Available online at www.sciencedirect.com
SCIENCE DIRECT
Analytica Chimica Acta 566 (2006) 81–98
ANALYTICA CHIMICA ACTA
www.elsevier.com/locate/aca

Properties of a genetic algorithm extended by a random self-learning operator and asymmetric mutations: A convergence study for a task of powder-pattern indexing

Wojciech Paszkowicz *

Institute of Physics, Polish Academy of Sciences, Al. Lotników 32/46, PL-02-668 Warsaw, Poland

Received 22 July 2005; received in revised form 1 February 2006; accepted 20 February 2006

Abstract

Genetic algorithms represent a powerful global-optimisation tool applicable in solving tasks of high complexity in science, technology, medicine, communication, etc. The usual genetic-algorithm calculation scheme is extended here by introduction of a quadratic self-learning operator, which performs a partial local search for randomly selected representatives of the population. This operator is aimed as a minor deterministic contribution to the (stochastic) genetic search. The population representing the trial solutions is split into two equal subpopulations allowed to exhibit different mutation rates (so called asymmetric mutation). The convergence is studied in detail exploiting a crystallographic-test example of indexing of powder diffraction data of orthorhombic lithium copper oxide, varying such parameters as mutation rates and the learning rate. It is shown through the averaged (over the subpopulation) fitness behaviour, how the genetic diversity in the population depends on the mutation rate of the given subpopulation. Conditions and algorithm parameter values favourable for convergence in the framework of proposed approach are discussed using the results for the mentioned example. Further data are studied with a somewhat modified algorithm using periodically varying mutation rates and a problem-specific operator. The chance of finding the global optimum and the convergence speed are observed to be strongly influenced by the effective mutation level and on the self-learning level. The optimal values of these two parameters are about 6 and 5%, respectively. The periodic changes of mutation rate are found to improve the explorative abilities of the algorithm. The results of the study confirm that the applied methodology leads to improvement of the classical genetic algorithm and, therefore, it is expected to be helpful in constructing of algorithms permitting to solve similar tasks of higher complexity.

Keywords: Diffraction pattern; Genetic algorithm; Indexing; Self-learning; Male driven evolution; Mutation; Operator; Premature convergence

1. Introduction

Global search and optimisation methods aim to find (or to approach) a global minimum or at least a 'satisfactory' solution in a large space characterised by a complex shape and many extremes. They are used for solving tasks, which cannot be treated neither in an analytical way nor by using 'hill-climbing' search routines. One of known valuable global-optimisation methods is provided by genetic algorithms (GAs) [1,2] forming a subset of broader classes of global-optimisation strategies called population-based methods and evolutionary algorithms. The application of GAs for optimisation has been initiated by De Jong [3]. GAs are helpful in solving many-parameter optimisation tasks in various domains of science, technology, medicine, etc., where the local 'hill-climbing' techniques (requiring a starting point close to the global solution) are useless.

The concept of GAs follows the old idea of minimizing human efforts in solving difficult scientific and technical problems by learning from nature (for discussion of this concept, see refs. [1,2,4]). The genetic computation proceeds in the space of variables. It mimics the evolution of living organisms represented by points in this space (individuals). In the beginning, initial population of individuals is generated. Next generations are successively created using simplified principles of (Dar-

Abbreviations: DPF, dynamic penalty function; GA, genetic algorithm; HGA, hybrid genetic algorithm; LCO, lithium copper oxide; LM, local minimum; MGA, mixed genetic algorithm; OFN, objective function; RGA, reduced genetic algorithm; SL, self learning

* Tel.: +48 228436601 3301/034; fax: +48 228430926.
E-mail address: paszk@ifpan.edu.pl.

0003-2670/$ – see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.aca.2006.02.055

winian) evolution. The calculation is terminated according to a stop condition such as predetermined fixed number of generations or some convergence criterion. The variables are organised in strings composed of real or integer numbers, vectors, matrices, logical variables . . .; frequently, the variables are binary coded. The basic genetic operators used in formation of each new population include selection, crossover (both may be classified as co-operative operators) and mutation (classified as self-adaptive one). The search in population-based methods is most frequently elitist (i.e. the principle of 'survival of 35 the fittest' prevents the already found "optimal" solution from extinction).

For the given optimisation task, a fitness function describing the population-member quality must be defined together with the variable ranges and other problem-specific constraints. The GA minimises (maximises) an objective function (OFN), which is either identical to fitness or is defined as linearly/nonlinearly scaled fitness. The selection operator ascertains a greater chance to pass the "good" genes to next generations for better-adapted individuals (solutions). This operator determines which individuals are chosen for crossover, whereas the crossover operator determines how the bits are exchanged between the individuals. Selection can be done on the basis of the OFN values (e.g., using the roulette-wheel method, on the ranking, or using the tournament selection). A part of individuals are submitted to crossover (with predefined crossover probability), the remaining ones are simply transferred to the next generation. A mutation operator typically consists of flipping a single bit with a small probability (mutation rate, m). Using this operator ascertains that virtually the whole space of variables is explored. On the other hand, the flipping of a randomly selected bit means a partial loss of genetic information acquired by previous generations. For $m = 0$, the GA performance is strongly limited because there is no means for the given generation to broaden the exploration field (it rather tends to be narrowed due to action of the selection operator). For m exceeding some limit, m_{lim}, the search becomes a 'population-based random-search' method and the search becomes ineffective because of very low speed of action (the value of m_{lim} is indicated as 50% [2]). A compromise m value is needed to ascertain exploration of the whole space with the optimum speed. In algorithms reported in literature, the value of m is either fixed (frequently at ∼1–5% level), or adjusted during calculations. Classical genetic algorithms are, typically, elitist. They exploit three operators: roulette selection, one-point crossover with probability = 1, and mutation.

Search and optimisation methods based on GAs are helpful in solving difficult static and dynamic global tasks in two following categories:

(i) finding a global extremum of a function of a low or moderate number of variables (for example in the static problem of fitting gaussian profiles to multiple peaks in a spectrum);
(ii) finding a solution being better than that which can be found by either deterministic methods or random search (for example a static problem of finding all possible atomic arrangements in very large multicomponent clusters, or a dynamic problem of weather forecast).

The former case concerns the problems where the space size, number of variables, number of local minimums (LMs), calculation time of the fitting-function value are not too large, and constraints are not too complex. The latter case covers the remaining, more difficult tasks.

The genetic approach to global search/optimisation frequently meets the problem of slow convergence (which may limit the real-time applications), of getting stuck in a local minimum, and of low accuracy of the optimum-position determination. Various ways exist which help in improving the GA action:

- using parallel processors;
- hybridising;
- adding new operators;
- modification of the GA through equipping it in some knowledge specific for the given mathematical/physical/engineering or other task.

For an objective function exhibiting many narrow LMs of comparable depth, the search tends to be stopped at one of such minima. For such case, several remedies are known. For example, there exist (related to each other) dynamic penalty function, DPF [5–8], and sequential niching [9] approaches permitting to identify numerous LMs.

Global search and optimisation methods based on the genetic approach find various applications in solving static as well as dynamic computational problems. Examples of surveys and bibliographies of GAs applications can be found, e.g., in refs. [10] (general), [11] (physics and chemistry), [12] (analytical chemistry), [13,14] (engineering), [15] (pattern recognition and machine learning), and [16] (control systems).

Applications of GAs in crystallography started about 10 years ago (see Table 1). Here, the GAs are used, e.g., for molecular design, structure prediction, and solving structures. A large number among the papers listed are devoted to indexing (finding the unit-cell size) and solving the structure (finding the atomic positions and site occupancies) from powder patterns, i.e., to these methods which contribute to determination of crystal structure of polycrystalline materials. Now, for some of the listed crystallographic purposes, commercial GA-based software is available.

Crystallographic studies performed with the use of intense synchrotron radiation sources provide powder-diffraction data of particularly excellent resolution and high counting rate. Solving those coming from large, low-symmetry unit cells frequently requires a considerable computational effort. One can measure a pattern in seconds, minutes or hours, but solving it may require days; some patterns remain unsolved due to limited resolution and computational barriers—that is why the indexing step is sometimes called a bottleneck in structure determination. The development in existing methods [74–78] focuses towards solving patterns of large unit cells, and of specimens including impurity phases. A future progress in indexing techniques is likely to be connected with development of global-optimisation methods.

Indexing is a good GA-efficiency testing object because of complex shape of the objective function: the shape is character-

Table 1
Applications of GAs in crystallography and related fields

Modelling	
Long chain molecules	[17]
Crystal structures	[18]
Structural optimization of clusters	[19–27]
Indexing crystal faces	[28]
Molecular design and drug design	[29–33]
Powder pattern indexing	[6,34–37]
Asymmetric diffraction peak fitting	[38]
Rocking curve simulation	[39,40]
Surface structure determination	[41]
Surface reconstruction	[42]
Prediction of	
Protein crystal structures	[43–45]
Inorganic crystal structures	[46,47]
Solving structures of	
Crystals	[48–55]
Pharmaceuticals	[56]
Proteins	[57]
Simulation of reflectivity data	
For liquid crystals	[58]
For Langmuir–Blodgett films	[59]
For thin solid layers and multilayers	[36,40,60]
General	[61]
Structure refinement	[62]
Thin layer structure	[63]
Determination and refinement of disordered crystal structures	[64]
Determination of texture	[65,66]
Genetic mapping	[67]
Control of growth of thin layers grown by molecular-beam epitaxy	[68]
DNA sequencing	[69]
RNA folding	[70]
Protein folding	[71]
Analysis of gene expression data	[72]
Spectrum similarity determination	[36]
Molecular replacement	[73]

ized by numerous deep but narrow LMs, which are difficult to locate using traditional computing methods. The shape complexity of functions applied in global-optimisation-based indexing algorithms has been illustrated, e.g., in refs. [35,79,80]. In order to ascertain finding the global minimum (true solution) it is desirable to identify all deep LMs. In a preceding study [6], a genetic approach for indexing has been described (working in the space of continuous variables), which permitted to identify the deepest LMs. This was a consequence of using the above-mentioned DPF approach. The solution at which no improvement was observed during a predefined number of generations was identified as a new LM and added to the list of minima, whereas the OFN value in a close vicinity of each identified LM was artificially increased. The DPF approach has enabled finding numerous LMs. The global (true) solution was one of about 15–20 of LMs found during the life of less than 250 generations. This approach permits for identification, storage and comparison of numerous LMs, and for ascertaining that (i) the algorithm stops at the given LM for a short time; (ii) the transition to the next LM occurs in a smooth way; and (iii) the algorithm does not return to already identified LMs. It is especially valuable, because, when using the DPF mechanism, the chance for success is less sensitive to the parameter values of GA operators. In particular, one can use the values providing somewhat enhanced selection pressure, leading to a higher accuracy of the solution position.

The purpose of this study is to analyse the convergence of a modified GA. The approach described improves the explorative properties of classical genetic algorithm and enhances its exploitative capabilities (convergence and accuracy of solution position). It will be shown, that combined use of asymmetric and variable mutation rate, and extension of a GA by a 'nongenetic' local-search operator can improve the global-optimisation efficiency (in order to reduce the number of algorithm parameters studied, the mentioned DPF approach is not implemented in this version of the algorithm).

2. Combining genetic algorithms with local search

One of principal ways of GAs development or extension is combining them with some nongenetic procedures or operators. Such extensions (termed as hybrid, mixed or memetic ones) which can be deterministic or non-deterministic, are aimed towards improving the properties (speed, convergence, accuracy) of GAs. A usual approach consists in combining a GA with another classical method suitable for local search. As noted in ref. [81], one of the ways of hybridisation is incorporating the local search into the basic loop of GA working together with crossover and mutation operators. The natural space for the local search is the real (uncoded) space (the local search can be performed in binary space, as well, see ref. [82] and refs. therein). The artificial intelligence based on hybrid systems is known to be a valuable tool for many applications, as noted, e.g., in refs. [2,13,83].

The local search can be connected with natural evolution processes. Namely, it may be treated as self-learning (or adjustment to the environment) that takes place during the lifetime of an individual; the notion of self-learning (a self-adaptive operator) should be distinguished from learning from others (co-operative operator).[1] For the sake of simplicity, the self-learning will be abbreviated below as 'SL' or learning.

