

TESLA REPORT 2006-02
TESLA THESIS 2006-003



WARSAW UNIVERSITY OF TECHNOLOGY
Faculty of Electronics and Information Technology
Institute of Electronic Systems

Krzysztof Kamil Korzunowicz
185691

Bachelor of Science Thesis
Application of a Genetic Algorithm to unfold
Thermoluminescence Dosimeter (TLD) Glow Curves
for use during radiation measurements for X-FEL
experiment in DESY, Hamburg

Mentors:
D. Sc. Ryszard Romaniuk
Ph. D. Krzysztof Poźniak
Ph. D. Bhaskar Mukherjee

Tutor:
M. Sc. Dominik Rybka

Warsaw, February 2006

Application of a Genetic Algorithm to unfold Thermoluminescence Dosimeter (TLD) Glow Curves for use during radiation measurements for X-FEL experiment in DESY, Hamburg

This thesis is a part of an effort to broaden the possibilities of measuring radiation doses in a mixed field radiation when using Thermoluminescent Dosimeters (TLD). The topic of study is Glow Curve of TLD that are sensitive to both neutrons and gammas. The Glow Curve has many overlapping peaks present, described by TL kinetics models. Since deconvolution can be described as a multidimensional search, the idea is to use a Genetic Algorithm to try to find out how each peak contributes to the whole curve.

Main use of this research effects is to easily, quickly and cheaply evaluate the doses of neutrons and gammas in the Tesla Test Facility (TTF) tunnel housing the Vacuum Ultra-Violet Free-Electron Laser (VUV-FEL). Since the technology created and refined for VUV-FEL will be used in X-Ray Free-Electron Laser (X-FEL), the data obtained with this thesis' effects can be later used for estimation of expected radiation levels of X-FEL. It can also be used for estimation of radiation effects on electronics housed in the tunnel. Based on those estimations radiation shielding and mitigation techniques for electronics will be chosen. They can also be tested with fruits of this thesis.

Acknowledgements

The author wishes to thank everyone involved in this thesis. Especially: D. Sc Ryszard Romaniuk, Ph. D. Krzysztof Poźniak, Dominik Rybka from Warsaw University of Technology and Ph. D Bhaskar Mukherjee, Ph.D. Stefan Simrock from Deutches Electronen Synchrotron (Hamburg) for their supervision and ideas.

Table of contents

Abstract.....	2
Acknowledgements.....	3
Table of contents.....	4
1. Introduction.....	5
1.1 Particle physics.....	5
1.1.1 X-FEL experiment.....	5
1.2 Radiation effects on electronic components.....	7
1.3 Thermoluminescence Dosimetry (TLD).....	7
1.4 Genetic Algorithms.....	8
2. Thesis' Objectives.....	10
3. Concept.....	12
3.1 Programming environment.....	13
3.2 Genetic Algorithm considerations.....	13
3.2.1 Data representation.....	13
3.2.2 Fitness function.....	15
3.2.3 Starting conditions.....	16
3.2.4 Mutation.....	16
3.2.5 Mutation as a variable.....	17
3.2.6 Crossover.....	18
3.2.7 Selection.....	18
4. Implementation.....	19
4.1 Files.....	20
4.2 Input and output data, program variables.....	20
4.3 Genetic Algorithm operations.....	22
4.4 Peak area computing.....	23
5. Instruction manual.....	24
6. Testing.....	28
6.1 Test 1: Shielding analysis.....	28
6.2 Test 2: Gamma responses of TLD.....	32
6.3 Convergence test.....	35
7. Summary and conclusions.....	39
7.1 Broader perspective.....	40
8. List of Figures.....	41
9. List of Tables.....	41
10. References.....	42
Appendix A: Example input files for the deconvolution program.....	44

1. Introduction

In the following chapters there will be a brief explanation of topics connected to this Thesis. The starting point will be where radiation measurements are needed, why they are important, then what effects radiation causes in electronics. Next topics will cover basics of TL dosimetry and a short description of Genetic Algorithms.

1.1 Particle physics

Particle physics is a branch of physics that studies the elementary constituents of matter and radiation, and the interactions between them. Modern particle physics research is focused on subatomic particles, which have less structure than atoms. These include atomic constituents such as electrons, protons, and neutrons (protons and neutrons are actually composite particles, made up of quarks), particles produced by radiative and scattering processes, such as photons, neutrinos, and muons, as well as a wide range of exotic particles. [1] It is also called High Energy Physics (HEP), because many elementary particles do not occur under normal circumstances in nature, but can be created and detected during energetic collisions of other particles, as is done in particle accelerators [1]. But that is not the only use of accelerators.

1.1.1 X-FEL experiment

The X-ray free-electron laser (X-FEL) is linear accelerator, which is being developed at the DESY (Deutsches Elektronen-Synchrotron) research center in Hamburg, in cooperation with European partners. The facility is scheduled to start operation in 2012.

This facility will produce high-intensity ultra-short X-ray flashes with the properties of laser light. This new light source will open up a whole range of new perspectives for the natural sciences. The X-ray laser opens up unsuspected perspectives for physics, chemistry, materials science and geophysical research, biological sciences and medicine. The radiation is also of considerable interest to industrial users [2].

X-FEL will be an innovatory project with only a single tunnel instead of two: one for accelerator and one for the electronics. Thus all of electronic components will be placed in the same tunnel as accelerator tube. Normally it is much cheaper and simpler to put the electronics out of the radiation field. But due to the time constraints of the system used in X-FEL, some electronics (front-end) has to be as close as possible to the accelerator (and also radiation). Also the rest of the electronics will be housed in the same tunnel as the machine (see Figure 1), due to costs of digging of another tunnel. Thus it will be also susceptible to radiation, just from a bigger distance. In order to minimize the possibility of radiation effects occurring, the electronics will be shielded. The effectiveness of the shielding has to be correlated with the doses.

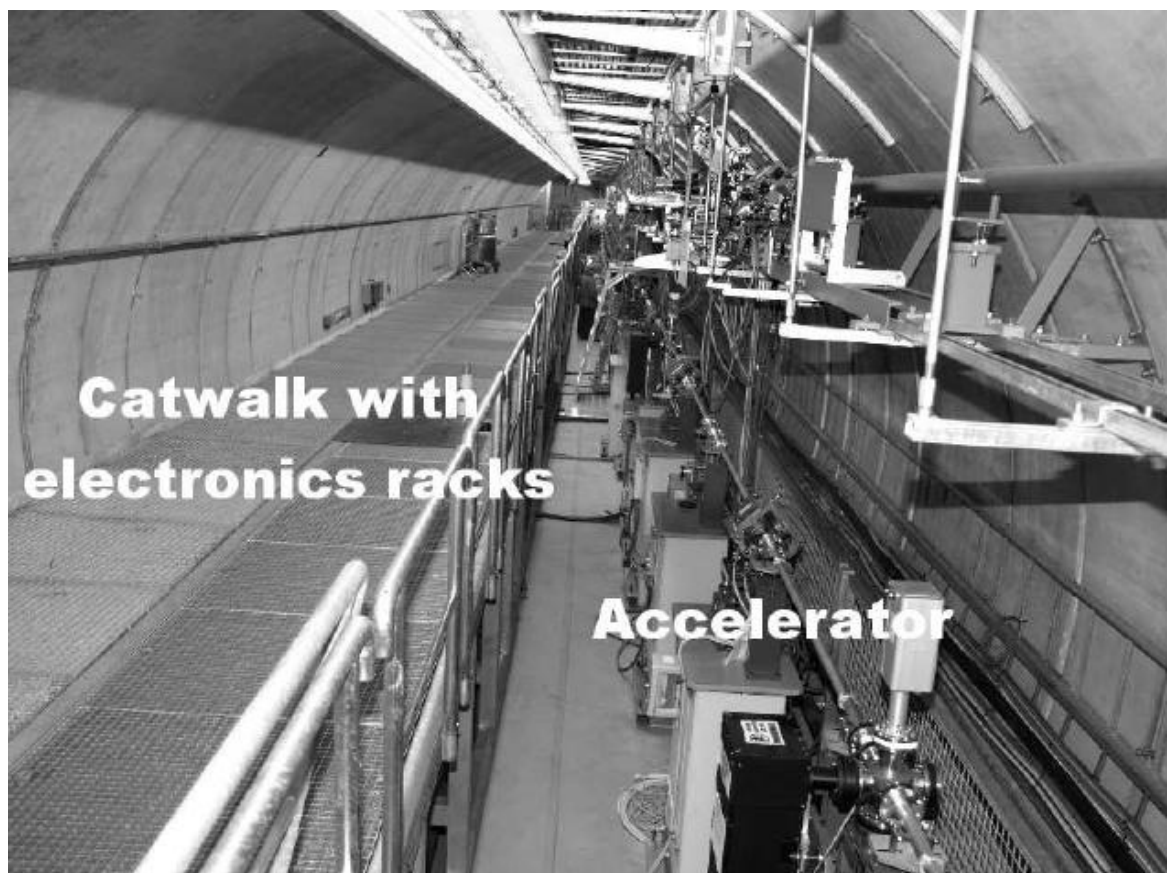


Figure 1: Tesla Test Facility tunnel housing VUV-FEL with catwalk for electronics.

1.2 Radiation effects on electronic components

Radiation causes damages in electronic components. The type and magnitude of these damages depend on irradiation environment. There are three main groups of mechanisms, the radiation effects in semiconductor materials [3][4]:

- Total Ionizing Dose (TID) effects
- Single Event Effects (SEE)
- Displacement damage due to NIEL - Non Ionizing Energy Loss

Effect	Caused by	Results in	Time scale
TID	neutrons, protons, heavy particles, gammas	Degradation of the device with ionizing radiation cumulation	long
SEE	high linear energy transfer particles recoiled by neutrons	device upset or destruction	instantaneous
NIEL	heavy or secondary particles	permanent damage of the semiconductor structure	medium/long

Table 1: Explanation of possible radiation damages in electronics.

1.3 Thermoluminescence Dosimetry (TLD)

It is a method used for radiation exposure monitoring. It can be used both for environmental monitoring and for staff personnel in facilities involving radiation exposure, among other applications.

