Eur. Phys. J. B **29**, 619–628 (2002)
DOI: 10.1140/epjb/e2002-00345-0

THE EUROPEAN PHYSICAL JOURNAL B

Genetic Algorithms as a tool in the study of aperiodic order, with application to the case of X-Ray diffraction spectra of GaAs-AlAs multilayer heterostructures

B. Leblanc[1], E. Lutton[1,a], and F. Axel[2]

[1] INRIA - Rocquencourt, BP 105, 78153 Le Chesnay Cedex, France
[2] Laboratoire de Physique des Solides[b], bâtiment 510, Université Paris Sud, 91405 Orsay, France

Received 18 March 2002 / Received in final form 3 July 2002
Published online 31 October 2002 –

Abstract. We present the first application of Genetic Algorithms to the analysis of data from an aperiodically ordered system, high resolution X-Ray diffraction spectra from multilayer heterostructures arranged according to a deterministic or random scheme. This method paves the way to the solution of the "inverse problem", that is the retrieval of the generating disorder from the investigation of the spectra of an unknown sample having non crystallographic, non quasi-crystallographic order.

PACS. 02.50.-r Probability theory, stochastic processes, and statistics – 05.90.+m Other topics in statistical physics, thermodynamics, and nonlinear dynamical systems – 61.10.-i X-ray diffraction and scattering – 61.43.-j Disordered solids

Introduction

Solving a particular problem can amount to the finding of the minimum of a function over a given space. One then considers an optimization problem. When the function has a certain type of regularity, a number of methods exist, most often based on gradient or generalized gradient computations (see for instance [1]). Generalized gradient methods work well when:

- "some sort" of gradient can be defined and computed at any point of the space of solutions (for instance, a directional derivative),
- the function of interest does not have too many local minima, or the value taken by the function at these minima is significantly greater than its value at the absolute minimum.

But for very irregular functions which do not satisfy these requirement, different methods have to be used for optimization. Most of them are based on stochastic schemes.

1. One of the most known stochastic algorithms is *Simulated Annealing*. It is a powerful technique for finding the global minimum of a function when a great number of parameters have to be taken into account. It is based upon an analogy with the annealing of solids, where a material is heated to a high temperature, then very slowly cooled in order to let the system possibly reach its ground state energy. The delicate point is to lower the temperature T not too rapidly, so as to avoid local minima.

   The Metropolis algorithm is then used: at "temperature" T, the jump from a state of energy E to a state of energy E' is made with probability one if E' is lower than E, that is the state of energy E' is "accepted", and with a probability proportional to $\exp((E-E')/T)$ if not [2–4].

   Theoretical results exist that prove the convergence of such a process, but reaching the optimal solution is guaranteed only if the "temperature" parameter is lowered at a logarithmic rate, implying a very large number of iterations in general.

   To solve optimization problems of other systems one uses a transposition to the statistical mechanics situation of simulated annealing along the lines of Table 1 [5].

   The combinatorial problem is therefore formulated as a statistical mechanical problem.
2. Another one is the *Replica Method* which has first been applied to spin glass systems. It had been originally proposed as a trick to simplify the computation of the average value of the free energy density. This is done by introducing n uncoupled replicas of the initial system of size N, then defining the partition function and the free energy of the n replicas as a function of n integer, that are simply related to the partition function and free energy of the initial system. Then extending these

[a] e-mail: Evelyne.Lutton@inria.fr
[b] CNRS UMR 8502

Table 1. Optimization and simulated annealing.

Optimization problem of a given system	Simulated annealing
Domain of the problem	Sample
Definition of configuration	State
Cost function for a configuration	Energy of a state
Optimal configuration	Ground state
Minimal cost	Ground state energy
Control parameter for optimization process	Temperature

to analytic functions of n, one takes the two limits $n \to 0$ and $N \to \infty$ in an adequate fashion, thus obtaining the desired average free energy.

The application to the solution of other optimization problems is straightforward [6,7].

3. In the following, we chose to experiment another optimization technique, *Genetic Algorithms* [8,9] which we selected for its efficiency in dealing with discrete codings, and describe in details below.

The systems under study, as explained below, are multilayers heterostructures composed of planar layers of two kinds (the two letter alphabet) arranged according either to deterministic algorithms, aperiodic substitutional or automatic sequences, or analogous systems where the disorder generating sequence is unknown.

In Section 1, we present the multilayer system under study and introduce the "inverse problem". In Section 2 an application of a Genetic Algorithm to this problem is presented for the investigation of calculated Xray diffraction spectra. Results are presented in Section 3.

1 X-Ray diffraction spectra and the inverse problem

1.1 X-Ray diffraction spectra analysis: the model

In their article [10], Peyrière, Cockayne and Axel present a theoretical and numerical study for the analysis of XRay diffraction spectra of Prouhet-Thue-Morse GaAs-AlAs multilayer heterostructures.

The experimental discovery of quasicrystals [11] in 1984 has opened a new field of research to both experimentalists and theoricians. In this field, the importance of deterministic structures having controlled aperiodic disorder is being increasingly recognised. This is why one-dimensional deterministic sequences generated by substitution or finite automata [12–14] have been widely used mathematical objects to build such structures. In particular, 3D multilayer heterostructures having two kinds of layers arranged according to the Fibonacci sequence were first defined and effectively made as early as 1985 by molecular-beam epitaxy [15,16] (MBE) and consequently investigated by X-ray and neutron diffraction [16–19] Raman scattering [20,21] etc.

Prompted by all of these studies, an extension of such methods to *nonquasiperiodic* systems soon began, with special interest in the Thue-Morse sequence, and its mathematical and physical properties. For instance, in 1987, a Thue-Morse superlattice heterostructure was made for the first time and investigated by Raman scattering [20].

The Prouhet-Thue-Morse sequence $\{\epsilon_n\}$ can be defined in several equivalent ways as follows:

- Let σ be a substitution acting on a two letter alphabet, for example $(0, 1)$:

$$\sigma \begin{cases} 0 \to 01 \\ 1 \to 10. \end{cases} \tag{1}$$

The sequence is then characterized by its initial conditions ϵ_0 and the number n of iterations of σ. Its length is $N = 2^n$.

