CMST 13(1) 13-22 (2007)

DOI:10.12921/cmst.2007.13.01.13-22

COMPUTATIONAL METHODS IN SCIENCE AND TECHNOLOGY **13(1)**, 13-22 (2007)

Optimization of Total Energy and Reduction of Computer Resources in Some Applications of the Ab-initio LMTO Package

Grzegorz Kamieniarz[1,2*]**and Paweł Sobczak**[1]

[1]*Institute of Physics, Adam Mickiewicz University, Poznań, Poland*
[2]*Max Planck Institute for the Physics of Complex System, Dresden, Germany*

**e-mail: gjk@amu.edu.pl*

(Rec. December 11, 2006)

Abstract: The computer resources needed to run the TB LMTO code have been reduced using a genetic algorithm in computations of the total energy requiring the interactive user-dependent mode. A computer program has been developed to search for the total energy minima and perform calculations in the background. The number of runs and output files is determined by the size of population and not by the number of scans of the configuration space.

Key words: computer resources optimization, LMTO ab-initio package, solid-state physics simulations

I. INTRODUCTION

The TB LMTO package [1] is extensively used in computational solid-state physics for the electronic structure determination. It consists of several modules and finds the total energy as the main result. However, sometimes it happens that the outcome of the LMTO program does not correspond to the observable physical properties so that the users are confronted with the problem of finding the global energy minimum or modifying the crystallographic unit cell. In view of the complexity of the problem, they used to proceed on the basis of trials which is not efficient and may use an excessive amount of computer memory and involve redundant computing runs.

Here we apply a genetic algorithm (GA) technique to optimize the computer resources needed to run efficiently the ab-initio calculations. The genetic algorithm is a search procedure based on the process of natural evolution, following the principles of natural selection, crossover and survival. The method has been proposed and developed by Holland [2]. Due to complex specification of initial conditions, GA is useful in searching for the optimal solution in situations where other optimization techniques may fail. A major characteristic of genetic algorithms is that they work on a population and can explore concurrently different regions of the solution space.

Genetic algorithms have successfully solved many optimization problems in business, engineering, and science [3-6], although in computational physics they have not attracted much attention.

The GA algorithm aiming at finding the minimal total energy, which depends on the atomic sphere radii, is described in the subsequent sections. There is no formal proof that the lowest energy criterion is correct in all the cases. We accept the lowest energy criterion provided that the results fulfil such constraints as the small amount of charge transfer or the similar value of the potential energy at the sphere boundary.

The structure and functionality of the LMTO simulation package, the computational optimization task of our program *ga_lmto* and some illustrative results are presented in the remaining part of the work.

II. LMTO PACKAGE STRUCTURE AND FUNCTIONALITY

The consecutive modules of the LMTO package enable finding the symmetry and volume of the crystallographic unit cell, atomic positions and atomic radii so that the volume constraints are fulfilled. When these data are obtained, the self-consistent calculations can be performed and usually the total energy is determined. The package is subject to improvement, e.g. recently the AIM (atoms-in-molecule) module is provided to integrate the charge density.

The physical properties calculated may be affected by several factors such as the exchange-correlation potential chosen, the crystal structure parameters imposed by experiment or the local energy minimum recovered by the computation approach. If the lattice structure has to be relaxed or a global minimum has to be searched, the multi-dimensional non-linear optimization problem is encountered.

Assuming that the crystal structure is correct, the total energy depends on the atomic radii. Their number N is equal to the number of non-equivalent atoms in the crystal unit cell and may reach tens for complicated structures. Searching for a minimum in such a parameter space is not trivial.

III. GENETIC ALGORITHM APROACH

In the GA optimization, any individual belonging to a given population and representing a replica of the system is defined by a set of parameters (genes in chromosomes) which express the potential solution of the problem. Each individual is evaluated according to the fitness function which is related in our problem to the total energy. The GA optimization starts with a random potential solution of the problem. The number of solutions is represented by the size of the population. Individuals with higher evaluation scores (better fitted, due to a smaller total energy value E_{tot}) are selected to generate a new population. Because of the genetic evolution, the individual (chromosome) with better fitness tends to yield a superior offspring which means it provides a better solution to the problem.