It is based on thermoluminescence properties of certain materials. The foundation of thermoluminescence is, that when ionizing radiation interacts with a special type of material (usually crystal) it causes electrons in the crystal's atoms to jump to a higher energy state, where they stay trapped due to impurities (usually manganese and titanium), until heated. Heating the crystal causes the electrons to drop back to their normal ground state, releasing a photon of specific frequency. The amount of light generated versus

temperature of heating is the Glow Curve of TLD. The temperature (energy) needed to free the electrons from trap states is dependant on Linear Energy Transfer of the impinging radiation. Unfortunately the radiation dose from each source cannot be told by simply looking at the Curve. The peaks overlap. That is the reason deconvolution is required to find the doses from each source. Deconvolution is based on TL kinetics models – physics of energy transfer between valence band and trap states, as well as retrapping and many other electro-physical aspects of the problem. [5]

There are many types of TLD materials, each with a different characteristic of sensitivity to kinds of radiation. The main are: TLD – 100 , 600 and 700. They are used in mixed radiation fields of gammas and neutrons. A good example of such radiation environment is the vicinity of a high-energy linear accelerator. [6]

TLD - 600 and 700 are mostly responsive to one kind of radiation. That is why it is quite easy to analyze them. Their disadvantage is that they are made out of specially enriched materials that need to be prepared. TLD - 100 type is abundant in nature, but responsive to both and has overlapping peaks, thus deconvolution is needed. [7][8][9]

1.4 Genetic Algorithms

A **genetic algorithm (GA)** is a technique used in computer science to find approximate solutions to optimization and search problems. Genetic algorithms are a particular class of evolutionary algorithms that use techniques inspired by evolutionary biology such as inheritance, mutation, natural selection, and recombination (or crossover). Genetic algorithms are typically implemented as a computer simulation in which a population of abstract representations (called *chromosomes*) of candidate solutions (called *individuals*) to an optimization problem evolves toward better solutions. The evolution starts from a population of completely random individuals and happens in generations. In each generation, the fitness of the whole population is evaluated, multiple individuals are stochastically selected from the current population (based on their fitness), modified (mutated or recombined) to form a new population, which becomes current in the next iteration of the algorithm (see Figure 2).

In many problems with sufficient complexity, GAs may have a tendency to converge towards local optima rather than the global optimum of the problem. The likelihood of this occurring depends on the shape of the fitness landscape: certain problems may provide an easy ascent towards a global optimum, others may make it easier for the function to find the local optimums. This problem may be alleviated by using a different fitness function, or by using techniques to maintain a diverse population of solutions.[10]

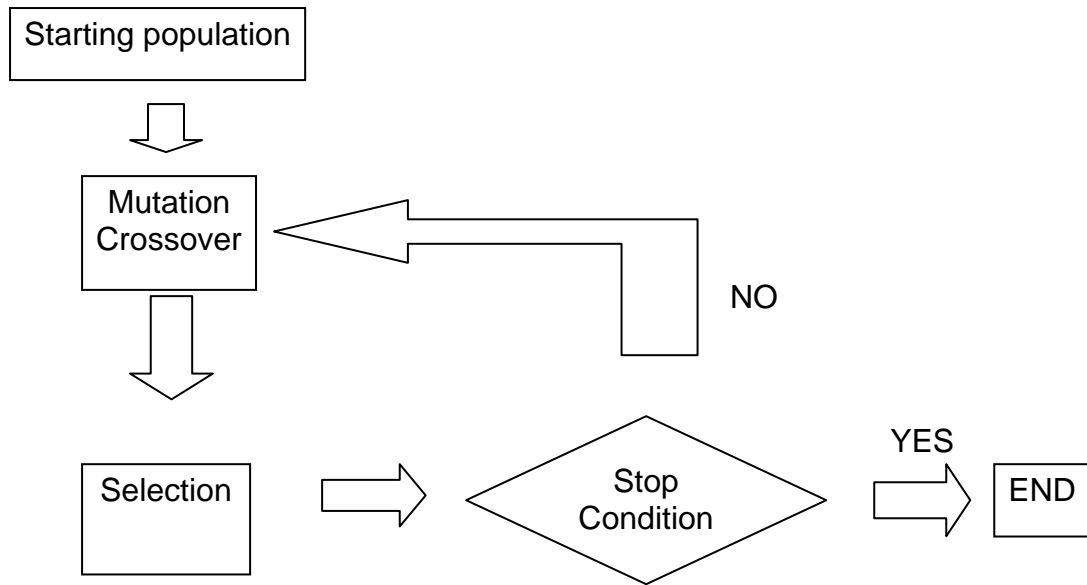


Figure 2: Genetic Algorithm cycle.

2. Thesis' Objectives

A significant number of electronic systems for controlling and diagnostics of modern HEP experiments such as X-FEL are usually located in the accelerator tunnel, next to the accelerating tube, in the radiation environment produced by the machine. Problems with these systems are expected to occur, due to that radiation. To optimize the design of electronic systems and to minimize failures during long-term operation [11] radiation experiments are conducted. They are divided into three steps [12]:

1. Radiation level measurements

2. Investigations of radiation effects on electronic components, circuits and complete systems:

- determination of radiation effects on different types of electronics
- prediction of performance and life-time of electronics in radiation environment

3. Counter measures:

- development of radiation mitigation concepts and evaluation of their performance in the presence of radiation

This thesis concentrates on radiation measurements. The scope of this research was to create software for deconvolution of TLD glow curves for use at DESY, with the following features:

- graphical user interface (GUI) – including data plotting
- convenient data input and output (I/O)
- ease of use
- no knowledge of genetic algorithms to operate
- fast operation
- platform independency

and without the defects that taint all existing software:

- guessing a starting value of the parameters that are sought for
- difficulty of use with increasing model complexity
- cost

These facts caused the typical software to be unreliable, hard to get and use.

Of course developed software could be used for any other kind of radiation measurements where TLDs are applicable, not only in HEP experiments.

3. Concept

The starting point of this project was an idea to consider the TLD glow curve not as an effect of irradiation of the TLD and emission of light due to a carefully prepared heating process with a lot of very complicated physics modeling behind it, but simply: a sum of a few smaller curves. The shape of the smaller curves (called peaks) is not random. They can be described with a varying degree of precision by first, second and general kinetic models. Knowing that it is possible to try to find the peaks, which after summing will result in the starting curve. The problem is one has to basically guess the solution. After brief calculations it is obvious, that a method checking all of the possible solutions is inadequate.

The brief calculations (very simplified):

Assuming there is a TLD glow curve with five glow peaks in it. The used peak description is the first kinetic model - it uses two values. They are called E (Activation Energy) and H (Peak Height). The values of E are between 0.2 and 5, while H is between 1 and 100. E will be sampled with a step of 0.1 and H with a step of 1. That gives 48 possible values of E and 100 for H. Considering there are five peaks the total number of possibilities is: $48^5 * 100^5$. That is approximately equal to $2 * 10^{18}$. Assuming we can do a check of ten possibilities a second, it gives $2 * 10^{17}$ seconds to finish the computations. That is about $80 * 10^9$ years.

Noticing the above it is obvious a much more sophisticated method needs to be used. Among them there are gradient and evolutionary methods. All current tools are based on gradient methods, just used in many variants. While Genetic Algorithms (one of the evolutionary methods) were already proven to be very useful in curve fitting, they were never actually tried to solve this particular problem[13][14]. It was decided to make use of them.

3.1 Programming environment

The considered programming languages/environments were C++, Java and Matlab. Very quickly it became obvious that Matlab allowed to achieve all of the requirements with the lowest time cost thanks to its enormous number of built-in functions: mathematical, graphical, I/O and efficient GUI making programs. It has only one drawback – it is not free. The decision was to use it, since it is considered a standard in scientific organizations and only people affiliated with those would be interested in using the program (i.e. DESY).

3.2 Genetic Algorithm considerations

The main drawback of Genetic Algorithms is that there is no single good instance of GA that can solve each and every problem. A lot of work has been done with trying to apply GAs to a very wide range of domains. After thorough study of [15] it becomes clear, that for different groups of problems different algorithms should be used. And even for similar problems the final GA can be quite different. Simply put a problem needs a GA to be “tailored” for it. Most of this topic is based on information from [15] and [16].

3.2.1 Data representation

There are many possible ways of representing the data for a Genetic Algorithm to use. The operations of mutation and crossover as well as the abilities of the Genetic Algorithm depend on the representation. For this particular problem two kinds of encoding were applicable – binary and real value.

- The binary representation is the same as an integer number representation in a computer’s memory.

The mutation is defined as switching a random bit with a preset probability (see Figure 3).

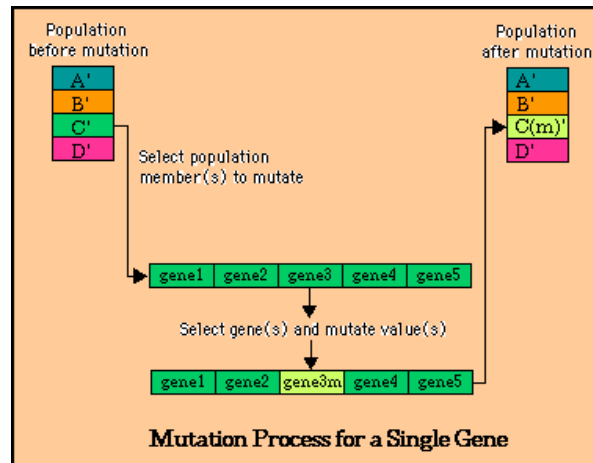


Figure 3: Mutation for a binary string [20]

Crossover as cutting two binary strings in the same place and switching one of the parts (see Figure 4).

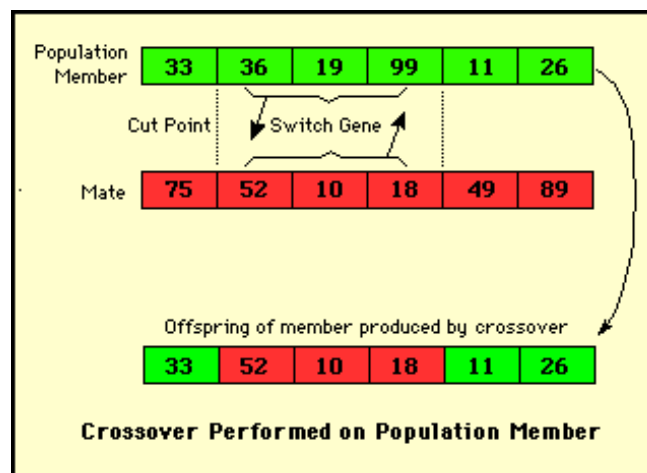


Figure 4: Crossover for a binary string [20]

The main advantage of the binary representation is, that the probability to reach any value represented by the binary string with one operation is not zero.

Main disadvantage is that the possible values of the string are limited by the number of bits, meaning that if very precise encoding is necessary - a very big string needs to be used. That also means that the bigger range of data, the bigger the string is.

- Real value representation means the variable is kept as a floating point in memory (32 bit, 64 bit or more). The mutation is usually defined as adding a distributed variable, with the average value of the distribution of zero. The crossover can only be some kind of averaging operation between two variables. That is the reason why it is not as necessary and useful as in binary representation. The main advantage of real value representation is that the precision is preset by the kind of floating point used. Also the range of data is big and there is no problem with enlarging it.

