Application of Genetic Algorithm with Real Representation to COULEX Data Analysis

Daniel Andrzej Piętak[1,2], Paweł J. Napiorkowski[2], Zbigniew Walczak[1] and Jacek Wojciechowski[1]

[1] Warsaw University of Technology, Institute of Radioelectronics, Warsaw, Poland
[2] Warsaw University, Heavy Ion Laboratory, Warsaw, Poland

Abstract. *The development of a new genetic algorithm designed for the data analysis in the nuclear physic experiment is presented. The calculations performed with the multidimensional test function identified the weakness of the real representation implemented to the genetic algorithm. The tests presented in this paper will result in upgrade of genetic operators which will be applied for the real physic case of the COULEX data analysis.*

1 Overview of COULEX data analysis

Coulomb Excitation (COULEX) is a well developed method to study the electromagnetic properties of nuclei in the ground and excited states. The most spectacular result of this experimental technique is shape determination of nuclei, what is an easy way to test different models of nuclear structure. That's why the main advantage of Coulomb Excitation lies in the fact that the interaction responsible for the process can be described by the well-established theory of the electromagnetic forces allowing nuclear structure to be studied in a model-independent way in contrast to nuclear reactions. The Coulomb Excitation is the preeminent probe of collective degrees of freedom in nuclear structure by a direct measure of the reduced electromagnetic matrix elements, which can be converted with use of some rules techniques into deformation parameters of the nuclear shapes.

A semiclassical theory of multiple Coulomb Excitation was developed in 1956 [1] and the first implementation of the method to the computer program COULEX was developed by Winther and de Boer in 1965 [2]. The code COULEX provided the first opportunity to calculate quantitatively multiple Coulomb Excitation amplitudes using an assumed set of the reduced electromagnetic matrix elements.

But the main difficulty for making a model-independent analysis lies in the large number of reduced matrix elements influencing heavy-ion excitation. The task is to fit the set of matrix elements as parameters of the theory to the collection of experimental observables. It is a typical approach in the experimental physics but a large number of parameters and a significant complexity of the multiple Coulomb excitation semiclassical theory formalism make the major difficulties.

From early eighties a semiclassical coupled-channel Coulomb Excitation least-squares search code, GOSIA [3] has become a basic tool for the data analysis. It has been developed by

Rochester-Warsaw collaboration to analyze the large sets of experimental data required to unambiguously determine the many electromagnetic matrix elements involved in heavy-ion induced multiple Coulomb Excitation.

The main advantages of GOSIA are:

- χ^2 fitting procedure (using a gradient method) to determine the best – with the lowest χ^2 value – set of electromagnetic matrix elements;
- fast approximation method developed to speed up the analysis of the complex experiments;
- extensive possibilities to take into account various geometries of experimental setup.

The main difficulty in GOSIA use is time consuming testing of alternative solutions.

New exotic beam facilities, delivering beams of higher intensity, together with a new generation of highly efficient detection systems will challenge the current method of analysis. With a new experimental tools more information will be collected, which makes the analysis more complex.

1.1 Jacob: the first approach to the genetic algorithm

The genetic approach to COULEX data analysis was developed to overcome the difficulty which lies in identifying local minima. In 2008 Jacob, a program exploiting genetic algorithm to determine matrix elements, was completed [4]. It was based on the binary representation of the parameters. And it was using concatenated pairs of bit-strings (first segment of the first string concatenated with second segment of the second string and vice versa) as a single point crossing-over operator for each parameter. As a selection operator the user could choose: roulette, tournament or truncation selection. The mutation relied on flipping random bits with: constant mutation probability or declining with the number of generations. It was independent for each bit of each chromosome. Jacob used as an objective function χ^2 values which were determined in GOSIA executions.

1.2 F7 function

An *a priori* unknown unique solution and a time consuming calculations of the full semiclassical formalism of multiple Coulomb Excitation were the reasons to use a test function to study the properties of the algorithm. One of the popular functions for testing optimization algorithms is F7, proposed by Schwefel in [5]. A shape of a multidimensional version of this test function is similar to χ^2 surface expected in real optimization problems of a Coulex data analysis. In order to achieve a formal agreement with the existing code, a scaling of the F7 function was applied.