In the GA applications three genetic operations are distinguished: reproduction (selection), crossover and mutations. In the reproduction process, a new population is generated from the individuals according to value of their fitness function. After that, the crossover follows. The individuals are paired, the chromosomes are split and an exchange of genes is performed. The progeny heritages some parent genes and the new chromosomes are close to those of their predecessors. The mutation is applied seldom and changes randomly one or more genes in a given chromosome. Both crossover and mutation are subject to the sampling procedure which leads to a crossover with much higher probability than to a mutation.

This scenario follows the idea of Holland [2] and is adapted in our calculations. The convergence of GA is defined by the number of steps N_s, whereas the size of the population (the number of chromosomes) is denoted by S, and the probability of crossover and mutation by p_c and p_m, respectively.

The idea of the algorithm can be expressed by the following scheme

```
Procedure Genetic Algorithm
begin
    t : = 0
    initialization $\mathbf{P}^0$
    estimation $\mathbf{P}^0$
    while (not condition to stop) do
    begin
        $\mathbf{T}^t$ : = selection $\mathbf{P}^t$
        $\mathbf{O}^t$ : = crossover and mutation $\mathbf{T}^t$
        estimation $\mathbf{O}^t$
        $\mathbf{P}^{t+1}$ : = $\mathbf{O}^t$
        t : = t + 1
    end
end
```

In the algorithm the two populations are processed: the basis population $\mathbf{P}^t$ and the progeny population $\mathbf{O}^t$, respectively. The copies of individuals from the $\mathbf{P}^t$ population are preserved in the temporary population $\mathbf{T}^t$ during the crossover and mutation. All the populations contain the same number of individuals. In the initialization step, the sample population is selected randomly and each chromosome is evaluated using a given fitness function. The evolution is determined within the loop. Each step leads to a new generation and is considered as an iteration. Within the final generation, the chromosome which reaches the highest value of the fitness function presents the solution.

The algorithm was implemented, assuming the following convention. The chromosomes correspond to different realizations of a given physical system. Each chromosome is then determined by the set of N atomic radii and its weight is estimated according to the total energy calculated by the LMTO module. The option for the modification of the radii is switched off in the *CTRL* file. Instead, GA assigns new values of atomic radii for the next iteration. We assume that the structure is densely packed, as the performance of the LMTO program is the best then. We put physical restrictions on the size of the radii and allow modification of $N-1$ radii, while the N-th radius is chosen from the condition of volume conservation.

IV. PROGRAM DESCRIPTION AND RESULTS

In the Tight Binding Linearized Muffin Tin Orbital (TB LMTO) method within Atomic Sphere Approximation (ASA) the atomic sphere radii are chosen in such a way that the sum of all atomic spheres volumes is equal to the volume of the unit cell. The potentials at the sphere boundary for the chosen sphere radii would be similar for all types of atoms. The LMTO calculations should begin with generation of the *CTRL* file for a given chemical compound. The *CTRL* file is generated by the program *lminit.run* having introduced fundamental information on the compound (number or symbol of symmetry group, lattice constants, types of atoms and their coordinates. The exemplary description of *CTRL* file generation is given in Ref. [1]. Then, the program *lmctl.run* is run to check if the elementary cell is 100% filled in *LM* file. If the filling is less than 100%, the program *lmes.run* should be run to search for empty spheres. If the filling is correct, the program *lmstr.run* should be run. After these preliminary steps the *lm.run* subroutine can be run.

The *lm.run* subroutine of the LMTO package can be run in two main modes controlled by the value of *SCLWSR* parameter from the *SCALE* category:

1. If we assume *SCLWSR=T*, the program is looking for the total energy E_{tot} for a given compound. The total energy E_{tot} is stored in *LM* file and the optimised atomic radii are stored in a modified *CTRL* file;
2. If we assume *SCLWSR=F*, then we can calculate the total energy E_{tot} for the atomic radii chosen. The atomic radii values are introduced into the *CTRL* file, and the energy value for the atomic radii assumed is written by the program for the *LM* file.

