The performance of minima hopping and evolutionary algorithms for cluster structure prediction

Sandro E. Schönborn,[1] Stefan Goedecker,[1,a)] Shantanu Roy,[1] and Artem R. Oganov[2,b)]
[1]*Departement Physik, Universität Basel, Klingelbergstr. 82, 4056 Basel, Switzerland*
[2]*Department of Materials, Laboratory of Crystallography, ETH Zürich, 8093 Zürich, Switzerland and Department of Geology, Moscow State University, 119899 Moscow, Russia*

(Received 7 October 2008; accepted 19 February 2009; published online 9 April 2009)

We compare evolutionary algorithms with minima hopping for global optimization in the field of cluster structure prediction. We introduce a new *average offspring* recombination operator and compare it with previously used operators. Minima hopping is improved with a *softening* method and a stronger feedback mechanism. Test systems are atomic clusters with Lennard-Jones interaction as well as silicon and gold clusters described by force fields. The improved minima hopping is found to be well-suited to all these homoatomic problems. The evolutionary algorithm is more efficient for systems with compact and symmetric ground states, including LJ_{150}, but it fails for systems with very complex energy landscapes and asymmetric ground states, such as LJ_{75} and silicon clusters with more than 30 atoms. Both successes and failures of the evolutionary algorithm suggest ways for its improvement. © *2009 American Institute of Physics*.
[DOI: 10.1063/1.3097197]

I. INTRODUCTION

To find the structural ground state of a cluster is a nontrivial global optimization task. One has to find the global minimum of the potential energy surface (PES), which is a function of all the atomic coordinates. Even for a relatively small cluster of 30 atoms, the configuration space has already 90 dimensions. Because knowing the structure is a prerequisite for the study of all other physical and chemical properties, the problem is of great importance and many algorithms have been developed to solve this global optimization problem. We compare *evolutionary algorithms* (EAs), which have successfully been used in many diverse fields with the *minima hopping* (MH) [1] method.

For the prediction of the ground state structure of crystals, the Universal Structure Predictor: Evolutionary Xtallography[2,3] (USPEX) method turned out to be extremely powerful and has already allowed material scientists to find interesting and unexpected new crystal structures.[4–8] Recently EAs have also been successfully used to predict surface phenomena such as steps on silicon crystals.[9,10] A widespread application of global optimization methods is the prediction of the structure of various clusters. In this field the majority of the work has been done with genetic or evolutionary methods as well.[11–14] We note that different EAs developed for various types of structure prediction problems (molecules, clusters, and crystals) have significant differences. Even for the same type of problem (e.g., crystal structure prediction) the previously proposed algorithms are very different in their construction and performance. The minima hopping method has been successfully applied to benchmark systems[1] as well as to silicon clusters and AFM tips.[15]

In the presented EA, we employ the ingredients which turned out to be successful for the crystalline case within USPEX and which can be transferred to the case of clusters. The version of minima hopping we are using is based on an improvement of the two key features of the original minima hopping method.[1] The feedback mechanism is enhanced and the Bell–Evans–Polanyi (BEP) principle[16] is exploited in a more efficient way by moving preferentially along soft directions in the molecular dynamics (MD) part of the minima hopping algorithm.

Our comparison of minima hopping and EAs is based on Lennard-Jones systems, especially the cluster with 55 atoms, which is an example of an easy one-funnel structure, and the 38-atom system, which is known to have a complicated double-funnel structure. We also apply the algorithms to more realistic systems, namely, silicon clusters described by a force field and gold clusters described by an embedded atom potential.

This paper is structured as follows: We first introduce the evolutionary method used. After a quick introduction to minima hopping and its modifications, we present the results section containing a comparison between minima hopping and the EA. Finally we also test different aspects of the EA and MH.

II. THE EVOLUTIONARY ALGORITHM

EAs implement a very simple model of biological evolution. They work on a set of samples—a *population*—which is gradually improved by selection and reproduction of fit members of the population—*individuals*. Each individual is a solution candidate. A single iteration step leads from a population to the next and is called a *generation*. The algo-

a)Electronic mail: stefan.goedecker@unibas.ch.
b)Current address: Department of Geosciences, Department of Physics and Astronomy, and New York Center for Computational Sciences, Stony Brook University, Stony Brook, NY 11794-2100, USA.

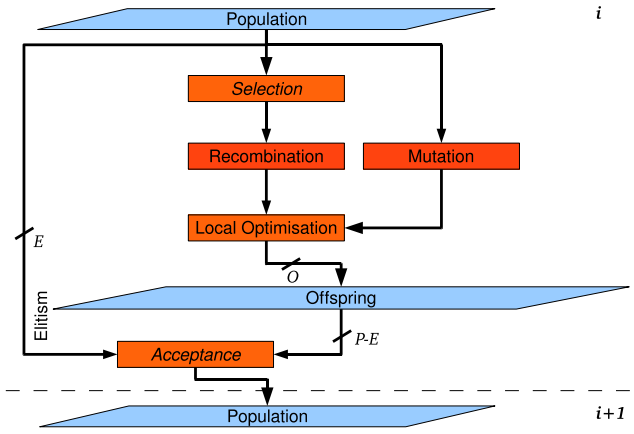

FIG. 1. (Color online) EA: A population i is evolved into the next generation $i+1$ using mutation and recombination operators. E: elitism (number of individuals kept from the old generation). O: total number of offspring produced. P: population size.

rithm optimizes the *fitness* function—in our context the negative energy of the configuration. The operators applied to the population to obtain the next generation are the heart of the algorithm as they determine its quality and properties.

