Application of genetic algorithms to model the structure of molecular crystals*)

Wojciech Łużny[1),)], Wojciech Czarnecki[1)]**

DOI: dx.doi.org/10.14314/polimery.2014.542

Abstract: In the case of complex organic molecules, very often happens that the powder diffraction pattern shows only few crystalline reflections and therefore powerful methods like the Rietveld procedure cannot work properly. Then in order to find the model of the structure, for which the calculated diffraction pattern shows an acceptable agreement with the experimental data, one may use one of the "artificial intelligence" techniques. Among them, the genetic algorithms (GA) method is one of the most widely used in a variety of scientific problems. Our idea was to use the unit cell (with its atoms content) as a phenotype of the GA. Each individual represents a random crystal structure. Its genotype contains: lattice constants: a, b and c, lattice angles: α, β, and γ, and orientation of molecule vs. the crystallographic axes. Its diffraction pattern is calculated and compared with the reference data. We have prepared a computer program named CrystalFinder. In its present shape, the CrystalFinder program is able to reproduce the crystalline structure of simple "toy" molecular crystals built of molecules containing up to ca twenty atoms. Usually 200—300 generations are sufficient to get a very good agreement between the reference diffraction pattern and the diffractogram obtained by the program as the best fitted individual. It follows that genetic algorithms can be used to model the structure of molecular crystals and more complex systems containing macromolecules.

Keywords: genetic algorithms, molecular crystals, computer modeling of crystalline structure.

Zastosowanie algorytmów genetycznych do modelowania struktury kryształów molekularnych

Streszczenie: W przypadku złożonych cząsteczek organicznych bardzo często zdarza się, że dyfraktogram proszkowy ma tylko kilka refleksów krystalicznych i z tego powodu metody modelowania typu Rietvelda nie dają poprawnych wyników. W celu znalezienia modelu struktury, który pozwoli na obliczenie dyfraktogramu wykazującego dobrą zgodność z danymi doświadczalnymi, można zastosować jedną z technik tzw. sztucznej inteligencji. Metoda algorytmów genetycznych (GA) — szeroko stosowana do rozwiązywania rozmaitych problemów naukowych — jest jedną z takich technik. Nasz pomysł polegał na użyciu komórki elementarnej wraz z zawartymi w niej atomami jako osobnika (fenotypu) występującego w GA. Każdy taki osobnik reprezentuje przypadkową strukturę krystaliczną. Jego chromosom (genotyp) zawiera: stałe sieci a, b i c, kąty sieciowe α, β i γ, oraz orientację cząsteczki względem osi krystalograficznych. Dla każdego osobnika obliczany jest jego dyfraktogram, który następnie zostaje porównany z danymi referencyjnymi. Idea ta wykorzystana została w napisanym przez nas programie CrystalFinder. W swojej obecnej wersji, program CrystalFinder jest w stanie poprawnie odtworzyć złożoną strukturę krystalograficzną prostego kryształu molekularnego, zbudowanego z molekuł zawierających do ok. dwudziestu atomów. Zwykle przeliczenie 200—300 pokoleń wystarcza do otrzymania dobrej zgodności pomiędzy dyfraktogramem referencyjnym, a dyfraktogramem uzyskanym dla najlepiej dostosowanego osobnika w populacji. Z przeprowadzonych obliczeń wynika, że algorytmy genetyczne mogą być zastosowane do modelowania struktury kryształów molekularnych i bardziej złożonych układów, zawierających również makrocząsteczki.

Słowa kluczowe: algorytmy genetyczne, kryształy molekularne, komputerowe modelowanie struktury krystalicznej.

[1)] AGH University of Science and Technology, Faculty of Physics and Applied Computer Science, Al. Mickiewicza 30, 30-059 Krakow, Poland.

*) Material contained in this article was presented at the IX International Conference "X-ray investigations of polymer structure", 3—6 December 2013, Zakopane, Poland.