In order to make a choice of representation the expected fitness landscape has to be defined: The global solution area seems to be quite big, with gentle ascent, with a very big number of local minima, but relatively easy to avoid. Another thing that needed is the input value range: One of the variables (Peak height) has very big range: $0 - 10^5$

The real value encoding is the simple choice. Representing such wide range of data with binary system would require some scaling of encoding depending on input. Also, since the global solution area should not be that hard to find, the biggest advantage of binary is not that useful here. Mutation as a parameter of the program should be enough to avoid local minima.

3.2.2 Fitness function

When using any algorithm there has to be defined the function that we are trying to optimize: either minimize or maximize. The function should be defined in the most simple possible way – to minimize the required computations, unless there is a reason not to. In this particular problem the simplest possible definition of the function is some kind

of distance between the starting Glow Curve and the sum of Peaks. Two such distances were tested (let V be distance between one point of the Curve and the corresponding point in the sum and N be the number of point in which the TL Signal is defined):

- $\sum_{k=1}^N |V_k|$
- $\sqrt{\sum_{k=1}^N (V_k)^2}$ - called the Euclid distance or RMS difference

Both worked. The latter was chosen because of suggestion from [16].

3.2.3 Starting conditions

Genetic Algorithms should start from a random group of solutions. With big input range there is a problem of choosing range of starting solutions. The decision was to keep the starting values close to the smallest possible input – it is much easier to allow the mutation to grow to get to the solution faster, than to wait for mutation to get smaller, while having solutions all over the search space.

3.2.4 Mutation

The most important operation when using real value representation is mutation. After choosing the representation the mutation has to be chosen from the corresponding mutation pool – like explained in chapter 3.2.1. In this case, it is a matter of choosing the distribution. The best according to [16] is the Cauchy, since the probability of choosing a value far away of the average of distribution does not drop to zero very fast (the “tail” is longer than in i.e. Normal distribution). That long tail gives the GA a much bigger area of movement in one generation. Unfortunately Matlab has only Normal distribution built-in and thus it was chosen.

3.2.5 Mutation as a variable

When input has a big range the mutation has to change accordingly, because: if mutation was very small for large input values - it would take very long for a noticeable change, on the other hand if it was large for small input - it would be more a random jumping algorithm, than a Genetic Algorithm. Thus a method of testing mutation has to be chosen. There were two tried:

1. mutation as a GA parameter
 - a. each variable with it's own mutation modifier
 - b. global modifier for all variables of the same type
2. arbitrary test of fitness change in a set number of generations

Possibility 1a allows a very quick change of the mutation of the Peak that has the biggest impact on the fitness. But since all other Peaks do not change much and their mutation is kept on a similar level, they stay small and their mutation is not enough to change fitness much after reaching some local minimum.

Possibility 1b worked quite good, but it sometimes caused the GA to get stuck in a local minimum and it took a long time before the mutation changed enough to move out of it.

Possibility 2 seems the best, since it is quite easy to test how much the fitness changed in the past generations and since the problem seems quite ascending one (the overall direction is easy to find). It is just a matter of setting the right mutation to avoid local minima. The arbitrary test method is faster than setting it as a GA parameter.

In order to change the mutation after deciding it is too small or too big there was a parameter made, to multiply it by the random value. That is "stretching" or "squeezing" the distribution depending on the parameter value. In the first case it is bigger than 1, in the other it is smaller than 1.

3.2.6 Crossover

As explained in chapter 3.2.1 the crossover is not really a necessity. It was used for test purposes and it was obvious, that not much has changed comparing to when it was not there. Only the time of reaching of the maximum of fitness was longer. So it was working against the optimization.

3.2.7 Selection

In order to decide which population members are moved to the next generation and which are removed a method of selection has to be chosen. There are many methods of selecting i.e. tournament, proportional, elite and possible mixes of them. For this problem the main job of the selection is to keep the population diverse enough to be able to move out of local minima. For that elite is inappropriate and tournament not very good. The best is proportional. The selection scheme is the following: each population member has his fitness function evaluated. Now, depending on whether the proportional selection is to be linear or nonlinear a cumulative chart is defined accordingly. Then there is a number of randomizations done equal to the population size. Each value is chosen randomly in the range of [0:1] with an uniform distribution. Cumulative chart is used to find out which population member is to be moved to the next generation depending on the randomization result.

4. Implementation

Implementation of the developed algorithm is presented on Figure 5.

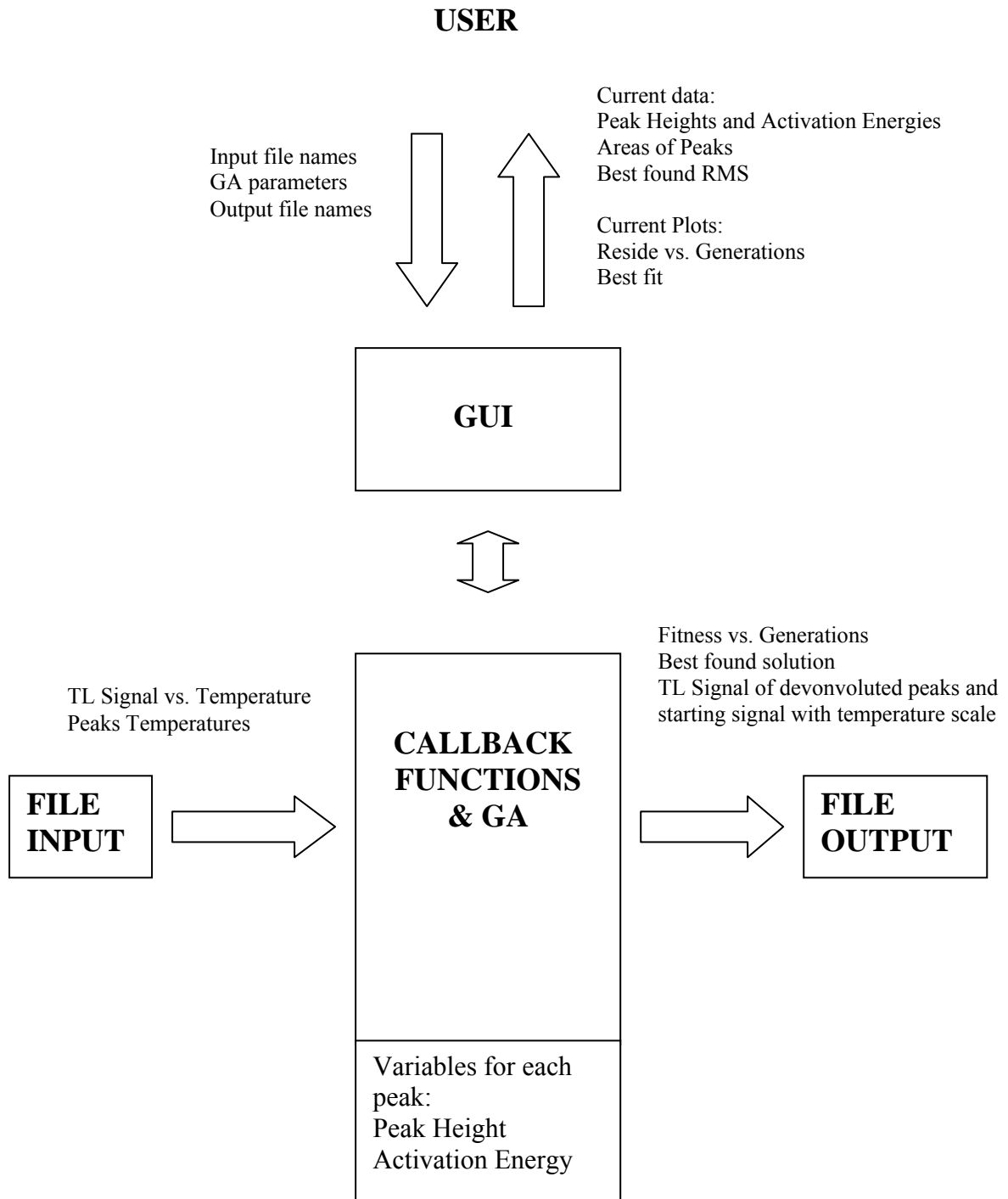


Figure 5: Implementation diagram

4.1 Files

The designed software is made up of two main files, typical for Matlab programs with GUI. One is the layout of the GUI (tld.fig) and the other is the set of callbacks to all the objects in the GUI (tld.m). There are two additional files, Error2.fig and Error2.m that define an additional message window.

4.2 Input and output data, program variables

The input of the program is of two kinds:

User input in GUI:

- Names of files from which data is to be loaded
- Genetic Algorithm parameters:
 - Number of generations
 - Population size
- Names of files into which output data is to be saved

The file input:

- Signal of the TLD (Glow Curve)
- Temperature in each point where Signal is defined.
- Temperature in which the peaks maximums are expected.

Exemplary input files are in Appendix A.

Since Matlab is optimized for operations on matrices all of the input is kept as vectors. The allowed input formats are Excel and comma delimited file. Excel is the most often used spreadsheet and that is why it is used. Comma delimited file is for those who do not use Windows and MS Office.

Variables defining the Peaks (first order Peak approximation by Podgorsak [17])

- Peak Height for each Peak for every member of the population
- Activation Energy for each Peak for every member of the population

Kept as a two 3-dimensional matrices.

The output of the program is also of two kinds:

User output:

- Information about the current best solution updated every 100 generations:
 - Peak Heights for all Peaks
 - Activation Energies for all Peaks
 - Area under every Peak
- Plots of:
 - Fitness (RMS) vs. Generations
 - Starting Glow Curve, all Peaks, sum of Peaks

The file output:

- Fitness vs. Generations
- Best found solution : set of Peak Heights and Activation Energies
- TL Signal of starting curve and TL signal of deconvoluted Peaks with the temperature scale

Format of the output is comma delimited file. It is very easy to export it to all the spreadsheets and it avoids the problem of lack of Excel output functions in non-Windows Matlab. More detailed explanation of usage and data formats is in chapter 5.

4.3 Genetic Algorithm operations

After loading of the right input data, the GA is ready to start. The cycle begins with setting random values of Activation Energies (E) and Peak Heights (P) for each Peak. They are not far from the lower boundary of possible values. Also starting values of mutation are attributed. E and P are mutated (see chapter 3.2.4), then their fitness is checked (see chapter 3.2.2). Then selection process is done (see chapter 3.2.7). Every 100 and 200 generations the percentage change of fitness is tested. The more often test is used to check if the mutation of E is not too small, based on how much the fitness of the best solution between these generations changed: If it was less than 2% than the mutation is multiplied by 1.3. That is because only E has a big input variance. P stays similar for all Curves. The 200 generations change of fitness is to check if mutation of E is not too big: the test checks whether the best solution did not get worse by 20%. If it did – the mutation is divided by 2.