In the *ga_lmto* program the parameter *SCLWSR=F*, so that the genetic algorithm is used to carry out the optimisation of E_{tot}. The data file for the *ga_lmto* program is named *INPUT* and has the following structure of records:

a. Number of iterations N_s being an integer variable in the 9d format,
b. Number of chromosomes *S*, being an integer variable in the 6d format,
c. Probability of crossover p_c, being a real variable in the 1.6f format from the range (0,1>,
d. Probability of mutation p_m, being a real variable in the 1.6f format from the range (0, 0.2>,
e. Number of inequivalent atoms in the compound studied N_r (the number of variables in the problem), being an integer variable in the 2d format,
f. Number of atoms in the elementary cell N_a, being an integer variable in 3d format,
g. Numbers of atoms distinguished in the elementary cell N_{an}, described by integer variables in the 1d format,
h. Volume of the elementary cell of a given compound *V*, being a real variable in the 8.8f format,
i. Constant *C* – being a real variable in the 8.6f format protecting against negative values of the total energy E_{tot} (the problem of E_{tot} minimisation should be replaced by that of E_{tot} maximisation),
j. Sequences containing 2 constants being real variables, defining the lower and upper bound of the radii searched in the 2.8f format. It is essential that the upper and lower bounds to the radius that ensures correct filling of the elementary cell were in the last line of the *INPUT* file. The genetic algorithm generates the radii of atomic spheres and searches the minimum value of E_{tot} in the ranges specified by the sequences. The exemplary *INPUT* file is given in Annex.

Correct execution of the program *ga_lmto* requires that the variable *PATH* is supplemented by a directory storing the *lm.run* subroutine. Let's assume that the program *ga_lmto* is localised in the working directory of the user, along with the files *INPUT* and *CTRL* and the file *OUTPUT* to be generated.

We also assume that the user has already generated in a standard way the *CTRL* file containing the physical parameters needed for solving the problem by the *lm.run* subroutine. The number of different atomic radii N_r is equal the number of genes of a given chromosome. The initial values of genes of each chromosome are generated at random from the range given in the *INPUT* file of *ga_lmto*.

The program can work in the background or the effects of its work can be observed on the screen. After the program completion the report with results is prepared in the *OUTPUT* file. The report gives the following information: the optimum value of the estimation function in a given population, the mean value of the estimation function in a given population, standard deviation, atomic radii and the minimum total energy E_{tot}.

Generation of subsequent generations of individuals follows according to the procedure of Genetic Algorithm. Our chromosome contains N_r inequivalent atomic radii: $r_1, r_2, \ldots r_{N_r}$. For each individual the *lm.run* subroutine calculating the energy value is called. The usefulness of a given individual in a generation is measured by its total energy value (the value of the function of purpose). The chromosomes of lower total energy have higher values of the fitness function. The function of purpose (estimation) determines the total energy of the individual E_{tot}, while the fitness function is supplemented with a constant *C*, replacing the problem of searching for a minimum by the problem of searching for a maximum. The next steps are: proportional selection, single-point crossing and uniform mutation.

The proportional selection can be illustrated in the following way. Let's make a virtual roulette and divide it into as many sectors as many individuals there are in a given population. Let's assign each sector of the roulette with one individual and let's make the are of the sector proportional to the fitness function of a given individual. Each chromosome from a given population is assigned to a subsection of the section [0, 1) (corresponding to a sector of the roulette wheel). For a population of the size *S* composed of the chromosomes c_i (i = 1, ..., S), the process of selection is the following:

1. If $f(c_i)$ is a fitness function of the ith chromosomes, the limits of the ith sections can be found in the way described below. Let's introduce the quantities:

$$F = \sum_{i=1}^{S} f(c_i), \quad p_i = \frac{f(c_i)}{F},$$

$$q_i = \sum_{j=1}^{i} p_j.$$

where the subsection assigned to a chromosome c_1 is $[0, q_1)$, and the subsections for chromosomes c_i (i = 2, ..., S) are $[q_{i-1}, q_i)$.

2. Any number drawn from the interval [0, 1) belongs to one of the above defined subsections and is equivalent to the choice of the chromosome assigned to it.
3. The numbers are drawn as many times as many chromosomes there are in a given population.

As a result of selection we have obtained the temporary population which is subjected to crossing.

The one-point crossing operation is realised in two steps:

step 1 – chromosomes from the parent pool are paired at random,

step 2 – each pair is subjected to crossing.

The process of crossing is random. When the decision on the crossing of a given pair of chromosomes is positive, a number of a gene k, that divides the chromosome into two parts, is drawn. The drawn gene of the number k denotes the point of the chromosomes crossing. All the genes of the numbers from k + 1 to N_r including the two limiting ones are exchanged in the pair and in this way two new chromosomes are obtained. When the decision on the crossing is negative, the chromosomes are left unchanged.