The learning may be exhaustive (i.e., a complete movement reaching the bottom of an adjacent valley), and not exhaustive (a small descending movement which typically does not reach any minimum). In other words, the notion 'exhaustive' is used for methods being able to precisely locate a local minimum. The contribution of SL must be limited because the local search creates an additional selection pressure, which results in weakening of exploration of the studied space. For a large SL rate the exploration tends to be dominated by exploitation, which manifests itself by a more or less strong tendency for premature convergence. There are several possible ways of combining the GAs with local search. The ways of introducing the local search into the calculations may be distinguished by kind of the local-

[1] In literature, learning is frequently connected with Lamarckian mechanisms of evolution; for a discussion of the Lamarckian aspects of evolution see ref. [84].

search method (exhaustive or not exhaustive), the learning rate (low or high) and by introduction (or not) of learning elitism. The GAs including a local-search stage can be divided into three categories: hybrid genetic algorithms (HGAs), reduced genetic algorithms (RGAs) and mixed genetic algorithms (MGAs).

2.1. Hybrid genetic algorithms

Hybrid genetic algorithms (HGAs) are widely used for various purposes. Their properties are discussed, e.g., in refs. [2,85,83]. The notion 'hybrid' can be understood as an indicator of the fact that the given local-search method joined with a GA has nothing in common with natural evolution processes. For the purposes of the present work, a GA will be called hybrid if the more or less classical genetic strategy is combined with an exhaustive (and usually elitist) method of local search. The elitism at the local-search stage ("learning elitism") is understood as "learning of the fittest". Such a strategy does not clearly follow any phenomenon known from nature (why only the fittest should learn?). For an HGA, the local search becomes a complementary calculation stage (either periodically during the search with a specific alternating period or at the very end of the algorithm). Typically, the genetic part of the HGA is a universal code (a hill-finding routine) which can be employed in a wide class of tasks, while the local-search part is a local optimisation routine (a hill-climbing routine) specific for the given task. The HGAs have found many applications and frequently they are concluded to have a better efficiency and/or convergence than those of classical GAs. The local-search step is frequently based on a gradient method or on the simplex algorithm; the latter is particularly interesting as it is a twin algorithm to GAs in the sense that both have the advantage of avoiding the use of OFN gradient; for discussion of such HGAs see, e.g., ref. [2], whereas some recent examples of applications may be found in refs. [24,86]. Recently, an interesting idea of using a 'microgenetic' search as a local-search step of the hybrid method has been elaborated [87]. For description and discussion of various approaches see ref. [2]. The hybrid genetic algorithms undoubtedly are of high practical meaning, as ready-to-use refinement (local search) methods usually exist for specific scientific/technical/medical/economic and other tasks. Creation of HGA through joining a GA with local search is easy in such cases.

2.2. Reduced genetic algorithms

The definition is similar to that of HGA, but the local search is exhaustive and is performed for all (or a large part of) population members or/and generations. Applying a local search to each solution belonging to the population is a known way of improving global-optimisation procedures. This variant has been termed a reduced genetic algorithm (RGA) [85]. In such case, the population is built mainly from individuals situated at (or close to) LMs in the space explored (for illustration see ref. [88], pp. 199–200). Therefore, the populations lose a large part of their diversity at the local-search step, and the improvement of next generations largely depends on number and distribution of LMs. Moreover, extensive local search may require a considerable processor time for multiple evaluation of the fitness at this stage. The RGA approach is 'less stochastic' than that of GA because the local search pushes all individuals (trial solutions) towards the nearest LMs (as large selection pressure is induced by extensive local search): at these conditions, the search space is efficiently explored mostly in the vicinity of initially found LMs.

2.3. Mixed genetic algorithms

The notion of mixed genetic algorithms (MGAs) will be used here for a case where a non-exhaustive local search is treated as an operator acting on a (stochastically chosen) small fraction of the population (cf. the discussion in ref. [89]). Various examples of application of operator of such kind may be found, e.g., in refs. [6,87,89–101]. The postulate for stochastic character of the local-search operator comes from observation of the nature: the living (especially higher) organisms learn individually from the contact with environment, but the amount of knowledge obtained is only partial and it remains individual dependent. Similarly, for an MGA, the local search remains a small perturbation of the (stochastic) genetic search. If this condition is fulfilled, the principal property of GA (random exploration of the whole space studied) should be virtually conserved and completed by advantages coming from the (deterministic or not deterministic) local search. Arabas ([88], p. 196) has discussed the opportunity of treatment of the local search as a genetic operator and has pointed out, that the local search does not need to be exact: it is enough that several steps of this search are performed, what causes acceleration of the extremum-position determination (this case corresponds to the above-mentioned not exhaustive self-learning). If a local-learning operator is used, a balance between genetic (explorative) and local (exploitative) search is a condition for success of calculations (as noted, e.g., in ref. [102]). With too large contribution of learning, the drawbacks characteristic for RGA (see above) are likely to appear.

3. Calculation scheme

In this work, an MGA approach employed earlier for indexing powder-diffraction patterns [6] is modified: the binary coding is used, crossover probability is 50% instead of 100%, and the DPF approach is not implemented. Mutation and self-learning are carried out for a fixed, low number of children (mutation is not allowed for the fittest). Dual variable representation is used. Mutation and crossover are performed on binary-coded variables, whereas both, learning and fitness calculation, operate on real (uncoded) variables. During the coding operation, the insignificant bits are eliminated, permitting for mutation and crossover operation only on these bits, which are important for the specific optimisation problem. The fitness function and the operators are defined as follows (details of the calculation scheme can be found in Table 2).

3.1. Fitness

For the indexing task, the fitness function, f, is assumed to be equal to the s_N function defined in ref. [79], representing a

Table 2
Main features of the genetic algorithm applied in the LCO tests

Space studied	3D space of unit-cell parameters a, b, c
Unit-cell size constraints	2 Å < a < 10 Å; 2 Å < b < 12 Å; 4 Å < c < 16 Å; 20 Å^3 < V < 1000 Å^3; $a < b < c$
Accuracy	0.0012, 0.00015, 0.00018 Å for a, b, c, respectively
Population structure	Two subpopulations of equal size, 100 individuals for each of two subpopulations
Co-operation operators	
Selection	'Genotypic' selection[a] based on fitness, probability determined by Fermi–Dirac function
Crossover	In binary space; two-point crossover
Self-adaptation operators	
Mutation	In binary space; effective mutation rate up to 5.5%, the fittest does not mutate
Self-learning	In real space; not exhaustive quadratic, rate 0–20%, self-learning length = 0.02 Å
Initial population creation	Random values of variables a, b, c
Each next population creation	50% generational replacement
Elitism	Survival of two the fittest individuals (one for each subpopulation)

[a] 'Genotypic selection', ascription of a child to subpopulation of either daughters or sons according to their Hamming distance to the parents (see text).

normalised measure of the difference between the experimental and the trial calculated diffraction pattern:

$$s_N(\vec{x}) = \left(\frac{A}{N}\right) \sum_{i=1}^{N} \frac{\varepsilon_i(\vec{x})}{\bar{e}_i(V)} \quad (1)$$

where the index N refers to the number of these initial reflections of the studied experimental pattern, which are taken into account in the calculation; $\vec{x}$ the vector of variables (for the orthorhombic system: lattice parameters a, b, c); A the normalizing factor; $\varepsilon_i(\vec{x})$ defined as the absolute difference between experimentally determined peak position, q_i, and position of the nearest calculated one, Q_i, i.e., $\varepsilon_i = |q_i - Q_i|$, the positions being expressed in (interplanar spacing)$^{-2}$ units; V the unit-cell volume; and $\bar{e}_i(V)$ the mean distance between the calculated peaks of the trial unit cell of volume V in the vicinity of the i-th experimental peak; the calculation is based on volume estimation method described in ref. [103].

Far from the solution, the s_N function values are of the order of 0.5, whereas the ideal one (in the global minimum) is zero (due to experimental errors, the real value is of the order of 0.01–0.05). For a particular case, the above-mentioned values may be somewhat different depending on the possible systematic extinctions and incompleteness of the pattern, on the presence of spurious peaks, or to experimental errors. In the present implementation, the objective function is defined as (not scaled) s_N.

3.2. Initial population

The initial population is created in the real space by random selection of the value of each variable, taking into account its considered range and constraints (including the convention $a < b < c$ adopted in all calculations). All below-described LCO tests started with the same initial population. The same concerns the NBS texts except of one case when additional constraints were applied (naturally, the OFN values ascribed to population members depend on the pattern).

3.3. Selection

The selection for a member of a randomly chosen pair of parents is based on the Fermi–Dirac function (earlier used, e.g., in refs. [6,104,105]) determining, for the given OFN value, how large is the probability of the given individual to have offspring.

3.4. Asymmetric mutation

For the purposes of the present paper, the population is divided into two equal-sized subpopulations, which differ by a single property. Namely, they are allowed to have different mutation rates (therefore, the adopted mutation scheme is called asymmetric mutation). The subpopulation of individuals characterised by lower mutation rate will be called females, the remaining ones, males. A criterion of 'Hamming distance being closer to the parent' of the given sex determines the subpopulation to which the given child is ascribed; this way of recombination is equivalent to the "genotypic" sex ascription applied earlier for related purposes in ref. [106].

The mutation is performed in the binary space for the given generation with the mutation-rate parameters m_m and m_f, defined as fixed percentage of mutated bits in the given male/female subpopulations, respectively. The 'effective mutation rate' is defined as $m_{eff} = (m_m + m_f)/2$, and the mutation ratio as $\eta = m_f/m_m$. The mutation ratio describes the mutation asymmetry: $\eta = 1$ for symmetric mutations and $\eta = 0$ for fully asymmetric ones.

Introduction of sexual differentiation has been suggested to lead to improvement of the GA efficiency (Section 5.3 in ref. [2]). Realisation of this idea by construction of two subpopulations characterised by different mutation rates is inspired by the postulate of greater differentiation among males in relation to females [107] and by observation of mutation difference in nature between males and females of various species [108–111]. A related asymmetrisation of mutation rates has been recently

considered in some GA implementations in refs. [112–115] (other ways of sexual chromosome differentiation in GAs are known, for example the asymmetrical selection described in refs. [116,117]).

3.5. Crossover

In this implementation, a simple two-point crossover in the binary space is used. It has been reported [118] to provide a better convergence than does a single-point one (see also refs. [119–121]). The chromosomes are considered as rings with the first and last bit positions connected. These rings are cut in two randomly selected sites and the resulting sub-rings are swapped.

The GA scheme is adopted involving the 50% generational replacement approach with the elitist search strategy independently applied to each of the two subpopulations.

3.6. Self-learning

The aim of using this operator was to improve the convergence. Self-learning is carried out in the real space with two parameters, r and l. Here, r is the self-learning rate defined as ratio (expressed in percent) of learning operations in respect to subpopulation size, K; l is the self-learning length (for definition see the text below). The self-learning is introduced in the sense of partial local refinement of a single individual (not in the sense of 'self-learning of the program'). The calculation scheme is as follows. Let us assume that the SL operator is applied to a randomly chosen (uncoded real) variable of the (also randomly) selected individual A. Two auxiliary individuals, B and C, along the corresponding axis are defined then, both at the self-learning length distance, l (AB = AC = l). For the purpose of the present paper, the l value is fixed at 0.02 Å for each of a, b, c lattice parameters. For individuals B and C, the fitness is calculated, and a quadratic interpolation based on points A, B, C is performed. If the parabola has a minimum at some point D, then the individual A is replaced by individual D. Otherwise, the fitter from among B and C replaces A. The whole 'quadratic self-learning' operation requires calculation of three (or two) fitness function values. Consequently, a learning rate of 1% would mean a 2–3% increase of computing time. In this study, the learning rates were tested of values up to 20%, corresponding thus to ~40–60% of processor time increase for a single generation. Such increase of total calculation time would be observed if the SL operator had no influence on the optimisation process.

