Scaling behavior of genetic algorithms applied to surface structural determination by LEED

M L Viana[1], W Simões e Silva[1], E A Soares[1], V E de Carvalho[1], C M C de Castilho[2] and M A Van Hove[3]

[1] Departamento de Física, ICEx, Universidade Federal de Minas Gerais, Caixa Postal 702, 30123-970, Belo Horizonte, MG, Brazil

[2] Grupo de Física de Superfícies e Materiais, Instituto de Física, Universidade Federal da Bahia, Campus da Federação, 40210-340, Salvador, BA, Brazil

[3] Department of Physics and Materials Science, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong

E-mail: `mviana@fisica.ufmg.br`

Abstract. Surface structural determination by Low Energy Electron Diffraction (LEED) requires a fitting procedure between the theoretical and experimental I(V) curves. This fitting procedure is quantified through an R-factor methodology. However, the R-factor space topology presents a large number of local minima. Thus, the task of identifying the global minimum, i.e. the task of finding the correct surface structure, requires a global optimization method that is able to determine the surface structure of complex systems. In this work we present the results of the application of genetic algorithms to three different systems, including performance tests and a comparison with another opmization method previously applied to the LEED problem, Simulated Annealing. We also present a scaling relationship of the computational effort versus the number of parameters to be fitted for the genetic algorithm method.

1. Introduction

Low Energy Electron Diffraction [1] is one of the most powerful techniques for surface structural determination. The LEED pattern can directly exhibit the symmetry and periodicity of a surface. The intensity of each diffracted beam as function of the incident electron kinetic energy, the so-called I(V) curve, contains information about the surface structural parameters such as layer relaxations and atomic displacements, as well as non-structural parameters like inner potential and surface Debye temperature. A set of I(V) curves, collected for distinct diffracted beams can be compared with theoretical curves calculated by assuming values for a set of parameters, these corresponding to both structural and non-structural physical quantities. The essence of the LEED structural determination is then to try to reproduce theoretically the experimental curves, by varying the parameters associated to the calculated curves. However, there is no reliable direct method to do that. Thus, a theoretical simulation is required. Starting from general information about the atoms that belong to the surface (atomic coordinates, lattice parameter, Debye temperature and inner potential), the I(V) curves are simulated and compared to the experimental ones. The structural and non-structural parameters are then determined in a fit procedure that searches for the set of parameters that optimizes the agreement between the theoretical and experimental curves, through minimization of the so-called R-factor [1]. However, the R-factor topology in the surface parameter space presents a complex configuration of local minima that becomes more complex as the number of parameters to be fitted is increased. With a growing interest in complex systems, such as oxide compounds, semiconductors or metallic alloys, the need for versatile search tools becomes essential.

In this work we propose the use of genetic algorithms [2, 3], a global optimization method based on species evolution, in the search for a set of parameters that maximizes the agreement between the experimental and theoretical curves or, in other words, minimizes the R-factor. To calculate the theoretical I(V) curves and to evaluate the R-factor we have used the SATLEED (Symmetrized Automated Tensor LEED) package [4], a set of codes that simulates a LEED experiment through multiple scattering calculations. In section 2 we present a brief description of the SATLEED code. In section 3 we show the general aspects of genetic algorithms, including some preliminary tests. In section 4 we give some details of our GA-SATLEED implementation. In section 5 we present the structural determination results of the tested systems, including a comparison with previous results. In section 6 we show the results of some performance tests to which the GA-SATLEED code was submitted, including a comparison with other optimization methods, and finally, in the last section, we offer our conclusions.

2. The SATLEED package

The symmetrized automated tensor LEED (SATLEED) package was developed by M.A. Van Hove and A. Barbieri [5] from the previous version of the automated tensor LEED

program (ATLEED) [5, 6]. Both codes perform an automated structural determination of surface structure, simulating theoretical I(V) curves through multiple scattering calculations using the tensor LEED approach [7]. SATLEED, the symmetrized version, allows the exploitation of symmetry to greatly increase computing speed: this relies on experimental data taken at normal incidence or with the incident beam in a mirror plane of the surface structure. After the I(V) calculation step, the SATLEED performs a fit procedure between the theoretical I(V) curves and the experimental ones by adjusting the structural and non-structural surface parameters through the minimization of the R-factor. The Simplex [8] and Powell [9] methods were implemented in the SATLEED package in order to perform the R-factor minimization. These optimization methods work well for local optimization, i.e. when the search starts from a point that is near to the global minimum. But for global optimization, it is necessary to use global optimization methods, such as simulated annealing or genetic algorithms.

3. Genetic algorithms

Although the simulated annealing approach has been applied to the global LEED optimization problem in a number of studies [10, 11], the genetic algorithm was only used in LEED analysis for a relatively confined problem [12], and recently it has been more extensively explored in a surface structure determination by the photoelectron diffraction technique [13].

Genetic algorithms (GAs) are a particular class of evolutionary algorithms [14] that use techniques inspired by evolutionary biology such as elitism, mutation, selection, and crossover (also called recombination). GAs are very useful when applied to optimization problems that require an extensive search in a parameter space that presents several local minima. In order to identify the global minimum among the local ones, the GAs start the search from several different points in the parameter space. Each point, corresponding to a random set of parameters, is treated, in the GA terminology, as an individual, and a set of individuals forms a population. The adaptation level or fitness of an individual is evaluated through the function to be optimized. At each step of the algorithm the population evolves to a new generation which, in general, is better adapted to the problem than the previous one, by evolutionary devices such as elitism, crossover and mutation.