In contrast with the original simple genetic algorithm, modern applications in the field of chemical structure prediction all use real value encoding instead of binary strings and phenotypical operators acting directly in real space instead of gene modification.[14,17] Also state of the art is the application of local optimization to each individual thus reducing the search space to basin bottoms. Local optimization is done using standard techniques such as steepest descent and conjugate gradient methods.

Figure 1 presents an overview of the EA used. It starts with a randomly initialized population, this is our generation 0. In each generation the algorithm goes through three steps: *selection*, *application of operators*, and *acceptance*. The step selection + operator produces new offspring and mutations and puts them into an intermediate population. In acceptance the next generation is selected out of all available offspring together with individuals from the old population.

The algorithm possesses many tunable parameters. The most important values are population size, the number of offspring produced, the number of individuals kept from the last generation (elitism), and the mutation rate (for a complete list see Table I).

As is often the case in EAs in this field an energy slot restriction allows only one candidate per energy interval energy slot in the population. This method of preventing multiple copies of the same configuration in the population may be dangerous for it might reject an important candidate having almost the same energy as an individual already known. Using force fields allow to calculate energy and forces with very high precision since the numerical noise is extremely low. It is thus easily possible to identify structures by their energy.

Recently, a different structure of EA, optimized for parallel machines, has been presented.[18] Instead of a stepwise evolution this approach handles a big pool of individuals, which is subject to continuous application of operators. This is closer to a biological population without sharply defined generation gaps and it solves the load balancing problem in parallel implementations of EA.

A. Operators

Operators are used to evolve a population to a next generation. We use two different kinds of operators, *heredity* operators, which take two individuals as input and produce a *child* sharing properties of both *parents*. The second kind is an operator applied to a single individual altering its configuration (*mutation*).

The selection step determines to which individuals the operators are applied; it is dependent on the operator. For a heredity operator there are two parents selected whereas for mutation operators only one individual is chosen. Selection is done using a linear ranking scheme. Individuals are sorted with respect to their fitness values and then assigned a probability depending linearly on the rank i. The probability of selecting the individual with rank i is in this case

$$P[i] = P_1 - (i-1)\frac{P_1}{c}, \tag{1}$$

where i is the rank, starting at 1, c is the parameter cutoff, and P_1 is the first selection probability determined by normalization constraint. The cutoff value is the last rank with a selection probability above zero; all following ranks are assigned zero selection probability. The same method is also used in USPEX method.[2,3] It turned out to be more efficient than Boltzmann selection where the selection probability fol-

TABLE I. Parameters of the EA.

Function	Name	Standard value
Population size	Population size	30
Number of offspring produced	Offspring	Population size
Number of individuals taken from former population	Elitism	$\frac{1}{3}$ population size
Last rank with selection probability >0	Cutoff	$\frac{2}{3}$ population size
Relative rate of offspring produced with *average* method	Avgoffspring	0.50
Only one individual allowed within this energy interval	Energy slot	10^{-4}
Total rate of mutation	Mutation rate	0.05
Random walk mutations (relative to total mutations)	Mutrwalk	0.60
Strain mutations (relative to total mutations)	Mutstrain	0.30
Probability of random rotation before recombination	Raterndrot	0.90
Convergence criterion for force norm in local optimizer	Fnrmtol	10^{-4}

lows a Boltzmann distribution depending on relative fitness values. The selection of the same individual serving as both parents is prevented. Mutation operators are applied randomly to the whole population.

The first heredity operator used is the cut and splice (*cutting-plane*) method introduced by Deaven and co-workers.[11,12] Both clusters are centered at their center of mass (COM). A randomly oriented plane cuts the two clusters apart. The new *offspring* cluster consists of one half of the first and the other half of the second cluster. Though we were able to obtain two offspring by this method, we only produced one. The plane usually contains COM and its cut preserves the total number of atoms in the child cluster.

A new way of producing offspring is implemented in the *average offspring* method. Both clusters are centered at COM and for each atom of the first cluster the closest lying atom of the other cluster is identified. The atom of the child is now placed randomly on the connecting line of the two parent atoms. The randomness of this operator is necessary to prevent producing a lot of identical offspring.

Before application of either heredity operator a random rotation can be performed on one of the parents. The frequency at which this rotation is used can be adjusted via raterndrot.

Mutations are introduced to keep the diversity of the population high and prevent premature convergence. Three different methods are used. The easiest of those is the *random walk mutation* where atoms are randomly displaced. The displacement is approximately normal distributed with a mean displacement in the order of the two-body potential equilibrium distance (bond length). A *strain mutation* applies a geometrical deformation to the whole cluster, inspired by mechanical stress. The deformations include (anisotropic) compression and shear. Such strain transformations gave increased efficiency in the USPEX method. If the compression is high this method relates to the *big-bang algorithm*, where configurations are relaxed from very high compressions.[19] We only apply a moderate compression. In a third type of mutation the cluster is cut into two pieces similar to the plane-cutting method. One of the pieces is rotated around an axis perpendicular to the cutting plane by a random angle. This method was introduced as *twinning mutation*.[20]

Since our operators are able to produce configurations with atoms lying very close to each other we use a prerelaxation method, which is essentially a steepest descent with a very small step size. After a few steps the real self-adjusting steepest descent is started until a nearly quadratic region around the local minimum is reached. The final minimization is then carried out by a conjugate gradient method.

If an intermediate set of offspring has been created the *accepting* step is triggered. In this step it is decided whether an offspring is accepted into the new population or is discarded. The algorithm first accepts the best individuals from the former population (*elitism*). The number of individuals chosen that way is given by elitism. In a second step the individual with the worst fitness value is replaced by the best offspring if energy slot constraints are fulfilled. This is repeated until all offspring are processed or the population is complete.

The algorithm is left running until a given limit of generations has been reached or the (known[13]) global minimum has been found.