The above definition of SL operator should enable virtual conservation of the stochastic character of the genetic approach, provided that the self-learning rate is small enough. The expected progress in deep-minimum detection and refinement is based on the opportunity of fast descent if the SL operator acts for a point located on a slope of a valley (the same operator has been implemented as a component of the tested genetic indexing algorithm [6] and rocking-curve fitting algorithm [39], but it has not been described in detail until now).

In the approach adopted, the SL acts in the real space which is equally used for evaluation of the fitness value, while the crossover and mutation operate on the binary-encoded variables. The encoding is performed with required (problem-specific) precision.

4. Results

4.1. General remarks

The calculations were carried out:

(i) for the lithium copper oxide (LCO) orthorhombic pattern;
(ii) sixteen high-quality NBS [122–126] orthorhombic patterns of various unit-cell volumes.

In the LCO tests, there were $K = 100$ 48-bit chromosomes in each subpopulation, making 9600 bits in the whole population The stopping criterion was 10^5 calls in these tests (equivalent to ~200–400 generations).

In the NBS tests, the strings of 39 bits length were used in a population of $2K = 2 \times 50$ chromosomes (total 3900 bits) and the stopping criterion was 10^6 calls. The number of bits was slightly smaller when a significant additional constraint was imposed to the c lattice parameter.

The properties of the described method are studied exploiting an example of indexing of powder diffraction test data of orthorhombic LCO, having three real variables (unit-cell parameters, a, b, c). Twenty-five initial peaks are taken for the calculation. For LCO pattern, the minimised function exhibits multiple narrow LMs in the space studied (the space bounds are indicated in Table 1). The global minimum is at (a_0, b_0, c_0) = (2.8573,[2] 5.7244, 12.4129 Å) with the function value $f_{min} = f(a_0, b_0, c_0) = 0.0188$. With the binary representation applied in this study, the minimum position is at a slightly less precise position (a_0, b_0, c_0) = (2.8572, 5.7231, 12.4111 Å), f_{min} value being equal 0.0195 in this case. The function shape around the global minimum is illustrated in Fig. 1. The concave part of the space around this minimum (this part represents the possible field of action of classical local-search methods) is evaluated to occupy about 3×10^{-5} fraction of the whole space studied. To find the global minimum with a rather poor 0.01 Å accuracy by random search, the number of calls needed is of the order of 10^9 (a single function call takes ~1 μs at a 2.66 GHz PC), whereas a single GA run frequently provides the true solution (with a much better accuracy) in 10^4–10^5 calls (total time less than 1 min, only).[3]

In the following subsections:

- the trends for nine examples of search-parameter combinations will illustrate the minimised function behaviour in the course of the global search (Section 4.2);
- the time (expressed through the number of function calls) necessary to achieve the convergence will be discussed for broad ranges of learning rate and mutation rate (Section 4.3);

[2] 1 Å = 10^{-10} m (SI derived unit).

[3] For crystals of lower symmetry, the following evaluation can give an idea of the computing time: the corresponding number of calls is by at least one to three orders of magnitude larger; moreover, in the low-symmetry even a single OFN calculation time is much longer.

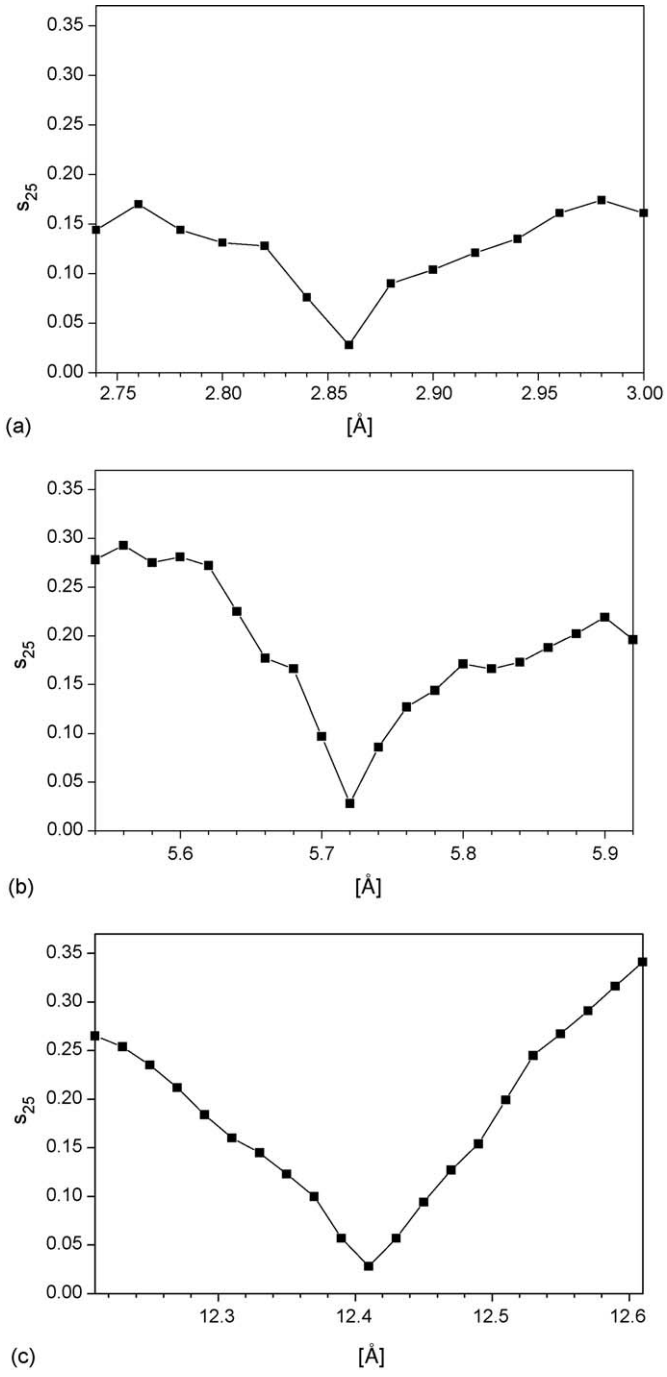

Fig. 1. Shape of the $f(a, b, c)$ function in the vicinity of the global minimum for the LCO pattern, illustrated by the sections along the three axes of the unit cell (lattice parameters a, b and c, values expressed in angstroms).

- the convergence will be studied as a function of effective mutation and learning rates (Section 4.4), and mutation asymmetry (Section 4.5);
- second set of tests are presented for sixteen NBS powder patterns of similar and larger cell volumes (Section 4.6), using a modified scheme of the algorithm.

4.2. Graphical illustration of population diversity and of algorithm convergence

The effect of the studied parameters in the LCO tests on the run of the fittest function value (f) and of the average function value (F) (averaged over the given subpopulation) is illustrated in the representative examples in Fig. 2. For nine examples the global minimum is reached (Fig. 2b, d–f, h–l), for six it is not (Fig. 2a, c, g, m–o). At each graph, the function runs are presented for both subpopulations. They permit to notify some connection between the population diversity: although F is not a precise measure of subpopulation diversity, its high value indicates that the population members are diverse whereas a low value shows that they are similar.

The inspection of the Fig. 2a–o leads to following observations:

(1) The run of f depends on the m_m and m_f values. For low m_{eff} (equal 2%, Fig. 2a–c), a tendency for premature convergence of f is seen. At the same time, the F_m and F_f values quickly decrease during the search (by a factor of 2.5–15 for these examples) what means that the necessary diversity level is lost quickly. Both, male and female subpopulations perform exploitation rather than exploration, so getting stuck into a local minimum is a frequent situation.
(2) The threat of premature convergence is much reduced for all higher m_{eff} values.
(3) For high or moderate m_{eff} value combined with high learning rate (Fig. 2f, i–l, n, o), f_f tends to be slightly lower than f_m during the run; this effect is attributed to the joint action of the crossover, asymmetric mutation and learning. The difference between the two fittest (male and female) individuals is never large: this is explained by the permanent bit exchange between subpopulations during frequent crossover operations.
(4) For the examples of Fig. 2g–l, where the effective mutation rates are moderate, the male population carries out an extensive exploration (consequently, during the whole life F_m is virtually equal, from the logarithmic scale perspective, to its starting value) whereas the females perform rather the exploitation (therefore, F_f decreases to a low value).
(5) For the examples of Fig. 2m–o, where mutation rate of males is still higher what results in enhanced exploration, the convergence is for both subpopulations slower (but systematic) than in remaining examples.
(6) The F_m and F_f behave in a characteristic way:
   - They are identical if $m_m = m_f$ (i.e., $\eta = 1$, "symmetric" case), see Fig. 2d–f.
   - They tend to be virtually identical and equal to their starting values for high mutation rates (m_{eff} about 5.5 or higher), meaning that the search approaches the random one, see Fig. 2m–o.
   - In remaining cases (all with $\eta < 1$) the F_m and F_f start to branch out at the beginning of the calculations, the difference between them increases with time; their values are stabilised usually around the moment when the global (local) f minimum is found.

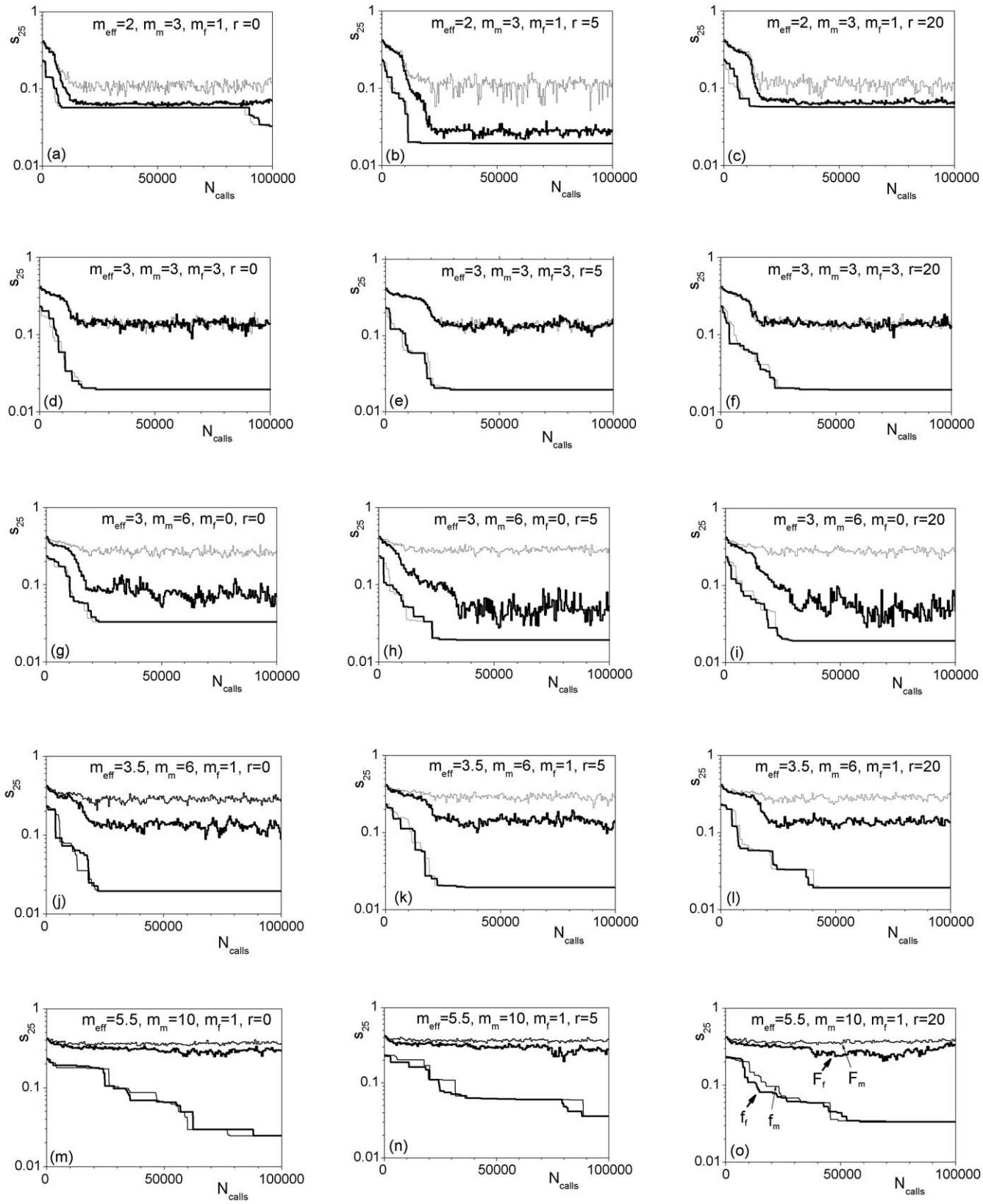

Fig. 2. Examples of the OFN-value evolution (f and F, the fittest and the average) for both subpopulations. The four curves (two for males and two females) are denoted by arrows in the right lower most figure. The examples are distinguished by the values of m_{eff}, m_{m}, m_{f} and r (expressed in percent). In the second row, the mutations are symmetric, in the remaining ones they are asymmetric. These examples of convergence of the global search for LCO indexing are given for a population built from 2×100 individuals. The abscissa represents the number of calls (linear, a relative time scale), the ordinate is the function value (logarithmic). The number of calls per generation varied between about 100 and about 300, depending on the r and m parameter values.