**) Author for correspondence; e-mail: Wojciech.Luzny@fis.agh. edu.pl

It is well known that there are several methods of finding crystalline structure models from the diffraction powder data (like the Rietveld procedure based on a full profile fitting, or so called „direct methods" based on the Patterson function). However, in the case of complex organic molecules, very often happens that one has only few crystalline reflections and these very powerful methods cannot work properly. Then in order to find the model of the structure, for which the calculated diffraction pattern shows an acceptable agreement with the experimental data, there are two possibilities: to use just „trial-and-error" techniques, or — better — to use one of the „artificial intelligence" techniques. Among them, the genetic algorithms (GA) method is one of the most widely used in a variety of scientific problems [1, 2].

GA is the method of optimization inspired by nature and it makes use of such terms as inheritance, mutation, selection and crossover. A population of candidate solutions (called individuals or phenotypes) to an optimization problem is evolved toward better solutions. Each individual represents the solution of a problem, and each candidate solution has a set of properties (its chromosomes or genotype) which can be mutated and altered. The evolution starts from a population of randomly generated individuals. In each generation, the fitness of every individual in the population is evaluated. The more fit individuals are stochastically selected from the current population, and each individual's genome is modified (recombined and randomly mutated) to form a new generation. The algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population. The genetic algorithm requires a fitness function to evaluate the solution domain.

Of course, the idea to apply evolutionary algorithms to study a variety of structural problems may be found already in the recent literature. For example, the program USPEX must be pointed out [3—5]. This is a very powerful and sophisticated computer tool, working effectively only on quick workstations. Very recently, two other papers in this field have been published and may be shown as the examples of application of GA based techniques to structure determination [6, 7]. Another attempt is reported in very interesting work [8]. However, in all the applications of structure solutions discussed in that paper, it is assumed that the unit cell parameters are already known from prior analysis of the experimental powder diffraction pattern. Therefore we have undertaken the new attempt in order to prepare a computer tool as universal as possible, and simultaneously being able to perform computations on PC computers in reasonable time.

Our idea is to use the unit cell (with its atoms content, of course) as a phenotype of the genetic algorithm. In general, we plan to do our work in three steps:

— a system of small molecules;

— a system of macromolecules;

— more complex systems, containing macromolecules and small molecules.

In this work we are able to show some results obtained at the first step.

Each individual represents a random crystal structure. The diffraction pattern for each individual is calculated and compared with the reference data. Basic assumptions of our simulations may be summarized as follows:

— full compatibility with PowderCell program is required [9];

— program must keep the shape of a molecule with the change of size and shape of a unit cell;

— program should allow full 3D visualization of structures;

— program should allow the rotation of a molecule around the unit cell axes.

As the input data one has to take the reference diffraction pattern and the file with information on atoms forming the molecule and its structure. As the fitness function we use one of Rietveld factors (it gives the number describing how the diffractogram calculated for a given individual is similar to the reference one).

COMPUTER PROGRAM DETAILS

Main characteristics

We have prepared a computer program named CrystalFinder. The most important steps of its work are listed below:

— initialization (input data, size of population, mutation probability, crossover probability, selection method, succession method);

— starting population: N individuals representing unit cells (with random parameters of their chromosomes);

— calculation of diffractogram for each individual;

— calculation of fitness for each individual;

— the main loop of GA consisting in selection of pairs of individuals for applying the genetic operators i.e., crossover, mutations and rotations of a molecule (the „child" population replaces the „parent" population);

— finalization of calculations.

Input data

The algorithm needs to operate on two types of input data. First, it needs a diffraction pattern, which will be used in the evaluation of solutions generated by the algorithm (so called reference data). On this basis the fitness of individuals will be determined. The program takes the diffraction patterns in the form of files, in which the diffraction data are stored in two columns. The first column is the value of the scattering angle; the second column contains the intensity of radiation. The second file contains information on the atoms making up the composition of the cell. Its file format is identical with the file format of PowderCell files (*.cel).