As in nature, the evolutionary devices in GAs act on the genotype of the individuals, changing their chromosomes. This requires that the individuals be encoded in strings like chromosomes. There are several ways to perform this codification, such as the classical or binary encoding [3, 14] and the real encoding [2]. In a binary encoding, each parameter is represented by a predefined number of bits (with values 0 or 1); that predefined number depends on the desired accuracy. The binary strings corresponding to each parameter join to form an unique chromosome that contains information about all the parameters for an individual, as we can see in table 1. In a real encoding each parameter is directly represented by a real number, thus the string (or chromosome) is

Table 1. Example of binary and real codifications in GAs.

	First parameter	Second parameter	Third parameter	Chromosome
Binary encoding	001111	110011	010101	001111110011010101
Real encoding	-0.23	0.85	0.45	-0.23 0.85 0.45

formed by a sequence of real numbers, each one representing a parameter, as can also be seen in table 1.

The main search device in GAs is the *crossover* [2, 3], which mixes the chromosomes of a pair of individuals to give rise to a new individual that will belong to the next generation. The choice of the pairs to be combined is based on the fitness of each individual in the present generation under the condition that the best individuals have a higher chance to be chosen. In order to simulate the natural behavior in mixing chromosomes, several ways to perform crossover in numerical strings have been proposed [15, 16]. As can be seen in figure 1-A, either for real or binary encodings, one of the simplest ways to perform crossover is to break the strings (chromosomes) at a point exactly between two parameters, such that the offspring inherit parameters from the parents, but no parameter has its value modified. Another possibility is to break the strings at points occuring "inside" a parameter as shown in figure 1-B for binary codification. In this case the value of the resultant offspring parameter is different from those of the parents, since it is built up by mixing the bits from the parents. In figure 1-C is presented an alternative way to perform the crossover. Taking the break point occuring "inside" a parameter, the offspring parameters can be constructed by a linear combination of the parent parameters, choosing the linear coefficients randomly. The examples presented in figure 1 are only some simple ways to perform crossover among many other alternative ways that have been proposed in the last years [15, 16]. However, all of them are based on the same principle: to create new individuals by inheriting properties from the parents. All the methods described above apply to both real and binary representations, however the real encoding has been used more frequently because it allows higher accuracy.

Another fundamental evolutionary device is *elitism* [2, 3] or survival of the best individuals. As the crossover does not guarantee that the offspring will have a better fitness than their parents, the survival of at least the most adapted individual helps to improve the next generation: it is simply copied (cloned) to the next generation, in order to speed up the convergence.

In the *mutation* [2, 3] process an individual is randomly chosen, and then one or more of its parameters is randomly modified. The aim of the mutation is to generate more diversity in the offspring, by creating offspring that are in some ways very different from their parents. Such mutation is applied relatively rarely, so as not to slow down the convergence too much.

4. The GA-SATLEED implementation

In our implementation one individual represents a set of surface structural and non-structural parameters, i.e. one specific trial geometry. The evaluation function is the R-factor calculated by the SATLEED code. The probabilities of selection and the elitism criterion are based on the R-factor values. We allow the parameters to vary within predefined ranges consistent with the assumed symmetry. The GA-SATLEED steps used in our implementation are as follows:

(i) An initial population of N individuals is chosen (N is an even number). Each individual is a vector containing the P parameters to be optimized in the structure analysis of the system. The value of each parameter is randomly chosen within a physically acceptable range for that system. Each individual is coded as a binary or real string that will contain the displacements to be added to the parameters of a reference initial surface structure.

(ii) The SATLEED code calculates the theoretical I(V) curves for one individual, which we call individual I. It starts from general information available about the system, such as atomic phase shifts, sample temperature, etc., but modifies the coordinates according to the displacements related to individual I. It makes the comparison to the experimental curves, and returns the R-factor value associated with individual I.

(iii) The probability of being selected for crossover is calculated based on the R-factor of the individual I. That probability should give to individuals with lower R-factors higher chances to be selected for the crossover process without to completely exclude the worst ones. In this implementation the probability for the individual I is calculated by $P_I = Ce^{-R_I}$, where C is a normalization constant, and R_I is the R-factor associated to the individual I.

(iv) The steps (ii) and (iii) are repeated for all other individuals in the current population.

(v) The search stops here if the best R-factor has not decreased after a preset number of generations or if a preset maximum number of generations is reached: the best individual in the last generation is selected as the best solution. Otherwise, the best individual is cloned to the next generation (by elitism) and the process continues.

(vi) N/2 pairs of individuals are chosen according to probabilities based on their R-factors obtained in step (iii). The crossover process then creates N new individuals for the next generation. The worst one is discarded, being replaced by the clone obtained in the previous step.

(vii) A number between 0 and 1 is chosen randomly. If this number is smaller than a previously chosen mutation rate, then a randomly chosen individual is subjected to the mutation process.

(viii) The new generation is ready, and the process restarts at step (ii) for the new generation.

Table 2. Best R-factor for the tested structural models for the $Ni(111)(\sqrt{3} \times \sqrt{3})R30° - Sn$ system.

	R-factor Standard LEED analysis [17]	R-factor GA-SATLEED
FCC overlayer	0.43	0.45
HCP overlayer	0.56	0.54
FCC substitutional	0.17	0.20
HCP substitutional	0.60	0.63
Top	0.46	0.45

Figure 2 shows a schematic flowchart for the GA-SATLEED code.

5. Structural determination results

In order to test the applicability of the method and to investigate its performance we used three systems previously studied using standard LEED analysis: Ni(111)($\sqrt{3} \times \sqrt{3}$)R30°-Sn, InSb(110), and CdTe(110). In this section we present their structure determination results, and a comparison with the previous results.

