

Gene Expression Programming Approach to Event Selection in High Energy Physics

Liliana Teodorescu

Abstract—Gene Expression Programming is a new evolutionary algorithm that overcomes many limitations of the more established Genetic Algorithms and Genetic Programming. Its first application to high energy physics data analysis is presented. The algorithm was successfully used for event selection on samples with both low and high background level. It allowed automatic identification of selection rules that can be interpreted as cuts applied on the input variables. The signal/background classification accuracy was over 90% in all cases.

Index Terms—Event selection, evolutionary algorithms, gene expression programming.

I. INTRODUCTION

ADVANCED data analysis algorithms have a gradually increasing presence in high energy physics. Neural Networks or Fisher Discriminant techniques are commonly used in many experiments. Other techniques such as Support Vector Machine, Kernel Estimation Technique or Evolutionary Algorithms have also been successfully tested in this field.

Evolutionary Algorithms, such as Genetic Algorithms (GA) [1] and Genetic Programming (GP) [2], are inspired by biological evolutionary theories. In these algorithms the solutions to a problem are represented as individuals which evolve throughout generations due to the interactions with other candidate solutions and the application of genetic operators that create genetic variation. Individuals are entities which encode candidate solutions to a problem (GA), or computer programs as candidate solutions to a problem (GP). In most of the implementations the same entity plays both the genotype (encoder) and the phenotype (candidate solution) roles, and evolution is performed directly on the phenotype, limiting the performance of the algorithms. In GP, for example, many syntactically invalid structures are produced through genetic variation, wasting computational resources.

Gene Expression Programming (GEP) [4] overcomes these limitations of GA and GP. The genotype and phenotype roles are played by different entities in GEP. The individuals of a population are non-linear entities of different size and shape (expression trees) which are encoded as strings of fixed length (chromosomes). This separation and the structural organisation of the chromosomes allow unconstrained genetic modifications, always producing valid expression trees. These characteristics

allow GEP to outperform GP in a remarkable way: more than two orders of magnitude for symbolic regression problems, and more than four orders of magnitude for classification problems [4]. The potential of GEP has started to be exploited in different fields such as data and text mining [5]–[8], electrical circuit modelling [9], and various engineering applications [10], [11].

In high energy physics, Genetic Algorithms were applied mainly to problems such as discrimination and parameter optimisation in both experimental and theoretical studies over the last ten years (for example, see [12]–[15]). Genetic Programming was only recently applied to event selection type problems [16]–[18]. The study presented in this paper is the first application of GEP to particle physics data.

II. GEP FUNDAMENTALS

The GEP algorithm is described in detail in [3] and [4]. The main ideas are summarised here.

A. Algorithm

The algorithm starts with the problem definition, the encoding of the candidate solution of the problem into a chromosome and the definition of the fitness function that describes how good the candidate solution is for the problem at hand. Then an initial population of chromosomes is randomly generated, the chromosomes are translated into expression trees, and the fitness function is evaluated for each chromosome. If a solution of adequate quality is not found, a set of chromosomes is selected and reproduced, creating a new generation of chromosomes. The process is repeated until an optimal solution to the problem is found or a given number of generations is produced.

B. Chromosome Encoding

The candidate solution is encoded into a chromosome composed of one or more genes of equal length. A gene is divided into a head composed of terminals (variables and constants) and functions, and a tail composed only of terminals. The length of the head (h) is an input parameter of the algorithm while the length of the tail (t) is given by:

$$t = h(n - 1) + 1 \quad (1)$$

where n is the largest arity of the functions used in the gene's head.

This head-tail partition of the gene ensures that every function of the gene has the required number of arguments available, making the chromosome correspond to a syntactically correct expression.

Manuscript received January 10, 2006; revised April 25, 2006. This work was supported by the U.K. Particle Physics and Astronomy Research Council (PPARC).

L. Teodorescu is with Brunel University, Uxbridge, Middlesex UBB 3PH, U.K. (e-mail: Liliana.Teodorescu@brunel.ac.uk).

Digital Object Identifier 10.1109/TNS.2006.878571

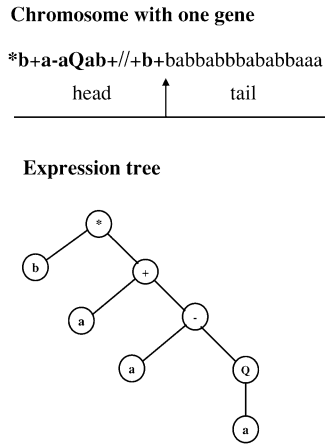


Fig. 1. Example of a chromosome and its corresponding expression tree.