With $\epsilon_0 = 0$, the sequence is:

$$\begin{array}{l} 0 \\ 01 \\ 0110 \\ 01101001 \\ 0110100110010110\ldots \end{array}$$

- A recursive definition. With ϵ_i the ith element in the sequence, one has:

$$\begin{cases} \epsilon_{2n} = \epsilon_n \\ \epsilon_{2n+1} = 1 - \epsilon_n \end{cases} \quad \text{with } \epsilon_0 \in \{0; 1\}\cdot \tag{2}$$

- A definition using an algorithmic machine known as a 2-automaton (see [10,22]).

Let us notice that for a given length there exist two possible sequences called *mirror* sequences corresponding to the two initial conditions $\epsilon_0 = 0$ and $\epsilon_0 = 1$.

In practice, superlattice heterostructures are grown on a GaAs(001) substrate by molecular-beam epitaxy (MBE). The deposition rate is about 1 Å/sec. The lattice simply consists of AlAs (A) and GaAs (B) layers. The values of d_A and d_B are designed to be $d_A = d_B = 5a_0$, where a_0 is the average constant of the cubic "zinc-blende" lattice of AlAs and GaAs.

Taking advantage of the specific properties of the Prouhet-Thue-Morse sequence and using the atomic structure factors of the GaAs and AlAs layers, and using kinematic diffraction theory, the authors [10] calculate a general formula for the diffraction amplitude $\hat{S}_n(q)$ with q the wave vector.

The intensity of the high resolution X-Ray diffraction spectrum is then:

$$I_n(q) = \left|\hat{S}_n(q)\right|^2 = \hat{S}_n^*(q)\hat{S}_n(q). \quad (3)$$

The authors have thus been able to successfully reproduce experimental high resolution X-Ray diffraction spectra from 2^7 and 2^{10} Prouhet-Thue-Morse multilayer heterostructures originally published in reference [23].

1.2 Generalization of the model

This model can be generalized for multilayer heterostructures having any kind of generating binary sequence

$$(s_N(k))_{k\in[0;N-1]}$$

with N the total length of the sequence.

With the symbolic association and using the notations of [10]:

0: codes GaAs layers, with thickness d_0 and diffraction amplitude $\hat{\mu}_0(q)$ as a function of wave vector q.
1: codes AlAs layers, with thickness d_1 and diffraction amplitude $\hat{\mu}_1(q)$ as a function of wave vector q.

Then the diffraction amplitude reads

$$\hat{S}_N(q) = \mu_{s_N(0)}(q) + \sum_{j=1}^{N-1} \mathrm{e}^{-2\mathrm{i}\pi q\left[\sum_{k=0}^{j-1} d_{s_N(k)}\right]}\mu_{s_N(j)}(q). \quad (4)$$

We note that for the calculation of $\hat{S}_N(q)$ one has to use a summation of $N = 2^n$ terms, whereas the calculation of $\hat{S}_N(q)$ in the Prouhet-Thue-Morse case requires only n factors. The generalization in the present context is at the price of going from $O(n)$ to $O(2^n)$ in computational time [10].

1.3 Nature and interest of the "inverse problem"

The goal is to retrieve from the experimental X-Ray diffraction spectrum of an unknown multilayer heterostructure sample the binary sequence after which the layers are arranged.

Such a problem is well resolved in "classical" crystallography where the possible lattices of crystalline samples have symmetries belonging to one of the 230 crystallographic groups. The analysis of the XRay diffraction spectrum then allows – the chemical composition being known – the complete retrieval of the structure. This is also true for quasicrystals [24,25], but not for materials where long range order is "less regular", such as aperiodic deterministic order generated by a substitutive sequence or for glassy materials where the disorder is in general thought of as being of random origin. The interest of finding or approaching a solution to this problem is obvious.

When dealing with difficult "inverse problems" as the previous one, where no analytical solution is known, a straightforward strategy is to try and deal with this problem as with an optimization problem. The optimization problem here is the minimization of a "distance" between the experimental spectrum and the computed spectrum according to equation (4), with respect to the generating binary sequence $(s_N(k))_{k\in[0;N-1]}$. The function to be minimized in this case is a very irregular and complex function and a well adapted stochastic optimization method must be used[1]. We present in the sequel a solution of this problem based on a Genetic Algorithms.

2 Use of a Genetic Algorithm

2.1 Genetic Algorithms

Genetic Algorithms – or more generally Evolutionary Algorithms – mimic Darwin's evolutionary model of survival-of-the-fittest, in evolving a set of potential solutions rather than a unique point. This is a main advantage over other stochastic schemes when optimizing irregular and difficult functions over large search spaces.

This method is based on two themes: the ability of simple representations (sequences on a two letter alphabet) to encode complicated structures, and the power of simple transformations to improve such structures. It has been shown [26] that with the proper control structure, rapid improvements of bit strings could be made to "evolve" as population of animals do. Recently established theoretical results [27–30] prove that, given appropriate conditions, genetic algorithms tend to converge onto solutions that are globally optimal, *i.e.* the limit distribution of the population when generations tends to infinity is concentrated on the global maximum (or maxima, if there are several) of the fitness functions[2].

In natural evolution, the characteristics of each individual are embodied in the composition of its chromosomes. Operations that alter this chromosomal composition specially occur when parents reproduce; among them are random mutation, *i.e.* a small alteration of its chromosomal material, and crossover, an exchange of chromosomal material between two parents' chromosomes. This feature of natural evolution – the ability of a population of chromosomes to explore its search space and simultaneously combine the best findings through crossover – is exploited during a Genetic Algorithms run.

Of course these notions are sufficiently simplified so they can be used in a computer program. The general structure of a Genetic Algorithm program is described in

[1] The size of the search space (see Sect. 2.2) without considering the irregularity of the function itself, is a sufficient reason to consider stochastic optimization methods.

[2] Other theoretical and experimental analysis proved a "weaker" convergence criterion, *i.e.* the best individual of the limit population is positioned on the global optimum (or on one of the global optima) of the fitness function.