- For the fully asymmetric mutation ($\eta = 0$, Fig. 2g–i), a large noise in the run for the female population is observed after finding the local or global minimum; this effect is apparently due to genetic information coming from explorative males (which frequently have high f values due to mutation rate as high as 6%) to exploitative (not mutating) females. When $\eta > 0$ (cf. the remaining examples), the noise is, consequently, much smaller.

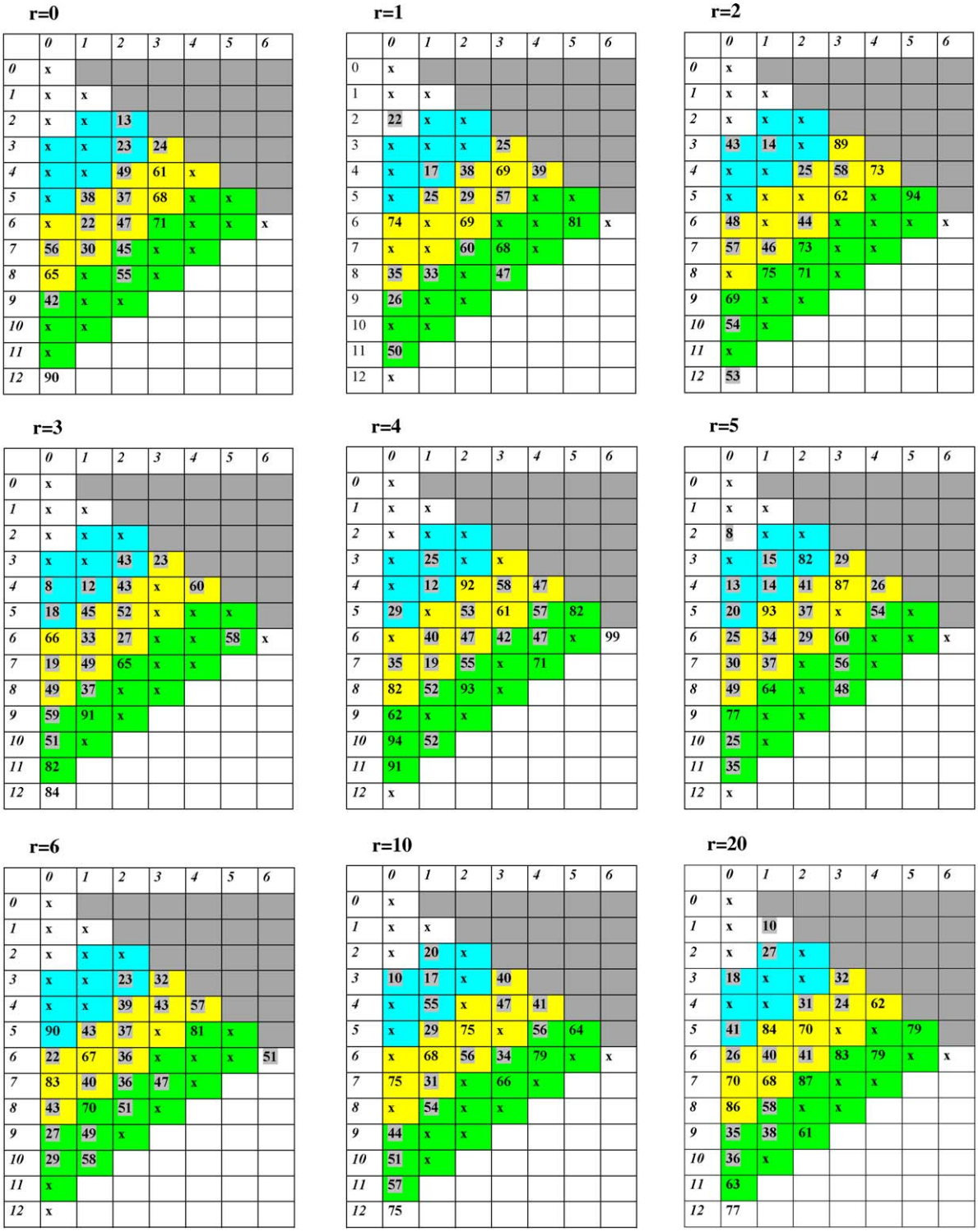

Fig. 3. Convergence of the algorithm for learning rates 0–6, 10 and 20%. The male and female mutation rate are given in italics in the upper row and in the leftmost column, respectively. The values quoted are the call numbers (corresponding to the relative calculation time) rounded to entire thousands of calls. Those not exceeding 6×10^4 calls are additionally marked (small squares with darker background). "x" indicates that the algorithm did not converge in 10^5 calls. The shaded areas represent three different domains of interest. The middle (bright shade) domain corresponding to average mutation rate from 3 to 4%, represents the calculations where the convergence is frequently observed before 10^5 calls. In the upper shaded domain, the convergence is less frequently observed. A similar observation is done for the lower shaded domain.

(7) The convergence to the global minimum is most frequently observed for learning rate equal 5% (four among five examples). The analysis below permits for more convincing consideration of convergence due to many parameter combinations studied.

One can conclude that the case of asymmetric mutation with appropriate m_m and m_f values ascertains a large diversity in the male subpopulation (permitting for exploration) while the female subpopulation (specialised in exploitation) is more conservative. A suitable combination of these parameters results in a fast convergence (fast decrease of f for both, the fittest male and female, see the example in Fig. 2k). Even without more detailed analysis, it is clear that the best combination of exploration and exploitation is reached when m_m is at the level of 4–6% and m_f at the level of 0–1%.

4.3. Analysis of convergence maps

A confirmation of the above reasoning may be found through analysis of convergence maps (Fig. 3). This figure illustrates the convergence of the algorithm at the diagrams constructed for nine selected values of the learning rate ($r = 0, 1, \ldots, 6, 10, 20$). Let us define N_{conv} as the number of calls necessary to reach the global minimum by the fittest individual of the whole (joint) population. The distribution of the N_{conv} values as a function of m_m, m_{eff} and r can be used to demonstrate how the mutation rate, mutation asymmetry and learning rate influence the convergence of the algorithm. At each of the diagrams, areas with different mutation rates are distinguished (A–E, with effective mutation rates m_{eff} from 0 to 1, 1.5 to 2.5, 3 to 4, 4.5 to 5.5, and above 5.5%, respectively). The numbers indicated in the fields of the figure represent the numbers of calls (if less or equal 10^5) for which the known global minimum has been reached. The following observations are done:

- The distribution of the N_{conv} values is not smooth. Such behaviour is due to the stochastic nature of the search method and to the fact that the process can stop for some time at each LM identified during the search. As the way to the solution goes through numerous LMs, the convergence time necessary for reaching the global minimum exhibits large fluctuations.
- In the domain C, the calculations exhibit the most frequent convergence before 10^5 calls. In the domain B, the convergence is less frequent, but if achieved, it is usually quite fast (N_{conv} can be even as low as 8×10^3 calls). In the domain A, the mutations are not frequent enough to ascertain satisfactory exploration of the search space. At the transition area from A to B, the convergence occurs accidentally, only. In the domain D, where the mutation rate is relatively high, the process has a significant contribution of a random search, making that the convergence becomes systematically slower.

Additional illustration of the above-discussed results is given in Table 3, where the cases of the fastest convergence are listed.

Table 3
List of solutions achieved with the independent parameters (m_m, m_f, r) for the stopping criterion $N_{calls} = 3 \times 10^4$

Mutation				Learning rate	N_{calls}		
m_m	m_f	m_{eff}	η	r	N_m	N_f	Min (N_m, N_f)
2	0	1	0	5	9	8	8
4	0	2	0	3	8	9	8
1	1	1	**1**	20	10	10	10
3	0	1.5	0	10	10	11	10
4	1	2.5	0.25	4	12	12	12
4	1	2.5	0.25	3	12	13	12
2	2	2	**1**	**0**	15	13	13
4	0	2	0	5	15	13	13
3	1	2	0.33	2	14	15	14
4	1	2.5	0.25	5	14	15	14
3	1	2	0.33	5	15	17	15
3	1	2	0.33	10	17	17	17
4	1	2.5	0.25	1	17	18	17
5	0	2.5	0	3	18	18	18
3	0	1.5	0.33	20	19	18	18
3	1	2	0.33	20	19	18	18
7	1	4	0.14	4	19	19	19
7	0	3.5	0	3	21	19	19
2	1	1.5	0.5	10	22	20	20
5	0	2.5	0	5	23	20	20
7	0	3.5	0	3	21	20	20
2	0	1	0	1	22	22	22
6	1	3.5	0.17	**0**	22	22	22
6	0	3	0	6	24	22	22
3	3	3	**1**	3	23	23	23
3	2	2.5	0.67	**0**	23	24	23
3	2	2.5	0.67	6	29	23	23
3	3	3	**1**	**0**	24	25	24
4	3	3.5	0.75	20	29	24	24
3	1	2	0.33	4	25	25	25
3	3	3	**1**	1	26	25	25
4	2	3	0.5	2	26	25	25
6	0	3	0.17	5	25	26	25
9	0	4.5	0	1	28	25	25
10	0	5	0	5	25	29	25
4	4	4	**1**	5	26	26	26
6	0	3	0	20	26	28	26
2	1	1.5	0.5	20	28	27	27
6	2	4	0.33	3	27	28	27
9	0	4.5	0	6	27	28	27
5	2	3.5	0.4	1	29	30	29
5	0	2.5	0	5	30	29	29
5	1	3	0.2	10	30	29	29
6	2	4	0.33	5	29	34	29
3	3	3	**1**	5	31	29	29
10	0	5	0	6	37	29	29
7	0	3.5	0	5	30	32	30

m_{eff} and η refer to the effective mutation rate and mutation ratio (derived parameters). The convergence is expressed in number of calls (in thousand-of-calls units) for males, and females subpopulations and for the whole population [N_m, N_f, min (N_m, N_f), respectively] at which the global minimum was fully refined. The classical η (symmetrical mutation, $\eta = 1$). and r (no learning, $r = 0$) values are distinguished by the bold font. N_m and N_f values lower than their counterparts are underlined. The table is ordered with increasing min (N_m, N_f) value.

An overview of this table gives an idea of the influence of algorithm parameters on convergence. In particular one can note that with high learning rate equal 10 or 20, the exploitative role of the female subpopulation is reflected in the fact that just in this

subpopulation the global minimum is reached earlier than in the male subpopulation: six times for females and two times for males (cf. a related observation noticed in Section 4.2 that, during the run, even for smaller r values f_f tends to be lower than f_m). However, as three parameters are involved (two mutation rates and the learning rate; the mutations can be also described in terms of effective mutation rate and mutation ratio), the influence of these parameters on convergence requires a more detailed discussion (for a more thorough analysis of data of Fig. 3 see the following subsections). It is noteworthy, that the m_{eff} values are around 1–2% in the starting rows of Table 3, but further analysis will lead to conclusion that these cases are isolated and the that optimum m_{eff} value is higher.