III. MINIMA HOPPING

Minima hopping is a recently developed global optimization algorithm,[1] which makes use of a BEP principle for MD trajectories.[16] The BEP principle states that low energy MD trajectories are more likely to enter the basin of a lower lying adjacent minimum than high energy trajectories. The algorithm also incorporates a history to repel it from previously visited regions. Using local optimization and MD simulation, it jumps between basins of attraction. The kinetic energy is kept as low as possible to escape the local minimum, but is increased if this minimum has already been visited before.

Minima hopping works with two self-adapting parameters, the kinetic energy of a MD escape step E_{kin} and an acceptance threshold E_{diff} for new minima to introduce a further downward preference. Starting from a local minimum a MD escape trial is started with kinetic energy E_{kin}. After a few steps it is stopped and the configuration locally optimized again. If escaped the new minimum is only accepted when the new energy lies at maximum E_{diff} higher than that of the previous minimum.

Minima hopping is adjusted by tuning five feedback parameters: α_1 decreases E_{diff} when a new minimum is accepted, whereas α_2 increases the threshold on rejection, β_1 increases the kinetic energy when a MD escape trial fails, β_2 increases E_{kin} when the new minimum is already known, and β_3 decreases the kinetic energy if the minimum is unknown. Each visited minimum is added to a history list and marked as known. The algorithm currently uses the energy value to identify different minima. For more details we refer to the original paper.[1]

Minima hopping was used with the standard parameter set presented in the original paper: $\alpha_1=1/1.05$, $\alpha_2=1.05$ and $\beta_1,\beta_2=1.05$, $\beta_3=1/1.05$. The algorithm is efficient since it inhibits revisiting the same configuration many times, which is likely to be the case in thermodynamically inspired methods such as *simulated annealing*.

We present two modifications of the original algorithm. The initial velocity vector of a MD escape trial is moved toward a direction with low curvature (*softening*)[16] and a stronger feedback (enhanced feedback) mechanism is used.

Softening. MD escape trials in the MH algorithm need an initial velocity distribution, which is then rescaled to fit the desired kinetic energy. The velocities are randomly directed for each atom with Gaussian distributed magnitudes. Regardless of the actual distribution chosen, it proved very useful to use *softening* to choose velocities along low-curvature directions. In this way one can typically find MD trajectories with a relatively small energy that cross rapidly into another basin of attraction. In the original MH method low kinetic energy trajectories could only be obtained by using large values for $\mathrm{md}_{\mathrm{min}}$, which results in long trajectories. A direction of low curvature is found using a modified iterative dimer method,

which only uses gradients; no second derivatives need to be calculated.[21]

Starting at a local minimum $\mathbf{x}$ with an escape direction $\hat{\mathbf{N}}$, the method calculates a second point $\mathbf{y}=\mathbf{x}+d\hat{\mathbf{N}}$ at a distance d along the escape direction. The forces are evaluated at $\mathbf{y}$ and the point is moved along a force component $\mathbf{F}^{\perp}$ perpendicular to $\hat{\mathbf{N}}$,

$$\mathbf{F}^{\perp} = \mathbf{F} - (\mathbf{F}\cdot\hat{\mathbf{N}})\hat{\mathbf{N}},$$

$$\mathbf{y}' = \mathbf{y} + \alpha\mathbf{F}^{\perp},$$

$$\hat{\mathbf{N}}' = \frac{\mathbf{y}'-\mathbf{x}}{|\mathbf{y}'-\mathbf{x}|}.$$

After a few steps the iteration is stopped before a locally optimal lowest curvature mode is found. Initial velocities for the MD escape are then chosen along the final escape direction $\hat{\mathbf{N}}$.

If the softening procedure is executed until it converges, the performance drops again. It is important not to overdo softening. Always escaping into the *same* soft mode direction of a given minimum reduces the possibilities of different escape directions and therefore weakens the method. A good indicator was the mean kinetic energy during a run. For a few softening iterations the value decreases whereas it starts to increase again at a certain number of softening iterations. We set the iteration count to the value where the mean kinetic energy was minimal. Typically 40 iterations are done with a step size α and a dimer length of $d=0.01$. Using softening as described above typically increases the minima hopping performance by a factor of 2 or more (see Table IV)

Enhanced feedback. In the original MH E_{kin} is increased by a factor β_2 if the current minimum has already been visited before, regardless of the number of previous visits. An enhanced feedback method uses a value of β_2 depending on the previous visits according to

$$\beta_2 = \beta_2^0(1 + c\ \log N), \tag{2}$$

where β_2^0 is the original value of 1.05 and N is the number of previous visits to this minimum. The parameter c has been set to 0.1 after tests on bigger Lennard-Jones clusters and gold systems. This feedback mechanism reacts slightly stronger if the minimum is visited many times. If the system has only one energy funnel, this enhanced feedback can even be slightly disadvantageous since it increases the kinetic energy too much and thus weakens the BEP effect of MD. The increased feedback mechanism improves the efficiency considerably for large systems where the system can be trapped in huge structural funnels. If a cluster has for instance both low energy icosahedral and fcc structures, it takes a very long time for the MH algorithm without enhanced feedback to switch from one structure to the other. Such a system is for instance the LJ_{100} cluster of Table IV where the enhanced feedback doubles the performance.

In contrast with the basin hopping method,[22] minima hopping is not a thermodynamic algorithm that creates some probability distribution. As a matter of fact after having found the global minimum, the system will visit higher and higher energy regions in very long minima hopping runs and finally explode. It maps thus out all low energy configurations and gives the configurational density of states reliably up to some upper limit, which increases with increasing computer time. Even though minima hopping uses, like Monte Carlo methods, an acceptance rejectance step, the acceptance rejectance is not based on the metropolis algorithm for sampling a thermodynamic distribution. As a consequence there is no temperature parameter in the minima hopping algorithm.