The list of functions and variables to be used in a gene is input information for the algorithm, while the constants are created by the algorithm itself in a range specified by the user. Each gene of a chromosome is translated (decoded) into an expression tree (ET) with the following rules:

- the first element of the gene is placed on the first line of the ET and constitutes its root,
- on each next line of the ET a number of elements equal to the number of arguments of the functions located on the previous line is placed,
- the process is repeated until a line containing only terminals is formed.

This decoding process can be seen as the analog of the expression process of the biological genes encoded in DNA into proteins.

The reverse process, the encoding of the ET into a gene, implies reading the ET from left to right and from top to bottom.

An example of a chromosome with the head length equal to 15 and made of five functions, Q , $*$, $+$ and $-$, (Q being the square root function) and two terminals, a and b , is shown in Fig. 1, with the corresponding ET.

It can be noticed that the ET ends before the end of the gene. This shows that the GEP genes can have non-expressed regions, just like the biological genes that can have regions non-expressed in proteins.

In the case of multigenic chromosomes, the expression trees corresponding to each gene are connected with a linking function defined by the user.

C. Reproduction

The reproduction of the chromosomes is done through two mechanisms: elitism and reproduction with modification.

Elitism is the process through which the best fitted chromosome is replicated unchanged into the next generation.

Reproduction with modification is the process through which the chromosomes are selected and modified with genetic operators producing offspring. It is important to emphasise that the genetic operators are applied on the chromosomes and not on the expression trees, as in GP. This fact, together with the head-tail

organisation of the genes, makes GEP always produce syntactically correct structures during the evolution process.

The chromosome selection is done with the roulette-wheel [19] method. This method allows the fitter individuals to have higher probability of leaving offspring.

In contrast with GA and GP which use mainly cross-over and mutation operators for producing genetic variation, GEP has several genetic operators.

The mutation operator randomly changes an element of a chromosome into another element, preserving the rule that the tails contain only terminals. In the head of the gene a function can be changed in another function or terminal and *vice versa*. In the tail a terminal can only be changed into another terminal.

The transposition operator randomly moves a part of the chromosome to another location in the same chromosome. In GEP there are three kinds of transposable elements:

- short fragments with a function or a terminal in the first position that transpose into the head of genes, except at the root. A sequence with the same number of elements is deleted from the end of the head in order to maintain the structural organisation of the gene.
- short fragments with a function in the first position that transpose to the root of the gene. A sequence with the same number of elements is deleted from the end of the gene head.
- an entire gene that transposes to the beginning of the chromosome.

The recombination or cross-over operator exchanges parts of a pair of randomly chosen chromosomes. In GEP there are three kinds of recombinations:

- one-point recombination in which the parent chromosomes are paired and split up at the same point. The material after the recombination point is exchanged between the two chromosomes, forming two new daughter chromosomes.
- two-point recombination in which the parent chromosomes are paired and two points are randomly chosen where the chromosomes are split. The material between the recombination points is exchanged between the two chromosomes, forming two new daughter chromosomes.
- gene recombination in which entire genes are exchanged between two parent chromosomes, forming two new daughter chromosomes containing genes from both parents.

The power of these operators was studied in detail in [4]. The mutation operator was found the most powerful one. Applied with small rates (0.01–0.1), it introduces diversity in the chromosome population that allows efficient evolution of the solution. Additional chromosome diversity is introduced by the transposition operator. Applied with rates of 0.1 or higher, it adds efficiency to the search process. The recombination operator, however, has an homogenising effect. Applied with moderate rates, around 0.3, and together with the other operators, it helps in maintaining the population steady.

D. Simple Example of GEP Application

GEP has been applied to problems like symbolic regression, parameter optimisation, classification, time series prediction and logic synthesis [4].

The algorithm will be illustrated here with a simple problem of symbol regression (function finding). The following polynomial function is chosen as a target:

$$g(x) = x^3 + x^2 + x + 1. \quad (2)$$

We consider a set of ten points (x_i, y_i) , where x_i was randomly generated in the interval $[-10, 10]$ and $y_i = g(x_i)$. GEP will be used to find the function that fits these points.