For the Ni(111)($\sqrt{3} \times \sqrt{3}$)R30°-Sn system we used the normal incidence experimental data collected, at 198 K, by the Woodruff group at Warwick. These data are the same used by Soares et al in [17]. We applied genetic algorithm fitness using 5 different reference models: FCC (ABC stacking sequence) and HCP (ABA stacking sequence) overlayer models, FCC and HCP substitutional models, and on top model. We found the best fit for the FCC substitutional model, shown in figure 3, in agreement with the previous work of Soares et al [17]. Table 2 shows the best R-factors obtained for all tested models. In table 2, we notice that the R-factor obtained with the Genetic Algorithm is not necessarily equal to or better than that of the earlier standard analysis. This illustrates the important fact that a GA is designed for efficient global searching and is not as efficient for local searching. For example, steepest-descent methods are more efficient for local searches than are GAs. This is why we propose in Section 6.3 a combination of global and local searching, giving a much higher efficiency for locating a global minimum. Table 3 shows the resulting improvement in the result.

We used, for the InSb(110) system, the same experimental data used by Soares et al in [18, 19] that includes 12 beams measured near normal incidence. For the fitness using the GA-SATLEED code we started from the "bulk structure" and used a fixed value of 180 K for the Debye temperature which we took from the previuos work of Soares et al [18, 19]. A total of 8 structural parameters were fitted: the Z coordinates (perpendicular to the surface plane) for the In and Sb atoms in the first and second atomic layers, and the X coordinates (parallel to the bulk In-Sb bond length) for the In and Sb atoms in the first and second layers. We found for the first layer a bond length rotation around 34°, which is in good agreement with the previous results [18, 19], in

Table 3. Structural determination results for the best tested model, the substitutional FCC, for the $Ni(111)(\sqrt{3}\times\sqrt{3})R30^\circ - Sn$ system, obtained by the standard LEED approach and GA-SATLEED including local optimization as described in Section 6.3. Δd is the interlayer distance change in angstrom or the percent difference from the bulk distance, and Θ_D is the surface Debye temperature.

	Standard LEED analysis [17]	GA-SATLEED
$\Delta d_{Sn}(\mathring{A})$	0.45 ± 0.03	0.44 ± 0.03
$\Theta_D^{Sn}(K)$	120 ± 50	130 ± 50
$\Delta d_{12}(\mathring{A})$	1.98 ± 0.02	2.00 ± 0.03
$\Delta d_{12}(\%)$	-2.52	-2.48
$\Theta_D^{Ni}(K)$	240 ± 50	240 ± 50
R_p	0.17 ± 0.03	0.17 ± 0.03

which was performed a standard LEED analysis by the ATLEED code [5]. Table 4 shows the complete structural determination results. For comparison the results from the previous analysis are also presented.

Table 4. Comparison between the structural determination results obtained by the GA-SATLEED code and the previous results for the system InSb(110), from Tensor LEED (TLEED).

	ATLEED [18, 19]	GA-SATLEED
First layer		
ΔZ_1^{Sb} $(\mathring{A})$	$\uparrow 0.23 \pm 0.05$	$\uparrow 0.25 \pm 0.05$
ΔZ_1^{In} $(\mathring{A})$	$\downarrow 0.59 \pm 0.07$	$\downarrow 0.55 \pm 0.06$
ΔX_1^{Sb} $(\mathring{A})$	-0.34 ± 0.15	-0.34 ± 0.09
ΔX_1^{In} $(\mathring{A})$	-0.54 ± 0.15	-0.57 ± 0.09
ω_1 $(^\circ)$	30	34
Rumple $(\mathring{A})$	0.82	0.80
$\Theta_D(K)$	180	180
Second layer		
ΔZ_2^{Sb} $(\mathring{A})$	$\downarrow 0.01 \pm 0.05$	$\downarrow 0.04 \pm 0.05$
ΔZ_2^{In} $(\mathring{A})$	$\uparrow 0.14 \pm 0.08$	$\uparrow 0.09 \pm 0.07$
ΔX_2^{Sb} $(\mathring{A})$	0.05 ± 0.14	0.02 ± 0.09
ΔX_2^{In} $(\mathring{A})$	0.08 ± 0.19	0.04 ± 0.08
ω_2 $(^\circ)$	5	2
Rumple $(\mathring{A})$	0.15	0.13
R_p	0.38 ± 0.06	0.33 ± 0.05

The third tested system was CdTe(110). We used the same experimental data obtained by Soares et al in [20]. The 10 inequivalent I(V) curves were collected at normal incidence from 20 to 150 eV. For this system, we fitted the Z coordinates (perpendicular to the surface plane) for the Cd and Te atoms in the first, second, and third atomic layers,

and the X coordinates (parallel to the bulk In-Sb bond lenght) for the Cd and Te atoms in the first, second, and third layers, totaling 12 structural parameters. We started from the "bulk structure" and used a surface Debye temperature of 140 K, obtained from the previous work [20], in which the least squares fitting method, included in the LEEDFIT code [21, 22], was used to determine the structural and non-structural parameters. The CdTe(110) system was also previously analysed by the Fast Simulated Annealing (FSA) search method [11] using the conventional Van Hove/Tong LEED code [23] to calculate the theoretical I/V curves. Also for this system a high bond-rotation angle, around 30 °, was found for the first layer. The structural results for the two first layers obtained by the GA-SATLEED and by a previous standard LEED analysis are shown in table 5. We omitted the results for the third layer because no changes were observed.

Table 5. Comparison between the structural determination results obtained by the GA-SATLEED code with previous results for the system CdTe(110).