Unit cell as an individual of the genetic algorithm

Each individual represents a random crystal structure. Its genotype contains: lattice constants: a, b and c, lattice angles: α, β, and γ, and three angles describing the orientation of molecule. All these parameters will be subject to the action of the genetic operators. The individual contains also atomic base, which gives complete information on atoms making up the molecule. On the basis of the unit cell and its contents, the intensity and the positions of crystalline reflections are calculated. The data obtained in this way, forming a set of single peaks, are then approached with the use of profiling functions: Gaussian, Lorentzian or pseudo-Voigt selected by the user.

Fitness function

To evaluate individuals, two Rietveld fit factors (R_F or R_{wp}) may be used alternatively to compare the input diffraction pattern and diffraction pattern generated by the genetic algorithm. To determine the fitness of a given member of the population from its R value, it is reasonable to calculate the following scaled ρ factor:

$$\rho = \frac{R - R_{\min}}{R_{\max} - R_{\min}} \quad (1)$$

where: R_{min}, R_{max} — the smallest and the largest values of the coefficient R in the population, respectively.

The value of ρ lies in the range $0 \le \rho \le 1$. Finally, the fitness of each individual is described by the function $F(\rho)$ which takes its highest value (equal to 1) when $\rho = 0$, and takes its lowest value when $\rho = 1$. A simple example of such function, possible to select by a user of CrystalFinder, is just $F(\rho) = \exp(-\rho)$. The values of R_{min} and R_{max} are continually updated as the population evolves, so the fitting function $F(\rho)$ is dynamically scaled during the GA calculations.

Coding methods

The genotype of each individual consists of one chromosome containing six parameters (a, b, c, α, β, and γ) describing the unit cell. The program CrystalFinder uses two types of coding. In binary coding, the sum of the numbers of bits, by which each of the parameters is encoded, is the total length of the chromosome. The other approach involves using arrays of real numbers instead of bit strings to represent chromosomes. A chromosome is a vector consisting of the real values of the actual parameters of the unit cell.

The genotype also contains the atomic base, or positions of all the atoms included in the unit cell. Positions of the atoms are not included in the chromosome because — according to the assumptions — the base is fixed, whereas the chromosome is subjected to genetic operators. Only base rotations are possible, which are discussed below. On the basis of information contained in an individual's genotype, its phenotype or diffraction pattern is calculated, which is subjected to „environmental assessment".

Used genetic operators

CrystalFinder uses standard genetic operators. Operators of crossover for binary encoding include: single point, multipoint, and uniform crossover. For real numbers coding the averaging operator is used. For binary encoding the mutation turns bits in the chromosome to the opposite value, whereas for real numbers coding it adds random values to the elements of the chromosome.

Another operator, designed specifically for CrystalFinder is a rotation of base, meaning rotation of the molecule inside the cell around crystallographic axes. The base is rotated around the center of mass of the molecule by a random angle, relative to a random line. Rotation of the base runs on the individual, who was not involved in the crossover operation and has not been subjected to mutation operator.

The general scheme of the algorithm

The algorithm used in CrystalFinder is not much different from the standard genetic algorithm scheme. After entering all necessary input data (reference diffractogram and base) and after setting all necessary configuration parameters of the program (see below), the algorithm takes its first step: the initiation of the population. N individuals representing unit cells are created. Each individual chromosome is initialized with random values — binary or real ones — depending on the encoding type. On the basis of unit cell parameters and the base, the diffractogram of each individual is calculated. The next step is to evaluate the population. For each individual, based on its diffraction pattern and the reference diffractogram, his adaptation is calculated using the fitness function.

The population prepared in this way is ready to use in the main loop of genetic algorithm. Individuals are selected from the base population using the selected method of reproduction and grouped in pairs. Then genetic operators (crossover, mutation and rotation of base) act with a selected probability and create a temporary population. With a base population and a temporary one, succession process creates the descendant population. It becomes a new base population, on which the algorithm operates in the next loop.