The set of functions and terminals that compose the chromosome is chosen by the user based on his knowledge about the problem and the expected solution. A set of four functions, $\{+, -, *, /\}$, and a terminal, $\{x\}$, were chosen for our example.

The structure of the chromosome is also chosen by the user. As no explicit rule exists for this choice, it is recommended to start with simple chromosomes and to increase their complexity gradually. A one gene chromosome was used in our example. The length of the gene head was varied between 1 and 30.

The fitness function that guides the search process heavily depends on the type of problem. Its form must take into account that GEP was developed to maximise the fitness. The following fitness function was chosen for our example:

$$f = 1000 \times \frac{1}{1 + E} \quad (3)$$

where E is the mean squared error:

$$E = \frac{1}{n} \sum_{i=1}^n (C_i - y_i)^2 \quad (4)$$

with C_i being the value of the function found by GEP for the point i and n being the number of points ($n = 10$ in our example).

The number of the chromosomes per generation was chosen 100. Again, no explicit rule exists for this choice. It must be considered, however, that the smaller this value, the higher the number of generations needed to find an optimal solution.

The genetic operator rates can be optimised for each problem. A detailed optimisation was performed in [4] and this example uses the values recommended there: 0.044 for mutation, 0.1 for transposition and 0.3 for recombination.

A common stopping criteria of the run of the algorithm is a maximum number of generations. The exact value depends of the complexity of the chromosome. The more complex the chromosome is, the more generations are needed for the algorithm to converge to a solution. The convergence is indicated by a plateau in the distribution of the fitness of the best candidate as a function of the number of generations. In our example the run of the algorithm was stopped after 1000–5000 generations.

Fig. 2 shows the fitness of the solution found by GEP as a function of the length of the gene head. The fitness gradually increases up to values very close to the maximum fitness. In Fig. 3 the fitness of the best candidate per generation is shown as a function of the number of generations for the run with the length of the gene head equal to ten. Table I summarises the solutions

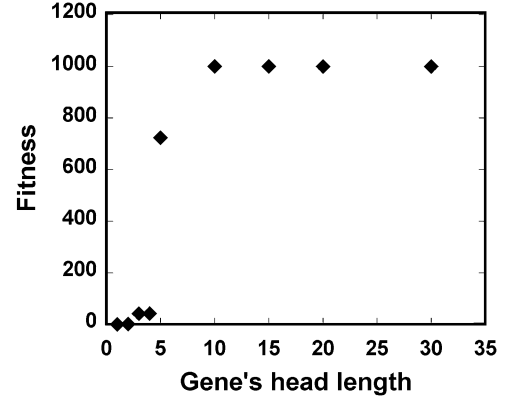


Fig. 2. Fitness of the solution as a function of the length of the gene head (function finding example).

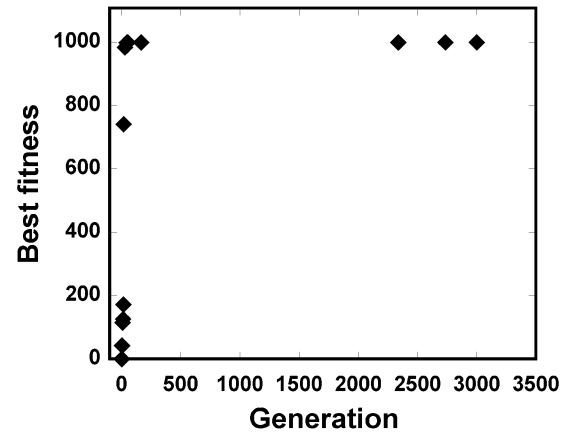


Fig. 3. Fitness of the best candidate as a function of the number of generations (function finding example).

TABLE I
EVOLUTION OF THE SOLUTION FOR THE FUNCTION FINDING EXAMPLE
(Gene Head = 10)

Generation	Fitness	Solution
1	0.004	$x(x - 9.357)/(-8.577) + 9.170x$
2	0.007	$(x - 8.310)/(x + 8.310) - (x + x)$
3	0.009	$((x^2 - x)/2.266 + x - x)x$
4	41.805	$(x^2 - x + 2x)x$
6	42.536	$(x^2 - 0.275/x + x + x - x)x$
9	114.261	$(x^2 + 0.409 + x + 1.1004/x)x$
14	125.596	$(x^2 + 1.531 + x)x$
15	172.201	$(x^2 + 1.409 + x/x + x)x$
17	741.339	$(x^2 + 0.409/x + x/x + x)x$
28	983.718	$(x^2 + 0.871/x + x/x + x)x$
44	998.589	$(x^2 + 1.038/x + x/x + x)x$
52	999.468	$(x^2 + 1.023/x + x/x + x)x$
166	999.964	$x(x^2 + 0.994/x + x/x + x)$
2337	999.965	$x(x^2 - 1.005/x + x + x/x)$
2735	999.982	$x(x/x + 0.996/x + x + x^2)$
3000	999.982	$x(x/x + 0.996/x + x + x^2)$

corresponding to each point showing how the algorithm evolves toward the final solution.