	LEEDFIT [20]	FSA [11]	GA-SATLEED
First layer			
ΔZ_1^{Te} (Å)	$\uparrow 0.15 \pm 0.05$	$\uparrow 0.17 \pm 0.05$	$\uparrow 0.18 \pm 0.05$
ΔZ_1^{Cd} (Å)	$\downarrow 0.65 \pm 0.05$	$\downarrow 0.62 \pm 0.05$	$\downarrow 0.66 \pm 0.05$
ΔX_1^{Te} (Å)	-0.13 ± 0.06	-0.11 ± 0.06	-0.10 ± 0.05
ΔX_1^{Cd} (Å)	-0.38 ± 0.06	-0.39 ± 0.06	-0.41 ± 0.06
ω_1 (°)	30.3	30.9	31.5
Rumple (Å)	0.80	0.79	0.84
Θ_1^{Te}	141 ± 200	140 ± 200	140 ± 200
Θ_1^{Cd}	(144 ± 200)	140 ± 200	140 ± 200
Second layer			
ΔZ_2^{Te} (Å)	$\downarrow 0.03 \pm 0.06$	$\downarrow 0.02 \pm 0.06$	$\downarrow 0.05 \pm 0.05$
ΔZ_2^{Cd} (Å)	$\uparrow 0.04 \pm 0.06$	$\uparrow 0.05 \pm 0.06$	$\uparrow 0.07 \pm 0.06$
ΔX_2^{Te} (Å)	0.06 ± 0.07	0.05 ± 0.07	0.04 ± 0.06
ΔX_2^{Cd} (Å)	0.02 ± 0.07	0.02 ± 0.07	0.01 ± 0.06
ω_2 (°)	2.4	2.5	1.2
Rumple (Å)	0.07	0.07	0.11
Θ_2^{Te}	144 ± 250	140 ± 200	140 ± 200
Θ_2^{Cd}	142 ± 250	(140)	140 ± 200
R_p	0.48 ± 0.06	0.44 ± 0.08	0.38 ± 0.06

The GA-SATLEED code showed itself to be a very useful tool in the surface structure determination by LEED since it was able to find the correct structures for the three systems we have tested. The next section presents the results of an attempt to find a scaling behavior for the GA-SATLEED code as well as a discussion of its performance when compared to another optimization method.

6. GA-SATLEED performance

In order to test the genetic algorithm performance and to compare it to simulated annealing [11], we have performed several evaluations. First of all, it is very important to know the scaling behavior of the method, or in other words, how the computational effort increases with the number of parameters to be fitted. That is particularly important because the main reason for using genetic algorithms is to allow an efficient analysis of complex systems with many parameters to be fitted and many local minima. In this way, we have studied the GA scaling behavior by minimizing two kinds of function, a multi-dimensional mathematical function and the LEED R-factor. Another required characteristic for a global optimization method is its capability to explore different regions of the parameter space, as is observed for the three applications we have studied. We have also applied the GA search using local refinement for the LEED problem: the search showed itself much faster in this case. In the next paragraphs we present details of these studies as well as a discussion of our results.

6.1. Scaling behavior

Since we did not know how the GA scales with the number of parameters to be fitted for any kind of problem, we have applied it to an artificial "multi-dimensional diffraction function", shown in equation 1, in which we can easily vary the number of parameters and quickly evaluate the cost function.

$$f(x_1, x_2, ..., x_n) = \prod_{i=1}^{n} \frac{sin^2 x_i}{x_i^2} \tag{1}$$

Only after this initial exploration did we apply the GA to the LEED problem, with its much more time-consuming cost function. The number of configurations to be tried, in an optimization procedure, should in fact depend not only on the number of parameters, but also on the sensitivity of the cost function, the physically allowed range, and the desired accuracy for each parameter. In addition, each kind of cost function should present a particular topography, in which the number and the depths of the local minima clearly make the search effort to identify the global minimum highly variable. We should therefore not expect the same scaling behavior for different applications. However, since the function of equation 1 presents a large number of local minima and a well defined global minimum at $x_1 = x_2 = ... = x_n = 0$, it is a harder test than the LEED problem, and works at least for comparison, since the GA scaling behavior is completely unknown. Figure 4 shows the scaling behavior for the GA being applied to the minimization of the function of equation 1 when fitting from 3 to 12 parameters. Each point is an average over 10 runs using different initial populations. We observe a scaling factor B = 1.7 which was obtained by fitting the function $f(x) = Ax^B$ to the data points in figure 4.

For the analysis of the GA scaling behavior applied to the LEED problem, we have used the CdTe(110) system in a theory vs. theory comparison. We have used as input

for the SATLEED code the parameter values as presented in table 5 and the resulting theoretical I(V) curves were used as "pseudo-experimental" curves to be fitted by the genetic algorithm. In this way, we were able to obtain R-factors very close to zero, which guarantees that the method was able to find the global minimum in the tests. We performed 10 runs for different random initial populations in which 12 parameters were fitted. Then 2 of those parameters were fixed at their optimum values and we performed the search for the other 10 parameters, again, for 10 different initial populations. We proceeded in the same way for 8, 6, 4, and 2 parameters, always taking an average over 10 different runs. The results for those tests are presented in figure 5. We fitted a curve like $f(x) = Ax^B$ to the points in the graph of figure 5, and we found a scaling factor B = 1.3.