This algorithm implements usage of some additional information. The expected deconvoluted Peak shape is that they grow with numbering. That is the reason for a restriction of the parameter E to a value of 1.5 in the first half of generations. That causes the Peaks to be tall rather than wide. During the other half the search space closer to the expected one is browsed first.

The cycle continues until the defined number of generations passes.

4.4 Peak area computing

The radiation dose is relative to the area under the Peak. To compute the area two methods have been tested:

- polynomial fitting and integration
- rectangular approximation

Polynomial fitting did work well in some cases. And the area was calculated very precisely then. But there were occasions, when the fitting ended with a certain tail that was totally wrong – went upwards instead of downwards. The only thing to remove the tail was to ignore the smallest points. But then the approximation was not so precise – there was a loss of area under the ignored points. And the rectangular fitting (multiplying length of each segment between temperatures by average of TL signal between those two points), with its small error seemed better and less time consuming than the polynomial fit.

5. Instruction manual

This is the instruction manual for the software.

TLD Glow Curve deconvolution software

1. Files of the program:

- tld.m – main script file
- tld.fig – GUI file of the main script
- tld2.m – main script file packed version (no different than tld.m – just names have to be different)
- tld2.fig – GUI file with all the icons smaller and packed up (for machines with very small screen resolutions)
- Error2.m – additional message box script file
- Error2.fig – GUI file of the message box

2. Program Requirements

- PC class computer
- Unix, Linux or Windows running Matlab 7.0

3. Program control panel overview (see Figure 6)

The program screen is divided into 5 main parts, they will show up in the order of numbering:

1. Data input
2. Genetic Algorithm parameters
3. Axis with glow curve, deconvoluted glow peaks and their sum and axis with RMS vs. Generations
4. Output parameters
5. Saving output

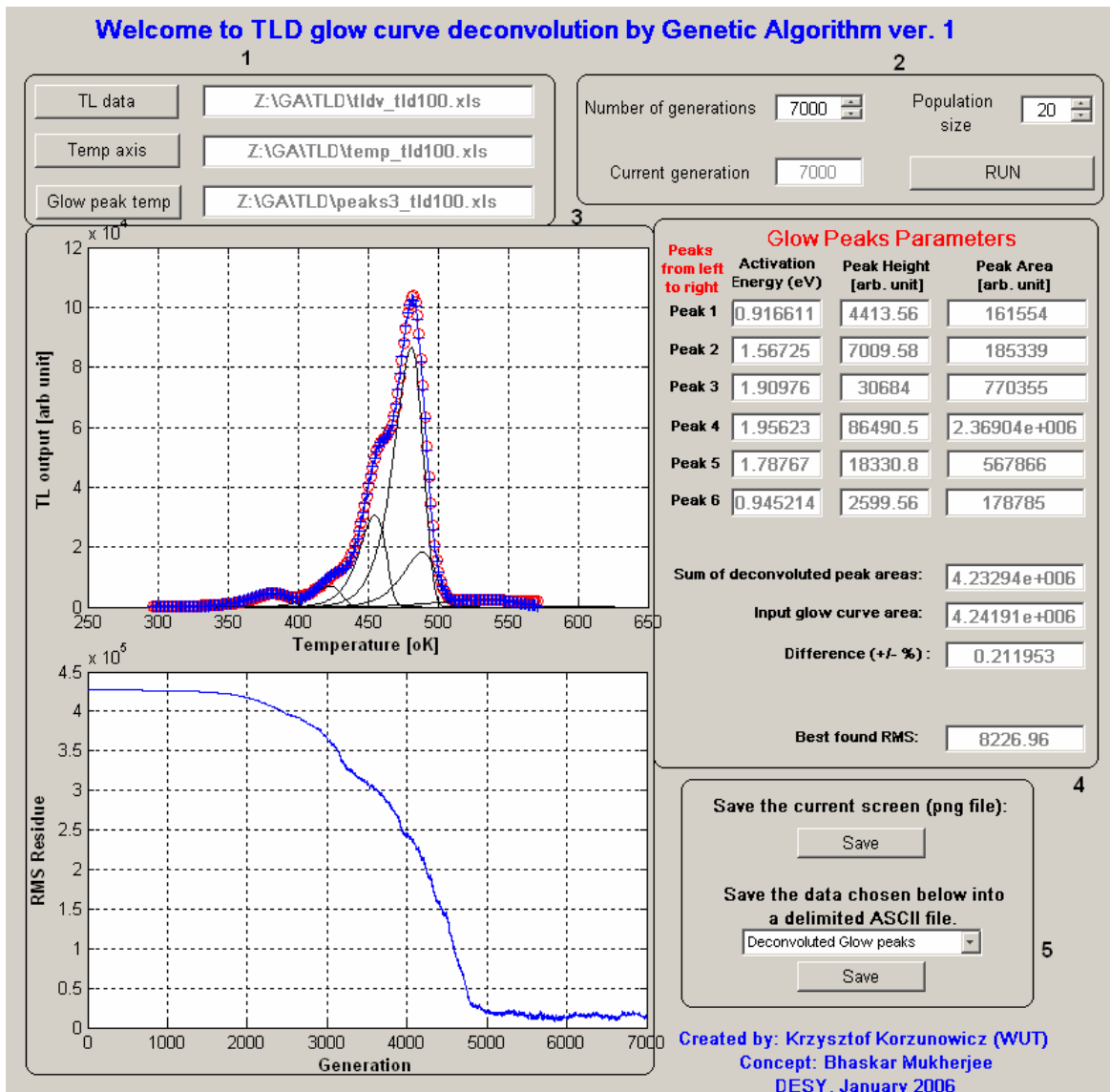


Figure 6: Program control panel overview

4. Input data requirements

The program requires 3 data vectors defined in 3 different files to run:

- A. Temperature axis data
- B. TL data – the values of TL signal in each temperature point defined
- C. Peak temperature values – the points on the temperature axis that the maximum of the deconvoluted Peaks will be.

Number of elements of vectors A and B has to be the same.

Number of elements of vector C has to be between 1 and 7.

A, B and C need to be COLUMN vectors in either Excel (*.xls) or comma delimited ASCII file format (*.csv)

A and C need be in the Celsius ($^{\circ}\text{C}$) scale.

5. Starting up the program

In order to start the program all of the files listed in point no. 1 have to be in the same folder. After starting Matlab it is necessary to change the Current Directory (using either the Current Directory window or Command Window) to the directory all of the program files are in. Startup command is: tld

The recommended number of generations is more than 2000. If it is not enough it should be made bigger. Population size – at least 5. The algorithm will run faster with smaller population size, but it might find a better solution with more.

6. Output data details

It is possible to have 2 kinds of output. Both available after the defined number of generations passes (not possible while the algorithm works).

A. Screen of the current status of the program window.

B. One of the following sets of data:

- Deconvoluted glow Peaks, and their sum (all points)
- RMS vs. Generations (all points)
- Glow Peaks FINAL parameters (each Peak Activation Energy, TL signal and area under each Peak) - same as the shown in part 4 of the screen.

File formats:

A - *.png

B - *.txt (comma delimited ASCII file)

It is very easy to import the data into Excel:

Open the txt file with Excel, it will automatically open the Import Wizard:

Choose file type: delimited

Click next

As delimiter choose ONLY comma

Click next

In Advanced choose decimal separator as comma (,) and thousands separator as: (`)
(it doesn't really matter what you choose as thousands separator, as long as its not the same as decimal separator)

6. Testing

Since this program has a totally different approach than any other available software and its use is supposed to be different, the testing was reduced to mainly checking how good the fitting process is for various curves. Many such tests were done, only a few will be shown here. A different kind of test was to check whether the Genetic Algorithm converges to the same point.

6.1 Test 1: Shielding analysis

Three PorTL cells (TLD dosimeters made in Hungary) [18] were irradiated with water moderated $^{241}\text{Am/Be}$ [19] neutron source (of average neutron energy 3.3 MeV) behind different kinds of shielding, which can be used to protect electronics. The exposure time was every time the same. To choose the shielding which leaves the lowest dose of thermal neutrons (high temperature Peak on TLD scale) the program was used:

1. No shielding – taken as reference (see Figure 7)
2. 6 mm of Polyboron shield (see Figure 8)
3. 6 mm of composite material shield (see Figure 9)