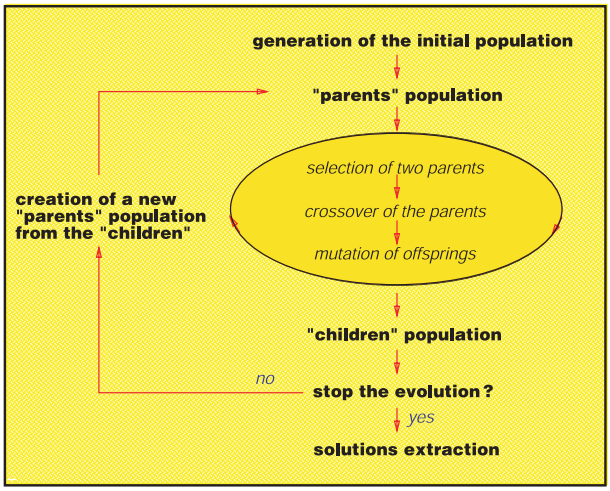

Fig. 1. General scheme of a Genetic Algorithm.

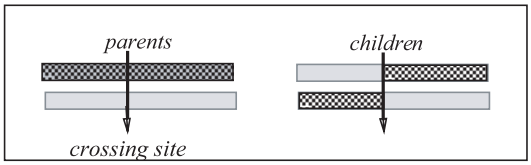

Fig. 2. The crossing over process: parts of the genome are exchanged between parents to form the offspring.

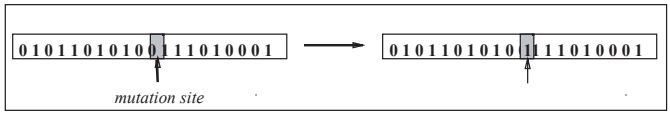

Fig. 3. The mutation process: a small perturbation of the genome. For example a bit is flipped ($0 \rightarrow 1$) on a randomly chosen position on the genome (mutation site).

Figure 1, the selection step is usually performed by a biased random shot, where the probability for an *individual* to be selected is proportional to its "*fitness*", *i.e.* a measure of the quality of the "solution" represented by this *individual* regarding the problem to solve. The "genetic" operators are usually two: crossover and mutation. They are stochastic operators, applied with some probabilities (p_c and p_m respectively). Figures 2 and 3 show two classical implementation of these operators.

In the application presented in this paper, we have made specific choices concerning these points, in order to perform an efficient statistical optimization. Of course, there exist many possible variations of this structure: we implemented a classical scheme, where the population size of each generation is constant, and the initial population is randomly generated on the search space. This application has been programmed with ALGON [31], a general Genetic Algorithm software which was developed at INRIA by two of the authors.

2.2 The standard method

For a given problem to be solved with the help of a Genetic Algorithm, it is necessary to define an efficient coding of its possible solutions (also called *individuals*) which could lend themselves to the action of genetic operators. It is also necessary to carefully design a *fitness function* that evaluates the qualities displayed by an *individual* confronted to the given problem.

In the present case:

- The coding of an *individual* is a binary sequence wich represents its *genotype* from which the diffraction spectrum (its *phenotype*) is calculated.
- The *fitness* function carries the "resemblance" of the spectrum generated by a given *individual* to the target spectrum. With N_e sample points (the values of the wave vector q for which the intensity $I_n(q)$ is calculated) in the spectrum, one defines the following "distance" between spectra and the *fitness* function:

$$distance(sp_i, sp_c) = \frac{1}{N_e} \sum_{k=0}^{N_e-1} \left[sp_i(k) - sp_c(k)\right]^2$$

$$fitness(i) = \exp\left(-distance(sp_i, sp_c)\right)$$

where $(sp_i(k))_{k\in[0;N_e-1]}$ are the values of the sampled spectrum associated to *individual* i and $(sp_c(k))_{k\in[0;N_e-1]}$ the values of the target spectrum. Note that this value is exactly 1 when the spectra sp_i and sp_c are identical, and goes to 0 as the two spectra are increasingly different.

An analysis has been performed with target spectra that are X-ray diffraction spectra numerically calculated using the general model of equation (4) with a given binary sequence of length N. Figures 12 and 13 show sufficiently well how very close they are to the experimental data. This binary sequence is then the exact solution and the *individual* which represents it has a *fitness* value equal to 1. One assumes furthermore that the sequence length N is known, which limits the search space to the set of binary chains of length N for a space size of 2^N.

We consider, as in [10], a Prouhet-Thue-Morse sequence of order 7, the size if the search space is then 2^{128}. Such a size alone would not be an obstacle for the convergence of an optimization algorithm if the $fitness$ function did not have a multimodal character, *i.e.* it has only one global optimum.

Preliminary computations show that a classical Genetic Algorithm must be improved due to difficulties arising from the multimodal character of the *fitness function* and the lack of specificity of the Hamming distance to characterise the "efficiency" of the *individuals* having the best *fitness* value in nearing the solution (the Hamming distance, that is the number of sites at which two binary chains of identical length are different, is the natural distance used in search spaces when using character strings or binary codes).

Moreover, the sequence length has a direct effect on the oscillating character of the corresponding spectrum. In order to have a correct sampling of the target spectrum, the number of sample points N_e has thus to be in proportion to the sequence length.

2.3 The shared Genetic Algorithm

Because of the above mentionned difficulties it has been found useful to use the *sharing* [32] method which allows to keep some genetic diversity in a population, so as to reduce the risk of premature convergence. (See [33] for a survey of the methods used for the conservation of genetic diversity among a Genetic Algorithm population.)

Sharing methods can briefly be described as follows: by analogy with the natural phenomenon of "niching", the Genetic Algorithm is modified in order to simultaneously explore all the promising zones it discovers in the search space. The concept of sharing stems from the following need: If the individuals of a same population subgroup have to share their resources, the growth of this population is limited. In case of overpopulation, the individuals will tend to look for new territories to be explored. The major way of controlling a Genetic Algorithm being through its fitness function, a simple strategy is to lower the fitness of an individual with respect to its neighbors in the current population. It is based on the use of a distance defined on the search space and computed either on chromosomes (genotypic distance) or on the search space itself (phenotypic distance). In the current problem, we shall keep the Hamming distance.