It can be observed by comparing figures 4 and 5 that the number of tries for the first application (equation 1) is around 100 times larger than for the LEED case. As we expected, the function of equation 1 is harder to minimize than the LEED R-factor in terms of configurations to be visited. However, the scaling behaviors for the two applications are in fact quite similar. The scaling factor values 1.7 ± 0.2 and 1.3 ± 0.4 for the "multidimentional diffraction function" and for the R-factor, respectively, are in agreement within the error bars. Our results are also in agreement with the previous work of Viana et al [13] where a scaling factor $B = 1.6$ was found for the GA applied to structure determination by photoelectron diffraction. In order to understand the meaning of this value we can compare it with previous scaling factors obtained for other optimization methods applied to the LEED structural analysis. For example, a steepest-descent algorithm exhibits approximately an N^2 scaling [6], where N is the number of parameters. The first application of the simulated annealing method to the LEED problem performed by Rous [10] suggested a N^6 scaling. Kottke and Heinz have proposed a steepest-descent method that performs global search by allowing non-vanishing probability for any parameter grid point [24], which results in a $N^{2.5}$ scaling. Also, the work of Nascimento and co-workers [11], using the fast simulated annealing approach, obtained a N^1 scaling, as shown in figure 6. That was the only global method showing a better performance, in terms of scaling behavior, than the genetic algorithm; however, as we discuss in the next section, the GA exhibits a better capability to explore different regions of parameter space.

6.2. Global search capability

A major characteristic of genetic algorithms is the fact that they use a *population* of candidate solutions. This allows a truly global search since each individual may start from a different region of parameter space. The FSA (Fast Simulated Annealing) method [11] starts the search from a single point and then allows parameter displacements to lead to other regions of the search space. Actually, both search methodologies are able to find the correct global minimum if given enough time, but sometimes the search could be very lengthy if the initial random structures are too far from the correct one. GAs

have the very desirable capability to get information about the system by exploring simultaneously different regions of parameter space, and by comparing them to each other. In addition, while the FSA requires R-factor evaluation of the present structure before making the next search step, the GA can perform simultaneous evaluations for all individuals in a generation, allowing the use of parallel computing, where several processor cores are used at the same time, dividing the computational time by the number of cores. For our particular implementation we can visualize in figure 7, for a theory vs. theory comparison using the CdTe(110) system, the "diversity" of regions being explored by their respective R-factors through different generations. We can also observe in figure 7 the trend that the population loses diversity after some generations, converging to the global minimum or to a local one. That trend is desirable in the case of convergence to the global minimum, but it should be avoided, by controlling the mutation rate, in the case of convergence to a local one. In any case, that trend can at least provide some useful information about the system by pointing out structures that could be better explored by local methods. In the next section we present the results of a simultaneous application of the GA and a local search method.

6.3. GA plus local refinement

What we really expect from a global optimization method, such as the GA, is to identify the "basin" in which the global minimum is located. Once inside the global basin, conventional local optimization methods like steepest-descent can work much more efficiently in finding the optimum. We can benefit from both methods by devising a hybrid approach in which we alternate the global search capability of a GA and the efficiency of a local optimization method. We have thus implemented a scheme in which each new trial structure generated by GA is first locally optimized before it is used in the next GA step of recombination (crossover). The local methods which we use are those that were already included in the SATLEED package, Simplex [8] and Powell [9]: they were implemented in a way that respects the desired symmetry of the sample in order to allow very fast refinement. We have tested this procedure for the $Ni(111)(\sqrt{3} \times \sqrt{3})R30^\circ - Sn$ system, and we found that the search speeded up by more than an order of magnitude. Figure 8 shows the scaling behavior for the GA without local refinement, and for the GA with simultaneous local refinement applied to the $Ni(111)(\sqrt{3} \times \sqrt{3})R30^\circ - Sn$ system. As can be seen, at least for this system, the search was much faster using the local refinement simultaneously with the global search of the GA.

7. Conclusions

The genetic algorithm method was applied to surface structural determination by LEED and proved to be a very helpful tool. We have tested it for three previously solved systems, $Ni(111)(\sqrt{3} \times \sqrt{3})R30^\circ - Sn$, InSb(110), and CdTe(110) and we found

structures which are in each case in close agreement with those previously published, with further improved R-factors. The scaling behaviour analysis suggests a scaling factor around 1.3, based on a theory vs. theory comparison for the CdTe(110) system. This scaling factor is very competitive when compared to the previously applied methods. In applying the GA with the SATLEED code we noticed that this approach is able, at least for the systems examined, to explore the parameter space in such a way as to indeed perform a real global search. In addition, the inclusion of a local refinement in the R-factor evaluation for each individual created during the GA search showed, for the SATLEED code, an order of magnitude improvement in the speed of the search, allowing therefore a more complete strucural analysis.

Acknowledgments

The authors would thank Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG), Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), and Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) (Brazilian Research Agencies) for financial support. MAVH was supported by City University of Hong Kong Grant No. 9380041.

References

[1] M A Van Hove, W H Weinberg, and C M Chan, Low Energy Electron Diffraction: Experiment, Theory and Structural Determination, Springer Series in Surface Sciences, vol. 6, Springer-Verlag (Berlin, Heidelberg, New York), 1986.

[2] David E Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Kluwer Academic Publishers, Boston, MA, USA, 1989.

[3] M L Viana, Busca Global em LEED usando Algorítmo Genético, Master Thesis, Universidade Federal de Minas Gerais, 2004.

[4] Barbieri/Van Hove SATLEED package http://www.ap.cityu.edu.hk/personal-website/Van-Hove_files/leed/leedpack.html

[5] M A Van Hove, W Moritz, H Over, P J Rous, A Wander, A Barbieri, N Materer, U Starke, and G A Somorjai, Surf. Sci. Rep. 19 (1993) 191.

[6] P J Rous, M A Van Hove, and G A Somorjai, Surf. Sci. 226 (1990) 15.