The algorithm terminates when it reaches the selected stop condition: either the set value of the fitting function is reached, or after a certain number of generations is prepared. It is possible also that the algorithm stops automatically, if the value of the fitting function for the best individual does not change during the set number of generations (this means that the algorithm is stuck in a local extreme of the fitness function).

How to use CrystalFinder

When the program starts, the main window, shown in Fig. 1, appears. In the middle part of the window three-dimensional visualization of unit cells found by the algorithm is displayed. At the bottom of the main window the field is located that displays text messages about the current state of the program. The right panel is used to guide the genetic algorithm calculations. It allows to load the input diffractogram file, to load the file containing the information on a molecule, to initialize the population, and to start (also to pause or to stop) the algorithm. The top menu bar consists of the following components:

— *Options* (related to the configuration of CrystalFinder — see below),

— *Genetic algorithm* (it allows to visualize the population and to see how the fitness changes in subsequent generations),

— *Diffractogram* (allows visualizing the input diffractogram).

Selecting from the menu bar option *Tools* and next *Visualize cell file* allows visualization of any *.cel file (see the sample in Fig. 2).

Clicking the right mouse button leads to displaying the context menu, shown as the inset in Fig. 1. It allows the user to use all functions related to displaying the resulting structures, diagrams and files.

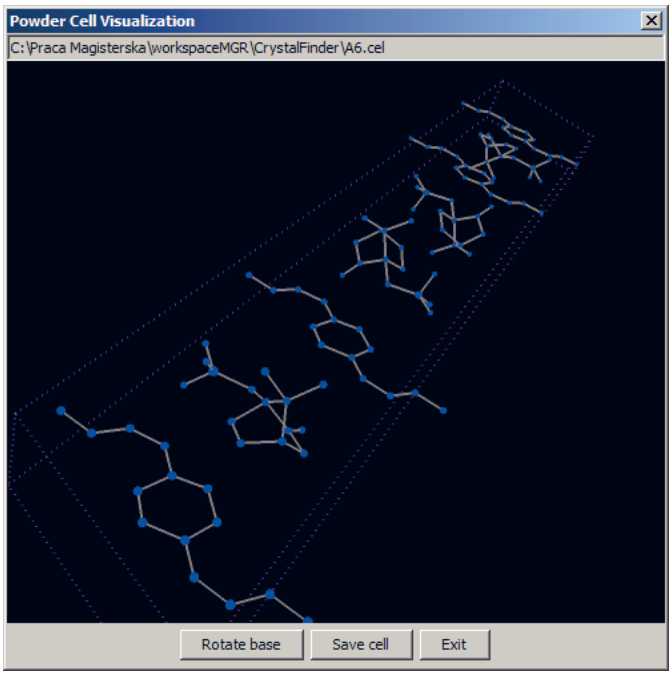

Fig. 2. Sample visualization of the *.cel file

Determining the configuration

It is well known that a genetic algorithm does not guarantee finding the optimal solution. The choice of algorithm parameters is the most important step in using