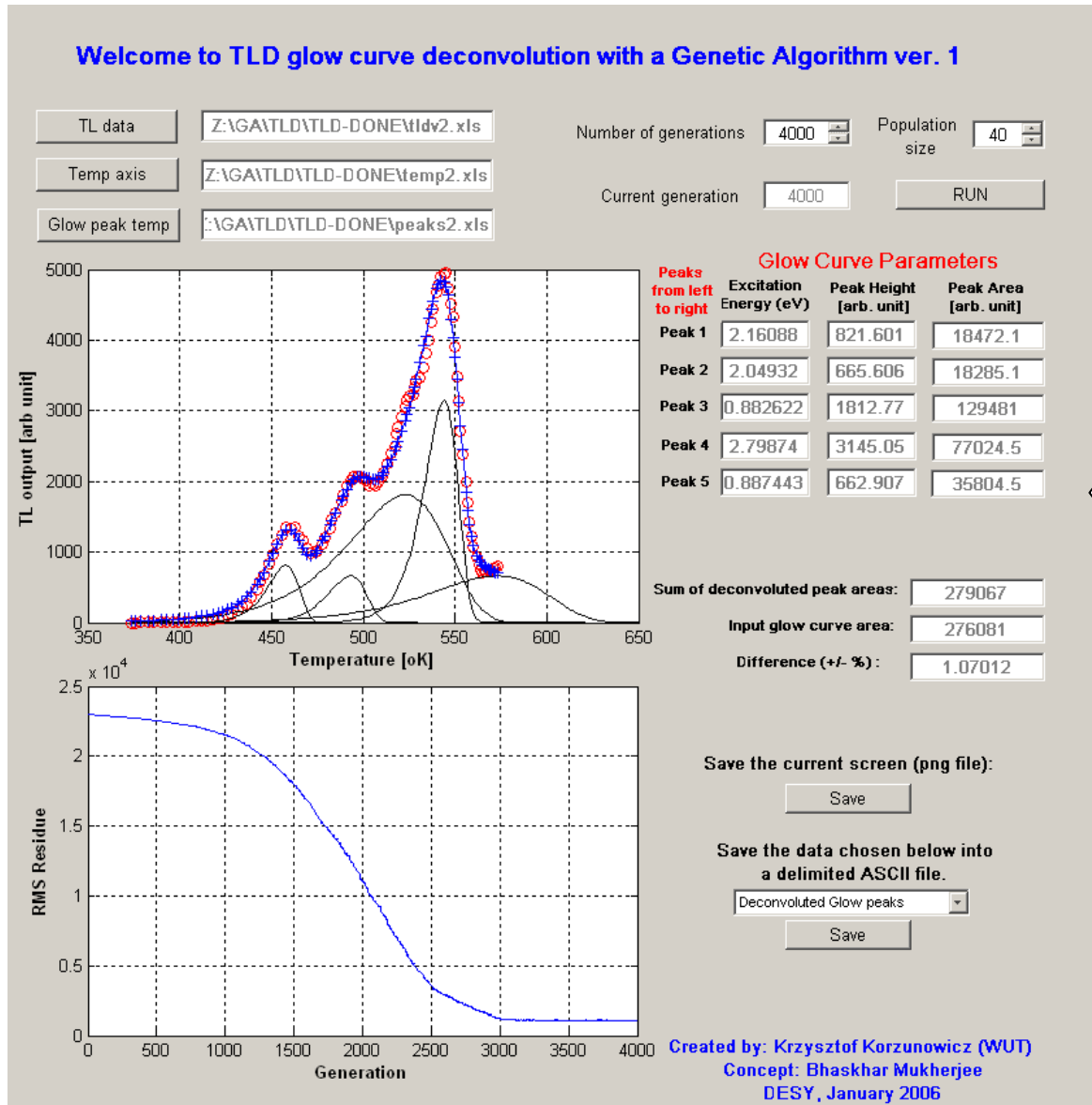


Figure 7: Deconvolution for unshielded TLD (Test 1) – marked neutron Peak

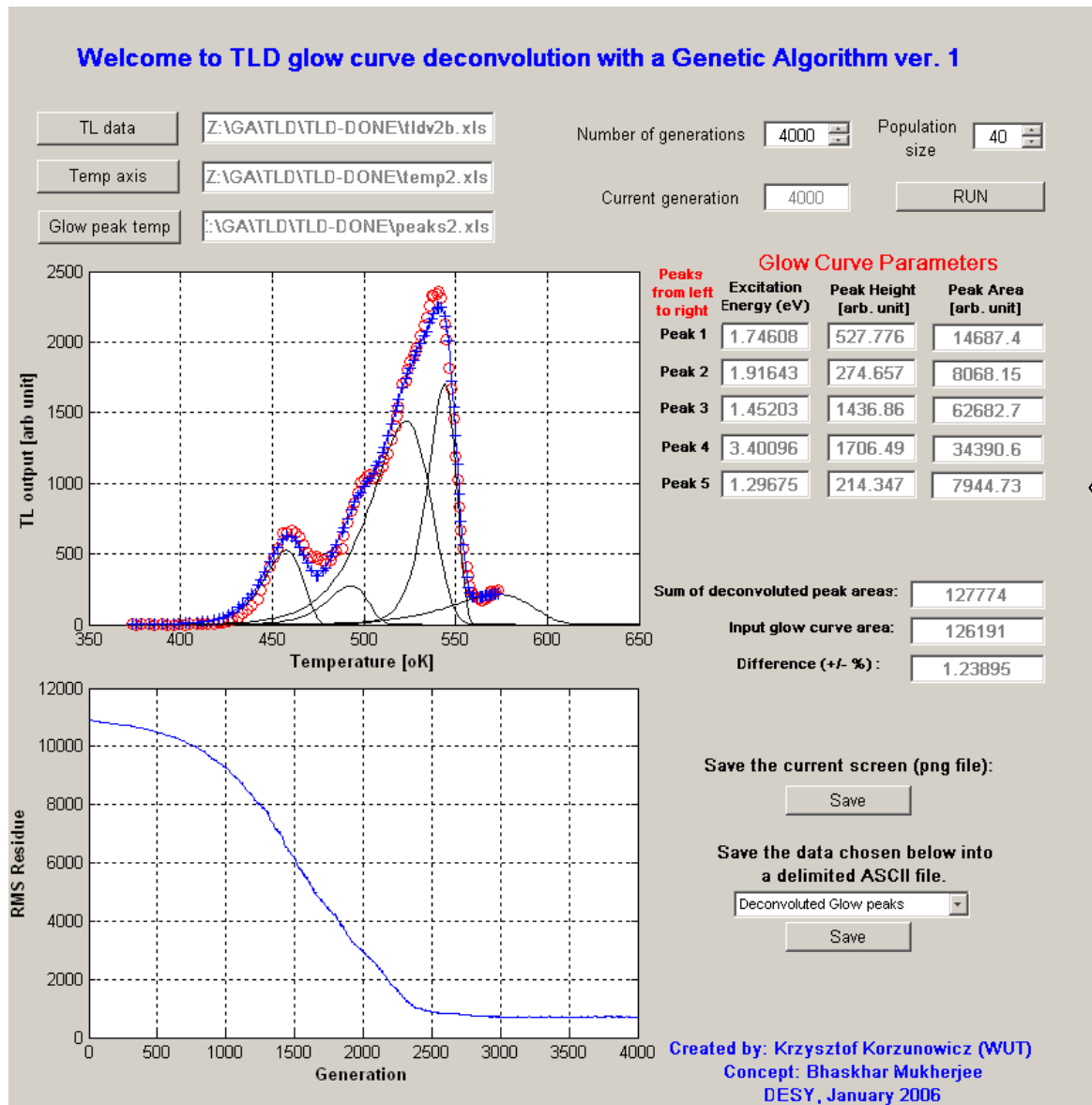


Figure 8: Deconvolution for Polyboron shield (Test 1) – marked neutron Peak

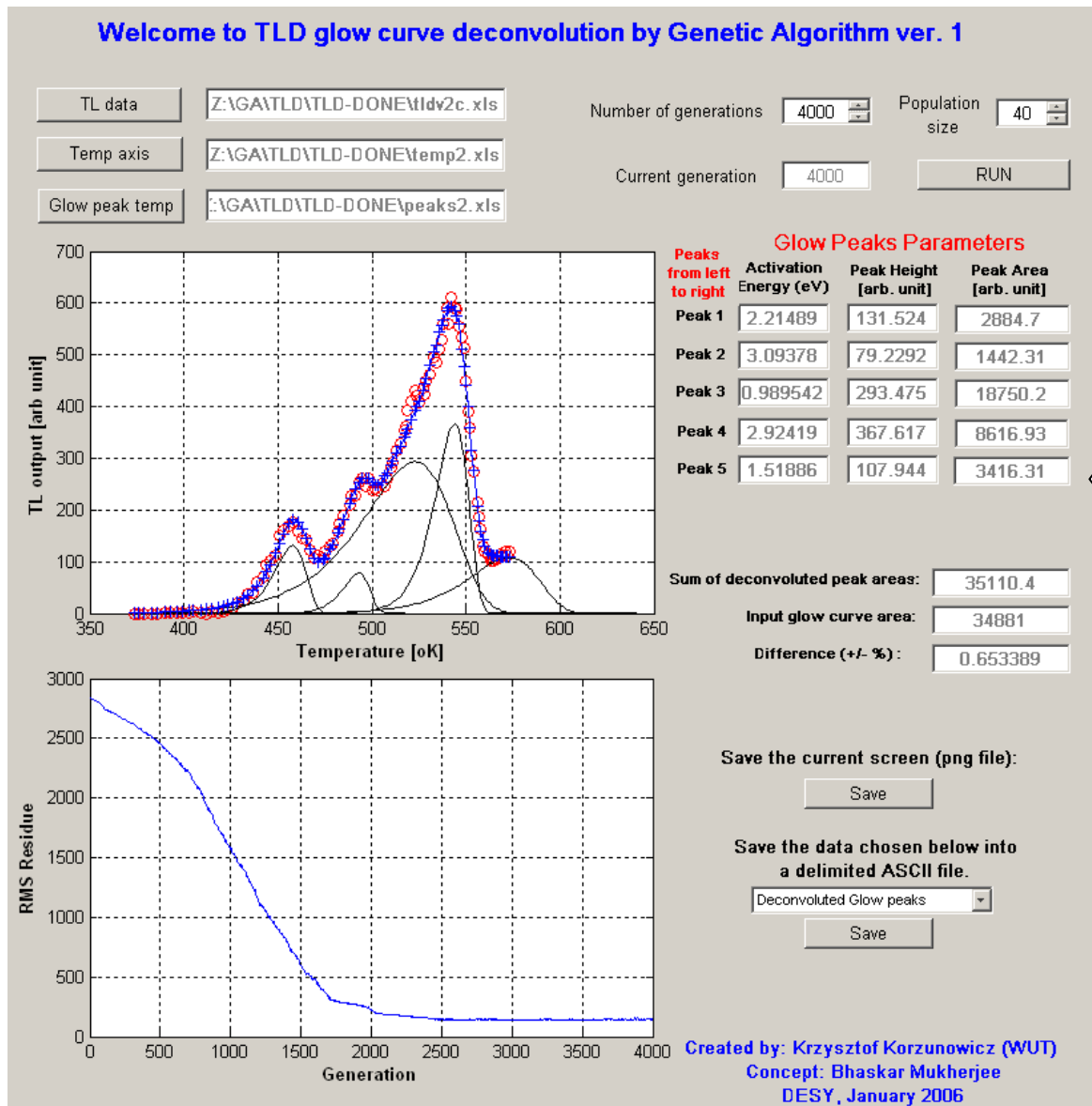


Figure 9: Deconvolution for Composite material (Test 1) – marked neutron Peak

	No shielding	Poly-boron shield	Composite material
Area under Peak 5 (thermal neutrons)	35804	7944	3416

Table 2: Comparison of thermal neutron Peak areas for different shielding.

One can conclude that the composite material is the best. It has lowered the neutron dose by a factor of 10. The ease with which the conclusion can be made is proof, that this software can be used to quickly assess the goodness of shielding.

6.2 Test 2: Gamma responses of TLD

Two PorTL cells were irradiated in Hera experiment at DESY with different gamma doses: 27.8 mGy and 462 mGy, to test if the fourth Peak area is linearly dependant on the gamma dose (see Figures 10,11).