[7] E A Soares, Estudos dos Sistemas Ag(111), Ag(111)-Sb, CdTe(110) e InSb(110) via Difracão de Elétrons Lentos (LEED), PhD Thesis, Universidade Federal de Minas Gerais, 1998.

[8] V B Nascimento, Processos de Otimizacão na Análise LEED e Estudo da Estrutura dos Sistemas Ag(110) e Ag(111)-Sb, PhD Thesis, Universidade Federal de Minas Gerais, 2001.

[9] William H Press, Saul A Teukolsky, William T Vetterling, Brian P Flannery, Foreword by Michael Metcalf, Numerical Recipes in Fortran 90, Volume 2 of Fortran Numerical Recipes, 2nd edition, 1997.

[10] P J Rous, Surf. Sci. 296 (1993) 358.

[11] V B Nascimento, V E de Carvalho, C M C de Castilho, B V Costa and E A Soares, Surf. Sci. 487 (2001) 15.

[12] R Döll, and M A Van Hove, Surface Science 355 (1996) L393.

[13] M L Viana, R Diez Muiño, E A Soares, M A Van Hove, and V E de Carvalho, J. Phys.: Condens. Matter 19 (2007) 446002.

[14] D Whitley, A Review of Models for Simple Genetic Algorithms and Cellular Genetic Algorithms, Department of Computer Science, Colorado State University, Fort Collins, Colorado, 80523, USA, 1995.
[15] N P Belfiore, and A Esposito, Journal of Optimization Theory and Applications 99 (1998) 271.
[16] S Baskar, P Subbaraj, and M V C Rao, Computers and Electrical Engineering 29 (2003) 407.
[17] E A Soares, C Bittencourt, E L Lopes, V E de Carvalho, and D P Woodruff Surface Science 550 (2004) 127.
[18] E A Soares, V E de Carvalho and C M C de Castilho, Surf. Rev. Lett. 5 (1998) 241.
[19] E A Soares, V E de Carvalho and C M C de Castilho, Surf. Sci. 367 (1996) 67.
[20] E A Soares, V E de Carvalho and V B Nascimento, Surf. Sci. 431 (1999) 74.
[21] H Over, W Moritz and G. Ertl, Phys. Rev. Lett. 70 (1993) 315.
[22] W Moritz and J. Landskron, Surf. Sci. 337 (1995) 278.
[23] M A Van Hove, S Y Tong, Surface Crystallography by LEED, Springer, Berlin, 1979.
[24] M Kottke, and K Heinz, Surf. Sci. 376 (1997) 352.

Figure 1. Examples of crossover scheme in Genetic Algorithms. The strings at the left side are the parents that are mixed to create the offspring at the right side. Each parameter, coded in the strings, is indicated by a different gray level. In the present examples, the parameters are real numbers lying between 0.0 and 1.0. The first example **(A)** shows, for binary and real encodings, the case in which the break point occurs between two parameters, such that there is no change in the parameter values. The second example **(B)** presents, for binary encoding, the case in which the break occurs "inside" a parameter, giving rise to new parameter values for the offspring. The numbers above the strings are the corresponding real values for each parameter. The last example **(C)** shows an alternative way to perform crossover by performing two steps: the first selects the parameter that will be modified, keeping the others only for exchange as in the first example; the second step is a random choice of the coefficient λ that is used to create the offspring parameters by a linear combination of the parents parameters.