4.4. Convergence probability and speed when varying the effective mutation rate and learning rate

Analysis of the distribution of N_{conv} values permits to understand the influence of effective mutation rate on the convergence. Fig. 4 shows that the maximum convergence probability, P, in 2×10^4 calls is achieved when m_{eff} is about 1.8% (short arrow). However, P is quite low (0.21) then, but it systematically shifts through $P = 0.32$ (for $m_{eff} = 2.5\%$ at 3×10^4 calls) and $P = 0.59$ (for $m_{eff} = 3.5\%$ at 6×10^4 calls) to, when permitting for 10^5 calls, the optimum m_{eff} becoming as high as 3.7% with the probability achieving a high level of $P = 0.8$ (long arrow). Extrapolating these results, one can argue that for the given pattern and using the described version of algorithm, at the level of at least 5–6% effective mutation the convergence with $P = 1$ is expected to occur for the stopping criterion markedly exceeding 10^5 calls.

In Table 4, the convergence probability is quoted as a function of m_{eff} and learning rate. With increasing m_{eff}, the chance for success rapidly grows between its values 1.5 and 3 to attain the higher values in the range 3–4 and decrease for those exceeding 4. The decrease observed for the highest m_{eff} values is softer than the increase for low ones. This behaviour is qualitatively the same for all learning-rate values studied. However, the fields with $P = 1$ are rare in the r range from 0–2, and are most frequent for $r = 5\%$. Judging after Table 4, the values of $m_{eff} = 3.5\%$ and

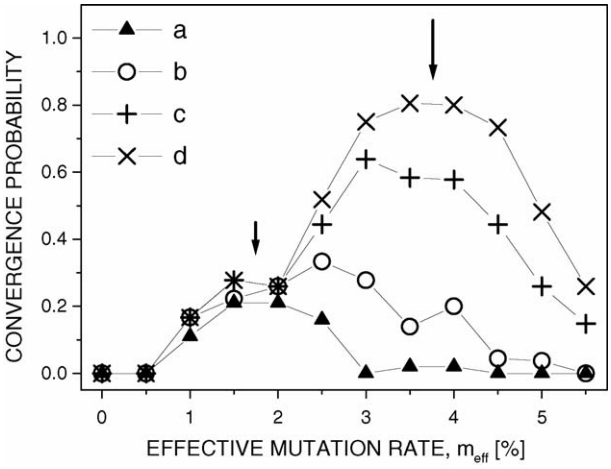

Fig. 4. Probability of successful convergence before 2×10^4 (a); 3×10^4 (b); 6×10^4 (c); and 10^5 (d) function calls, averaged for the points exhibiting the given mutation rate. The distribution was calculated for all fields of all of nine maps, i.e., they show an average over the learning rates 0–20. The short and long arrows indicate the optimum m_{eff} values, 1.8 and 3.7, found for the condition of convergence before 2×10^4 and 10^5 calls, respectively.

$r = 5\%$ give optimal conditions for the algorithm convergence for the applied stop condition. As concerns the optimal m_{eff} values, their 3–4.5% range leads to quite good convergence, but, as indicated above, this interval would shift to higher values with a less severe stop condition.

4.5. Convergence probability when varying the mutation asymmetry

Fig. 5 illustrates the influence of mutation ratio, η, on the convergence. To construct this figure, the nine diagrams discussed in Section 4.3 were divided into seven subdomains according to distinct η values: 0 (the strongest asymmetry), from 0.125 to 0.14, 0.17 to 0.20, 0.29 to 0.40, 0.50 to 0.60, and 1. For construction of the figure, additional constraint for m_{eff} was imposed, $3 \leq m_{eff} \leq 4.5$, according to the results presented in Table 4: the convergence outside this range is typically slow or very slow. Fig. 5 shows that the highest probability of conver-

Table 4
Probability, P, of reaching the global minimum for the LCO data in 10^5 function calls, shown for all studied effective mutation rates, m_{eff}, and learning rates, r (m_{eff} and r are expressed in percent units)

m_{eff}	$P(r=0)$	$P(r=1)$	$P(r=2)$	$P(r=3)$	$P(r=4)$	$P(r=5)$	$P(r=6)$	$P(r=10)$	$P(r=20)$
0	0	0	0	0	0	0	0	0	0
0.5	0	0	0	0	0	0	0	0	0
1	0	0.5	0	0	0	0.5	0	0	0.5
1.5	0	0	0.5	0	0	0	0	1	1
2	0.33	0	0.33	0.33	0.33	1	0	0.33	0
2.5	0.33	0.33	0	1	0.67	1	0.67	0.33	0.33
3	0.75	1	0.75	1	0.25	1	1	0.5	1
3.5	1	0.5	0.5	0.75	1	1	1	1	1
4	0.8	0.8	0.8	0.8	1	0.8	0.8	0.6	0.8
4.5	0.6	0.6	0.6	0.6	1	0.8	0.8	0.8	0.8
5	0.17	0.17	0.5	0.33	0.67	0.5	0.67	0.67	0.67
5.5	0	0.5	0	0.33	0.5	0.33	0.17	0.17	0.33

Shaded are the fields where the probability exceeds 0.5: $P = 1$ (the darkest), $P = 0.75$ or 0.8 (medium) and $P = 0.6$ or 0.67 (the lightest).

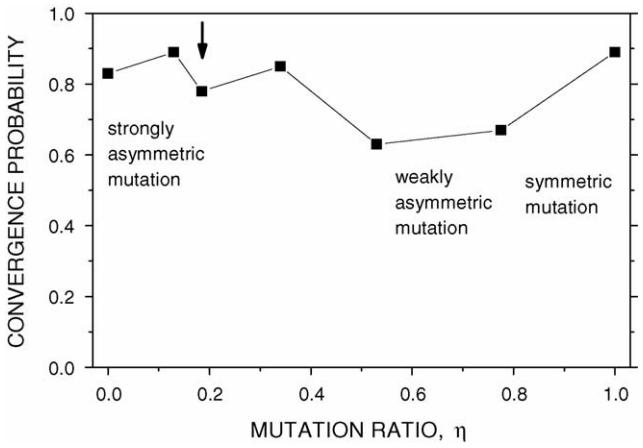

Fig. 5. Probability of successful convergence at the stop condition of 10^5 function calls, plotted against the mutation ratio value, η. Summed data for nine maps of Fig. 3 with m_{eff} belonging to the interval [3.0, 4.5] are classified into six domains distinguished by the mutation ratio, 0 (the strongest asymmetry), from 0.125 to 0.14, 0.17 to 0.20, 0.29 to 0.40, 0.50 to 0.60, and 1 (symmetric mutations). The points plotted represent the η and N_{conv} averaged over the given domain. The middle of the area of higher convergence probability for low mutation ratio is indicated by an arrow.

gence (in 10^5 calls) is found for small η values (around 0.2) indicated by the arrow as well as for η equal unity. The case of $\eta = 1$ is additionally illustrated in Fig. 6: the probability distribution is quite sharp while for the strongly asymmetric case ($m_{\mathrm{f}} = 0$) it is apparently more diffuse. The difference between the two curves suggests that the optimal mutation rates are different for symmetric (experimental points $m_{\mathrm{f}} = m_{\mathrm{m}} = 3$ or 4, evaluated range 2.5–4.5) and fully asymmetric cases ($m_{\mathrm{f}} = 0$ and $m_{\mathrm{m}} = 5$–12). The fact that the distribution for asymmetric case is broader than for the symmetric one demonstrates that a GA with

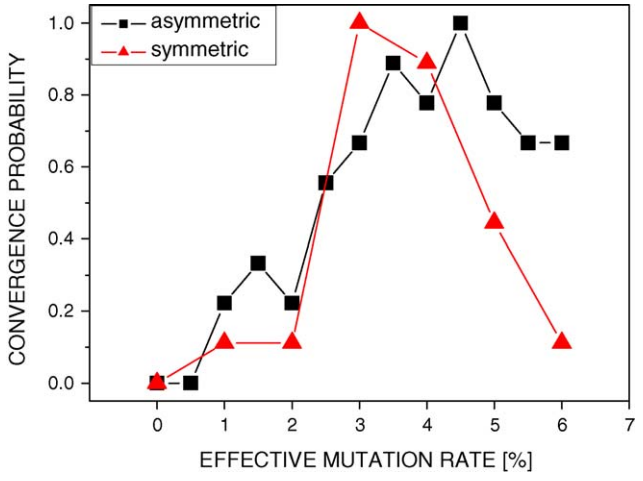

Fig. 6. Comparison of convergence probability, for symmetric mutation and the most asymmetric ($m_{\mathrm{f}} = 0$) mutation. The probability (based on nine maps of Fig. 3) is plotted against the effective mutation rate, η. The stop condition applied is 10^5 function calls. The probability exceeding 0.5 is observed for m_{eff} belonging to the interval [∼2.5, ∼4.5] in the symmetric case (triangles) while the corresponding interval in the asymmetric case is much broader, [2.5, ≥6] (m_{f} from 5 to ≥12, squares).

asymmetric mutation is less sensitive to the choice of mutation-rate-parameter values.

4.6. Tests for NBS patterns

With the aim of verifying the influence of the operator parameters on the convergence of the algorithm, further tests were performed with 16 high-quality NBS patterns [122–126] for orthorhombic substances having unit-cell size up to 15 Å, and the space studied was 2.5 Å < a < 10 Å, 3.5 Å < b < 15 Å, 4 Å < c < 15 Å. In one case ($Al_{18}B_4O_{33}$) the largest lattice parameter is beyond these limits (15.01 Å) but it did not prevent finding the solution (the output lattice parameter was 14.999 Å). As it is concluded that the stopping criterion imposed in ±4.2–±4.5 (low value of 10^5 calls) results in underestimation of the optimal m_{eff} value, the tests in this subsection were performed using the stopping criterion of 10^6 calls.

Trial calculations for NBS patterns using the algorithm described in Section 3 have shown that the time used for finding the global optimum increases rapidly with growing range of the optimised variables and with rising unit-cell volume, mainly due to tendency for trapping the population by one of local minima. Assuming the above defined a, b, c ranges, the true solution was found for all seven cells of volume less than ∼350 Å, whereas the same stopping criterion (10^6 calls) was not big enough for five among nine cells of larger volume (see Table 5, column MCZ5). This observation gave a motivation for further work on the method. Modifications of the algorithm (and of its parameters) can improve the efficiency. For example, additional problem-specific constraints can be imposed which reduce the space. Namely, the largest of the orthorhombic a, b, c variables cannot be smaller than the interplanar spacing of the first reflection. Moreover, analysis of the peak density across the pattern permits to estimate the lower and upper bounds between which the unit-cell volume should be contained. However, maintaining the population diversity at a sufficiently high level during the whole run is the most important condition enabling finding the true solution. Various ways can be considered to improve the diversity, such as using higher m_{m} and m_{f} values, or reducing the stagnation time at each LM through the use of DPF approach. Below, calculations are presented which prove that the convergence can be improved without the (already checked in ref. [6]) DPF approach.