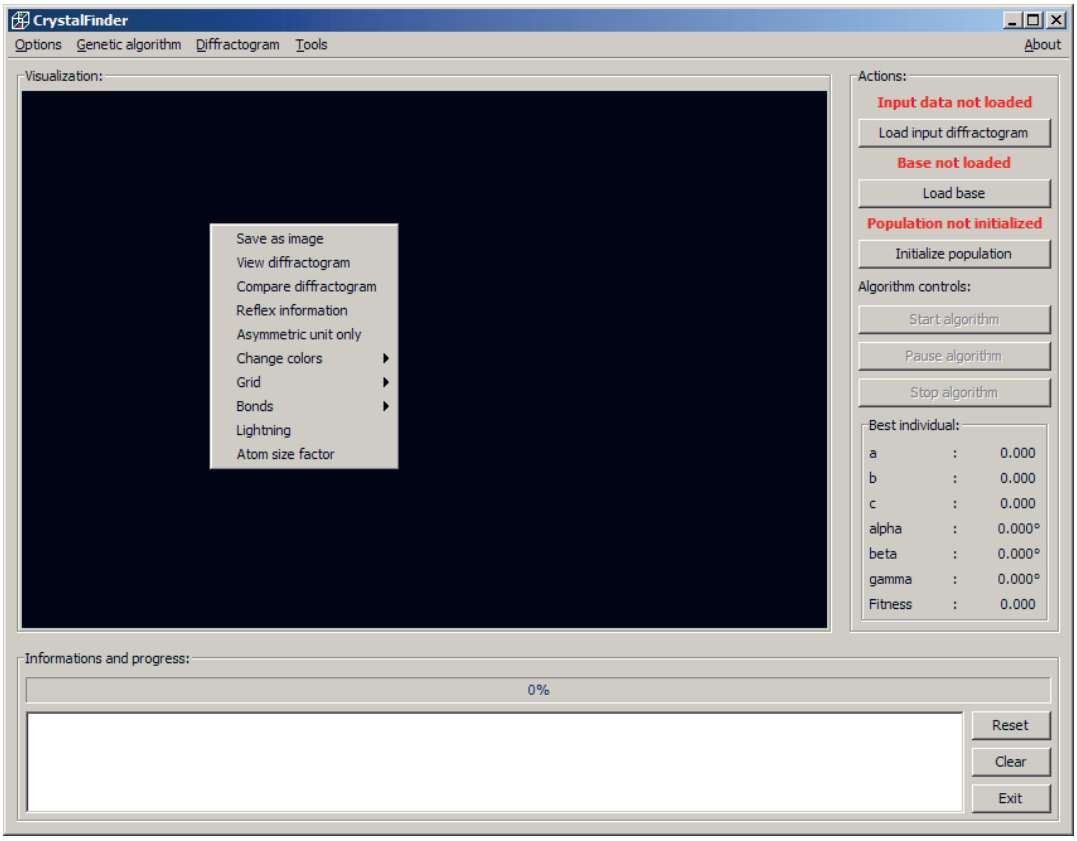

Fig. 1. The main window of the CrystalFinder program and inset showing a context menu of the program

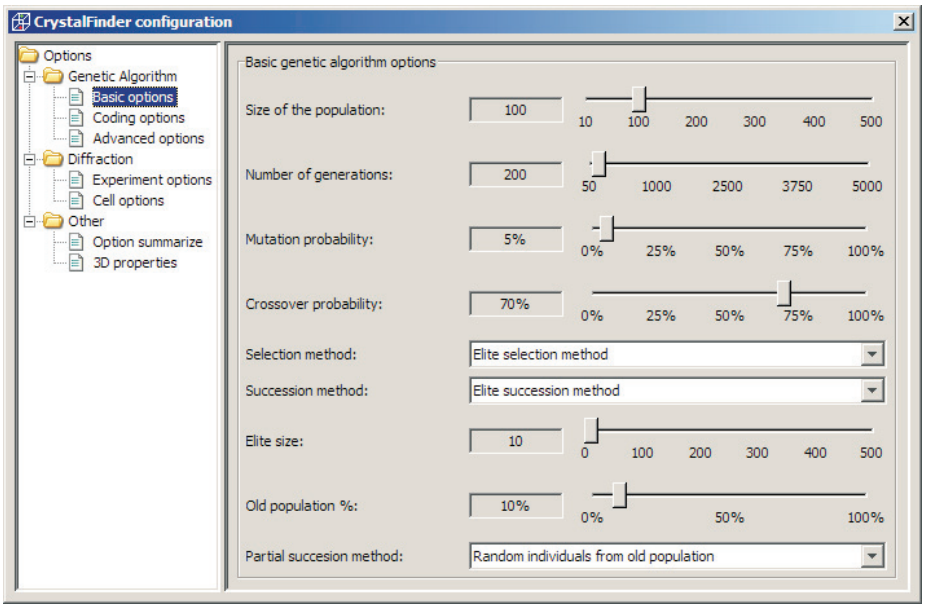

Fig. 3. The panel for configuring the basic genetic algorithm options

genetic algorithms. The skill and speed of finding solutions by the algorithm depends on this choice. Therefore we put the large emphasis on the possibilities of configuring CrystalFinder. It allows modification of every parameter with a significant impact on the operation of the genetic algorithm.