The operation of mutation is performed at random for each gene separately, which may lead to a change in one or more genes in a given chromosome. The aim of mutation is to cause a sporadic and random change in the value of genes in a chromosome. The GA System is based on flow-point coding.

It should be emphasised that Our problem is not an ordinary mathematical problem of optimisation, as apart from searching for a minimum certain physical requirements must be preserved such as the overlapping of spheres, filling of the elementary cell and admissible values of the atomic radii.

The results are illustrated for the alloy of nickel, manganese and germanium Ni_2MnGa, whose elementary cell L12 is composed of 4 fcc sublattices [7]. There are two sublattices populated by Ni (the same atoms), one fcc sublattice populated by Mn and one populated by Ga, and the elementary cell contains 4 atoms. For the calculations to have physical sense the sum of atomic spheres of these 4 atoms must be equal to the elementary cell volume. The program *ga_lmto* changed the values of 3 atomic radii in the elementary cell that is 2 radii of Ni atom and one radius of Mn atom, while the radius of Ga was adjusted so that the elementary cell filling would be 100% when the elementary cell volume was disturbed by the crossing and mutation. It was assumed that the radii could change up to ±5% of the expected value and in this range AG generated the radii. When the atomic radius of Ga exceeded the assumed range, the solution was treated as unphysical and the individual was punished by a zero fitness function. The individual with the zero fitness function has no chances of getting into the next population.

The performance of the *ga_lmto* program is illustrated for Ni_2MnGa. At first the *lm.run* subroutine from the LMTO package is run with *SCLWSR=T* from the SCALE category, which automatically searches for the atomic radii that correspond to full filling of the unit cell. The *NKABC* parameters from the Bz category were taken as 20 20 20. The *CTRL* file generated in on initiation of the *lm.run* subroutine, used as the basis of the calculations is given in Annex. The result obtained in the *LM* file was: the total energy E_{tot} = `-12270.81351425` to the accuracy of DE_{tot} = `0.00000209;` E_{tot} is given in Rydbergs. The details essential for the calculations whose results are given in the *LM* file are displayed in Table 1.

The lengths of the atomic radii by the *lm.run* subroutine taken from the modified *CTRL* file expressed in [a.u.] are:

$$r_{Ni} = 2.65529373$$

$$r_{Mn} = 2.67611954$$

$$r_{Ga} = 2.86355211$$

The *lm.run* program was run once again with *SCLWSR* = *T* for the modified value of *NKABC* = 100 100 100, to verify the total energy and atomic radii obtained. The result of the repeated calculations was E_{tot} = `-12270.80302505` to the accuracy DE_{tot} = `0.00000019`. Important fragments of the resulting LM file are presented in Table 2.

The total energy E_{tot} (ETOT variable) for *NKABC* = 100 100 100 is consistent with its values for *NKABC* = 20 20 20. The atomic radii calculated by the *lm.run* subroutine are the same as those found for *NKABC* = 20 20 20.

The calculations in the *ga_lmto* program were made assuming that the radii calculated earlier could differ from

Table 1. Fragment of the *LM* file generated by the *lm.run* subroutine
for the parameters *SCLWSR =T* and *NKABC* = 20 20 20

```
      VOL=   335.476     VOLSPH=   335.476     VOLSPH/VOL=100.0%

      ATOMSC: CLASS          QTOT          VMAD         VHART(WSR)      V(WSR)
                Ni         0.1039116    -0.076699      0.001568    -0.766140
                Mn        -0.3324050     0.221507     -0.026916    -0.791968
                Ga         0.1245818    -0.068109      0.018903    -0.741130

   ITER 17 OUT OF 30:   MAG MOM= -3.93258525   ETOT=  -12270.81351425
                          RMS DQ=  0.00000343  DETOT=       0.00000209
```

Table 2. Fragment of the *LM* file generated
by the *lm.run* subroutine for *SCLWS R=T* and *NKABC* = 100 100 100

```
      VOL=   335.476     VOLSPH=   335.476     VOLSPH/VOL=100.0%

      ATOMSC: CLASS          QTOT          VMAD         VHART(WSR)      V(WSR)
                Ni         0.1038710    -0.076669      0.001567    -0.766142
                Mn        -0.3325990     0.221626     -0.026942    -0.791985
                Ga         0.1248570    -0.068287      0.018917    -0.741137

      ITER  8 OUT OF 30:   MAG MOM= -3.95158783   ETOT=  -12270.80302505
                             RMS DQ=  0.00000589  DETOT=       0.00000019
```