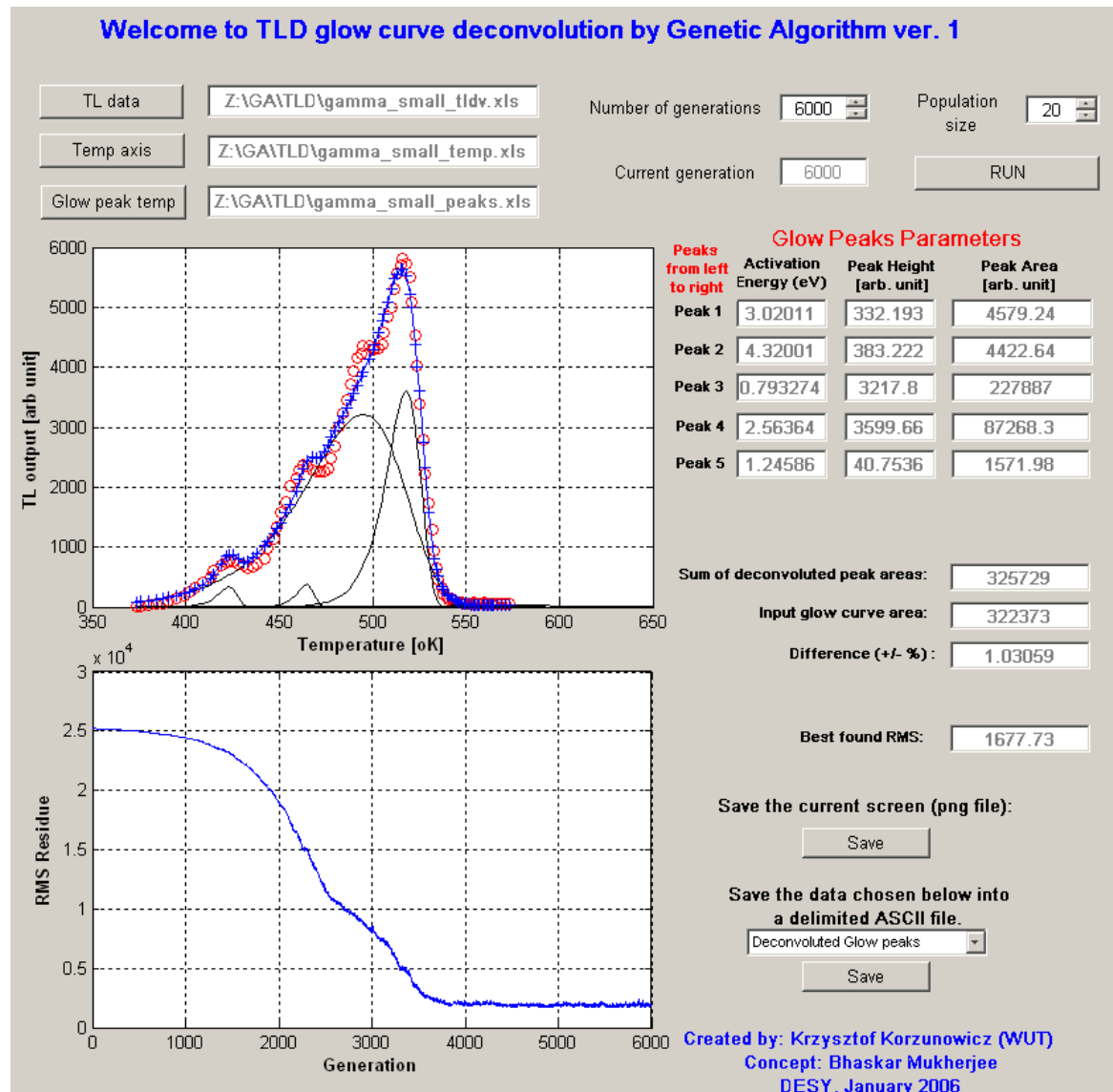


Figure 10: Deconvolution of Gamma data – dose 27.8 mGy

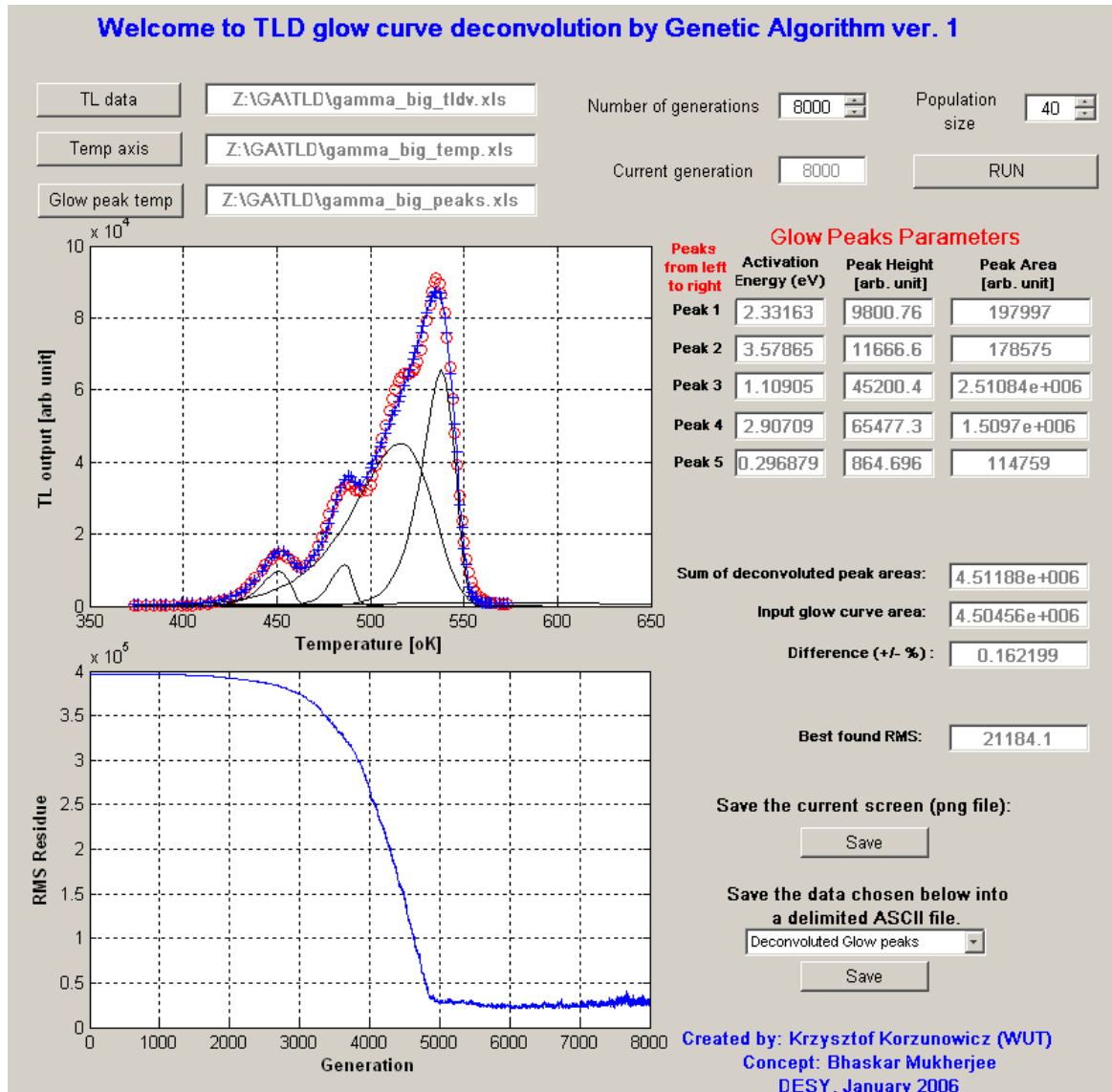


Figure 11: Deconvolution of Gamma – dose 462 mS

Dose (mGy)	27.8	462
4 th Peak Area (arb unit)	87268	1509000
4 th Peak Area/Dose (arb unit)	3139	3266

Table 3: Comparison of Peak area to dose rates for different doses.

The Table 3 shows that the Peak area grows linearly with the dose. But unfortunately that depends on how good the deconvolution was. I.e. the first dose Glow Curve had two solutions of very similar fitness (see Figures 10 and 12). The fitness is different by just a few points. That is the reason, why the solutions have to be checked thoroughly before they are considered final.

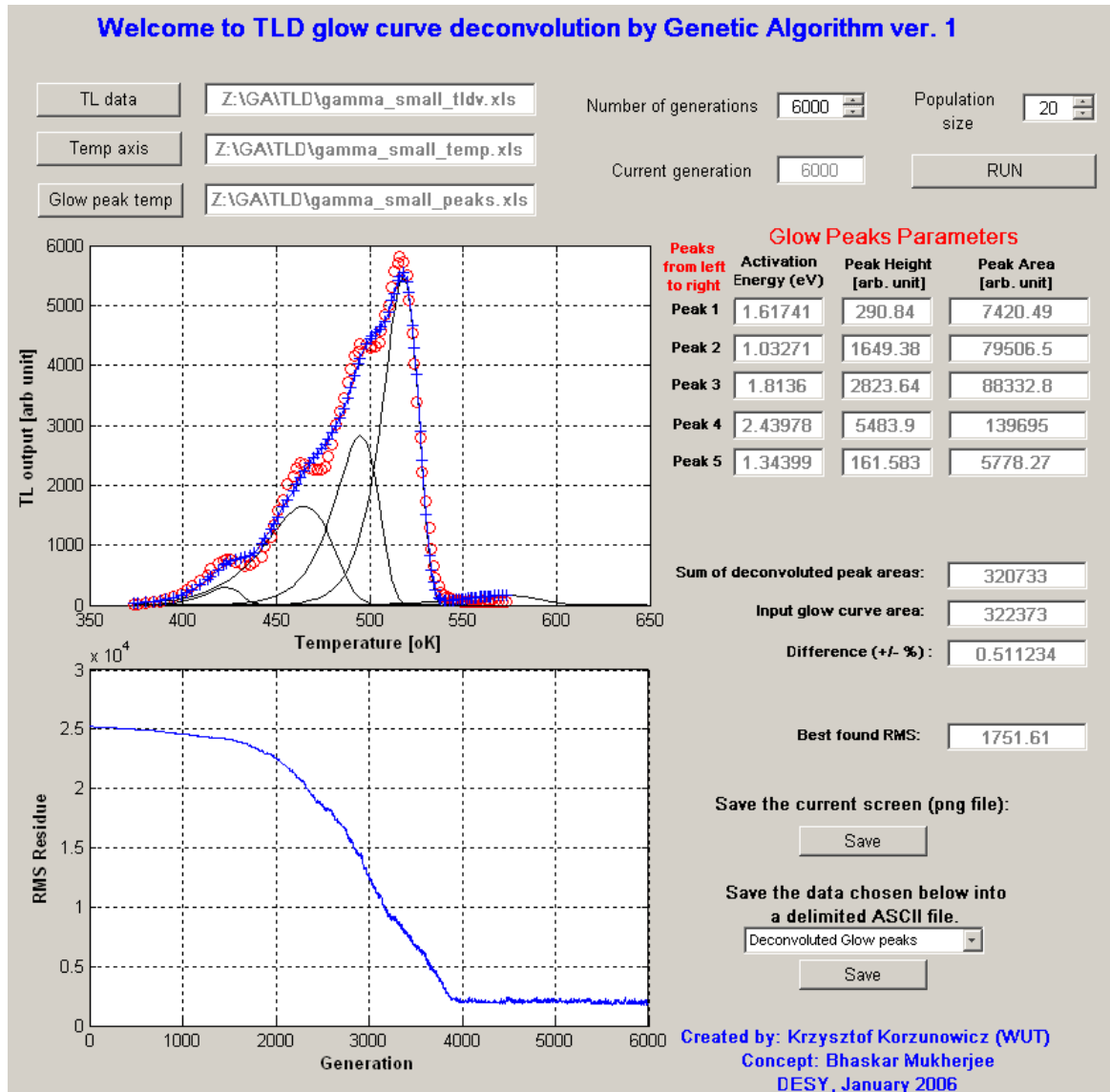


Figure 12: Deconvolution of Gamma – dose 27.8 mGy (2).