Minima hopping can be transformed into basin hopping by the following steps. The numbers listed in the following comparison were obtained by tests with a LJ_{150} cluster.

- The thresholding in the acceptance rejectance step has to be replaced by a metropolis acceptance rejectance step with a Boltzmann factor. For the temperature (=energy) parameter E in the Boltzmann factor of the metropolis step, we can in principle use the same feedback mechanism that we use for the E_{diff} parameter in the thresholding step of the minima hopping method.[1] If this is done the efficiency of the resulting method is roughly equal to the one of the original minima hopping. The average value found by the feedback mechanism for E_{diff} and the temperature E (expressed in units of energy) is also virtually identical in both cases and has the value of .28. If, however, one switches off the feedback mechanism for $E=E_{\mathrm{diff}}$ and fixes the value at .28 the performance in terms of the number of geometry optimization degrades by a factor of 3. The reason for this is that small values of E are optimal to go down in a funnel, but much larger values are needed to jump out of a wrong funnel. With the feedback mechanism of minima hopping, the values for E_{diff} are adjusted according to the circumstances and take on the optimal values for whatever situation is encountered. By choosing other constant values of E, the overall performance could be improved somewhat, but it is clear that with a constant value one can never reach the performance of a flexible value.
- In the standard version of basin hopping the moves are done with random displacements instead of MD. Choosing a step size for the random displacements of $\pm.38$ allows to leave the current local minimum in half of all cases. This is only slightly worse than the 30% change of not being able to leave the current minimum in a single escape trial step of the minima hopping method. If one wants to accept also half of the new configurations, one has to choose a value of E of 10 in the metropolis acceptance/rejectance step. With such a large value it is virtually impossible to find the global minimum of a large system such as the LJ_{150} cluster because one will sample mainly the astronomically large number of high energy structures. In minima hopping one chooses actually typically first a much smaller temperature E and adjusts then the step size such that half of all configurations are accepted. The resulting step size is also close to the value of. 38 because in a first approximation all the escapes that lead out of the

current minimum are rejected whereas the unsuccessful escapes that lead back to the current minimum are accepted. This is again very inefficient since one is not able to escape from a current minimum during many trial escape steps while one has to perform many geometry optimizations just to find the same local minimum again and again. Because of these problems, the basin hopping method using random displacements is not very efficient for large clusters. We started 20 runs at two temperatures of $E=10$ and $E=1$ and none of these runs were able to locate the global minimum within a run time, which was 10 times longer than the average run time of minima hopping. In the MD based escape trials of the minima hopping method, one can achieve both a high escape probability and a high acceptance probability of new local minima configurations since the local minima found by low energy MD trajectories have on average much lower energy than the minima found by random displacements.[16] The problems of basin hopping are less severe for small systems where the basin hopping method allows to find global minima[23]

Parallel minima hopping. Parallelizing minima hopping is straightforward. On each processor an own MH process is started, all sharing the same history list. Only energy values of visited minima have to be shared. Due to the feedback mechanism and the common history list, overlap of search areas is penalized in this parallel setup and running on several processors can thus yield an almost linear speedup in runtime. A further effect can be exploited in parallel runs. A single run might easily get trapped in a metastable funnel with high escape time, but the probability of all processors getting trapped in this local minimum is exponentially reduced with the number of processors running in parallel. The total expected runtime until success is therefore less influenced by the long escape times of relatively stable local minima in parallel runs. On the other hand there is a minimal number of local minima that has to be visited before the global optimum can be found. This minimal number of hops renders the use of too many processors less efficient again leading to an optimal number of processors depending on the structure of the PES. The idea of parallel sampling is known to have a positive effect.[24]

IV. COMPUTATIONAL EXPERIMENTS ON CLUSTERS

Atomic clusters are an ensemble of bound atoms, bigger than a dimer but smaller than bulk matter. They show interesting properties in the transition region of single atoms to bulk matter. From a global optimization point of view they are complicated multidimensional systems usually difficult to optimize since they contain a lot of local minima. Therefore they are of interest to test the capabilities of an optimization algorithm.

A simple model of chemical interaction of two noncharged atoms is the Lennard-Jones potential. The potential well depth and the equilibrium distance are the only parameters; both are set to 1.

Such models represent rare-gas clusters reasonably well. Lennard-Jones clusters are thoroughly studied model systems with well-known global minima up to 1000 atoms. Some of those clusters have a nontrivial multifunnel PES where the global minimum is not easily found. The use of the Lennard-Jones two-body potential is fast compared to more accurate potentials or even density functional theory (DFT) calculations. For these reasons they are well suited to test global optimization methods. Our algorithms are applied to Lennard-Jones systems consisting of 38, 55, 75, 100, and 150 atoms. All clusters chosen show different aspects: LJ_{55} is a very easy system, LJ_{38} has a double funnel structure, LJ_{75} is a very hard double-funnel system, and LJ_{100} and LJ_{150} are examples of bigger clusters. Additionally, we perform single runs on LJ_{39}, LJ_{74}, and LJ_{98} to find the ground state motif dependence of the algorithms. For these additional clusters we have pairs of similar-sized clusters with very different ground state geometries.