Table 3. Fragment of the *LM* file important for the solution found by *ga_lmto* assuming admissible fluctuations of ± 2.5%.
In the *CTRL* file *SCLWSR =F* and *NKABC* = 20 20 20

```
      SCLWSR: cell volume=   335.47608  sum of sphere volumes=   335.47608

      ATOMSC: CLASS          QTOT          VMAD         VHART(WSR)      V(WSR)
                Ni         0.1835517    -0.135483      0.001331    -0.764627
                Mn        -0.2911086     0.203648     -0.011973    -0.776317
                Ga        -0.0759949     0.067319      0.012882    -0.749385

      ITER 14 OUT OF 30:   MAG MOM= -3.94199843   ETOT=  -12270.82143051
                             RMS DQ=  0.00000042  DETOT=       0.00000221
```

Table 4. Fragment of the *LM* file important for the solution found by *ga_lmto* assuming the admissible fluctuations of ± 5%.
In the *CTRL* file *SCLWSR = F* and *NKABC* = 20 20 20

```
    SCLWSR: cell volume=   335.47608  sum of sphere volumes=   335.47608

    ATOMSC: CLASS          QTOT          VMAD         VHART(WSR)      V(WSR)
              Ni         0.3253461    -0.240145     -0.001891    -0.764979
              Mn        -0.4356484     0.310049     -0.018707    -0.784696
              Ga        -0.2150439     0.170241      0.013682    -0.750563

    ITER 16 OUT OF 30:   MAG MOM= -3.93774602   ETOT=  -12270.82463482
                           RMS DQ=  0.00000074  DETOT=       0.00000085
```

the expected value by ±2.5% and ±5%. Assuming the admissible fluctuations of ±2.5%, the lower and upper bounds to the atomic radii in the *INPUT* file were:

r_{Ni} :	2.58891138-2.72167607
r_{Mn} :	2.60921655-2.74302252
r_{Ga} :	2.79196330-2.93514091

The program was run using the *CTRL* files shown in the Annex (assuming *SCLWSR* = *F*) and *INPUT*. The results obtained by *ga_lmto* in the *OUTPUT* file were E_{tot} = `-12270.82143051 and` DE_{tot} = `0.00000221`, with the atomic radii of :

$$r_{Ni} = 2.68321626$$

$$r_{Mn} = 2.70019461$$

$$r_{Ga} = 2.79204054$$

The fragment of the *LM*, important for the calculation of the solution found by *ga_lmto* is given in Table 3.

Increasing the admissible fluctuation to ± 5% the lower and upper bounds for atomic radii in the *INPUT* file were:

$$r_{Ni} : \ 2.52252904\text{-}2.78805841$$

$$r_{Mn} : \ 2.54231356\text{-}2.80992551$$

$$r_{Ga} : \ 2.72037450\text{-}3.00672971$$

After the calculations by the genetic algorithm the *OUTPUT* file gave: E_{tot} = `-12270.82463482` and DE_{tot} = `0.00000085`, while the atomic radii were:

$$r_{Ni} = 2.73108819$$

$$r_{Mn} = 2.65028068$$

$$r_{Ga} = 2.74713582$$

The information from the LM file important for the solution found by *ga_lmto* are given in Table 4.

The values of E_{tot} obtained by the *ga_lmto* program for Ni_2MnGa (Table 3 and 4) are in perfect agreement with the correct ones found by the *lm.run* subroutine from the standard LMTO package (Table 1 and 2). It is expected that the *ga_lmto* program will be useful when the standard *lm.run* program does not give a correct solution. It should be emphasised that in the calculations with *ga_lmto* the potential at the border of the spheres V(WSR) takes comparable values for all atoms presented and these values do not differ by more 10%. The potential VHART(WSR) does not exceed 0.05, so the conditions related to the admissible fluctuations of the potential at the border of the sphere are satisfied. The calculations performed by *ga_lmto* indicate that for a densely packed Ni_2MnGa, the genetic algorithm correctly finds the minimum total energy.

In the optimizing program searching for the minimum value of E_{tot} for at least a few inequivalent atomic radii, which requires assuming *SCLWSR = F* the *ga_lmto* program reduces the memory and time requirements of computer work. For an arbitrary accuracy of the *lm.run* subroutine for a given number of individuals *S*, the *ga_lmto* program generates only *S LM* files. The standard running of the *lm.run* subroutine with *SCLWSR = F*, would be much more expensive in terms of computer memory and time required for the analysis of the results.