6.3 Convergence test

The test is to check whether the Genetic Algorithm ends its searching in the same point. The basis of this test was the 462 mGy dose Glow Curve from Test 2. It was deconvoluted 3 times with the same starting Peak temperatures. (see Figures 12,13,14).

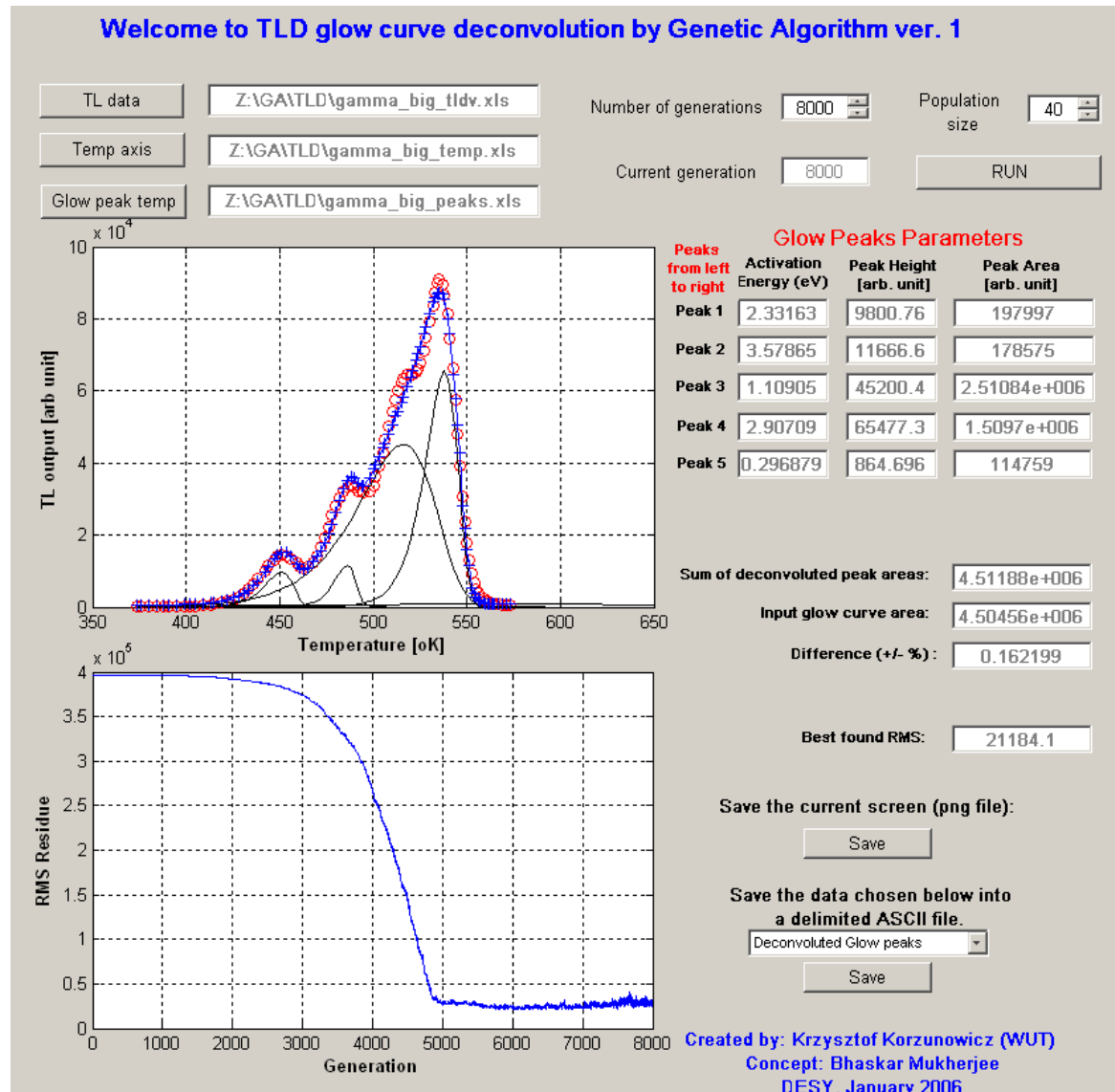


Figure 13: Convergence test: Deconvolution no. 1

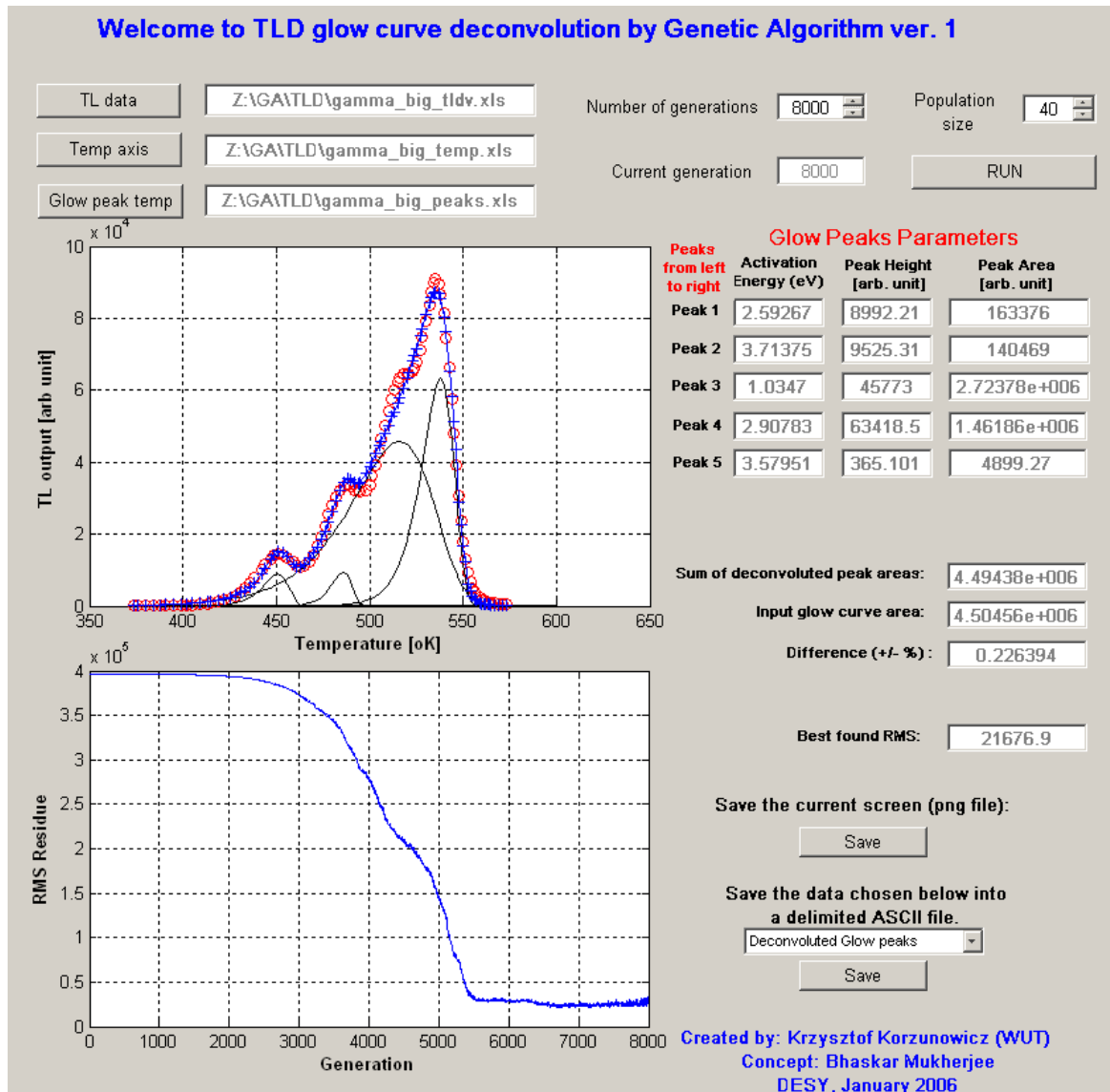


Figure 14: Convergence test: Deconvolution no. 2

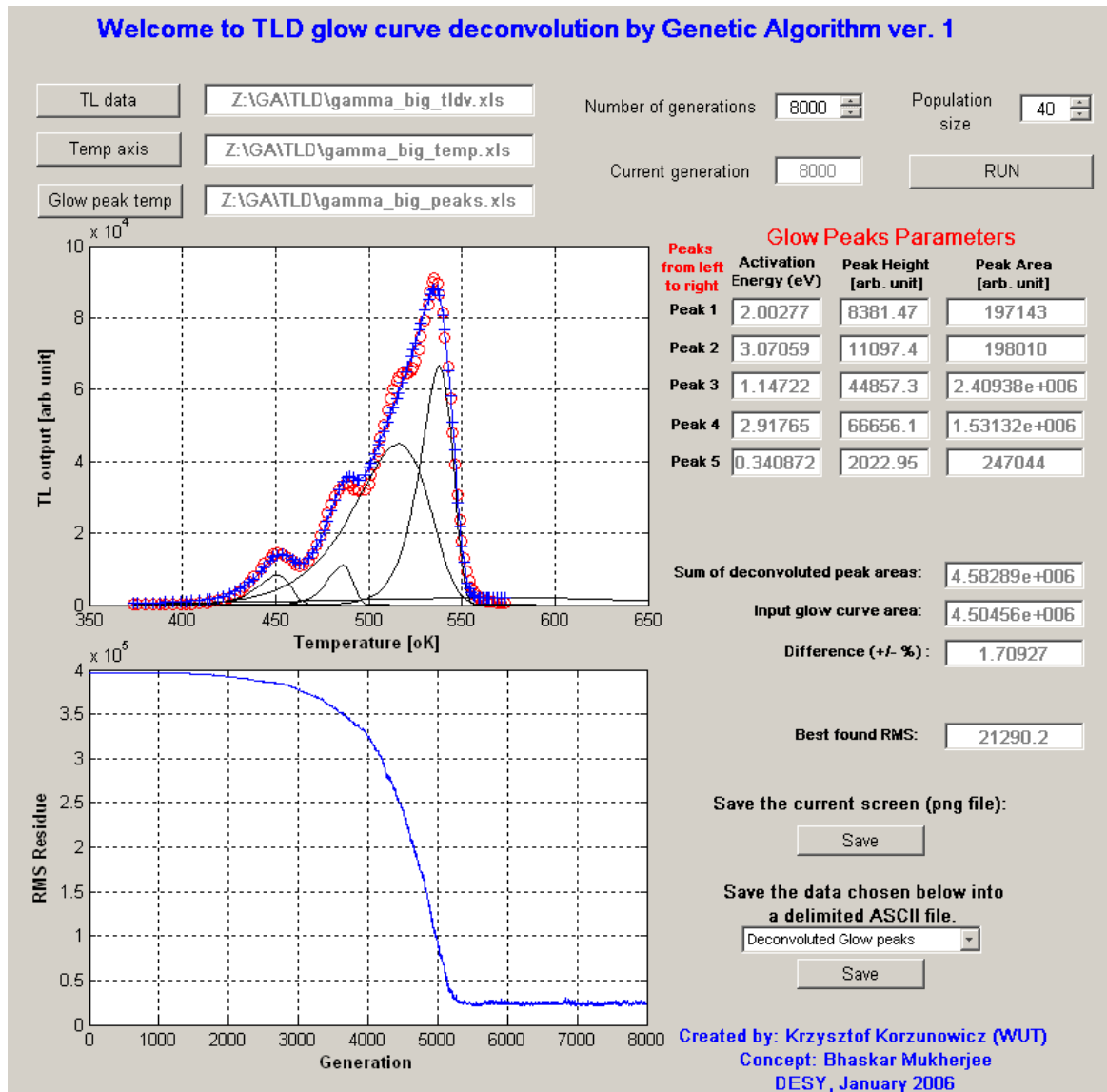


Figure 15: Convergence test: Deconvolution no. 3

Deconvolution no.	Best RMS found	Peak 4 area (*10 ⁶) (arb unit)	Difference between deconvoluted area and starting area (%)
1	21184.1	1.50970	0.16219
2	21676.9	1.46186	0.22639
3	21290.2	1.53132	1.70927
Mean	21383.73	1.50096	0.699283
Std deviation	259.3729	0.035545	0.875263

Table 4: Three different algorithm runs – comparison of solutions

The obtained values of three different algorithm runs are very similar to each other, the standard deviation of the values in Table 4 are small. The only value that changes by a bigger margin is the difference of areas in percentage, but that is caused by dependency on all Peak areas and is already a percentage. The algorithm finds very similar solutions each run. But that does not have to be true for every problem. As shown in the previous test it is possible that there will be many solutions of similar fitness, that look completely different. In such cases user should be very careful, not to conclude about the rightness of the algorithm too fast.

7. Summary and conclusions

The project was a success – the created software allows to deconvolute Glow Curves with a measurable precision. The Genetic Algorithm allowed to make the program easy to use, without many additional information necessary.

All the requirements for the program were met:

- graphical user interface (GUI) – including data plotting

User-friendly Matlab based GUI

- convenient data input and output (I/O)
- platform independency

Excel and ASCII delimited file format accepted as input, only the latter given as output to allow Unix and Linux operation (Matlab on those platforms has no Excel format write functions)

- ease of use (very basic genetic algorithms knowledge to operate)

The only required input from user is population number and number of generations. Both require just a little explanation to be understood.

- fast operation

The typical Glow Curve is deconvoluted in less than 10000 generations. On a Pentium 4 type processor that is less than 3 minutes.

The user of this program should know that Genetic Algorithms never give a guarantee, that the solution found is the best possible. That danger is especially true if the solution area is very big and flat, which can cause the GA to find quite different solutions each time. They would not hold much value then. It is recommended to run the algorithm a few times to check that.

7.1 Broader perspective

The chosen approach - assuming nothing about the Glow Curve itself, using only the Genetic Algorithm to find the fitting solution, creates two disadvantages. One is that only a physicist will know if the deconvolution gave a result that is acceptable, since all of these curves have many possible deconvolutions with a preset precision, based on the starting Peak positioning. That leads to the second disadvantage: because of the Podgorsak approximation the deconvolution is very dependant on the choice of Peak positioning. Even a few degrees difference can give results that differ by more than a margin. To alleviate that, the next steps should be, to use a more complex model and try to use the additional information known due to physical nature of the phenomenon.

8. List of Figures

Figure 1: Tesla Test Facility tunnel housing VUV-FEL with catwalk for electronics.....	6
Figure 2: Genetic Algorithm cycle.	9
Figure 3: Mutation for a binary string [20].....	14
Figure 4: Crossover for a binary string [20]	14
Figure 5: Implementation diagram.....	19
Figure 6: Program control panel overview	25
Figure 7: Deconvolution for unshielded TLD (Test 1) – marked neutron Peak.....	29
Figure 8: Deconvolution for Polyboron shield (Test 1) – marked neutron Peak.....	30