4.6.1. Calculation details

The population diversity was modified through reduction of the elitism and rising the average mutation rate (but maintaining the principal results from the LCO tests). The following modifications were done in the algorithm described in Section 3:

- The mutation rates were cyclically varied, resulting in alternating ‘more explorative’ and ‘more exploitative’ stages of calculation (variable mutation has been earlier an element of the calculations in a GA based study of reconfigurable logic arrays [127]). Such variation can be either regular with fixed period(s) or random. Here, the simplest possible periodical

Table 5
Results of calculations for sixteen NBS patterns

	compound	space group	unit cell volume [Å^3]	MCZ5	LAN0	LAN2	LAN5	LAN10	LVS5	LFS5	LBN5	LAZ5	EAN5	Ref.
m_{av} (m_{avs})				6 (0)	13 (7.5)	13 (7.5)	13 (7.5)	13 (7.5)	13 (7)	13 (7)	13 (7.5)	13 (0)	13 (7.5)	
1	Fe_3C	*Pnma*	155	330	198	49	18	776	7	1000 $	31	1000 #	19	[126] p.72
2	Cr_3C_2	*Pnam*	180	3	15	5	20	12	14	980	379	75	15	[126] p.60
3	Cu_3AsS_4	$Pnm2_1$	293	152	40	262	67	1000 &	131	433	1000	1000	10	[126] p.68
4	Fe_2SiO_4	*Pmnb*	308	246	1000	1000	563	2	129	1000	612	187	375	[125] p.59
5	$CaMgSiO_4$	*Pmnb*	340	22	68	30	418	7	11	371	96	70	138	[125] p.30
6	$Fe(NbO_3)_2$	*Pcan*	343	597	40	921	41	228	259	1000	1000	1000	18	[125] p.56
7	$BaTi_4O_9$	*Pnma*	347	12	5	31	28	24	7	89	427	4	10	[125] p.13
8	$CoTi_2O_5$	*Bbmm*	366	1000	1000	1000	611	1000	1000	1000	1000	1000	1000	[126] p.66
9	$Al_{18}B_4O_{33}$	*A****	653	204	349	961	51	374	42	92	1000	1000	289	[123] p.5
10	$Cd(BrO_3)_2 \times 2H_2O$	$P2_12_12_1$	714	1000	1000	1000	2	1000	945	1000	411	1000	3	[123] p.14
11	thallium hydrogen phtallate $C_8H_5O_4Tl$	$Pca2_1$	857	8	10	4	76	17	38	68	106	28	3	[122] p.75
12	$KTiOPO_4$	$P2_1nb$	869	1000	1000	1000	419	1000	1000	1000	1000	1000	4	[126] p.110
13	MgB_4O_7	*Pbca*	939	568	273	118	61	407	340	114	1000	118	57	[123] p.47
14	$C_4H_6O_6U \times 2H_2O$	$Pna2_1$	978	12	527	634	26	1000	1	314	997	5	7	[124] p.76
15	$NH_4B_5O_8 \times 4H_2O$	*Bba2*	1155	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	[123] p.7
16	$Na_2MoO_4 \times {}^{\times}2H_2O$	*Pcab*	1241	1000	1000	1000	1000	1000	1000	1000	1000	1000	162	[125] p.87
N_{av}				447.13 (163)	470.31 (171)	563.44 (205)	275.06 (100)	552.94 (201)	370.25 (135)	653.81 (237)	691.19 (251)	592.94 (216)	194.38 (71)	
n_1				5	6	5	10	5	7	3	2	5	10	
n_2				11	10	10	14	9	12	8	8	7	14	

The numbers refer to number of calls of the objective function, N_{conv} (in thousand units). "1000" means that the solution is not found during 10^6 calls. Meaning of symbols in the first column: m_{av} = time-averaged mutation rate; m_{avs} (in parentheses) = time-averaged selective mutation rate; numbers 1–16, number of compound; N_{av}, average number of calls for sixteen compounds (1000 is taken if the convergence was not successful) with normalised values added in parentheses; n_1, number of cases with convergence before 10^5 calls; n_2, the same but for 10^6 calls.

Deep derivative solution found: $, *a*, 2*b*, *c*; #, *a*, *b*, 3*c*; &, *a*/2, *b*, *c*.

Calculation code *xyzn* is built from four components: *x*, cycle code: L, five generations with full female elitism and with elitism for male subpopulation limited to the cycle duration, i.e., a less elitist option; E, as above but with extra constraints; M, infinite cycle duration, full elitism for both subpopulations, i.e., a more elitist option); *y*, mutation code: A, variable asymmetric mutation, level (m_m, m_f) = (10, 2) during four generations and (m_m, m_f) = (60, 20) during the fifth one; B, (m_m, m_f) = (30, 2) and (30, 20); V, variable symmetric mutation (m_m, m_f) = (6, 6) and (40, 40); C = constant asymmetric mutation (m_m, m_f) = (10, 2); F, constant symmetric mutation (m_m, m_f) = (13, 13); *z*, selective, mutation code: N, asymmetric for males and females (10, 5); S, symmetric (7, 7); Z = not applied (0, 0); *n*, self-learning code (the number corresponds to the *r* value).

Additional constraints applied in EAN5 set, (V_{min} [Å^3], V_{max} [Å^3], c_{min} [Å]): Fe_3C: (87, 1065, 3); Cr_3C_2: (107, 1347, 4.5); Cu_3AsS_4: (77, 993, 6); Fe_2SiO_4: (154, 1933, 5); $CaMgSiO_4$: (161, 2014, 4.5); $Fe(NbO_3)_2$: (166, 1355, 7); $BaTi_4O_9$: (232, 2392, 7); $CoTi_2O_5$: (126, 1585, 4.5); $Al_{18}B_4O_{33}$: (175, 2233, 7); $Cd(BrO_3)_2 \times 2H_2O$: (561, 6770, 6); thallium hydrogen phtallate $C_8H_5O_4Tl$: (453, 5763, 12.5); $KTiOPO_4$: (377, 4646, 6); MgB_4O_7: (295, 3682, 6.5); $C_4H_6O_6U \times 2H_2O$: (392, 4906, 7.5); $NH_4B_5O_8 \times 4H_2O$: (310, 3827, 5.5);and $Na_2MoO_4 \times 2H_2O$: (381, 4731, 6.5).

changes were adopted: the cycle length was after several preliminary tests fixed at five generations (this value is 10 times smaller than that used in the cited reference), the $m_m = 10$ and $m_f = 2$ values were used with a ratio giving $m_{eff} = 6\%$ (the value concluded from LCO tests), being raised in each fifth (explorative) generation to $m_m = 60\%$ and $m_f = 20\%$, respectively (i.e., m_{eff} and η were 40% and 0.33 in the last generation of the cycle); for comparison, symmetric cases were also studied. The mutation rate averaged over the five generations of the cycle was thus 13%.

- In order to reduce the selection pressure, the elitism of the whole population was weakened, namely it was left for the females but for the male population a short-term elitism was adopted (it was in force during the cycle, only).
- A 'selective-mutation' operator (classified as a self-adaptive one) was added, consisting in multiplication or division of *a*, *b* or *c* by a factor such as 2 or 3, in order to facilitate switching between derivative local minima (it is a problem-specific operator). The selective mutation rate is defined as the percentage of individuals (randomly) submitted to these operations.
- The constraints mentioned at the beginning of this subsection were imposed on the largest lattice parameter and on the unit-cell volume. The volume range was estimated according to an earlier elaborated method [103] using the updated program described in ref. [128]. The evaluated range was based on assumptions of Bravais lattice of any kind (primitive or centred) and of the coverage factor higher than 0.5.

Table 6
Comparison of calculation results performed with different parameter sets

Set 1	Set 2	Feature	Observations
LAN5	MCZ5	Comparison with algorithm used for basic (LCO) tests	Convergence for optimised set (:LAN5) is much better for larger cells than for the basic set (MCZ5)
LAN5	LAN0, LAN2, LAN10	Influence of learning rate	Convergence for optimised set (:LAN5) is considerably faster than for sets with lower and higher learning rates
LAN5	LAZ5	Influence of selective mutation operator	Convergence for optimised set (:LAN5) is much better for larger cells than convergence of the algorithm without selective mutation
LAN5	LBN5, LVS5	Comparison with different mutation asymmetry	Convergence with other (than for LAN5) mutation asymmetries is considerably slower
LVS5	LFS5	Influence of temporal variation of mutation rate	If the mutation is symmetric, mutation rate variable in time gives a better convergence
LAN5	EAN5	Influence of additional constraints	Joint physical constraints on c and V result in fastening of calculation

Set 1, optimised or partially optimised and set 2, not optimised.

4.6.2. Results

Among the parameter sets studied and presented in Table 5, two sets, LAN5 and EAN5, exhibit the best results: for both, the convergence in 10^5 calls is achieved for 10 patterns, whereas in 10^6 calls for 14 patterns, i.e., the convergence is much better than for other parameter sets tried. The average number of calls necessary to find solution (average value for 16 patterns), N_{av}, is 275×10^3 for LAN5. For other parameter combinations except EAN5 it is considerably higher.

For some patterns the success (convergence before 10^6 calls) does not depend on parameter choice, for some it is strongly dependent. This is primarily influenced by the OFN-landscape complexity which is higher for cells with large volume. The above-described algorithm modification results in a better convergence for larger cells as documented by comparison of LAN5 set in Table 5 with the MCZ5 set obtained for unmodified algorithm.

The observations concluded on the basis of Table 5 are collected in Table 6. In particular, among the calculation schemes with learning rates $r = 0$–10 (LAN0, LAN2, LAN5, LAN10), the best one is LAN5, providing the global solution for all patterns except two characterised by the largest cell volume. Exploiting the property, that some deep LMs for orthorhombic system may have interdependent positions, results in considerable fastening of the calculation when adding the selective mutation operator (LAN5 versus LAZ5). The calculations confirm that proper asymmetrising the mutation (leaving the same m_{eff} value) leads to a considerable improvement of the convergence (LAN5 versus LBN5 and LVS5). Introduction of mutation variability also results in a change in convergence. This is shown for the case of symmetric mutation: if the mutation rate is variable in time (set LVS5), the convergence is apparently faster, by 45% in terms of N_{av}, than for constant mutation rate (LFS5 set). The physical constraints imposed on c and on V reduce the calculation time (the speed is by ∼30% larger, see the EAN5 set). In particular, for a large cell of 16th example, where none of parameter sets led to solution within 10^6 calls, the true solution is found when the constraints are applied.

5. Discussion

The tests performed in this study permitted for determining the optimum values of mutation and learning parameters. The calculations were proceeded for a number of datasets, using stopping criterion between 2×10^4 and 10^6 calls.

High mutation rate ascertains a genetic diversity of the population during the search, while low mutation rate conserves the property of the genetic algorithms of memorising (in the bits of the whole population) the long-term information about the features of many earlier populations. The calculations for LCO show (Figs. 2 and 4, Table 4) that for the studied computational task the optimal m_{eff} value is about 4% for 10^5 calls criterion and is concluded to be slightly higher (5–6%) if the stop criterion would be less restrictive. In the following NBS tests, the 6% value was successfully applied during the main part of each cycle.

The calculations carried out for LCO data strongly indicate that suitable mutation asymmetry ($\eta = 0.2$) enhances the explorative capabilities of the algorithm, what is manifested by acceleration of the global search. The resulting increase of convergence probability is not very large (0.8 for strongly asymmetric and ideally symmetric instead of 0.6 for weakly asymmetric mutations, as indicated in Fig. 5) supported by results of NBS tests of Table 5 where using the asymmetry caused a ∼25% acceleration of the calculation. The asymmetry seems to be worth further investigation the more so as (i) the reason of improvement is understandable according to the nature-based arguments presented on male/female diversities and mutation rates, e.g., in refs. [107–109] and (ii) the recent attempts of introducing the mutation asymmetry into genetic-algorithms appeared successful [112–114]. In the latter studies,

a combination of conservative females with innovative males proved useful as it resulted in easier escaping from local optima during the global search. Although the mutation in this study is introduced in a different way than in the cited approach (fixed common mutation rate versus tuned individual mutation rate, respectively), in both cases the mutation asymmetrisation is concluded to be a valuable genetic-algorithm modification. Using other words, the male subpopulation with a higher mutation rate ascribed can be interpreted as explorative one, whereas the female (exploitative) subpopulation, characterised by a low mutation rate, better conserves the genetic information from previous generations and exploits the LMs already discovered. This differentiation leads to the observed effects. The best η value quantitatively describing the mutation ratio is of the order of 0.2 as concluded from the LCO tests and supported by NBS tests (this value coincides with that existing in nature for a number of real-world species, cf. refs. [108,109]). The presence of two tendencies (exploration and exploitation) in the male/female subpopulations, respectively, creates favourable conditions for finding and refining the minima. A related, observed in nature, phenomenon [129–131] is usually called "male driven evolution".

The calculations of LCO tests show that self-learning at the level of about 5% (expressed in percentage of learning individuals) permits for improving the optimisation results, owing to, in particular, the simple construction of the learning operator (quadratic interpolation) requiring a small computational effort. The NBS tests confirm this conclusion, as from among the models with SL rate 0–10%, that with $r = 5\%$ is the most effective. This result qualitatively correlates with the finding, saying that 5% is the optimal percentage of individuals submitted to Lamarckian search (mentioned in ref. [88], p. 200). Further studies are needed to determine how general the findings of the present work are and how the values or variations of the learning length parameter would influence the calculation efficiency (intuitively, the l value should be at least several times smaller than the width of the valley where the solution is located). In the model adopted here, the self-learned information is inherited in a Lamarckian way by next generations. This way does not directly reflect the common understanding of evolution (it is noteworthy that some opinions exist that the reality is not so simple, namely that in nature there exists some way of inheritance of learned information, see ref. [132]).