Another simple way to maintain genetic diversity and to avoid fitness recalculation of the same individual, is to forbid identical individuals inside the current population. The application of a then so-called "elitist" strategy has also proven to be efficient in order to reduce the effects of fitness sensitivity to a small genetic change. In this case, only a part of the population is replaced at each generation, this part being generally made of individuals having the lowest fitness values. This elitist strategy has also the advantage of keeping mainly those individuals that represent "good" solutions in the current population.

Moreover, the joint use of a *sharing* and an elitist strategy raises the problem of the preservation of some genetic diversity inside the selection process. The *sharing* method presented by Miller and Saw in [34], thereafter called *Dynamic Niche Sharing*, seems to us particularly well adapted to combine the advantages of elitism and sharing strategies. An improved version of this technique is used here (see Appendix for details).

Finally, according to Baker [35], who has compared several selection methods, the *Stochastic Universal Sampling* selection method (instead of a classical *Roulette Wheel Selection*) seems to us more appropriate in the present problem. The Roulette Wheel Selection has a larger variance with respect to the number of offspring of a given *individual*, while these methods both produce

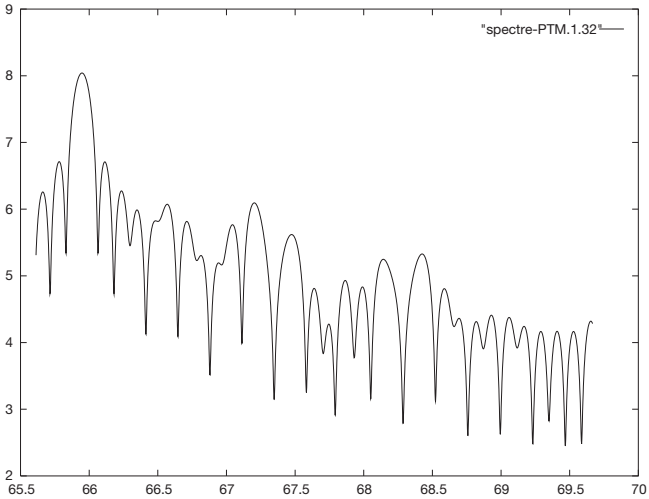

Fig. 4. Spectrum $I_n(q)$ generated with the Prouhet-Thue-Morse sequence of size 32 (see text).

in average the same expected number of offspring $p(i)$ that is for individual i of a population of size N:

$$p(i) = \frac{fitness(i)}{\sum_{j=1}^{N} fitness(j)}. \tag{5}$$

3 Results

Three types of sequences with three different length have been tested to generate the *target* spectrum for the Genetic Algorithm: a Prouhet-Thue-Morse sequence, a periodic sequence (strict alternance of 0 and 1), and a randomly generated sequence. Their sizes have been successively 32, 64 and 128.

3.1 Sequences of length 32

Figures 4, 5 and 6 show the intensities $I_n(q)$ as a function of the wave vector q for the *target* spectra corresponding to the three different types of sequences of size 32.

The Genetic Algorithm parameters are:

- Population size: 160 *individuals*.
- Use of two points crossover, with a crossover probability $p_c = 0.85$.
- Mutation probability $p_m = 0.02$.
- Use of the *dynamic niche sharing*, with the number of niches to be identified $P = 8$, the parameter $\sigma_{share} = 12$, that controls the mean radius of a niche (the Hamming distance is used for a metric on the search space).
- The survival rate $t_s = 0.5$. It corresponds to the proportion of *individuals* in a population subset being automatically transferred to the next generation.

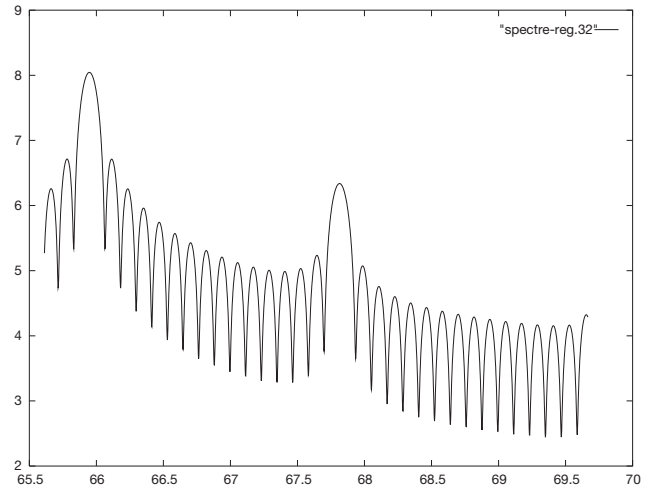

Fig. 5. Spectrum $I_n(q)$ generated with the periodic sequence of size 32 (see text).

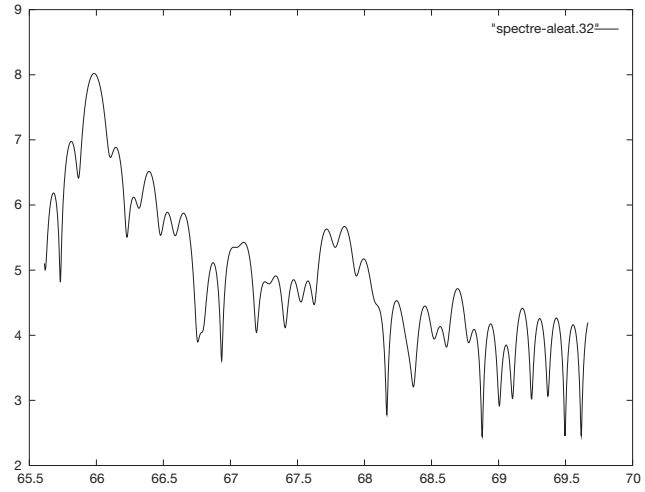

Fig. 6. Spectrum $I_n(q)$ generated with the random sequence of size 32 (see text).