III. METHODOLOGY

Symbolic regression, parameter optimisation or classification are common tasks in high energy physics data analysis. GEP can, potentially, provide alternative methods for solving these

tasks for particle identification, jet calibration, vertex reconstruction or event selection, for example.

The first application of GEP to high energy particle physics presented here is for an event selection problem. Using a statistical learning approach, GEP is used to extract selection criteria for the signal/background classification.

The study was performed using APS (Automatic Problem Solver) 3.0 [20], a Windows based commercial software for function finding, classification and time series analysis with GEP.

The data sample was a set of Monte-Carlo events for K_S production in e^+e^- interaction at ≈ 10 GeV from the BaBar experiment [21]. $e^+e^- \rightarrow q\bar{q}$ events (q being a quark and \bar{q} an antiquark) were generated using JETSET [22] (for q being the u, d, s and c quarks) and EvtGen [23] (for q being the b quark) simulation packages. The generated events were passed through the detector response simulation package [24] and reconstructed with the BaBar analysis software. The reconstructed K_S particles were considered signal if they were associated with the generated K_S particles and their reconstructed π daughters were associated with the π daughters of the generated K_S particles. All the other reconstructed K_S particles were considered background.

The classification rules were extracted from training samples containing 4000–5000 events and tested with other samples containing a similar number of events. The number of events was limited by the processing capabilities of the APS 3.0 software.

Each event contains a set of variables usually used in a standard cut-based analysis for $K_S \rightarrow \pi^+\pi^-$ selection:

- $doca$ —distance between the two π daughters of K_S at the point of closest approach,
- R_{XY} —radius of the cylinder that defines the e^+e^- interaction region,
- $|R_Z|$ —half length of the cylinder that defines the e^+e^- interaction region,
- $|\cos(\theta_{hel})|$ —absolute value of the cosine of the K_S helicity angle,
- SFL — K_S signed flight length defined as the projection of the vector from interaction point to K_S decay vertex on the K_S momentum direction,
- $Fsig$ —statistical significance of the K_S flight length,
- $Pchi$ — χ^2 probability of K_S vertex,
- $Mass$ — K_S reconstructed mass.

These variables, together with the functions from Table II (given as input to the algorithm) and with floating point constants in the range of $(-10, 10)$ (range also given as input) were used to construct the GEP chromosomes. The set of functions were chosen from the functions implemented in the APS 3.0 software in order to construct cut-type selection rules.

Other GEP input parameters were: the length of the gene's head (between 1 and 5), the number of chromosomes per generation (100) and the maximum number of generations (between 500 and 2000, depending on the complexity of the chromosomes). The genetic operator rates were kept as recommended in [4]: 0.044 for mutation, 0.1 for transposition and 0.3 for recombination.

The fitness function was the number of hits (the number of events correctly classified as signal or background). While other

TABLE II
GEP INPUT FUNCTIONS

Function	Definition
AND	
AND1	if $x \geq 0$ AND $y \geq 0$ then 1 else 0
AND2	if $x \leq 0$ AND $y \leq 0$ then 1 else 0
<, >	
\leq , \geq	
$=$, \neq	

fitness functions might be of even more interest for an event selection problem, like the statistical significance of the signal or the purity of the sample, the number of hits was the only fitness function adequate for this problem implemented in the APS 3.0 software.

The performance of the algorithm was analysed in terms of the following parameters:

- classification accuracy Ac_{class} defined as the ratio of the total number of events correctly classified (signal and background) to the total number of events of the sample; this is the parameter optimised by the algorithm.
- signal (background) efficiency, $Ef_{sig}(Ef_{bkg})$ defined as the ratio of the correctly classified signal (background) events to the total number of signal (background) events of the sample,
- purity (P) defined as the ratio of the correctly classified signal events to the total number of events classified as signal, correctly or incorrectly.