The χ^2 value is never less than zero and the ranges of parameters are typically in the interval (–10.0, 10.0). In the application to our problem domain of F7 was scaled to fit the range of each parameter independently and shifted up by the value of the total global minimum. To have multi-optimum objective surface, F7 was scaled by multiplying each argument (matrix element) by 100. The transformed one-dimensional function F7 is shown on Fig. 1.

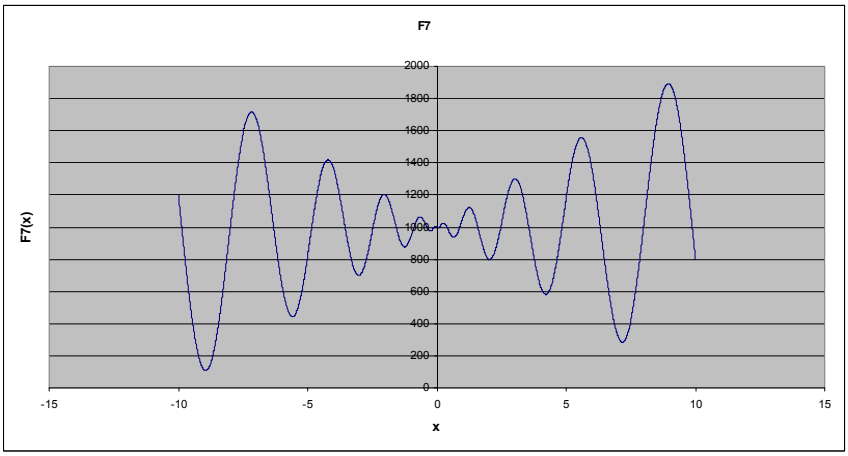


Figure 1. Scaled F7 one-dimensional function in the range (-10.0, 10.0), with shift up = 1000
F7(100*x) = (-1) * (100*x) * sin(sqrt(abs(100*x))) + 1000

2 Jacob2 : genetic algorithm with real representation to the Coulex data analysis

The recent approach to the Coulex optimization was established by using a real representation for genetic algorithm. Additionally, code refactorization was performed and its flexibility to apply new genetic operators or representations was improved.

Jacob v2.0, in its present implementation (Fig. 2), exploits the representation based on real numbers. The selection operator may be chosen from: roulette, tournament and truncation selection. The crossing-over is carried out by drawing a number from the range of the selected parents' parameters with a uniform distribution. The mutation is defined by changing the matrix elements vector with a Gaussian distribution (when it occurs, it is applied to the whole chromosome).

The use of the JACOB2 code to find a solution of the real physics case found some difficulties to reproduce the solution received with previous technique of the optimization. To identify the source of the problem the test procedure with the use of test function was preformed.

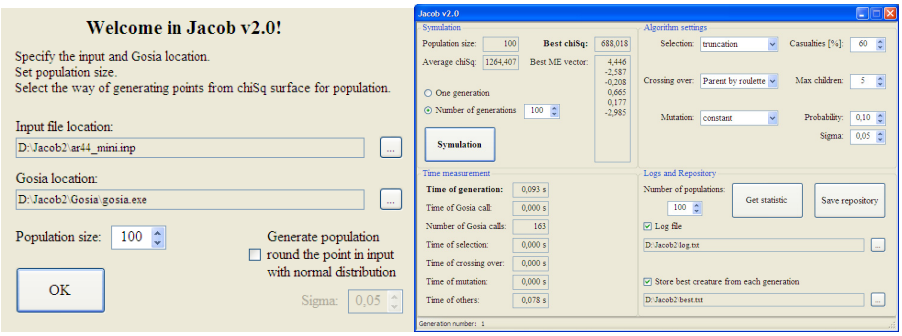


Figure 2. Jacob2

3 Jacob 2 tests with F7

For Jacob2 tests with six dimension F7 function was chosen. Ranges of parameters were defined as not symmetrical to reproduce conditions of the real case. In particular three different groups of parameters' ranges were chosen:

1. (0.1 , 5.0)
2. (-3.0, 3.0)
3. (-0.3 , 0.3)

Each chromosome is a vector of six parameters: one in the first range of variability, next three in the second one, and finally last two parameters are defined in the third range shown above. This happened because in the first range the global minimum is deep and more or less in the middle of the range. In the second – the global minimum is on the periphery and there is more than one minimum closer to the middle of the range. In the third case the minima are not very deep; also the range is fairly narrow. These are the reasons to manifold some ranges – to check the repeatability of finding minima.

Two crucial questions are as follows.

1. How often does the algorithm find the correct (global) minimum after a fixed number of generations?
2. What is the sampling scheme of the algorithm during minimization process?

To find the answers, test simulations were carried out. One thousand times a population of fifty chromosomes evolved for one hundred generations. The chromosomes for populations were generated with uniform distribution on the candidate solutions surface. Two kinds of data were stored:

1. repository of all points (chromosomes) for which objective function value (F7 value) was evaluated,
2. the best point (chromosome) from each population reached during the evolution process.

Genetic operators which are applied are as follows:

- truncation selection with 60% threshold level in each generation,
- mutation with constant mutation probability equals to 10% and standard deviation equals to 0.05.

In the first range (Fig. 3 and 4) the global minimum was easily detected. Almost all starting populations converged to correct minimum. Also the highest density of sampling was around the global minimum – in other minima only slightly increased.

In the second range (Fig. 5 and 6) the global minimum is close to the left limit of the range. Frequency of reaching the global minimum is twice the frequency of reaching the second. The same is with the sampling density.

In the third range (Fig. 7 and 8), perhaps because of the narrow range, the sampling is very much diffused. The most often found minimum is very shallow and close to the middle of the range. The global minimum is again at the limit of the range and only an increase of sampling may be noticed there.