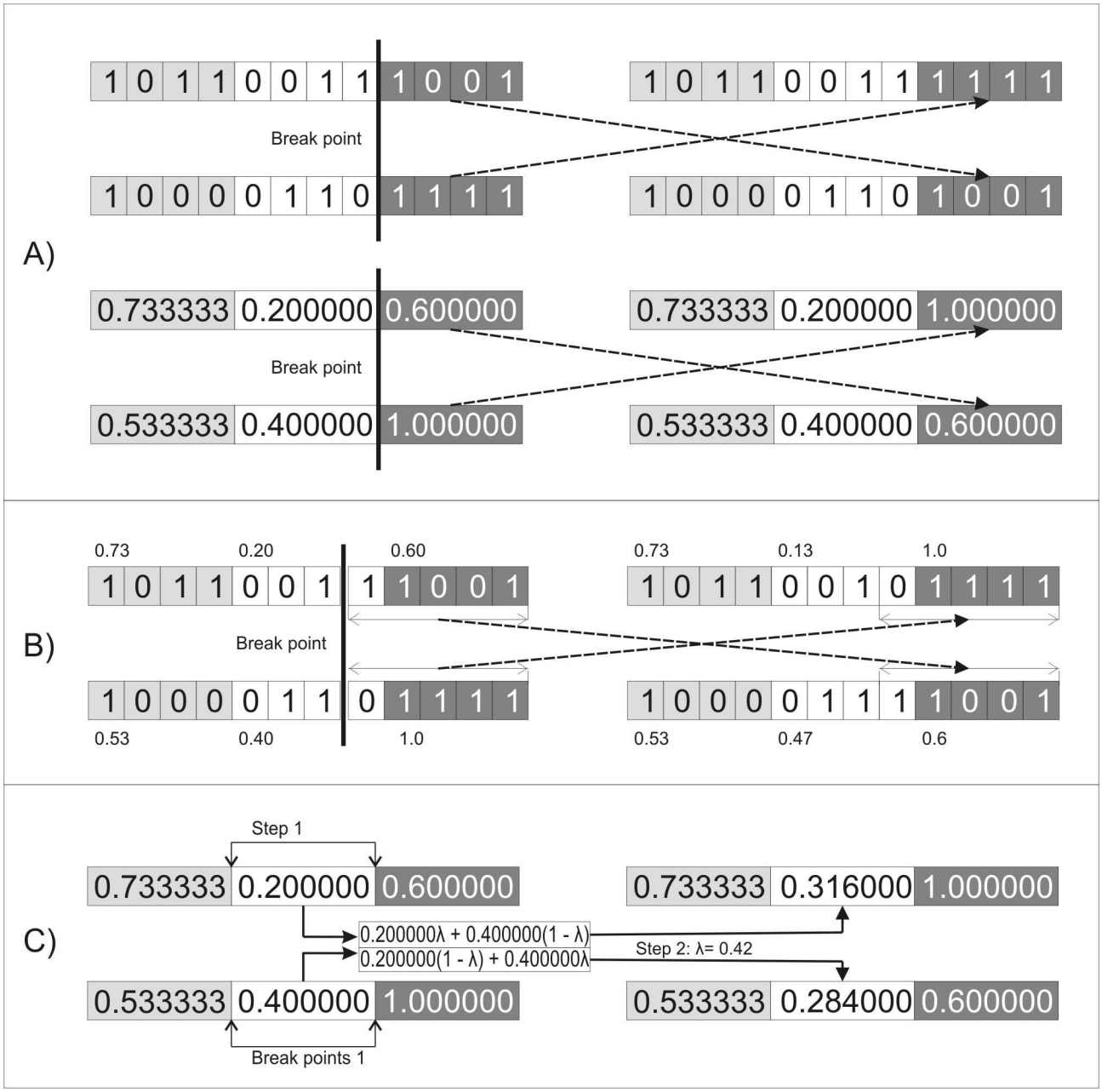

Figure 2. Schematic flowchart for the GA-SATLEED code.

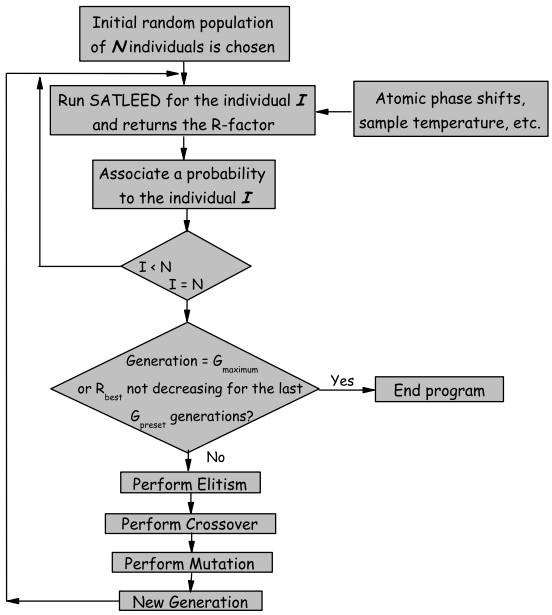

Figure 3. Top and side views of the substitutional fcc-stacking model for the $Ni(111)(\sqrt{3} \times \sqrt{3})R30^\circ - Sn$ system.

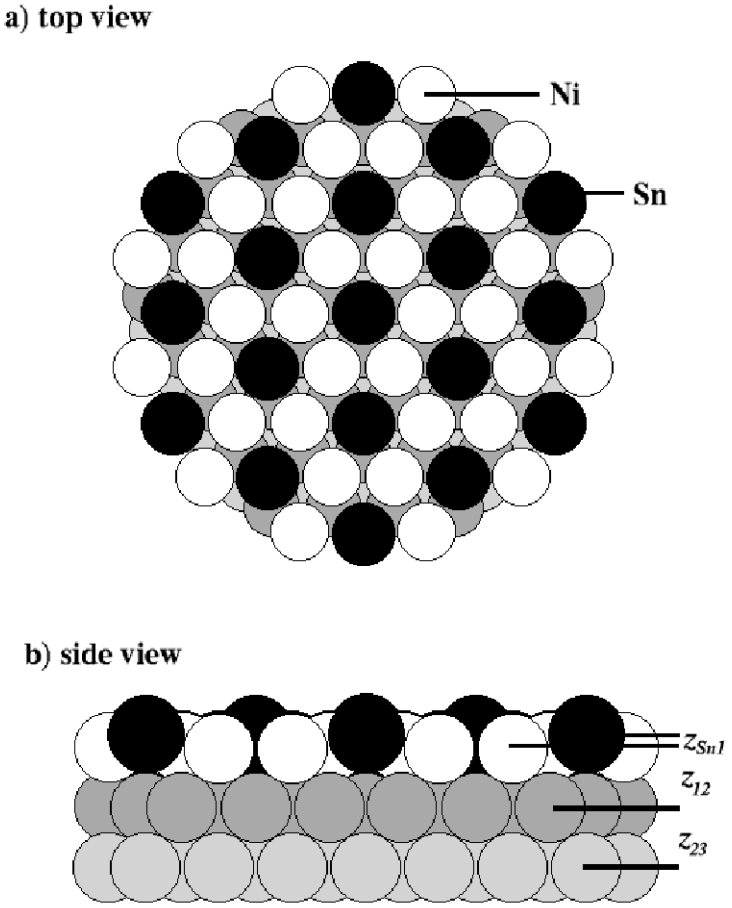

Figure 4. Scaling relationship for the genetic algorithm method applied to the multi-dimensional diffraction function of equation 1.