Variable mutation rate is shown to lead to improvement of the convergence. The simple scheme applied involves periods composed of four generations of relatively low mutation rate (exploitative stage) and followed by one generation of much higher mutation rate (explorative stage) making that the average mutation rate is approximately doubled. Periodical changes of mutation rate apparently facilitate, in a different way than the asymmetrisation does, escaping from local minima.

6. Conclusions

The present study describes properties of operators of a genetic algorithm, which is a largely modified version of algorithm earlier applied for indexing powder-diffraction patterns. The modifications consist in employing the binary coding instead of real one and in revision of operator structure and adding new operators. The tests performed in this work involve investigation of convergence when varying such parameters as mutation rate and asymmetry, and self-learning rate.

Classical genetic algorithms, when employed for global-optimisation tasks, may suffer from specific drawbacks such as long calculation time, tendency for premature convergence, and slow convergence in a vicinity of the solution. Difficulties may be connected with the choice of optimal values of the algorithm parameters. These problems are more or less obstructive depending on the specific task and on its properties such as the space size, the landscape of the fitness function, computing time of the fitness function value.

The proposed GA extension involving a learning operator and mutation asymmetry, both inspired by observation of nature, is found to improve the calculation efficiency of the studied LCO example. In tests for sixteen NBS patterns some other modifications (variable mutation rate, selective mutation operator) causing a further convergence improvement were considered.

According to the present calculations, at least for some kinds of global-optimisation tasks, the choice of the mutation rate must be careful as it is of high importance for the efficiency of GA work. Therefore, possible ways of taking it into account are careful testing of the mutation values for the given task before the calculations, adjusting the mutation rate during the calculation, and considering the mutation asymmetrisation. Moreover, the tests show, that introduction of periodic changes of the mutation rate during the run provides a supplementary tool against trapping of the population at local minima.

The obtained results give evidence that the quadratic self-learning operator introduced according to the employed definition is useful and can provide a valuable completion of the global search procedures based on the genetic principles. Its use means a minor deterministic (but acting on randomly selected individuals) contribution to the (stochastic) genetic search. It is sufficient to apply it to a small fraction of the population (such as 5%) to get an improvement in the convergence.

In summary, the examples discussed in this study demonstrate that completion of a classical GA by periodically varying asymmetric mutation and by learning operator, improves the optimisation convergence towards the global minimum. The described genetic-algorithm modifications are believed to be worth combining with earlier studied dynamic-penalty-function approach preventing long stagnation of the algorithm at local minima. Joint effect of various modifications of genetic algorithms is expected to lead to improvement of the calculation efficiency in various global-optimisation tasks, including those of indexing for low-symmetry unit cells.

Acknowledgements

The author is indebted to Dr. Marek W. Gutowski and Dr. Jerzy B. Pełka (both from Institute of Physics PAS) for

critical remarks. Two anonymous referees are acknowledged for reports being helpful in creation of the final version of the manuscript. This work was supported in part within State Committee for Scientific Research (Republic of Poland) Grant No. 72/E-67/SPB/5.PR UE/DZ 27/2003–2005.

References

[1] J.H. Holland, Adaptation in Natural and Artificial Systems, The University of Michigan Press, Ann Arbor, MI, USA, 1975.
[2] D.E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, Reading, MA, 1989.
[3] K.A. De Jong, An analysis of the behavior of a class of genetic adaptive systems, Ph.D. thesis, University of Michigan, USA, 1975.
[4] J.H. Holland, Genetic Algorithms, Academic Press, New York, 1992.
[5] J. Joines, C. Houck, in: Proceedings of the First IEEE International Conference on Evolutionary Computation, IEEE Press, Orlando, 1994, pp. 579–584.
[6] W. Paszkowicz, in: Proceedings of the Fourth European Powder Diffraction Conference, Chester, UK, 10–14 July 1995, Mater. Sci. Forum 228–231, 1996, pp. 19–24.
[7] S. Kazarlis, V. Petridis, in: Proceedings of the Fifth International Conference on Parallel Problem Solving from Nature, Berlin, Germany 1998, Lecture Notes Comp. Sci., vol. 1498, Springer Verlag, 1998, pp. 211–220.
[8] J. Grefenstette, United States Patent Application 20050038762, February 17, 2005.
[9] S. Mahfoud, in: Proceedings of the Sixth International Conference on Genetic Algorithms (ICCGA), Pittsburgh, PA, USA, July 15–19, L.J. Eshelman (Ed.), Morgan Kaufmann Publishers Inc., San Francisco, CA, 1995, pp. 136–143.
[10] S. Forrest, Science 261 (5123) (1993) 872–878.
[11] J.T. Alander, Indexed bibliography of genetic algorithms in chemistry and physics. Report 94-1-CHEMPHYS, University of Vaasa, Department of Information Technology and Production Economics, Vaasa, 1995.
[12] B.K. Lavine, A.J. Moores, Anal. Lett. 32 (1999) 433–445.
[13] G.N. Bullock, M.J. Denham, I.C. Parmee, J.G. Wade, Des. Stud. 16 (4) (1995) 507–524.
[14] M. Gen, R. Cheng, Genetic Algorithms and Engineering Design, John Wiley & Sons, New York, NY, 1997.
[15] S.K. Pal, P.P. Wang (Eds.), Genetic Algorithms for Pattern Recognition, CRC Press, Boca Raton, FL, 1996.
[16] M. Jamshidi, L. dos Santos Coelho, R.A. Krohling, P.J. Fleming (Eds.), Robust Control Systems with Genetic Algorithms, CRC Press, Boca Raton, 2003.
[17] T. Brodmeier, E. Pretsch, J. Comput. Chem. 15 (1994) 588–595.
[18] V.E. Bazterra, M.B. Ferraro, J.C. Pacelli, J. Chem. Phys. 116 (2002) 5984–5991.
[19] D.M. Deaven, K.O. Ho, Phys. Rev. Lett. 75 (1995) 288–291.
[20] W.J. Pullan, Comput. Phys. Commun. 107 (1997) 137–148.
[21] F.R. Manby, R.L. Johnston, C. Roberts, Commun. Math. Comput. Chem. 38 (1998) 111–122.
[22] K. Michaelian, Chem. Phys. Lett. 293 (1998) 202–208.
[23] C. Roberts, R.L. Johnston, N.T. Wilson, Theor. Chem. Acc. 104 (2000) 123–130.
[24] M. Iwamatsu, Int. J. Mod. Phys. C 13 (2002) 279–295.
[25] L.D. Lloyd, R.L. Johnston, C. Roberts, T.V. Mortimer-Jones, Chem. Phys. Chem. 3 (2002) 408–415.
[26] V.E. Bazterra, O. Ona, M.C. Caputo, M.B. Ferraro, P. Fuentealba, J.C. Facelli, Phys. Rev. A 69 (5) (2004) 053202-1-7.
[27] B. Hartke, Struct. Bond. 110 (2004) 33–53.
[28] K.Y. Tam, R.G. Compton, J. Appl. Crystallogr. 28 (1995) 640–645.
[29] D.E. Clark, D.R. Westhead, J. Comput. Aided Mol. Des. 10 (1996) 337–358.
[30] D.J. Maddalena, G.M. Snowdon, Expert Opin. Ther. Patents 7 (3) (1997) 247–254.
[31] S. Kamphausen, N. Höltge, F. Wirsching, C. Morys-Wortmann, D. Riester, R. Goetz, M. Thürk, A. Schwienhorst, J. Comput. Aided Mol. Des. 16 (2002) 551–567.
[32] D.E. Clark (Ed.), Evolutionary Algorithms in Molecular Design (Methods and Principles in Medicinal Chemistry, vol. 8), R. Mannhold, H. Kubinyi, H. Timmerman (Series Eds.), Wiley-VCH, Weinhein, 2000.
[33] V.J. Gillet, Struct. Bond. 110 (2004) 33–152.
[34] B.M. Kariuki, H. Serrano-Gonzàlez, R.L. Johnston, K.D.M. Harris, Chem. Phys. Lett. 280 (1997) 189–195.
[35] B.M. Kariuki, S.A. Belmonte, M.I. Mc Mahon, R.L. Johnston, K.D.M. Harris, R.J. Nelmes, J. Synchrotron Radiat. 6 (1999) 87–92.
[36] J.A. Hageman, Optimal Optimisation in Chemometrics, Ph.D. thesis, Katholieke Universiteit Nijmegen, 2004.
[37] J.A. Hageman, R. Wehrens, R. De Gelder, L.M.C. Buydens, J. Comput. Chem. 24 (9) (2003) 1043–1051.
[38] W. Paszkowicz, unpublished, 1996.
[39] W. Paszkowicz, M. Leszczyński, in: Proceedings of the First Polish Meeting on High-Resolution X-ray Diffractometry and Topography (Szklarska Poręba, Poland, 14–17 IX 1996), High Pressure Research Centre, Warsaw, 1996 (in Polish).
[40] M. Wormington, C. Panaccione, K.M. Matney, D.K. Bowen, Philos. Trans. R. Soc. Lond. A 357 (1999) 2827–2848.
[41] E. Landree, C. Collazo-Davila, L.D. Marks, Acta Crystallogr. B 53 (1997) 916–922.
[42] R.T. Fu, K. Esfarjani, Y. Hashi, J. Wu, X. Sun, Y. Kawazoe, Science reports of the Research Institutes Tohoku University Series A, Phys. Chem. Metall. 44 (1) (1997) 77–81.
[43] T. Dandekar, J. Mol. Model. 3 (1997) 312–314.
[44] P. Saxena, I. Whang, Y. Voziyanov, C. Harkey, P. Argos, M. Jayaram, T. Dandekar, Biochim. Biophys. Acta 1340 (1997) 187–204.
[45] R. Unger, Struct. Bond. 110 (2004) 153–175.
[46] S.M. Woodley, P.D. Battle, J.D. Gale, C.R.A. Catlow, Phys. Chem. Chem. Phys. 1 (1999) 2535–2542.
[47] S.M. Woodley, Struct. Bond. 110 (2004) 95–132.
[48] K. Shankland, W.I.F. David, T. Csoka, Z. Kristallogr. 212 (8) (1997) 550–552.
[49] R. Csoka, W.I.F. David, K. Shankland, Mater. Sci. Forum 278–281 (1998) 294–299.
[50] K.D.M. Harris, R.L. Johnston, B.M. Kariuki, Acta Crystallogr. A 54 (1998) 632–645.
[51] O.J. Lanning, S. Habershon, K.D.M. Harris, R.L. Johnston, B.M. Kariuki, E. Tedesco, G.W. Turner, Chem. Phys. Lett. 317 (2000) 296–303.
[52] G.W. Turner, E. Tedesco, K.D.M. Harris, R.L. Johnston, B.M. Kariuki, Chem. Phys. Lett. 321 (2000) 183–190.
[53] S. Habershon, K.D.M. Harris, R.L. Johnston, G.W. Turner, J.M. Johnston, Chem. Phys. Lett. 353 (2002) 185–194.
[54] S. Habershon, K.D.M. Harris, R.L. Johnston, J. Comput. Chem. 24 (2003) 1766–1774.
[55] K.D.M. Harris, R.L. Johnston, S. Habershon, Struct. Bond. 110 (2004) 55–59.
[56] B.M. Kariuki, K. Psallidas, K.D.M. Harris, R.L. Johnston, R.W. Lancaster, S.E. Staniforth, S.M. Cooper, Chem. Commun. 17 (1999) 1677–1678.
[57] P. Chacon, J.F. Diaz, F. Moran, J.M. Andreu, J. Mol. Biol. 299 (2000) 1289–1302.
[58] D.J. Mikulin, D.A. Coley, J.R. Sambles, Liq. Cryst. 22 (1997) 301–307.
[59] J.W. Choi, K.S. Cho, W.H. Lee, H.S. Lee, Thin Solid Films 327 (1998) 273–277.
[60] A. Ulyanenkov, K. Omote, J. Harada, Physica B 283 (2000) 237–241.
[61] A.D. Dane, A. Veldhuis, D.K.G. de Boer, A.J.G. Leenaers, L.M.C. Buydens, Physica B 253 (1998) 254–268.
[62] R. Wehrens, E. Pretsch, L.M.C. Buydens, Anal. Chim. Acta 388 (1999) 265–271.
[63] M.C. Vargas, J. Torres-Jiménez, J.C. Ruiz-Suárez, J. Phys. A 35 (2002) 3865–3876.