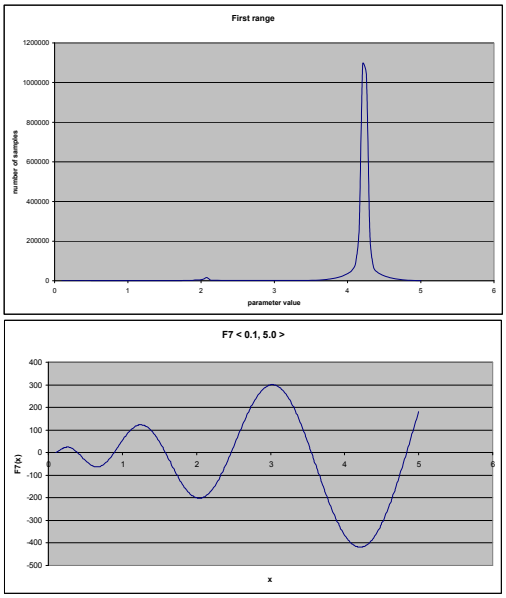


Figure 3. Sampling in repository

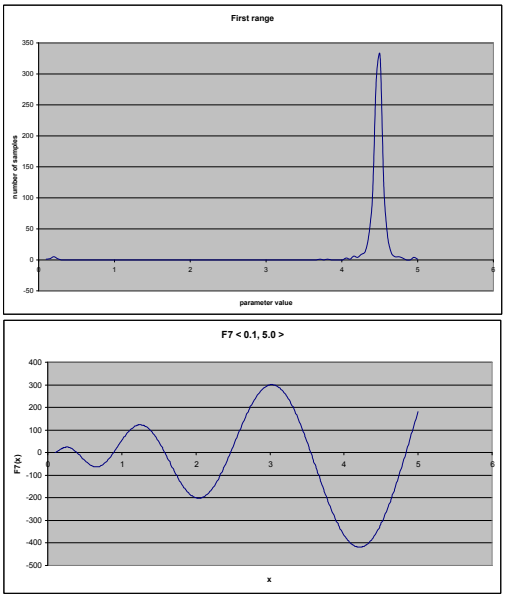


Figure 4. Best chromosome after 100 generations

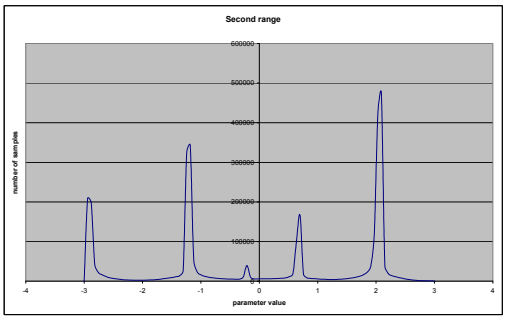


Figure 5. Sampling in repository

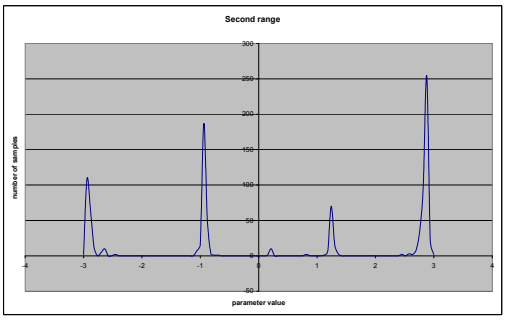
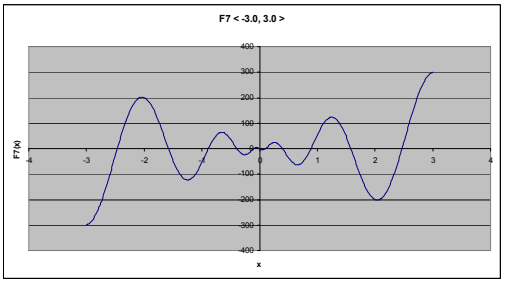
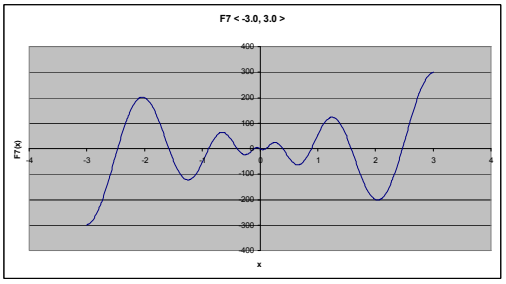