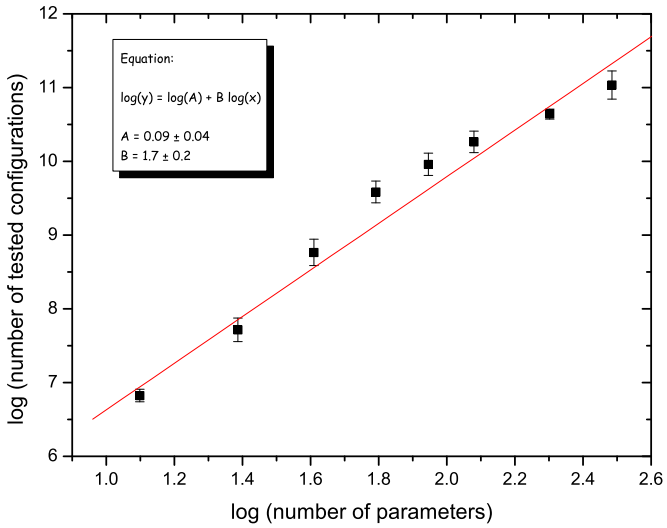

Figure 5. Scaling relationship for the genetic algorithm method applied to the surface structure detemination of the CdTe(110) system in a theory vs. theory comparison.

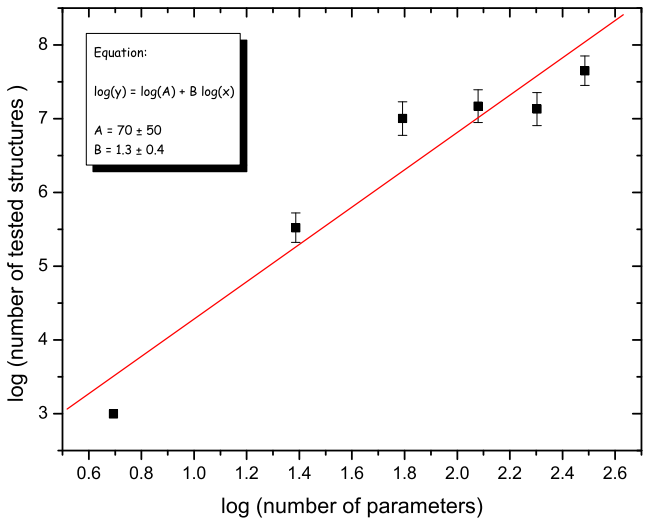

Figure 6. Scaling relationship for the simulated annealing method applied to the surface structure detemination of the CdTe(110) system in a theory vs. theory comparison [11].

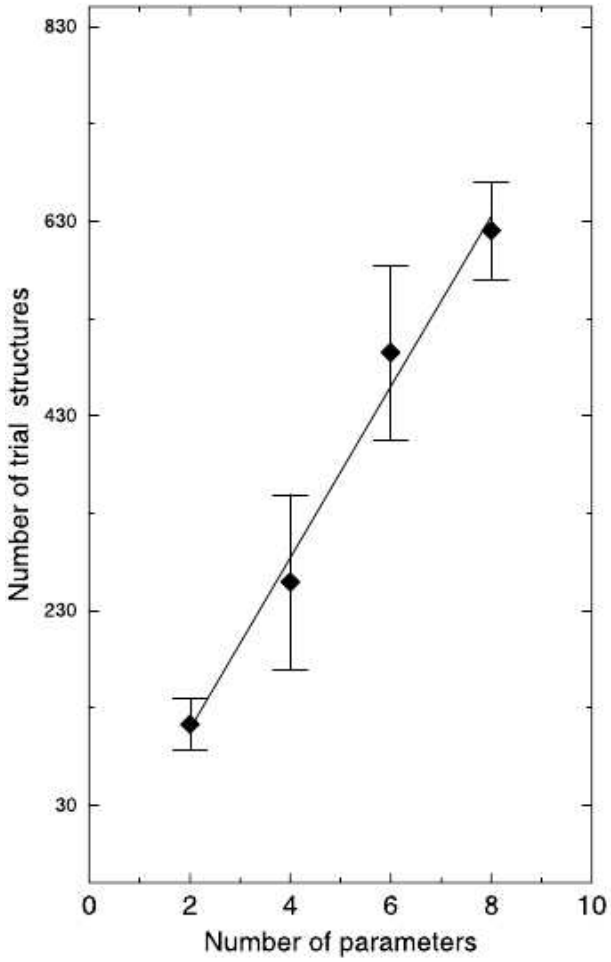

Figure 7. Evolution: R-factor for all individuals in the population for the 1^{st}, 10^{th}, 30^{th}, and 40^{th} generations.

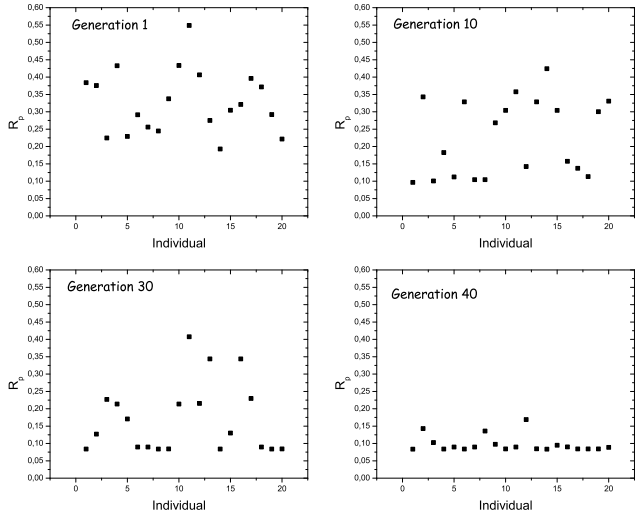

Figure 8. Scaling behavior for the GA alone, and for the GA with simultaneous local refinement applied to the $Ni(111)(\sqrt{3}\times\sqrt{3})R30^{\circ}-Sn$ system.