The use of a genetic algorithm means that much less time is needed to get results. The time complexity of the standard LMTO calculations described by the number of runs of the *lm.run* subroutine is expressed by a polynomial taking into regard the number of points in the ranges r_{min} and r_{max} for each radius of the atomic sphere and increases with increasing number of inequivalent N_r radii.

The time complexity of the *ga_lmto* program is expressed by the size of the population *S* and the number of iterations N_s. These quantities somehow depend on N_r. The character of $S(N_r)$ and $N_s(N_r)$ relations is not known but we expect at most a linear dependence which implies that the complexity is expressed at most by a square function of N_r.

For the exemplary compound Ni_2MnGa the *ga_lmto* program called *lm.run* 20 × 500 = 10 000 times. Changing the radius at a step of 0.01 within the range of ±5 % of the expected value in the genetic algorithm, in the standard approach the *lm.run* subroutine should be called 23 000 times. Each run of the *lm.run* subroutine would generate an output file *LM* and would require storage of the *CTRL* file.

V. DISCUSSION AND CONCLUSIONS

There is a number of problems which have to be addressed when processing the complex computing packages: the structure, interconnections, functionalities and constraints which have to be considered interactively by the user. In our calculations we were confronted with the constraint on the 5-10% overlap of atomic spheres and complete filling of the unit cell. As it can be handled for the densely packed systems, we have excluded the loosely packed ones, which does not seem a serious limitation as the application of the LMTO is not recommended for them.

The problem of efficient management of the available computer resources performing computations with the LMTO package is important when the global total energy of a physical system has to be found using the *SCLWSR = F* calculation mode. As the configuration space is probed automatically by the GA program, the interactive part of the LMTO calculations is reduced to minimum. The user does not have to store large intermediate files in order to analyse them and the demand on the memory does not depend on the number of parameters determining the physical problem (i.e. the number of atomic radii).

The program prepared by us can be successfully used for other compounds of densely packed structure. After modifications the program can be applied for other calculation/computation packages for solving physical problems. The program can be easily generalised to tackle the problem of structure optimisation. Then the lattice constants would play the role of chromosomes.

In conclusion, we have worked out the program which finds the total energy of an electronic system by using the LMTO subroutines *lm.run* with the atomic radii found by the genetic algorithm. The LMTO programs are performed in the background and the disc space is reduced to that needed to store *S* output files *LM* irrespective of the number of iterations. Results obtained by *ga_lmto* have been illustrated for the ordered alloy Ni_2MnGa.

At present only the compiled version of *ga_lmto* is available for registered users of the LMTO package on request addressed to gjk@amu.edu.pl.

Acknowledgements

The authors are grateful to Prof. A. Jezierski for fruitful and clarifying discussions related to the LMTO package. This work was supported by the Polish Ministry of Science and Education grant 4T11F 014 24 and the EU NoE project NMP3-CT-2005-515767. Computations were partially performed on the PCSN supercomputing platforms.

References

[1] G. Krier, O. Jepsen, A. Burkhardt and O. K. Andersen, *The TB – LMTO – ASA program,* Max – Planck – Institut für Festkörperforschung Heisenbergstr. 1, D-70569 Stuttgart, Federal Republic of Germany.

[2] J. H. Holland, *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*, University of Michigan Press, Ann Arbor 1975.

[3] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Publishing Company, Inc., Massachusetts, 1989.

[4] D. E. Goldberg, *The Design of Innovation: Lessons from and for Competent Genetic Algorithms*, Kluwer Academic Publishers, 2002.

[5] C. W. Ahn and R. S. Ramakrishna, *A Genetic Algorithm for Shortest Path Routing Problem and the Sizing of Populations*, IEEE Transactions on Evolutionary Computation **6(6)** (2002).

[6] R. Wyrzykowski, J. Dongarra, M. Paprzycki and J. Waśniewski, (Eds.), *Parallel Processing and Applied Mathematics*, 5th International Conference, PPAM 2003 Częstochowa, Poland, September 2003 Revised Papers, Springer LNCS 3019.

[7] A. Ayuela, J. Enkovaara and R. M. Nieminen, *Ab initio study of tetragonal variants in Ni2MnGa alloy*, J. Phys.: Condens. Matter **14**, 5325-5336 (2002).