Figure 9: Deconvolution for Composite material (Test 1) – marked neutron Peak.....	31
Figure 10: Deconvolution of Gamma data – dose 27.8 mGy	32
Figure 11: Deconvolution of Gamma – dose 462 mS.....	33
Figure 12: Deconvolution of Gamma – dose 27.8 mGy (2).	34
Figure 13: Convergence test: Deconvolution no. 1	35
Figure 14: Convergence test: Deconvolution no. 2	36
Figure 15: Convergence test: Deconvolution no. 3	37

9. List of Tables

Table 1: Explanation of possible radiation damages in electronics.....	7
Table 2: Comparison of thermal neutron Peak areas for different shielding.....	31
Table 3: Comparison of Peak area to dose rates for different doses.....	33
Table 4: Three different algorithm runs – comparison of solutions	37

10. References

- [1] Wikipedia: Particle physics, <http://en.wikipedia.org/wiki/Particle_physics>
- [2] What is XFEL?,
<<http://www.desy.de/pr-info/desyhome/html/presse/hginfos/xfel/was.en.html>>
- [3] L. Dusseau, Introduction to Radiation Issues of Commercial-Off-The-Shelf (COTS) Components, CERN Training 2000
- [4] M. Dentan, Radiation Effects on Electronic Components and Circuits, CERN Training 2000
- [5] Wikipedia: Thermoluminescence Dosimeter, Dosimetry
<http://en.wikipedia.org/wiki/Thermoluminescent_dosimeter>
- [6] B. Mukherjee, D. Makowski, S. Simrock, Dosimetry of high-energy elektron linac produced photoneutrons and the bremsstrahlung gamma-rays using TLD-500 and TLD-700 dosimeter pairs.
- [7] Y. S. Horowitz, D. Yossian, Computerized Glow Curve Deconvolution: Application to Thermoluminescence Dosimetry, Radiation Protection Dosimetry, Vol. 60, No. 1, 166
- [8] M. Ranogajec-Komor, Thermoluminescence Dosimetry – Application in Environmental Monitoring, Radiation Safety Management Vol.2, No.1 (2-16)
- [9] B. Mukherjee , Glow curve analysis of TLD-700 dosimeters exposed to fast neutrons and gamma rays from isotopic sources, Nuclear Instruments and Methods in Physics Research A 385 (1997) pp. 179-182
- [10] Wikipedia: Genetic Algorithm, <http://en.wikipedia.org/wiki/Genetic_algorithm>
- [11] A. Leuschner, S. Simrock, Radiation field Inside the Tunnel of the Linear Collider TESLA, Laboratory Note, DESY D3-113, 2000
- [12] D. Rybka, Praca dyplomowa magisterska: Integrated Measurement Systems for Electronic Devices Operating in Radiation Environment, 2005, Warsaw University of Technology
- [13] K. Messa, M. Lybanon, Curve Fitting Using Genetic Algorithms, Final Report October 1991, Naval Oceanographic and Atmospheric Research Lab Stennis Space Center

- [14] J. Pittman, C. A. Murthy, Optimal Line Fitting Using Genetic Algorithms, Technical Report, July 1997, Pennsylvania State Univ. University Park Center For Multivariate Analysis
- [15] Thomas Bäck, David B. Fogel, Zbigniew Michalewicz, Handbook of Evolutionary Computation (Computational Intelligence Library), Institute of Physics Publishing and Oxford University 1997
- [16] J. Arabas, Wykłady z Algorytmów Ewolucyjnych (Lectures on Evolutionary Algorithms), Wydawnictwa Naukowo-Techniczne 2001
- [17] C. A. Perks, T. M. Marshall, Techniques for Thermoluminescence Glow Curve Analysis, Radiation Protection Dosimetry 38 (1991), pp. 261-269
- [18] S. Deme, I. Apathy, L. Bodnar, A. Csoke, I. Feher, T. Pazmandi: PorTL® - a compact, portable TLD reader for environmental and personal dosimetry. *ibid.*, 91-96.
- [19] B. Mukherjee, Development of a simple neutron irradiation facility with variable energy using a light water moderated ²⁴¹-Am/Be source. Nuclear Instruments and Methods A, 363 (1995) pp.616
- [20] Generator – Genetic Algorithm software manual.
<http://www.nli-ltd.com/products/genetic_algorithms/generator.htm>

Appendix A: Example input files for the deconvolution program

Example input files:

- with temperature data is presented on Listing 1
- with TL Signal data is presented on Listing 2
- with Peak locations is presented on Listing 3

24.16	58	26.9
25.54384	39	32.3
26.92768	77	40.7
28.31152	60	44.9
29.69535	46	47.6
31.07919	57	57.4
32.46303	89	
33.84687	60	
35.23071	76	
36.61455	80	
37.99838	90	
39.38222	93	
40.76606	119	
42.1499	101	
43.53374	121	
44.91758	166	
46.30141	160	
47.68525	206	
49.06909	204	
50.45293	229	
51.83677	300	
53.22061	343	
54.60444	338	
55.98828	421	
57.37212	430	
Listing 1	Listing 2	Listing 3