The user of CrystalFinder has to select the parameters of configuration by using the system of windows, opening on demand from the panel of *Options* and next *Configuration* (see Fig. 3). The following tools are available to the user:

— *Basic options* i.e., size of the population, number of generations, mutation probability, crossover probability, selection method (one may choose between: elite, roulette or tournament ones), succession method (the choice is: elite, partial or full succession, respectively);

— *Coding options* i.e., binary or real coding, binary crossover type, real coding mutation type, real mutations percentage;

— *Advanced options* i.e., fitness function [$F(\rho)$], Rietveld parameter (R_F or R_{wp}), base rotation probability, maximum base rotations angle, method of the algorithm's stopping;

— *Experiment options* i.e, diffractogram profiling function (Gaussian, Lorentzian or pseudo-Voight), ranges of the Miller indices, diffraction peaks' width parameter, X-ray wavelength;

— *Cell options* i.e., the ranges for lattice constants and cell angles, the accuracy of their encoding.

EXAMPLE RESULTS OF THE PROGRAM CRYSTALFINDER

To test CrystalFinder, diffraction data generated by PowderCell were used. The real experimental data were not necessary for tests, because finding a crystal structure based on generated diffraction data is sufficient as a proof of correctness of the genetic algorithm used by our program.

Here we show the sample results obtained for the artificial crystal of camphorsulfonic acid molecule (CSA) built of 15 atoms (all hydrogen atoms have been omitted for simplicity of calculations). The triclinic unit cell contains only one CSA molecule. The structure of this molecule is real, but the parameters of the unit cell used in the reference structure are selected arbitrarily.

If the configuration of CrystalFinder is determined properly, the program is able to reproduce the crystalline structure of such "toy" molecular crystal quite well. Usually 200—300 generations are sufficient to get a very good agreement between the diffraction pattern calculated for the reference structure (treated as the "experimental data") and the diffractogram obtained by the program as the best fitted individual. Example comparison of the calculated diffraction pattern with the reference data is shown in Fig. 4. The typical number N of individuals in the population is in the range between 100 and 300. The reference unit cell parameters and their analogs found by CrystalFinder are compared in Table 1.

T a b l e 1. Comparison of the reference unit cell parameters and their analogs found by CrystalFinder

Unit cell parameters	a, Å	b, Å	c, Å	α, °	β, °	γ, °
Input	9.00	3.50	6.00	90.00	70.00	80.00
Generated	9.014	3.507	6.015	90.06	70.05	80.04

The typical dependence of the fitness function versus the number of generations is presented in Fig. 5.