IV. ANALYSIS AND RESULTS

Two analyses, of data samples with different level of background, were performed. The signal to background ratio was 5.7 and 0.5 for the low and high background data samples, respectively.

A. Analysis of the Low Background Data Sample

The training sample contained a total number of 3985 events, 3390 being signal events and 595 being background events. Each event contained five variables: $doca$, R_{XY} , $|R_Z|$, $|\cos(\theta_{hel})|$ and SFL .

The classification rules found by the GEP algorithm, using chromosomes consisting of one gene with the head varying from 1 to 5 symbols in length, are summarised in Table III, together with algorithm performance parameters. A classification accuracy of around 92% is obtained in all cases indicating that simple chromosome configurations are sufficient for finding good solutions to this problem. For this classification accuracy, the signal efficiency was around 99% and the purity around 92%.

The best result is obtained with chromosomes containing one gene with the head made of two elements. For this case, the fitness of the best candidate as a function of the number of generations is shown in Fig. 4. A highly fit candidate is obtained very quickly, in less than 200 generations. In each generation, 100 candidate solutions are evaluated. Then a plateau is reached indicating the convergence of the search.

Similar selection criteria (cuts on R_{XY} and $doca$) are used in a standard cut based analysis of K_S decay data in BaBar experiment [25]. In this standard analysis a high purity sample was

TABLE III
RESULTS OF THE ANALYSIS OF THE LOW BACKGROUND DATA SAMPLE

Head length	Classification rule	Best fitness	$Ac_{class}(\%)$	$Ef_{sig}(\%)$	$Ef_{bkg}(\%)$	$P(\%)$
1	$R_{XY} \leq 0.27cm$	3673	92.2	99.1	52.8	92.3
2	$R_{XY} \leq 0.27cm$ AND $doca \leq 1.00cm$	3681	92.4	98.8	56.1	92.8
3	$R_{XY} \leq 0.25cm$	3669	92.1	98.9	53.3	92.3
4	$R_{XY} \leq 0.20cm$	3666	92.0	98.5	54.9	92.6
5	$R_{XY} \leq 0.32cm$	3661	91.9	99.2	49.9	91.9

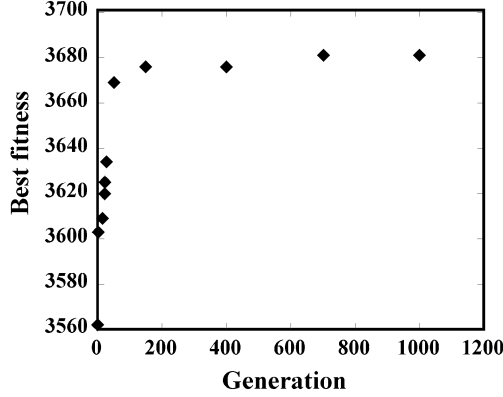


Fig. 4. Fitness of the best candidate as a function of the number of generations (low background data sample).

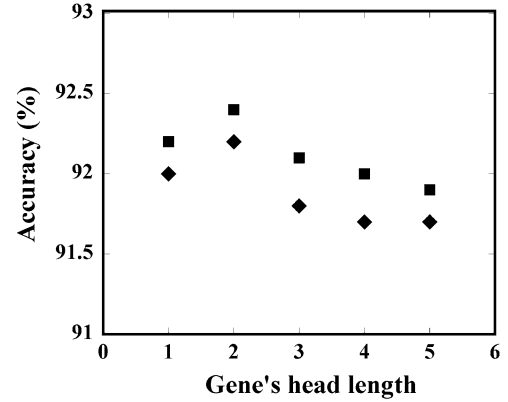


Fig. 5. Classification accuracy for training (squares) and test (diamonds) data samples (low background sample).