The *dynamic niche sharing* parameters have been chosen in order to take into account several facts:

- Two binary chains of length l uniformly randomly chosen differ on the average by $l/2$. A value of σ_{share} too close to or larger than $l/2$ would induce too many overlaps.
- A value σ_{share} too low would induce a too large number of population subsets considering the number of desired niches. The majority of the population *individuals* would then belong to the non-peaks category.
- For a given size of population subset, a too large number of niches would induce an average population subsets size too low.

The parameter P is thus chosen as a function of the population size, so that the population subsets be not too small. The parameter σ_{share} has thus been fixed in order to perform a classification of the major part of the population.

Table 2. Parameter setting of the GA for sequences of length 32.

Population size	160
p_c	0.85
p_m	0.02
Number of niches P	8
σ_{share}	12
Survival rate t_s	0.5
Number of sample points N_s	210

Table 3. Number of generations to find the target for sequences of length 32.

Sequence	N_g minimum	N_g maximum	N_g average	Standard deviation
Periodic	24	47	35.5	7.5
PTM	33	163	66	35
Random	46	447	104	90

Finally, the number of sample points N_s is 210, this number directly influences the computation time, but a subsampling of the spectrum would diminish the efficiency of the *fitness* function.

These parameters are summarised in Table 2.

For 20 tests performed according to these conditions, we present in Table 3 the number of the generation N_g used to find the *target* sequence, or its mirror, whose spectrum in the case of the Prouhet-Thue-Morse and periodic sequence as quasi identical to the *target* spectrum.

The computation time for 100 generations is near 5 minutes on a dec-alpha station (DEC-3000-M300X).

It is interesting to note the extent to which, the more the spectrum depends on a regular underlying structure, the more efficient the Genetic Algorithm is.

3.2 Sequences of length 64

The results are shown in Figures 7, 8 and 9. The Genetic Algorithm parameters are presented in Table 4.

The average computation time for 100 generations is 40 minutes on a dec-alpha station. For 5 tests we obtain the results of Table 5.

For this parameter setting, the performances of the Genetic Algorithm on a random sequence are very bad, confirming the fact that spectra based on sequences having a regular structure are more easily inversed (we have already noticed this with the experiments on length 32 sequences). A modified parameters setting (see Tab. 6) led to a computation time of almost 2 hours for 100 generations, results are presented in the last raw of Table 5.

For comparison, the Genetic Algorithm needs around 20 000 *fitness* evaluations to converge in the case of a periodic sequence, whereas it needs 50 000 *fitness* evaluations

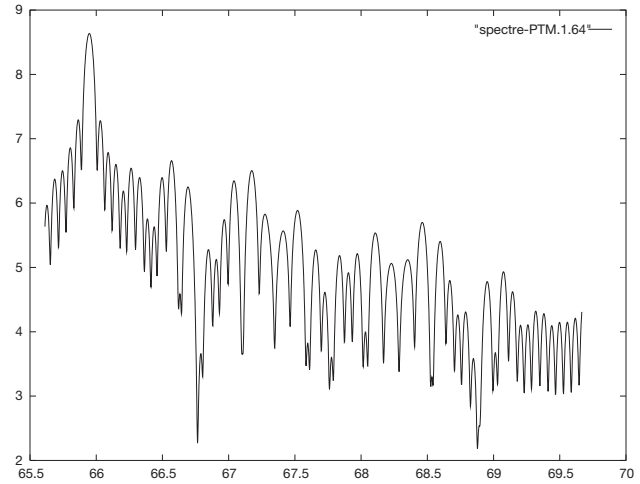

Fig. 7. Spectrum $I_n(q)$ generated with the Prouhet-Thue-Morse sequence of size 64 (see text).

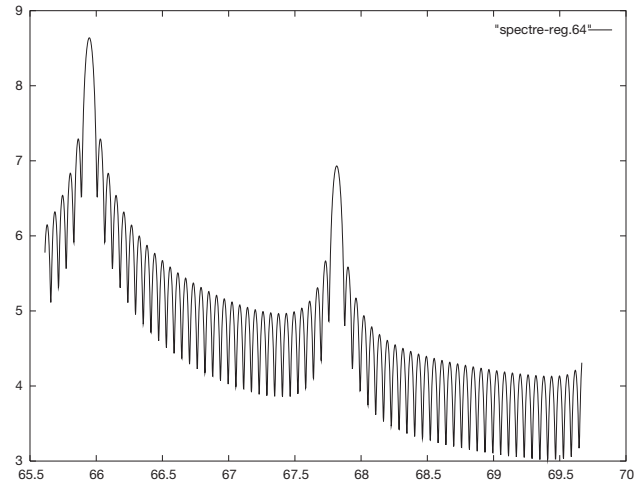

Fig. 8. Spectrum $I_n(q)$ generated with the periodic sequence of size 64 (see text).

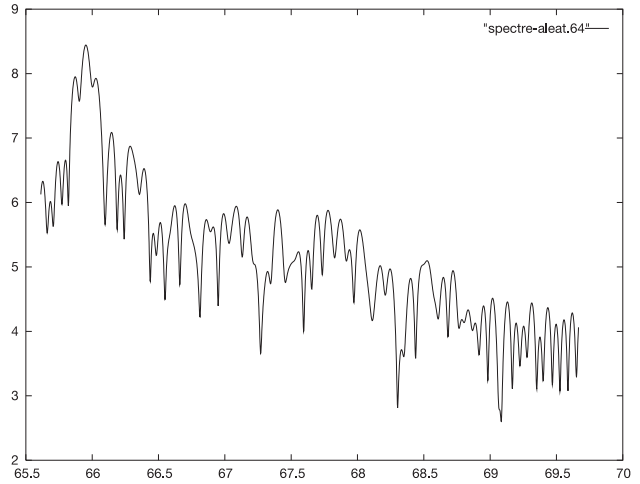

Fig. 9. Spectrum $I_n(q)$ generated with the random sequence of size 64 (see text).

Table 4. Parameter setting of the GA for periodic and PTM sequences of length 64.

Population size	400
p_c	0.85
p_m	0.01
Number of niches P	16
σ_{share}	24
Survival rate t_s	0.5
Number of sample points N_s	416

Table 5. Number of generations to find the target for sequences of length 64.