Figure 6. Best chromosome after 100 generations





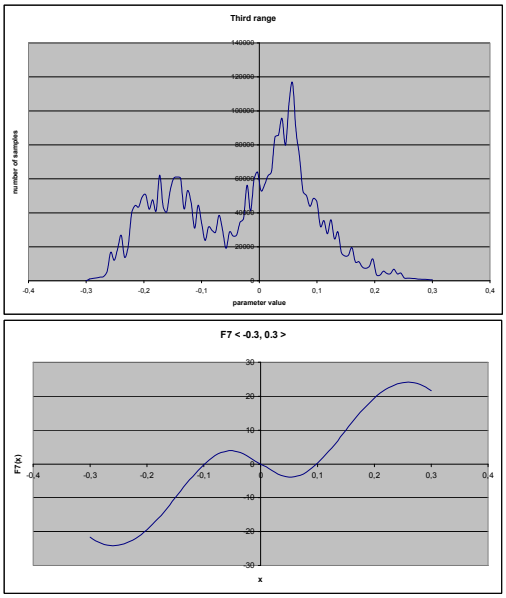


Figure 7. Sampling in repository

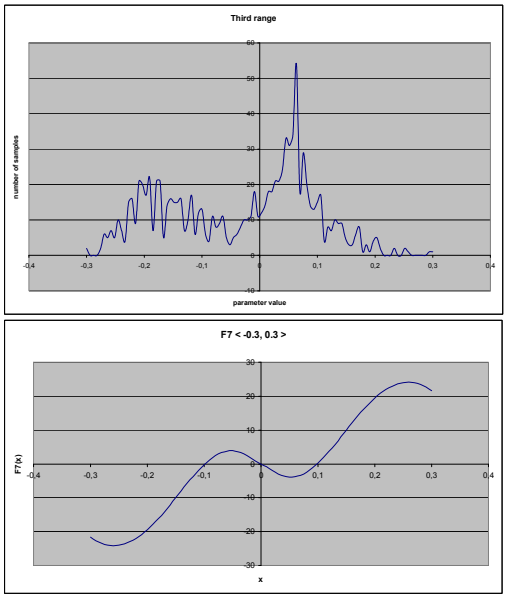


Figure 8. Best chromosome after 100 generations

4 Sampling on the constant objective function

The global minimum in the previous example is never reached simultaneously for all six parameters. Most often it is only the first one and three out of the rest five. What is the reason?

To explain it, we should take a deeper inside view at the genetic algorithm. Let us consider a constant objective function. Then the sampling of the surface should be uniform because there is no reason to increase sampling density in any part of the sampling space.

To check this hypothesis a population of five hundred chromosomes was generated one thousand times. The surface sampling is presented on Figs. 9, 11, and 13. Also one thousand times a population of one hundred chromosomes evolved for fifty generations and it is presented on Figs. 10, 12, and 14).

It is observed (Figs. 9-14) that only the starting distribution is uniform – it is based on a pseudo-random numbers generator. But it is seen that the evolving process concentrates on the sampling in the middle of the range. This effect is independent from the range. It means that the genetic operators cause this phenomenon. Which of them, then?

The selection (truncation selection) kills the chromosomes with too low objective function value. But in this case the objective function value is always constant, so the selection works randomly. There is no concentration ability in it – the sampling should still be uniform.

The mutation randomly changes selected chromosomes with Gaussian distribution. But chromosomes are randomly distributed at the beginning, so the sum of distributions will be a uniform distribution.

The crossing-over is defined as uniformly generated number from the range of parents' parameters values. So the expected value is the average from parents' parameters values and because parents are distributed uniformly, it concentrates in the middle of the range. This

explains the mystery. The crossing-over operator makes not authorized concentration of sampling.