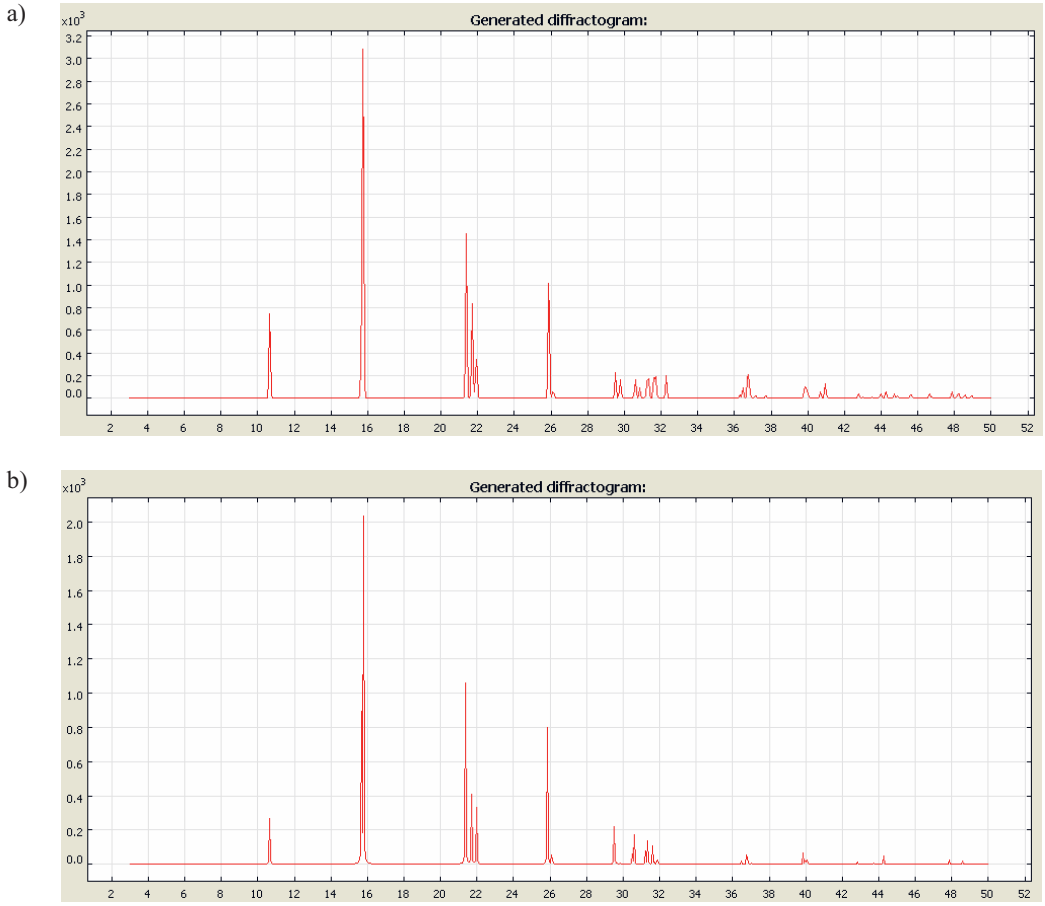

Fig. 4. Comparison of the reference diffraction pattern (a) with diffraction pattern found by CrystalFinder as the best fit individual (b)

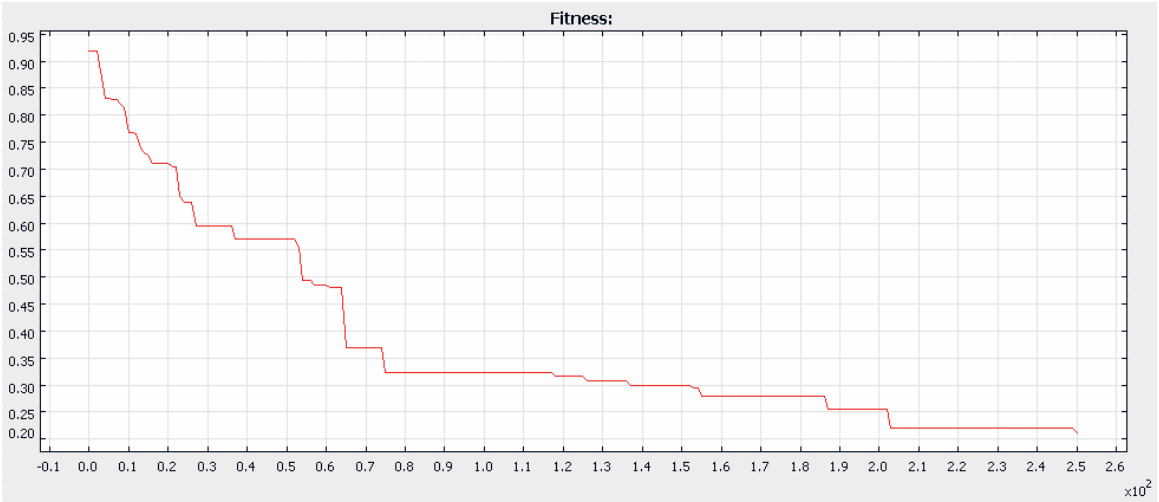

Fig. 5. Typical dependence of the fitness function versus the number of generations (result of the CrystalFinder program)