ANNEX

1) *INPUT* file

```
500          # number of iterations (Ns)
20           # number of chromosomes (S)
0.7          # probability of crossover (pc)
0.1          # probability of mutation (pm)
3            # number of inequivalent atoms in a compound (Nr)
4            # number of atoms in the elementary cell (Na)
2 1 1        # number of atoms distinguished in the elementary cell (Nan)
335.47608    # volume of the elementary cell of a given compound (V)
–12270.5      # constant C
2.58891138   2.72167607 # The sequences containing two constants describing the upper and lower bounds of the atomic radii searched
2.60921655   2.74302252 # The upper and lower bounds ensuring the maximum packing of the elementary cell for a given compound
2.79196330   2.93514091 # should be in the last line
```

2) *CTRL* file

```
HEADER    MnGaNi2, cubic face-centred
VERS      LMASA-48
IO        VERBOS=30 HELP=F WKP=F IACTIV=F ERRTOL=2 OUTPUT=LM ERR=ERR
SYMGRP    NGEN=3 GENGRP=I R4X R3D
          SPCGRP=Fm-3m USESYM=F
STRUC     ALAT=11.02995784
          PLAT=0.0 0.5 0.5
```

```
               0.5 0.0 0.5
               0.5 0.5 0.0 FIXLAT=T
DIM       NBAS=4 NCLASS=3 NL=3 LDIM=31 IDIM=5 NSYMOP=48 NKP=72
OPTIONS   NSPIN=2 REL=T CCOR=T NONLOC=F NRXC=1 NRMIX=2 CORDRD=F
          NITATOM=30 CHARGE=F FATBAND=F AFM=F SEWALD=F FS=F
          CARTESIAN=T WRIBAS=F Q=---- SPORB=F
CLASS     ATOM=Ni Z=28 R=2.65529373 LMX=2 CONF=4 4 3 4 IDXDN=1 1 1
          ATOM=Mn Z=25 R=2.67611954 LMX=2 CONF=4 4 3 4 IDXDN=1 1 1
          ATOM=Ga Z=31 R=2.86355211 LMX=2 CONF=4 4 4 4 IDXDN=1 1 2
SITE      ATOM=Ni POS=0.25 0.25 0.25
          ATOM=Ni POS=-.25 -.25 -.25
          ATOM=Mn POS=0.50 0.50 0.50
          ATOM=Ga POS=0.00 0.00 0.00
SCALE     SCLWSR=T OMMAX1=.16 .18 .20 OMMAX2=.40 .45 .50
STR       KAPPA2=0 RMAXS=3.2 NDIMIN=350 NOCALC=F IALPHA=0
          DOWATS=F DELTR=.1 LMAXW=8
          ATOM=Ni SIGMA=.7 .7 .7
          ATOM=Mn SIGMA=.7 .7 .7
          ATOM=Ga SIGMA=.7 .7 .7
START     NIT=30 BROY=T WC=-1 NMIX=1 BETA=.1
          FREE=F CNVG=.00001 CNVGET=.00001 BEGMOM=T CNTROL=T
          EFERMI=-.06730825 VMTZ=-.76544744
          ATOM=Ni  P=4.67604363 4.41153865 3.86455990
                     4.69502044 4.42825917 3.84142696
                   Q=0.30603564 -.00802874 0.01028102
                     0.36510325 0.00015225 0.00972494
                     4.53792835 0.01804703 0.03172471
                     0.32428069 -.00709734 0.01238225
                     0.37699778 0.00272184 0.01146309
                     4.19358267 0.01651295 0.02292684
              enu   =-.50003776 -.34800839 -.22105072
                     -.46591699 -.31616934 -.21132180
              c     =-.35680116 0.65478577 -.19994624
                     -.35598888 0.65368140 -.17617785
              sqrdel=-.41544227 0.39368209 0.09970383
                     -.41573740 0.39315876 0.10147370
              p     =0.04765721 0.02555325 2.12240320
                     0.04850835 0.02594122 2.03685561
              gamma =0.41626973 0.10581695 -.00300931
                     0.41555995 0.10550190 -.00246489
          ATOM=Mn  P=4.66009604 4.39212609 3.86714273
                     4.62260248 4.36858380 3.55234178
                   Q=0.28791846 -.00188771 0.00868114
                     0.33008388 0.00604774 0.00817623
                     4.38855752 0.02694687 0.04427164
                     0.25881041 -.00204887 0.00707852
                     0.31983505 0.00545213 0.00808377