Sequence	N_g minimum	N_g maximum	N_g average	Standard deviation
Periodic	63	136	103.6	28
PTM	221	341	262	47
random	567	1712	1068	462

Table 6. Parameter setting of the GA for random sequences of length 64.

Population size	1000
p_c	0.85
p_m	0.01
Number of niches P	20
σ_{share}	24
Survival rate t_s	0.5
Number of sample points N_s	520

for the Prouhet-Thue-Morse sequence, and 500 000 for the random sequence (the search space is always of size $|S| = 2^{64} \simeq 1.84 \times 10^{18}$).

3.3 Sequences of length 128

In this case, experiments on the Prouhet-Thue-Morse sequence of order 7 and on a periodic sequence have been performed (Figs. 10 and 11 show the associated *target* spectra). The size of the search space increases in an exponential way with respect to the preceding test, and at the same time, the computation time of the spectra also increases in a significant way. If we increase the number of sample points proportionally to the sequence length, the complexity would be $O(n^2)$. Experiments prove that increasing the population size is sufficient to make the Genetic Algorithm converge to the solution.

The Genetic Algorithm parameters are summarised in Table 7.

The Genetic Algorithm finally found the Prouhet-Thue-Morse sequence after 884 generations, *i.e.* near 900 000 *fitness* evaluations (due to rounding errors during the computation of the number of survivors of a population subset, an average of 1000 *individuals* are created at each generation). For the periodic sequence, 184 generations were necessary, *i.e.* near 200 000 *fitness* evaluations.

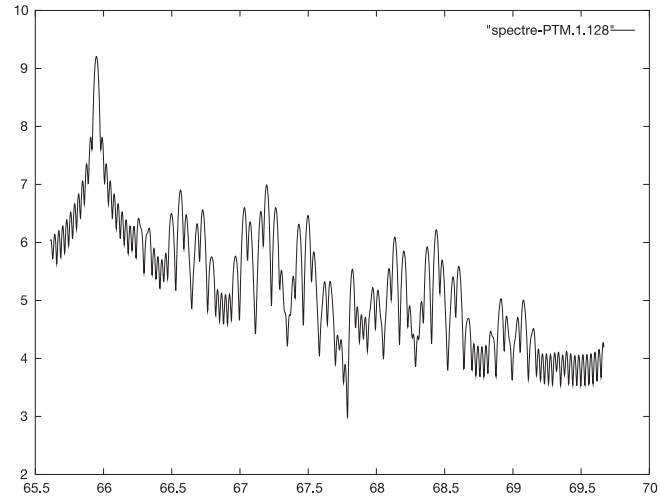

Fig. 10. Spectrum $I_n(q)$ generated with the Prouhet-Thue-Morse sequence of size 128.

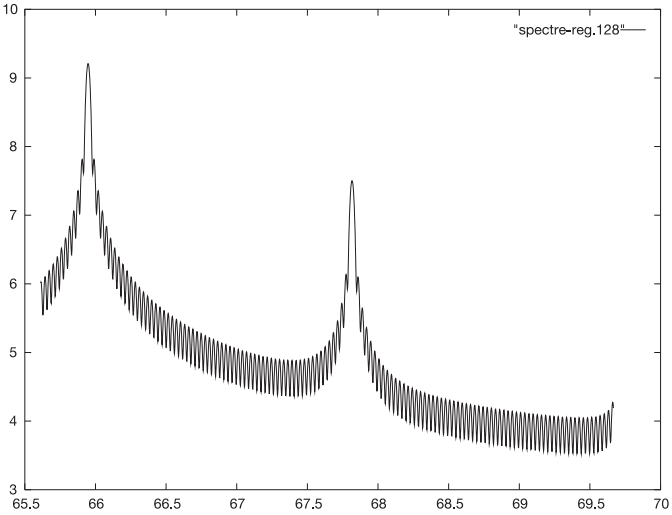

Fig. 11. Spectrum $I_n(q)$ generated with the periodic sequence of size 128.

Table 7. Parameter setting of the GA for sequences of length 128.

Population size	2000
p_c	0.85
p_m	0.01
Number of niches P	30
σ_{share}	48
Survival rate t_s	0.5
Number of sample points N_s	624

The computations time was near 1 hour for 10 generations. The Genetic Algorithm parameter setting certainly needs further adjustments (also the choice of the number of sampling points), but it is obvious that such computation times discourage systematic tests. We can still consider as quite satisfactory the fact that it is definitely possible to solve this problem with a Genetic Algorithm.

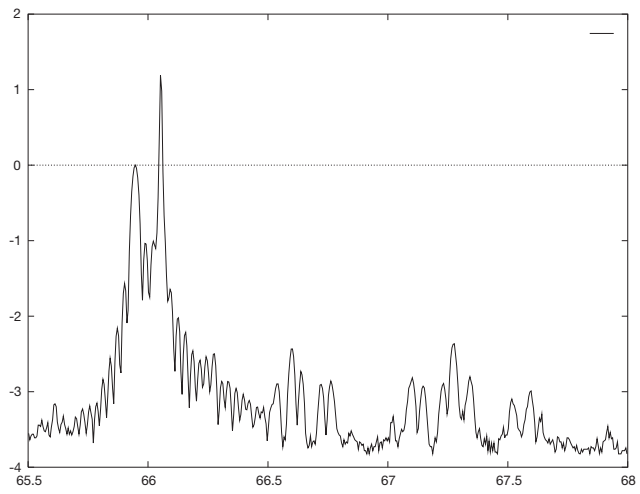

Fig. 12. Experimental spectrum, showing the diffracted intensity $I(q)$ in arbitrary units as a function of wave vector q, for a Prouhet-Thue-Morse GaAs-AlAs multilayer with $N = 128$ (courtesy F. Laruelle, L. Leprince and J. Schneck, CNET-Bagneux).