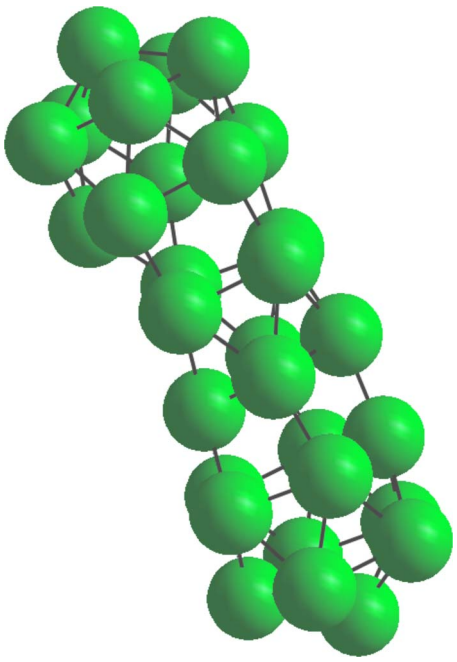

FIG. 2. (Color online) Si_{30} ground state with EDIP.

Silicon clusters are of more practical interest as silicon is so widely used in research and technology. To obtain realistic results usable in those fields, it would be necessary to calculate energies using at least DFT methods. However those methods are consuming a lot of CPU time and one should highly optimize the algorithms applying DFT. Especially in the field of global optimization it is of importance to use an algorithm which is efficient to save computing time. To investigate the behavior of the presented algorithm, we use only a force field to evaluate the energy of a configuration. Silicon systems are chosen as model systems since they possess directed bonds and show frustrated behavior. They have nontrivial minimum structures which are in general neither compact nor spherically symmetric. Silicon systems containing 18, 22, 30, and 60 atoms are explored. The force field used is Bazant and co-workers'[25–27] *environment-dependent interatomic potential* (EDIP).[28] This force field has been chosen because it tends to elongated minimal structures, which are not spherical (Fig. 2). There exist other force fields pre-

ferring spherical ground states. However the performance with spherical ground states is already investigated in Lennard-Jones and gold systems. Such highly elongated structures might not be very realistic as more accurate DFT calculation suggests only slightly elongated ground state configurations.

In gold systems different configurations can have very similar energy and the global optimum is often only very slightly lower than the next-best solution. This leads to a situation where the global geometry of the gold cluster can change completely by adding only a single atom. The energies of gold clusters are calculated using an empirical many-body potential by Rosato, Guillope, and Legrand.[29] Gold clusters with 28, 76, and 102 atoms are investigated. Additionally, Au_{79} and Au_{101} are added to the test set in order to have different ground state geometries at similar sizes.

To compare different algorithms we measure two quantities, the total number of calls to the function calculating energy and forces and the total number of local geometry optimizations. The local optimization is the most expensive step in both algorithms and the speed is mainly determined by its number. To obtain meaningful numbers, we performed between 20 and 100 runs on each problem. The numerical convergence criterion for local optimizations is $\|\mathbf{F}\| < 10^{-4}$ for the total Euclidean force norm, which leads to an energy precision of almost $\sim 10^{-8}$.

V. RESULTS

We gathered results for total performance on the investigated systems. Additionally, we also did comparative runs concerning the new modifications mentioned above. The numerical results of the different performance runs can be found in Table III, whereas special test cases are addressed later. The table shows the results obtained using the best parameter set known to us. Those values are usually a combination of both heredity methods and all mutations. Minima hopping found the global minimum in all problems investigated, whereas the EA could not find the global minima for LJ_{75}, LJ_{98}, Si_{30}, Si_{60}, Au_{79}, and Au_{102}.

A. General performance

The comparison shows a good applicability of minima hopping in all problems. The EA is capable of finding most of the ground states but fails in some cases. Where it is successful it can even outperform MH. Problematic for the EA are the nonicosahedral ground states, such as LJ_{75}, LJ_{98}, and the elongated silicon clusters.

The failure of the EA to find the elongated ground state structure of silicon systems shows a clear advantage of minima hopping; it is not geometry dependent as the current EAs are. Moving via MD it is applicable to any system for which forces are available. In the past there have been approaches inspired by genetic algorithms which used standard crossover and in the beginning even binary strings as coordinate representation. Those ideas seem to be a bit less geometry dependent than the nowadays applied heredity methods. However the original simple genetic methods have been shown to be less effective than more elaborate geometrical operators in Lennard-Jones systems.[17]

TABLE II. Performance of the EA depends on the motif of the ground state configuration. Finding nonicosahedral ground states in systems bigger than 38 atoms is problematic with current EA. Minima hopping was able to find all of the listed configurations. A *no* means failure to find the ground state structure within 100 000 local optimizations.

System	Structure	Success
LJ_{39}	Icosahedron	yes
LJ_{38}	Fcc	yes
LJ_{74}	Icosahedron	yes
LJ_{75}	Marks decahedron	no
LJ_{100}	Icosahedron	yes
LJ_{98}	Tetrahedron	no
Au_{76}	Icosahedron	yes
Au_{79}	fcc	no
Au_{101}	Icosahedron	yes
Au_{102}	fcc	no

When comparing two clusters of similar size but with different ground state structures, it is obvious that the EA can in general not reproduce nonicosahedral ground states with the operators presented here. The results of these tests can be found in Table II. This table contains results of comparative runs without enough statistics to compare performance. The problem has already been identified and solved by *niches* to prevent domination of the whole population by only one geometry type.[30] Using niches it is possible to find the ground state of all the Lennard-Jones cluster in Table II for which the standard EA fails. When implementing niches one has to introduce some measure that allows differentiation between different types of clusters such as icosahedral clusters and octahedral clusters. A niche is then a certain interval of values of this measure. During the genetic algorithm the population of one niche is not allowed to grow so strongly such as to annihilate the population of other niches. In spite of the significant improvement that can be obtained by the introduction of niches, we renounced implementing this most sophisticated variant of EAs for the following reasons. To define niches we need already some knowledge about the system. We have to know initially for example that the system will take on either icosahedral or fcc-like structures. For this reason we would presumably have been able to locate as well the ground states of the gold clusters in Table II, but we could have missed some more complicated ground states with unexpected structures.[31] The introduction of niches would probably not allow to find the ground states of the silicon clusters in Table III with EAs. Second, implementing niches is not trivial and for this reason the majority of EAs renounce using niches. Third and last, it is to be expected that also other global minimization methods including minima hopping could be made more efficient by introducing niches.