                     1.08277214 0.00623541 0.01015328
              enu   =-.45966740 -.33145793 -.22878788
                     -.44617752 -.31233374 -.22860187
              c     =-.27843473 0.74316694 -.20170550
                     -.19584083 0.83673808 0.01980990
              sqrdel=-.43153694 0.40867237 0.11985275
                     -.43706105 0.41464391 0.14167800
              p     =0.03715848 0.02324053 1.45307553
```

```
                     0.03386285 0.02232289 0.87705329
              gamma =0.42494169 0.11150518 -.00065781
                     0.42708565 0.11285212 0.00975284
          ATOM=Ga  P=4.80624268 4.58006095 4.18541289
                     4.80562811 4.60153024 4.19078288
                   Q=0.57141506 -.01563231 0.01226015
                     0.76228843 -.01974795 0.01153425
                     0.19080617 0.00000001 0.00199534
                     0.57620670 -.01420854 0.00891854
                     0.82994863 -.02116259 0.01119650
                     0.19350053 0.00000000 0.00171179
              enu   =-.61596575 -.34914323 -.26912873
                     -.62254791 -.32792381 -.24001235
              c     =-.70997643 0.15686909 2.61202651
                     -.71545062 0.14987477 2.44683235
              sqrdel=-.35157627 0.36357743 -.65223964
                     -.35168956 0.36267718 -.61636468
              p     =0.15388682 0.06761374 0.23716790
                     0.15340226 0.06924184 0.21283874
              gamma =0.42540057 0.12028863 0.13374068
                     0.42551878 0.11958232 0.12572305
CHARGE    LMTODAT=T ELF=F ADDCOR=F SPINDENS=F CHARWIN=F EMIN=-2 EMAX=2
PLOT      ORIGIN=0 0 0
              R1=1 0 0 NDELR1=12
              R2=0 1 0 NDELR2=12
              R3=0 0 1 NDELR3=12
          FORMAT=1
BZ        NKABC=20 20 20 TETRA=T METAL=T TOL=.000001
          N=0 W=.005 RANGE=5 NPTS=1001
EWALD     NKDMX=250 AS=2 TOL=.000001
RHOFIT    FIT=F KAPPA2=0 RMAXS=3.5
          ATOM=Ni LMXRHO=2 SIGMA=.7 .7 .7
          ATOM=Mn LMXRHO=2 SIGMA=.7 .7 .7
          ATOM=Ga LMXRHO=2 SIGMA=.7 .7 .7
SCELL     PLAT=0.0 0.5 0.5
               0.5 0.0 0.5
               0.5 0.5 0.0 EQUIV=T
HARTREE   BEGATOM=T LT1=2 LT2=2 LT3=2
DOS       NOPTS=801 EMIN=-.9 EMAX=2.7
SYML      NQ=35 Q1=0.50 0.50 0.50 LAB1=L
                Q2=0.00 0.00 0.00 LAB2=G
          NQ=40 Q1=0.00 0.00 0.00 LAB1=G
                Q2=0.00 1.00 0.00 LAB2=X
          NQ=20 Q1=0.00 1.00 0.00 LAB1=X
                Q2=0.50 1.00 0.00 LAB2=W
          NQ=30 Q1=0.50 1.00 0.00 LAB1=W
                Q2=0.50 0.50 0.50 LAB2=L
          NQ=25 Q1=0.50 0.50 0.50 LAB1=L
                Q2=0.00 0.75 0.75 LAB2=K
          NQ=45 Q1=0.00 0.75 0.75 LAB1=K
                Q2=0.00 0.00 0.00 LAB2=G
FINDES    RMINES=.9 RMAXES=4 NRXYZ=48 48 48
```

GRZEGORZ KAMIENIARZ is a professor in theoretical and computational physics at the Institute of Physics of the Adam Mickiewicz University in Poznań. His field of interest covers statistical mechanics, phase transitions, phenomenological modeling and simulations of the low- dimensional magnetic systems including molecular nanomagnets.

PAWEŁ SOBCZAK received MSc degree in physics and computer science applications, at the Adam Mickiewicz University, Faculty of Physics Poznań (2005). He is a PhD student at the same University and the same Faculty (2005). His field of interest covers genetic algorithms, molecular magnets, DMRG technique and computer simulation.