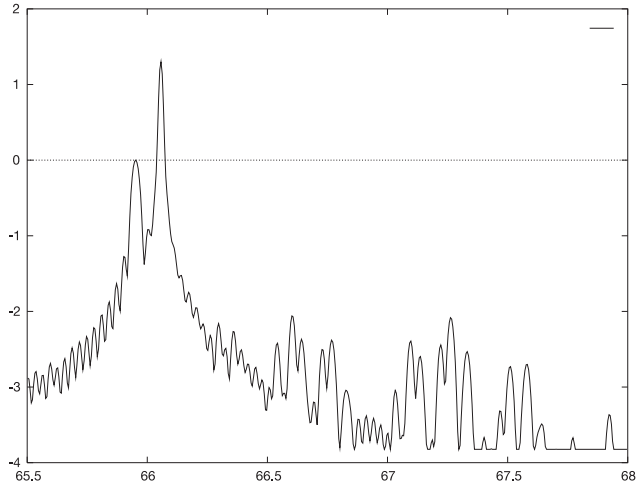

Fig. 13. Computed spectrum from the model of [10] with a simple substrate model.

4 Conclusion

We presented an application of Genetic Algorithms to the analysis of high resolution X-Ray diffraction spectra of binary multilayer heterostructures. The method actually allows the accurate retrieval of the binary generating sequence from the analysis of spectra of an unknown multilayer sample up to a layer number of 128.

Other target spectra should be investigated, so as to enable us to know the extent to which this technology allows satisfactory structure determination in systems with aperiodic order.

The authors thank E. Cockayne for sending them his computer program of reference [10], F. Laruelle, L. Leprince and J. Schneck for stimulating discussions, and the anonymous referees for their remarks.

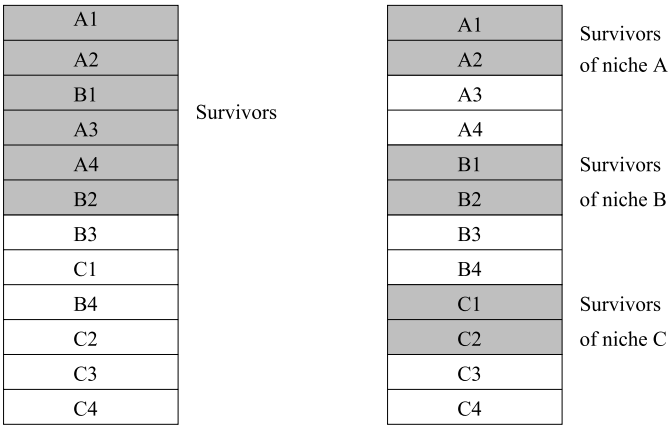

Fig. 14. An elitist strategy for dynamic niche sharing.

Appendix: Dynamic niche sharing

This method rests on the two hypotheses usually made for *sharing*:

1. the number of peaks P can validly be estimated;
2. the peaks are all situated to within a minimum relative distance of $2 \times \sigma_{share}$ (this is the threshold value for the distance between two *individuals* under which they are considered to belong to the same niche).

The essential characteristic is that for a Genetic Algorithm using sharing this way, the *individuals* gradually occupy the *niches* as generations develop. The *dynamic niche sharing* tends to identify P peaks of this forming niches and uses the dynamically identified peaks to classify all the *individuals* as belonging either to one of the dynamic niches (because they are within a distance smaller than σ_{share} of a dynamic peak), or belonging to the class of the "non-peak".

Within this framework, the modified *fitness* value F_{share} of an *individual* belonging to a dynamic niche equals its initial *fitness* divided by the size of the dynamic niche population. The *fitness* of the *individuals* belonging to the category of non-peaks is divided by the meter m_i used for the classical sharing (see [32]). So the new fitness value F_{share} is given by

$$F_{share}(i) = \frac{fitness(i)}{m_{dsh,i}}$$

where the dynamic niche meter, $m_{dsh,i}$ is given by:

$$m_{dsh,i} = \begin{cases} n_j & \text{if } i \text{ belongs to the dynamic niche } j, \\ m_i & \text{if } i \text{ belongs to the "non-peak"class).} \end{cases}$$

with

$$m_i = \sum_{k=1}^{N} Sh(d_{ik}).$$

The peak identification technique proposed by Miller and Shaw (*Greedy Dynamic Peak Identification*) can be described as follows:

1. Sorting of the population based on decreasing initial *fitness*.
2. Extraction of *individual* i from this sorting.
3. If i belongs to one or several niches, it is classified in the niche the dynamic peak of which is the closest to i. Else i becomes a new dynamic peak and the number of identified niches increases by 1.
4. If the number of identified niches is less that P or if i is less that the population size, one goes back to step 2.
5. Else: end.

Essentially, the *dynamic niche sharing* is different from the classical sharing by the distribution of the population inside a niche. In this last case, the count of the *individuals* inside a given niche is done separately for each of the *individuals*. And since it knocks down the shared *fitness*, the Genetic Algorithm tends to reduce this count with the effect of making more uniform the population distribution inside a given niche. This process then slows down the convergence speed of this population subset towards the optimum of the considered niche (under the hypothesis, however, that inside a given niche, there can be only one optimum). On the contrary the dynamic niche sharing identify the niches in a more global manner and knocks down identically all the *individuals* of a given niche. The selective pressure inside a given niche then depends only on the initial *fitness* value of the *individuals*, yet unchanged by sharing.

This value easily lends itself to the application of an "elitist" strategy, since one indeed can use the explicit division of the population into subsets: one subset per identified niche and possibly the subset of unclassified *individuals*. The operational choice to keep a fixed proportion of the population then naturally extends to each population subset.

Figure 14 is an illustration of this idea.

- The left table shows a population classified in decreasing order (A1 has the best *fitness* value, C4 the lowest). The population is divided into three population subsets corresponding to niches A, B and C, where A1 is the best *individual* of niche A, A4 the worst, etc ... This example illustrates the case where the whole population is taken into account for the choice of survivors

(the proportion of survivors is 0.5, survivors being *individuals* selected to be transfered directly from the former the the latter population class). All the *individuals* of niche A survive, contrary to those of niche C.

- The right table illustrates the case when the choice of survivors is performed, still with a proportion of 0.5, out of the *individual* population subsets. The *individuals* are gathered in each niche and internally classified in decreasing order. Each niche having the same number of *individuals*, they all have equal numbers of survivors.