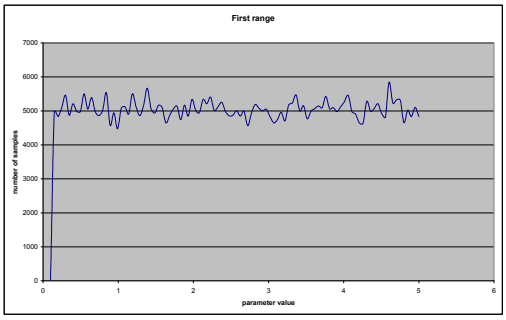


Figure 9. Starting population

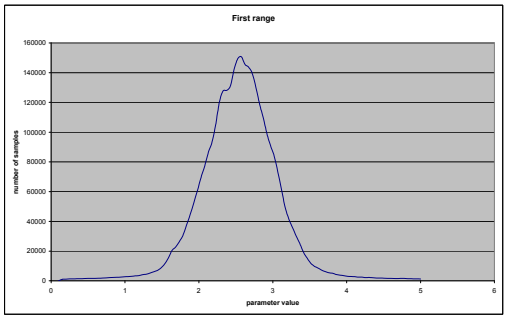


Figure 10. Population after 50 generations

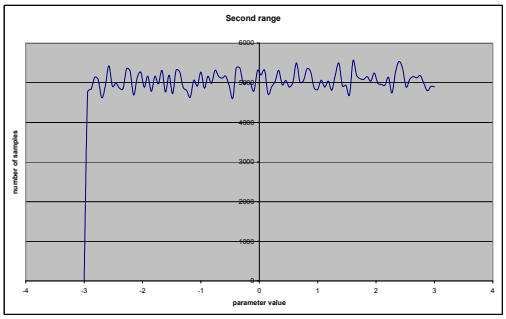


Figure 11. Starting population

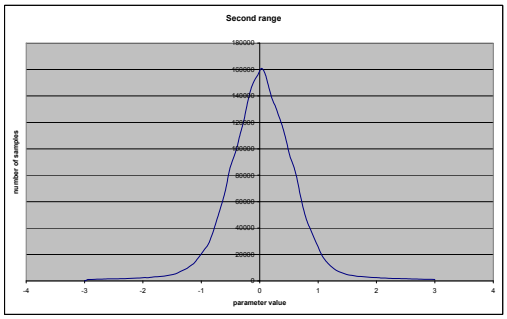


Figure 12. Population after 50 generations

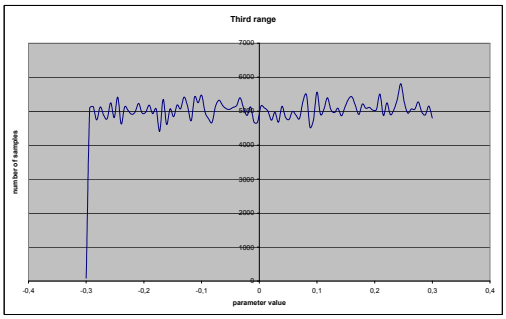


Figure 13. Starting population

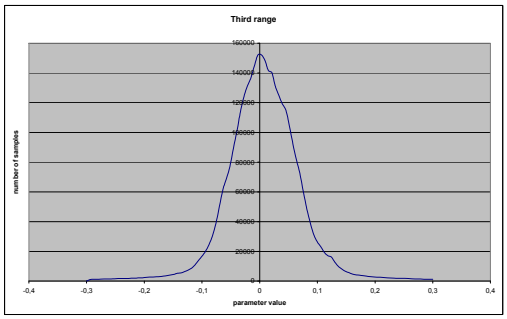


Figure 14. Population after 50 generations

5 Conclusions

The genetic algorithm operators shouldn't impose any increase of sampling resulting form their construction. The sampling density should only be related to the algorithm convergence.

In the presented example, the convergence of crossing over operator perturbs finding the minimum. Global minimum was never reached because in the second range (second, third and forth parameter) global minimum lies on one of the limits of the range. In the third range (fifth and sixth parameter) the convergence which is a result from the depth of minimum is weaker than the convergence of the crossing over operator. So the convergence of crossing-over operator caused concentration of sampling in other part of the surface and as the result disallowed finding of the global minimum.

During construction of genetic algorithm the convergence of genetic operators should be avoided; except of the cases where it is intentional.

The calculations performed with the multidimensional F7 function identified the weakness of the real representation implemented to the genetic algorithm applied for the physics optimization problem. The presented tests procedure can be used to improve the genetic operators and compare the performance with the binary representation of the problem used in the Jacob 1.

The use of the F7 function allows performing the test with an *a priori* known solution of the optimizing problem. It is important since the full calculation of the χ^2 surface in the physical case is not possible due to time consuming multidimensional χ^2 determination. The tests presented in this paper will result in upgrade of genetic operators which will be applied for the real physics case of the COULEX data analysis.

Bibliography

[1] K. Alder, A. Bohr, T. Huus, B. Mottelson, A. Winther - Rev. Mod. Phys. 28:432 (1956)

[2] A. Winther and J. de Boer, 1965, A Computer Program for Multiple Coulomb Excitation, California Institute of Technology, Technical Report,

[3] T. Czosnyka, D. Cline, C.Y. Wu - Bull. Am. Phys. Soc. 28:745 (1983)

[4] D. A. Piętak, P. J. Napiorkowski, A. Pająk, J. Srebrny: An implementation of a genetic algorithm to the Coulex data analysis. Poster on EURISOL-EURONS Town Meeting, Helsinki, 2007

[5] Schwefel, H.-P. Numerical optimization of computer models. Chichester: Wiley & Sons, 1981.