Minima hopping is able to find the nonicosahedral motifs without further modification in the standard configuration though with a decreased performance comparing to the icosahedral configuration of similar size. The results show

TABLE III. Best Results. Highlighted in bold face are the problems where the EA performed better. All values are averaged over multiple runs, indicated in the column *runs*. Standard deviations are as big as the mean values.

Cluster		Minima Hopping				EA			
	Energy	GO no.[a]	Calls (GO)[b]	Calls (MD)[c]	Runs[d]	GO no.	Calls	Configuration[e]	Runs
LJ_{26}	−108.315 62	96	50 610	8300	100	**56**	**34 200**	10-4-6	100
LJ_{38}	−173.928 42	1190	688 500	85 930	100	1265	732 900	25-0-20	100
LJ_{55}	−279.248 47	190	74 840	14 700	100	**100**	**54 900**	10-3-3	100
LJ_{75}	−397.492 33	27 375	12.3×10^6	2.11×10^6	20	⋯	⋯	⋯	⋯
LJ_{100}	−557.039 82	5960	1.87×10^6	417 000	42/50	5908	3.89×10^6	60-20-40	35/40
LJ_{150}	−893.310 26	9490	3.72×10^6	758 000	45/50	**7980**	4.36×10^6	60-20-40	17/20
Si_{11}	−44.912 23 eV	29	11 790	4190	60	61	31 300	10-3-4	50
Si_{18}	−74.884 19 eV	110	23 100	10 700	93	195	110 100	10-4-6	40
Si_{22}	−92.480 90 eV	370	187 400	52 100	21	3300	1.74×10^6	10-4-6	1/5
Si_{30}	−126.095 2 eV	5050	3.98×10^6	750 000	100	⋯	⋯	⋯	⋯
Si_{60}	−253.055 09 eV	23 300	14.1×10^6	2.80×10^6	44/49	⋯	⋯	⋯	⋯
Au_{28}	−99.951 15 eV	54	26 210	3510	100	87	68 910	10-3-6	50
Au_{76}	−279.347 91 eV	1124	526 500	70 800	98	2680	1.60×10^6	25-5-13	19/20

[a]Average number of local optimizations over all runs.
[b]Calls to energy function during local optimization including softening.
[c]Calls to energy function from MD escape steps.
[d]If two numbers are present some runs had to be stopped due to too long runtime, the first number indicates the successful runs.
[e]Configurations of EA: populationsize-elitism-cutoff.

that a further adaption of EA to some specific features of clusters is necessary to enable the EA to work as efficiently for clusters as they do for periodic solids.

Minima hopping tends to need in total more calls to the energy and forces function than the EA in the cases where the EA succeeded (at least in smaller systems). This is clearly due to the use of MD and softening in minima hopping. However we remark that it is not necessary to perform MD escape with full accuracy. Within a DFT based global geometry optimization one can do the MD escape part for instance using a reduced basis set and therefore with significantly less computer time per force evaluation. For this reason we separately listed the number of force evaluation for the geometry optimization and MD part in Table III. On the other hand the EA produces structures that are usually far from local minima and initial stages of their local geometry optimization could be done within a DFT calculation at reduced precision or even using force fields. Within such a scheme the number of force evaluations for the EA in Table III would overestimate the required CPU time.

On the other hand minima hopping never produces energetically awful candidates since it moves via MD. Since heredity and mutation operators do not necessarily generate chemically reasonable configurations, the local geometry optimization requires more force evaluations in the EA. Since in DFT applications the early stages of local optimization do not need high precision they could even be done using force fields thus reducing the computational demand.

B. Heredity methods

While comparing both heredity methods used we observed that average offspring method performed better in systems with a compact optimal structure. The plane-cut method could only produce offspring as good as the other method by choosing a very low cutoff. When Daven *et al.*[12] applied this method they produced every single combination of two parents from a very small population, so having actually a very low cutoff level too. With the lower cutoff level the *plane-cut* method performed well overall.

The samples obtained using the average offspring method are energetically closer to the fittest member of the population where the *cutting plane* method samples broader with more diversity [Figs. 3(a) and 3(b)]. It is a known fact that decreasing diversity in the population can lead to premature convergence.[32] On the other hand, if samples in the complete energy range are allowed, the method resembles too much a random search and loses efficiency. To observe a well evolving population it is necessary to have a balanced distribution of selected individuals.[32] It seems therefore necessary to lower the cutoff parameter in plane-cut runs to select more of the fitter individuals as parents. A mix of both heredity methods delivered the best results [Fig. 3(c)].

The plane-cut method is better suited to a general application of the algorithm; it can partially solve geometries less compact than the average offspring method. In big clusters it proved useful to combine all operators available. The results, especially LJ_{100}, were best with a combination of all presented methods. In general the mixture was at a 1:1 ratio or even more preferring plane cuts in systems with known tendency toward nonspherical ground state (e.g., Si_{18} in Table IV).

We also tested different plane-cut setups with a slightly modified method where a minimal distance between the two cluster halves is enforced. This method performed poorly and was always weaker than all different methods tested. Another modification where COM is not enforced to lie in the plane was also considered and dropped since there were no improvements.