It can be clearly seen that the first method goes against the aim of sharing which is to keep each niche populated, or at least the interesting ones if there are too many.

References

1. A. Bihian, C. Lemarechal, J.J. Strodiot, *On a bundle algorithm for nonsmooth optimization*, in *Non-Linear Programming 4*, edited by O.L. Mangasarian, R.R. Meyer, S.M. Robinson (Academic Press, 1981), pp. 245–282
2. E. Aarts, P. Van Laarhoven, *Simulated annealing: a pedestrian review of the theory and some applications*, NATO ASI Series Vol. F30, PCO, Elcerlyclaan 2, B-3090 Overijse, Belgium, 1986
3. S. Kirkpatrick, C.D. Gelatt Jr, M.P. Vecci, Science **220**, 671 (1983)
4. N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, E.J. Teller, J. Chem. Phys. **21**, 1087 (1953)
5. S. Geman, D. Geman, IEEE Trans. Pattern An. Machine Intelligence **6**, 712 (1984)
6. M. Mezard, G. Parisi, J. Phys. Lett. **46**, L-771 (1985)
7. M. Mezard, G. Parisi, M.A. Virasoro, *Spin Glass Theory and Beyond*, Lecture Notes in Physics, Vol. 9 (World Scientific, 1987), and references therein
8. L. Davis, *Genetic Algorithms and Simulated Annealing* (Pittman, London, 1987)
9. D.A. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning* (Addison-Wesley, January 1989)
10. J. Peyrière, E. Cockayne, F. Axel, J. Phys. I France **5**, 111 (1995)
11. D. Schechtman, I. Blech, D. Gratias, J.W. Cahn, Phy. Rev. Lett. **53**, 1951 (1984)
12. G. Christol, T. Kamae, M. Mendès-France, G. Rauzy, Bull. Soc. Math. France **108**, 401 (1980)
13. M. Dekking, M. Mendès-France, A. van der Poorten, Math. Intellig. **4**, 130 (1982); **4**, 173 (1982); **4**, 90 (1982)
14. J.P. Allouche, Expositiones Mathematicae **5**, 239 (1987)
15. R. Merlin, K. Bajema, R. Clarke, F.Y. Juang, P.K. Bhattacharya, Phys. Rev. Lett. **55**, 1768 (1985)
16. H. Terauchi, S. Sekimoto, K. Kamigaki, H. Sakashita, N. Sano, H, Kato, M. Nakayama, J. Phys. Soc. Jpn **54**, 4576 (1985)
17. J. Todd, R. Merlin, R. Clarke, K.M. Mohanty, J.D. Axe, Phys. Rev. Lett. **57**, 1157 (1986)
18. H. Terauchi, Y. Noda, K. Kamigaki, S. Matsunaka, M. Nakayama, H. Kato, N. Sano, Y. Yamada, J. Phys. Soc. Jpn **57**, 2416 (1988)
19. H. Terauchi, K. Kamigaki, T. Okutani, Y. Hishihata, H. Kasatani, H. Kasano, K. Sakane, H. Kato, N. Sano, J. Phys. Soc. Jpn **59**, 405 (1990)
20. R. Merlin, K. Bajema, J. Nagle, K. Ploog, J. Phys. France, Colloq. **48**, C5-503 (1987)
21. F. Laruelle, V. Thierry-Mieg, M.C. Joncour, B. Etienne, J. Phys. France, Colloq. **49**, C5-529 (1988)
22. M. Mendès France, J-P Allouche, in *Beyond Quasicrystals*, edited by F. Axel, D. Gratias, Collection "Centre de Physique de Houches" (EDP Sciences/Springer Verlag, 1995)
23. F. Axel, H. Terauchi, Phys. Rev. Lett. **66**, 2223 (1991), and **73**, 1308 (1994) (Reply)
24. M. Cornier-Quiquandon, M. Quivy, S. Lefebvre, E. Elkaim, G. Heger, A. Katz, D. Gratias, Phys. Rev. B **44**, 2071 (1991)
25. W. Steurer, Acta Cryst. B **45**, 534 (1989)
26. J.H. Holland, *Adaptation in Natural and Artificial System* (Ann Arbor, University of Michigan Press, 1975)
27. R. Cerf, *Asymptotic convergence of genetic algorithms*, in *Artificial Evolution, European Conference, AE 95, Brest, France, September 1995, Selected papers*, Lecture Notes in Computer Science, Vol. 1063 (Springer Verlag, 1995), pp. 37–54
28. T.E. Davis, J.C. Principe, *A Simulated Annealing Like Convergence Theory for the Simple Genetic Algorithm*, in *Proceedings of the Fourth International Conference on Genetic Algorithm, 13-16 July, pages 174-182, 1991*
29. G. Rudolph, IEEE Trans. Neural Networks **5**, 96 (1994)
30. Y. Landrin-Schweitzer, E. Lutton, *Perturbation theory for Evolutionary Algorithms: towards an estimation of convergence speed*, in *Parallel Problem Solving from Nature - PPSN VI 6th International Conference, 2000* (Springer Verlag, 2000), LNCS 1917
31. B. Leblanc, E. Lutton, *ALGON Users guide,* in `http://www-rocq.inria.fr/fractales/`, 1997
32. D.E. Goldberg, J. Richardson, Genetic algorithms with sharing for multimodal function optimization, in *Genetic Algorithms and their Applications*, edited by J.J. Grefenstette (Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1987), pp. 41–49
33. S.W. Mahfoud, *Niching Methods for Genetic Algorithms*, Ph.D. thesis, University of Illinois at Urbana-Champaign, 1995. `http://GAL4.GE.UIUC.EDU/illigal.home.html`
34. B.L. Miller, M.J. Shaw, *Genetic algorithms with dynamic niche sharing for multimodal function optimization*, Technical report, IlliGAL Report, University of Chicago, 1995, in `http://GAL4.GE.UIUC.EDU/illigal.home.html`
35. J.E. Baker, *Reducing bias and inefficiency in the selection algorithm*, in *Genetic Algorithms and their applications: Proceedings of the Second International Conference on Genetic Algorithms, Cambridge, Mass, 1987, pp. 14–21*