCO_3$	150 GPa
S	12 GPa
$CaCO_3$	50, 80, 150 GPa
H	200, 600 GPa
CO_2	50 GPa

An overview of systems with unknown structures for which we have performed calculations and discovered new structures can be found in Table 3. An example is discussed in Section 3.1.

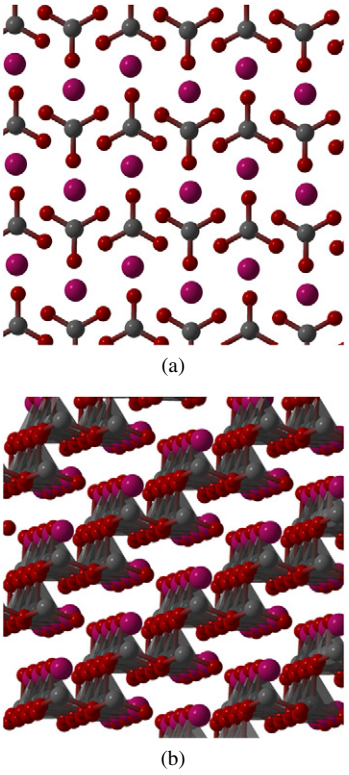
3.1. $CaCO_3$

Chronologically the first application of USPEX for solution of relatively complex and hitherto unknown structures was the study of high-pressure phases of $CaCO_3$ [20]. Experimental studies found a new phase, post-aragonite, to be stable above 40 GPa [21], but could not solve its structure. The structure found by USPEX (see Fig. 5(a)) closely reproduces experimental diffraction pattern; in agreement with experiment, calculations show that this structure is more stable than aragonite above 42 GPa.

Above 137 GPa, our simulations predicted stability of another new structure (space group $C222_1$, see Fig. 5(b)) containing carbon in the tetrahedral coordination. This prediction has been subsequently verified experimentally [22].

4. Conclusions and outlook

USPEX can very reliably find the most stable crystal structure of systems with up to several dozen atoms/cell. Due to local optimization and the process of exploiting promising regions, many highly competitive metastable structures are found during the search. Furthermore, the generated structures and their respective free energies yield information on the chemical regime at the given conditions. Besides identifying stable phases, this method can thus be used for materials design, both in finding promising structures to synthesize and in giving information on what conditions would be best suited for synthesis.


(a)

(b)

Fig. 5. (a) Post-aragonite phase of $CaCO_3$ identified by USPEX. (b) Phase I (orthorhombic $C222_1$) identified by USPEX.

Local optimization and spatial heredity seem to be the key issues for the success of USPEX. Strong changes of free energies due to slight changes of interatomic distances render evaluation of not locally optimized structures unreliable. Therefore information carried by such structures is close to inexploitable. The quality of locally optimized structures per contra is well captured by their free energies. Furthermore, locally optimized structures are a good basis for further creation of new structures. Therefore, in accordance with our experience, local optimization seems crucial for an effective global optimization based on methods exploiting information from a set of structures, like evolutionary algorithms. Local optimization also provides the abundant metastable structures.

Spatial heredity is likely to be the essential variation operator. The quick decay of atomic interactions with distance suggests a partial separability of the given problem. This separability is exploited in spatial heredity. Local optimization further enhances this operation, leading to heredity of close to locally optimal fractions of structures. Due to the prior shifting of the structures, successful 'substructures' can quickly manifest themselves anywhere within the unit cell, leading to an immanent diversification.

Other features complement USPEX to the powerful method it has become. Lattice mutation is an efficient way of search-

ing the neighborhood of promising structures in a way hard to achieve with other operations. In systems where many highly competitive metastable structures differ essentially in a distorted lattice from the stable structure, lattice mutation is very powerful. Permutation seems useful for cases with many different types of atoms involved and/or in cases where different types of atoms have similar properties. 'Seeding' the first generation—i.e. providing structures to incorporate in the first generation—can cut computational costs significantly, where good structures are available. These structures can be chosen from known structures of similar systems, or from previous USPEX runs. Potential strategies are, for example, to include structures from different runs with very small populations, where convergence is fast[20] or from runs with smaller unit cells, where resulting structures are 'blown up' to the desired size.[21] Seeding increases the bias of the calculation which not only can be helpful, but can misdirect the search completely and should therefore be treated with care.

Even on a qualitative level, system chemistry has a huge impact on the landscape shape. The landscapes of, e.g., molecular and ionic systems are very dissimilar. Therefore the optimal parameter setting will vary from system to system. However we have observed that USPEX is capable of finding the stable structure both for different systems using identical parameter settings and for different parameter settings used on the same system. This indicates a high robustness.

Many further developments are envisioned. Major projects are to extend the method, enabling whole-molecule-handling (for molecular crystals) and dealing with variable stoichiometries. Operating on whole molecules greatly decreases the dimensionality of molecular systems. Instead of three dimensions for each atom, only three coordinates and three angles per molecule would remain, at least for molecules, where distortions of interatomic distances and torsion angles can be identified by local optimization. If necessary some of these variables can be included in global optimization. From operating on whole molecules we therefore expect a major impact on molecular systems.