[64] T. Weber, H.B. Bürgi, Acta Crystallogr. A 58 (2002) 526–540.
[65] P. Sałek, J. Tarasiuk, K. Wierzbanowski, A. Baczmański, Mater. Sci. Forum 273-275 (1998) 139–143.
[66] J. Tarasiuk, K. Wierzbanowski, B. Bacroix, Comput. Mater. Sci. 29 (2) (2004) 179–186.
[67] C. Gaspin, T. Schiex, in: Proceedings of Third European Conference on Artificial Evolution, AE'97, Nimes, France, 22–24 October 1997, Lecture Notes Comp. Sci. 1363 (1998) 145–155.
[68] Z. Meng, W.T. Taferner, Q. Yang, K.G. Eying, Y.-H. Pao, in: Y.H. Pao, S.R. LeClair (Eds.), Artificial Intelligence in Real-Time Control, Elsevier, Kidlington, UK, 1998, pp. 155–166.
[69] J. Blazewicz, M. Kasprzak, W.W. Kuroczycki, J. Heuristics 8 (2002) 495–502.
[70] K.C. Wiese, E. Glen, Biosystems 72 (1–2) (2003) 29–41.
[71] G.A. Cox, T.V. Mortimer-Jones, R.P. Taylor, R.L. Johnston, Theor. Chem. Acc. 112 (3) (2004) 163–178.
[72] C.H. Ooi, P. Tan, Bioinformatics 19 (2003) 37–43.
[73] G. Chang, M. Lewis, Acta Crystallogr. D53 (1997) 279–289.
[74] J.W. Visser, J. Appl. Crystallogr. 2 (1969) 89–95.
[75] A. Altomare, C. Giacovazzo, A. Guagliardi, A.G.G. Moliterni, R. Rizzi, P.-E. Werner, J. Appl. Crystallogr. 33 (2000) 1180–1186.
[76] A. Boultif, D. Louër, J. Appl. Crystallogr. 37 (2004) 724–731.
[77] M.A. Neumann, J. Appl. Crystallogr. 36 (2003) 356–365.
[78] J. Bergmann, A. LeBail, R. Shirley, V. Zlokazov, Z. Kristallogr. 219 (12) (2004) 783–790.
[79] W. Paszkowicz, J. Appl. Crystallogr. 20 (1987) 166–172.
[80] S.A. Belmonte, B.M. Kariuki, M.I. McMahon, R.L. Johnston, K.D.M. Harris, R.L. Nelmes, Commiss. Powder Diffr. Newslett. 21 (1999) 4–5.
[81] R. Cheng, M. Gen, Y. Tsujimura, Comput. Ind. Eng. 36 (1999) 343–364.
[82] D. Whitley, Inf. Software Technol. 43 (2001) 817–831.
[83] A. Tsakonas, G. Dounias, in: Proceedings of the Second Hellenic Conference on Methods and Applications of Artificial Intelligence, Thessaloniki, Greece, April 11–12, 2002, Lecture Notes Comp. Sci. 2308 (2002) 494–511.
[84] E. Jablonka, M.J. Lamb, E. Avital, Trends Ecol. Evol. 13 (1998) 206–210.
[85] A. Jaszkiewicz, Eur. J. Operat. Res. 137 (2002) 50–71.
[86] R. Chelouah, P. Siarry, Eur. J. Operat. Res. 148 (2) (2003) 335–348.
[87] S.A. Kazarlis, S.E. Papadakis, J.B. Theocharis, V. Petridis, IEEE Trans. Evol. Comput. 5 (2001) 204–217.
[88] J. Arabas, Wyklady z algorytmow ewolucyjnych (Lectures on Evolutionary Algorithms, in Polish), Wydawnictwa Naukowo-Techniczne, Warsaw, 2001.
[89] M. Lozano, F. Herrera, N. Krasnogor, D. Molina, Evol. Comput. 12 (3) (2004) 273–302.
[90] B. Freisleben, P. Merz, in: H.-M. Voigt, W. Ebeling, I. Rechenberg, H.-P. Schwefel (Eds.), in: Proceedings of the Fourth International Conference on Parallel Problem Solving from Nature, Springer, Berlin 1996, pp. 890–899.
[91] B. Dengiz, F. Altiparmak, A.E. Smith, IEEE Trans. Evol. Comput. 1 (3) (1997) 179–188.
[92] H. Kopfer, D.C. Mattfeld, in: Proceedings of the First International Conference on Operations and Quantitative Management, vol. 2, Jaipur, India, January 5–8, 1997. University of Baltimore, Baltimore, MD, 1997, pp. 498–505.
[93] R. Dorne, J.-K. Hao, in: A.E. Eiben, T. Back, M. Schoenauer, H.-P. Schwefel (Eds.), Proceedings of the Fifth International Conference on Parallel Problem Solving from Nature, PPSN V, Lectures Notes Comp. Sci., vol. 1498, Springer, Berlin, 1998, pp. 745–754.
[94] F. Vavak, K.A. Jukes, T.C. Fogarty, in: Proceedings of the Third Annual Conference on Genetic Programming, J.R. Koza, W. Banzhaf, K. Chellapilla, K. Deb, M. Dorigo, D.B. Fogel, M.H. Garzon, D.E. Goldberg, H. Iba, R.L. Riolo (Eds.), Morgan Kaufmann Publishers, San Francisco, CA, 1998, pp. 602–608.
[95] L.D. Whitley, A.E. Howe, S. Rana, J.P. Watson, L. Barbulescu, in: Proceedings of the SMC'98 IEEE International Conference on Systems, Man, and Cybernetics, vol. 3, IEEE, New York, NY, 1998, pp. 2466–2471.
[96] E.K. Burke, A.J. Smith, IEEE Trans. Power Sys. 15 (1) (2000) 122–128.
[97] A. Jaszkiewicz, M. Hapke, P. Kominek, in: Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization, EMO, E. Zitler, K. Deb, L. Thiele, C.A. Coello Coello, D. Corne (Eds.), Lecture Notes Comp. Sci., Springer, Berlin, 2001, vol. 1993, pp. 241–255.
[98] F.B. Pereira, E. Costa, in: Proceedings of the EvoWorkshop 2001. Lecture Notes Comp. Sci. 2037 (2001) 421–430.
[99] T.H. Guo, C.-D. Mu, in: Proceedings of the Fourth World Congress on Intelligent Control and Automation, vol. 3, IEEE, Piscataway, NJ, 2002, pp. 1840–1843.
[100] H.R. Mashhadi, H.M. Shanechi, C. Lucas, IEEE Trans. Power Sys. 18 (3) (2003) 1181–1186.
[101] C.G. Koh, Y.F. Chen, C.Y. Liaw, Comput. Struct. 81 (2) (2003) 107–117.
[102] H. Ishibuchi, T. Yoshida, T. Murata, IEEE Trans. Evol. Comput. 7 (2003) 204–223.
[103] W. Paszkowicz, J. Appl. Crystallogr. 20 (3) (1987) 161–165, 529.
[104] M.W. Gutowski, J. Phys. A 27 (1994) 7893–7904.
[105] B. Örçal, E. Tüzel, V. Sevim, N. Jan, A. Erzan, Int. J. Mod. Phys. C 511 (2000) 973–986.
[106] S.R. Bonissone, R. Subbu, in: Proceedings of the SPIE Annual Meeting, vol. 4787, July 9–10, Seattle, WA, 2002, pp. 9–22.
[107] V.A. Geodakian, Nauka i Zhizn' (Science and Life), 1965 (in Russian) (English translation at Complexity Digest, http://www.comdig.com), arXiv:CS.NE/0408006 v1, 2004.
[108] H. Ellegren, Nature Genet. 31 (1) (2002) 9–10.
[109] H. Ellegren, A.K. Fridolfsson, J. Mol. Evol. 56 (4) (2003) 458–463.
[110] J.F. Crow, Trends Genet. 16 (12) (2000) 525–526.
[111] M. Kirkpatrick, D.W. Hall, Evolution 58 (2) (2004) 437–440.
[112] K. Omori, Y. Fujiwara, S. Maekawa, H. Sawai, S. Kitamura, in: N. Baba, L.C. Jain, R.J. Howlett (Eds.), Knowledge-Based Intelligent Information Engineering Systems and Allied Technologies, vol. 1, IOS Press, Amsterdam, 2001, pp. 122–126.
[113] K. Omori, Y. Fujiwara, S. Maekawa, H. Sawai, S. Kitamura, Transactions of the Institute of Systems, Control Inf. Eng. 15 (8) (2002) 422–429.
[114] K. Omori, S. Maekawa, H. Tamaki, S. Kitamura, Electr. Eng. Jpn. 150 (1) (2005) 42–49.
[115] J. Sánchez-Velazco, J.A. Bullinaria, in: J.M. Rossiter, T.P. Martin (Eds.), Proceedings of the UK Workshop on Computational Intelligence, Bristol, UK, University of Bristol UKCI-2003, 2003, pp. 217–223.
[116] K.S. Goh, A. Lim, B. Rodrigues, Artif. Intell. Rev. 19 (2) (2003) 123–152.
[117] C. García-Martínez, M. Lozano, D. Molina, A.M. Sánchez, Technical Report, University of Granada, Granada, 2005 (http://decsai.ugr.es/~lozaano/chrom-dif.pdf.).
[118] K.G. Khoo, P.N. Suganthan, Patt. Recognit. Lett. 23 (13) (2002) 1589–1597.
[119] D.J. Cavicchio, Adaptive Search Using Simulated Evolution, Ph.D. thesis, University of Michigan, Ann Arbor, MI, 1970.
[120] R. Biesbroek, http://www.estec.esa.nl/outreach/gatutor/crossover.htm.
[121] W.D. Potter, R.W. Robinson, J.A. Miller, K.J. Kochut, D.Z. Redys, in: Proceedings of the 7th International Conference on Industrial & Engineering Applications of Artificial Intelligence and Expert Systems (IEA/AIE'94),Austin, Texas, June 1994.
[122] M.C. Morris, H.F. McMurdie, E.H. Evans, B. Paretzkin, J.H. deGroot, C.R. Hubbard, S.J. Carmel, NBS Monograph 25, Sec. 16 (1979).
[123] M.C. Morris, H.F. McMurdie, E.H. Evans, B. Paretzkin, C.R. Hubbard, S.J. Carmel, NBS Monograph 25, Sec. 17 (1980).
[124] M.C. Morris, H.F. McMurdie, E.H. Evans, B. Paretzkin, H.S. Parker, N.C. Panagiotopoulos, C.R. Hubbard, NBS Monograph 25, Sec. 18 (1981).

[125] M.C. Morris, H.F. McMurdie, E.H. Evans, B. Paretzkin, H.S. Parker, N.P. Pyrros, C.R. Hubbard, NBS Monograph 25, Sec. 20 (1984).
[126] M.C. Morris, H.F. McMurdie, E.H. Evans, B. Paretzkin, H.S. Parker, W. Wong-Ng, D.M. Gladhill, C.R. Hubbard, NBS Monograph 25, Sec. 21 (1985).
[127] Z.M.B. Saleeba, A Self-Reconfiguring Computer System Ph.D. thesis, Monash University 1998, sec. 5.1.2.
[128] W. Paszkowicz, J. Appl. Crystallogr. 22 (1989) 186–187.
[129] L.D. Hurst, H. Ellegren, Nature 420 (2002) 365–366.
[130] K.D. Makova, W.H. Li, Nature 416 (6881) (2002) 624–626.
[131] A. Bartosch-Harlid, S. Berlin, N.G.C. Smith, A.P. Moller, H. Ellegren, Evolution 57 (10) (2003) 2398–2406.
[132] W.W. Gibbs, Sci. Am. 12 (2003) 78–85.