DISCUSSION AND CONCLUSIONS

Of course, not all combinations of configuration parameters give good results. During testing, where attempts were made to configure the program to give the best results we found out that the elite methods, both for the selection and succession, did not apply in solving the problem set for this work. Too soon there occured a problem of a small population differentiation, which led to stopping the algorithm in the local minimum. The best results were obtained when we used the tournament method of selection and succession with partial replacement. Also important was the choice of probability of mutations and crossover, which affect the ability of the algorithm to search the space of solutions of the problem.

It turned out that the biggest problem was the fitness function. For unit cell described by six parameters, func-

tion *F* is in fact the function of nine variables (a, b, c, α, β, γ, and three angles describing orientation of a molecule around the crystallographic axes). If too wide ranges of variability of these parameters were set, the space in which the algorithm was looking for solutions dramatically raised and the program very rarely has been able to find a solution close to the ideal. Suitable restriction ranges in such a way that the algorithm has to search a smaller solution space had a significant impact on the probability of finding a solution.

The use of the Java language to write the program was associated with CrystalFinder requirements for a visualization of unit cells, which thanks to the use of the Java3D library was extremely simple to implement. However, the Java language is not the ideal solution for numerical computations. The computational complexity of the program is increasing rapidly with the number of atoms making up the composition of the unit cell, and with the larger ranges of the (*h*, *k*, *l*) indices. Therefore we plan to prepare the next versions of CrystalFinder (being able to work with macromolecules) in another language, like C++. It will allow us to think about making use of parallel computing.

The results obtained in this study showed that with the right choice of variability ranges of the unit cell parameters, the genetic algorithm used in CrystalFinder was able to find the correct solution, which differs only slightly from the ideal solution. It follows that the genetic algorithms can be used to model the structure of the molecular crystals and — hopefully — more complex systems containing macromolecules.

This work was supported by the Polish Ministry of Science and Higher Education and its grants for Scientific Research.

REFERENCES

[1] Michalewicz Z.: „Genetic Algorithms + Data Structures = Evolution Programs", Springer-Verlag, 1996.

[2] „Applications of Evolutionary Computation in Chemistry. Structure and Bonding" (Ed. Johnston R.L.), Springer-Verlag, vol. 110, 2004.

[3] Oganov A.R., Glass C.W.: *J. Chem. Phys.* **2006**, *124*, 24 470, http://dx.doi.org/10.1063/1.2210932

[4] Glass C.W., Oganov A.R., Hanses N.: *Comp. Phys. Comm.* **2006**, *175*, 713, http://dx.doi.org/10.1016/j.cpc.2006.07.020

[5] Lyakhov A.O., Oganov A.R., Stokes H.T., Zhu Q.: *Comp. Phys. Comm.* **2013**, *184*, 1172, http://dx.doi.org/10.1016/j.cpc.2012.12.009

[6] Altomare A., Cuocci C., Giacovazzo C., Moliterni A., Rizzi R., Corriero N., Falcicchio A.: *J. Appl. Cryst.* **2013**, *46*, 1231, http://dx.doi.org/10.1107/S0021889813013113

[7] Wu S.Q., Ji M., Wang C.Z., Nguyen M.C., Zhao X., Umemoto K., Wentzcovitch R.M., Ho K.M.: *J. Phys.: Condens. Matter* **2014**, *26*, http://dx.doi.org/10.1088/0953-8984/26/3/035402

[8] Harris K.D.M., Johnston R.L., Habershon S.: „Applications of Evolutionary Computations in Structure Determination from Diffraction Data" in „Applications of Evolutionary Computation in Chemistry, Structure and Bonding" (Ed. Johnston R.L.), Springer-Verlag, **2004**, *110*, 55. http://dx.doi.org/10.1007/b13933

[9] Krauss W., Nolze G.: PowderCell for Windows, v.2.4, Federal Institute for Materials Research and Testing, Berlin 2000.