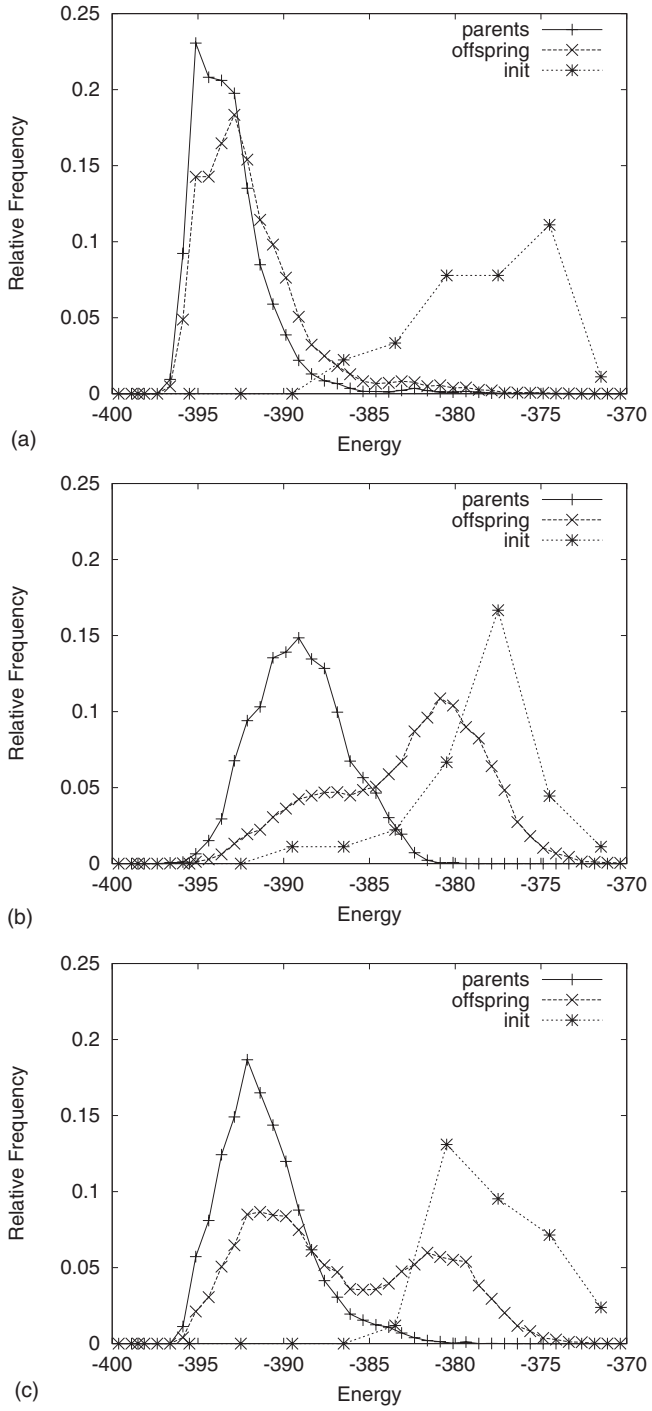

FIG. 3. Candidate samples in LJ_{75} after 10 000 minima. *Parents*: all energy values of candidates selected as parents. *Offspring*: all produced samples. *Init*: initial population. The global optimum is located at −397 5. (a) *Average offspring* method with a relative cutoff of 1, (b) *Plane-cut* method with a relative cutoff of 1/3, (c) Combined 1:1 with relative cutoff 2/3.

The random rotation before recombination is necessary for average offspring method, but only of advantage for the plane-cut method (see Table V). If the system prefers nonspherical configurations raterndrot should be small.

C. Parameter tuning

In systems with a double-funnel structure, where the global maximum is located in the narrower funnel, it might turn out advantageous to disable elitism or (better) include in elitism only sufficiently different structures. In our tests LJ_{38} performed best when filling the population with offspring only. However we should remark that average offspring method has a rather preserving character often reproducing candidates already known.

TABLE IV. Heredity methods in direct comparison.

	Average offspring		Plane-cut	
Cluster	GO no.	Configuration	GO no.	Configuration
LJ_{55}	119	10-2-8	100	10-3-3
LJ_{38}	1265	25-0-20	1595	25-0-10
Si_{18}	322	10-3-6	195	10-4-6
Au_{28}	87	10-3-6	88	10-2-6

A drawback of the EA is the need to tune many parameters. A solution working with good performance on many different systems without adjustment would definitely be of interest. Minima hopping performed better in this aspect: it never needed additional tuning and all runs were done using the same set of standard parameters. Using a standard set (Table I) for all problems resulted in performance loss of the EA. The problems still converged but with performances down to half of the tuned versions shown in Table III. A possible way to overcome this limitation is to fix the relative rates of elitism and mutation, etc., and only adjust the population size to the specific problem. Another possibility would be an automatically self-adapting version, which tunes the parameters during runtime. In this case a stable and efficient scheme of parameter adaptation would be needed, which is clearly not a trivial task.

We note that for crystals with up 30–60 atoms in the unit cell, the USPEX algorithm[2,3] proved to perform very well with essentially a universal set of parameters without any parameter tuning. Evolutionary optimization of clusters, which are more complex systems, is more sensitive to parameter values.

D. Modification of minima hopping

Minima hopping has been considerably improved using softening in all studied cases. The use of the enhanced feedback method is advantageous in large or multifunneled systems, but can even have a negative effect in easy systems as

TABLE V. Number of local optimizations for different random rotation rates before heredity operator application for LJ_{55} systems. For each setup 20 runs were performed. Random rotation is crucial for *average offspring* method (AO) but not for *plane-cut* method (PC).

Rotation[a] (%)	AO	PC
0	>1000	280
10	343	142
50	174	142
75	123	127
90	121	115
100	130	147

[a]Frequency of random rotation.