Variable stoichiometries would allow simultaneous optimization of structure and composition. This is especially important for metallic alloys, where it is very difficult to rationalize or predict *a priori* stable stoichiometries. Optimizing stoichiometries has been pursued in [23], where an evolutionary algorithm was used to find stable alloys. However, in that work the structure was fixed (fcc- and bcc-structures). Simultaneous optimization of structure and composition would certainly be a major challenge.

Acknowledgements

Colin W. Glass thanks his dear friend Jo A. Helmuth for many fruitful discussions. Calculations were performed at ETH Zurich and CSCS (Manno). We thank T. Racic, G. Sigut and O. Byrde for computational assistance, and M. Valle for developing the STM3 library for the visualization of our results. This work is supported by the Swiss National Science Foundation under grant 200021–111847/1.

[20] Possible speed-up for system with separated competitive regions in the search space.

[21] This implicitly assumes that successful structures of different unit cell sizes have similarities.

References

[1] J. Pannetier, J. Bassasalsina, J. Rodriguez-Carvajal, V. Caignaert, Prediction of crystal structures from crystal chemistry rules by simulated annealing, Nature 346 (1990) 343–345.
[2] J.C. Schön, M. Jansen, First step towards planning of syntheses in solid-state chemistry: Determination of promising structure candidates by global optimization, Angew. Chem. Int. Ed. 35 (1996) 1287–1304.
[3] R. Martoňák, A. Laio, M. Bernasconi, C. Ceriani, P. Raiteri, F. Zipoli, M. Parrinello, Simulation of structural phase transitions by metadynamics, Z. Krist. 220 (2005) 489–498.
[4] R. Martoňák, A. Laio, M. Parrinello, Predicting crystal structures: The Parrinello–Rahman method revisited, Phys. Rev. Lett. 90 (2003) 075503.
[5] S. Gödecker, Minima hopping: An efficient search method for the global minimum of the potential energy surface of complex molecular systems, J. Chem. Phys. 120 (2004) 9911–9917.
[6] T.S. Bush, C.R.A. Catlow, P.D. Battle, Evolutionary programming techniques for predicting inorganic crystal structures, J. Mater. Chem. 5 (1995) 1269–1272.
[7] S.M. Woodley, P.D. Battle, J.D. Gale, C.R.A. Catlow, The prediction of inorganic crystal structures using a genetic algorithm and energy minimization, Phys. Chem. Chem. Phys. 1 (1999) 2535–2542.
[8] S.M. Woodley, Prediction of crystal structures using evolutionary algorithms and related techniques, Structure and Bonding 110 (2004) 95–132.
[9] J. Maddox, Crystals from first principles, Nature 335 (1988) 201.
[10] G.M. Day, et al., A third blind test of crystal structure prediction, Acta Cryst. B 61 (2005) 511–527.
[11] J.M. Soler, E. Artacho, J.D. Gale, A. Garcia, J. Junquera, P. Ordejon, D. Sanchez-Portal, The SIESTA method for ab initio order-*n* materials simulation, J. Phys.: Condens. Matter 14 (2002) 2745–2779.
[12] G. Kresse, J. Furthmüller, Efficient iterative schemes for ab initio total-energy calculations using a plane wave basis set, Phys. Rev. B 54 (1996) 11169–11186.
[13] Z. Michalewicz, D.B. Fogel, How to Solve It: Modern Heuristics, Springer, Berlin, 2004.
[14] H. Mühlenbein, M. Gorges-Schleuter, O. Krämer, Evolution algorithms in combinatorial optimization, Parallel Comput. 7 (1988) 65–85.
[15] D.M. Deaven, K.M. Ho, Molecular geometry optimization with a genetic algorithm, Phys. Rev. Lett. 75 (1995) 288–291.
[16] M. Valle, STM3: a chemistry visualization platform, Z. Krist. 220 (2005) 585–588.
[17] A.R. Oganov, S. Ono, Theoretical and experimental evidence for a post-perovskite phase of $MgSiO_3$ in Earth's D" layer, Nature 430 (2004) 445–448.
[18] M. Murakami, K. Hirose, K. Kawamura, N. Sata, Y. Ohishi, Post-perovskite phase transition in $MgSiO_3$, Science 304 (2004) 855–858.
[19] S. Ono, T. Kikegawa, Y. Ohishi, High-pressure phase transition of hematite, Fe_2O_3, J. Phys. Chem. Solids 65 (2004) 1527–1530.
[20] A.R. Oganov, C.W. Glass, S. Ono, High-pressure phases of $CaCO_3$: crystal structure prediction and experiment, Earth Planet. Sci. Lett. 241 (2006) 95–103.
[21] S. Ono, T. Kikegawa, Y. Ohishi, J. Tsuchiya, Post-aragonite phase transformation in $CaCO_3$ at 40 GPa, Am. Mineral. 90 (2005) 667–671.
[22] S. Ono, Pers. comm.
[23] G.H. Jóhannesson, T. Bligaard, A.V. Ruban, H.L. Skriver, K.W. Jacobsen, J.K. Nørskov, Combined electronic structure and evolutionary search approach to materials design, Phys. Rev. Lett. 88 (2002) 255506.
[24] A.R. Oganov, C.W. Glass, Crystal structure prediction using *ab initio* evolutionary techniques: Principles and applications, J. Chem. Phys. 124 (2006), art. 244704.