TABLE VI. Geometry optimizations with and without *softening* in minima hopping with different feedback parameters. n/a denotes not tested.

	No Softening	Softening		
System	$c=0$ [a]	$c=0$	$c=0.1$	$c=0.2$
LJ_{38}	2062	1217	1190	990
LJ_{55}	320	140	190	198
LJ_{100}	9100	7300	4700	5800
LJ_{150}	n/a	14 900	9111	11 630
Au_{28}	167	44	44	56
Au_{76}	n/a	979	890	1024

[a]Parameter c is defined in Eq. (2).

LJ_{55} (Table VI). Parameter c in Eq. (2) should not be chosen too large. We used softening and enhanced feedback with $c=0.1$ in the comparison runs.

VI. CONCLUSION

We tested an EA capable of finding ground state structures of atomic clusters. In spite of the success of EA for periodic systems and on surfaces the current EA is overall less efficient than minima hopping in the current implementation. It is not yet able to find global minima with geometrically difficult structures such as elongated silicon clusters and nonicosahedral ground states without the concept of niches. In contrast, minima hopping was able to find all ground states. Where the EA succeeds its performance is comparable to or even better than that of MH.

Further improvements in the EA could make it superior to MH for cluster optimization if the specifics of cluster systems are taken into account, as it was already done for periodic systems in the USPEX algorithm. The MH algorithm, on the other hand, shows a high stability and little need for further adaptation, at least for homoatomic systems.

Minima hopping was improved by doing escape steps in directions with relatively low curvature of the PES and by using an enhanced feedback method.

[1] S. Goedecker, J. Chem. Phys. **120**, 9911 (2004).
[2] A. R. Oganov and C. W. Glass, J. Chem. Phys. **124**, 244704 (2006).
[3] C. Glass, A. Oganov, and N. Hansen, Comput. Phys. Commun. **175**, 713 (2006).
[4] G. Gao, A. R. Oganov, A. Bergara, M. Martinez-Canales, T. Cui, T. Iitaka, Y. Ma, and G. Zou, Phys. Rev. Lett. **101**, 107002 (2008).
[5] A. R. Oganov, C. W. Glass, and S. Ono, Earth Planet. Sci. Lett. **241**, 95 (2006).
[6] A. R. Oganov, S. Ono, Y. Ma, C. W. Glass, and A. Garcia, Earth Planet. Sci. Lett. **273**, 38 (2008).
[7] A. Oganov, Y. Ma, C. Glass, and M. Valle, Psi-k Newsletter **84**, 142 (2007), (http://www.psi/)-k.org/.
[8] Y. Ma, A. R. Oganov, and C. W. Glass, Phys. Rev. B **76**, 064101 (2007).
[9] F. Chuang, C. Ciobanu, C. Predescu, C. Wang, and K. Ho, Surf. Sci. **578**, 183 (2005).
[10] R. M. Briggs and C. V. Ciobanu, Phys. Rev. B **75**, 195415 (2007).
[11] D. M. Deaven and K. M. Ho, Phys. Rev. Lett. **75**, 288 (1995).
[12] D. Daven, N. Tit, J. R. Morris, and K. M. Ho, Chem. Phys. Lett. **256**, 195 (1996).
[13] D. J. Wales, J. P. K. Doye, A. Dullweber, M. P. Hodges, F. Y. Naumkin, F. Calvo, J. Hernndez-Rojas, and T. F. Middleton, The Cambridge cluster database, http://www-wales.ch.cam.ac.uk/CCD.html.
[14] R. L. Johnston, Dalton Trans. **2003**, 4193.
[15] S. A. Ghasemi, S. Goedecker, A. Baratoff, T. Lenosky, E. Meyer, and H. J. Hug, Phys. Rev. Lett. **100**, 236106 (2008).
[16] S. Roy, S. Goedecker, and V. Hellmann, Phys. Rev. E **77**, 056707 (2008).
[17] W. J. Pullan, Comput. Phys. Commun. **107**, 137 (1997).
[18] B. Bandow and B. Hartke, J. Phys. Chem. A **110**, 5809 (2006).
[19] R. H. Leary, J. Global Optim. **11**, 35 (1997).
[20] M. Wolf and U. Landman, J. Phys. Chem. A **102**, 6129 (1998).
[21] G. Henkelman and H. Jonsson, J. Chem. Phys. **111**, 7010 (1999).
[22] Z. Q. Li and H. A. Scheraga, Proc. Natl. Acad. Sci. U.S.A. **84**, 6611 (1987).
[23] R. Gehrke and K. Reuter, Phys. Rev. B **79**, 085412 (2000).
[24] M. R. Shirts and V. S. Pande, Phys. Rev. Lett. **86**, 4983 (2001).
[25] M. Z. Bazant and E. Kaxiras, Phys. Rev. Lett. **77**, 4370 (1996).
[26] M. Z. Bazant, E. Kaxiras, and J. F. Justo, Phys. Rev. B **56**, 8542 (1997).
[27] J. F. Justo, M. Z. Bazant, E. Kaxiras, V. V. Bulatov, and S. Yip, Phys. Rev. B **58**, 2539 (1998).
[28] S. Goedecker, Comput. Phys. Commun. **148**, 124 (2002).
[29] F. Baletto, R. Ferrando, A. Fortunelli, F. Montalenti, and C. Mottet, J. Chem. Phys. **116**, 3856 (2002).
[30] B. Hartke, J. Comput. Chem. **20**, 1752 (1999).
[31] K. Bao, S. Goedecker, K. Koga, F. Lancon, and A. Neelov, Phys. Rev. B **79**, 041405 2009.
[32] D. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning* (Kluwer Academic, Boston, MA, 1989).