TABLE IV
PERFORMANCE PARAMETERS OF THE GEP AND THE CUT BASED ANALYSIS
SELECTION CRITERIA

Selection criteria	$Ac_{class}(\%)$	$Ef_{sig}(\%)$	$Ef_{bkg}(\%)$	$P(\%)$
$R_{XY} \leq 0.27cm$	92.2	99.1	52.8	92.3
$R_{XY} \leq 0.27cm$ $doca \leq 1.00cm$	92.4	98.8	56.1	92.8
$R_{XY} \leq 0.25cm$ $doca \leq 1.00cm$ $ R_Z \leq 3.2cm$	92.4	98.7	56.5	92.8
$R_{XY} \leq 0.25cm$ $aoca \leq 1.00cm$ $ R_Z \leq 3.2cm$ $SFL > 0cm$	92.4	98.7	56.5	92.8
$R_{XY} \leq 0.25cm$ $doca \leq 1.00cm$ $ R_Z \leq 3.2cm$ $SFL > 0cm$ $ \cos(\theta_{hel}) $	84.3	86.5	71.8	94.6

required and additional cuts on the SFL, $|R_Z|$ and $|\cos(\theta_{hel})|$ variables were applied. The effect of these cuts, applied one after the other, on the performance parameters used in the GEP analysis is summarised in Table IV. It can be seen that the cuts on the $|R_Z|$ and SFL variables do not significantly improve the performance parameters, indicating that GEP found only the most powerful selection criteria (cuts on the R_{XY} and $doca$ variables in this case). The cut on the $|\cos(\theta_{hel})|$ variable slightly increases the purity of the sample (from 92.8% to 94.6%) with a reduction in the signal efficiency from 98.7% to 86.5% and in the classification accuracy from 92.4% to 84.3%. As GEP uses the number of hits as the fitness function, maximising the classification accuracy, this last cut is not included in the optimal solution found by the algorithm.

The predictive power of the selection criteria found by GEP was tested on a test data sample containing the same number of events as the training data sample. The algorithm performance

parameters obtained after the test were similar with those obtained after the training. An example is shown in Fig. 5 where the classification accuracy for both training and test data samples, obtained with different chromosomes configurations, is presented.

B. Analysis of the High Background Data Sample

The training sample contained a total number of 5000 events, comprising 1264 signal events and 3736 background events. Each event contained all eight variables described in Section III.

The two most powerful selection criteria (cuts on the F_{sig} and R_{XY} variables in this case) are again found by the algorithm using simple chromosome configurations. The results are summarised in Table V. The classification accuracy is 90–94% for which the signal and background efficiencies are over 90%. The purity is lower in this case compared to the previous analysis, between 75% and 85%. This indicates that a very high purity sample can not be obtained from a high background sample using the number of hits as a fitness function.

The best two results (identical) are obtained with chromosomes containing one gene with the head made of two and five elements, respectively. The evolution of the fitness of the best candidate with the number of generations is shown in Fig. 6 for chromosomes where the length of the gene's head equals two. Convergence is reached later for this case, after 1400 generations.

A high predictive power of the selection criteria was also found by testing them on a test data sample containing 5000 events. The classification accuracy for both training and testing data samples, obtained with different chromosome configurations, is presented in Fig. 7. The results obtained for both low and high background samples required very little computing resources. For the most complex configuration, approximately 20

TABLE V
RESULTS OF THE ANALYSIS OF THE HIGH BACKGROUND DATA SAMPLE

Head length	Classification rule	Best fitness	$Ac_{class}(\%)$	$Ef_{sig}(\%)$	$Ef_{bkg}(\%)$	$P(\%)$
1	$Fsig \geq 5.40$	4549	91.0	91.9	91.0	77.0
2	$Fsig \geq 5.34 \text{ AND } R_{XY} \leq 1.00cm$	4697	93.9	91.3	94.8	85.7
3	$Fsig \geq 4.80$	4548	91.0	92.8	90.3	76.5
4	$Fsig \geq 4.50 \text{ AND } R_{XY} \leq 1.00cm$	4701	94.0	92.7	94.5	85.0
5	$Fsig \geq 5.35 \text{ AND } R_{XY} \leq 1.00cm$	4697	93.9	91.3	94.8	85.7

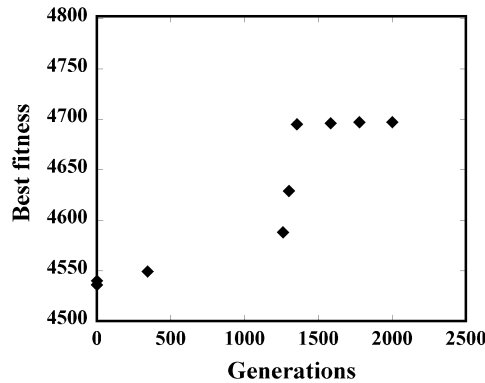


Fig. 6. Fitness of the best candidate as a function of the number of generations (high background data sample).

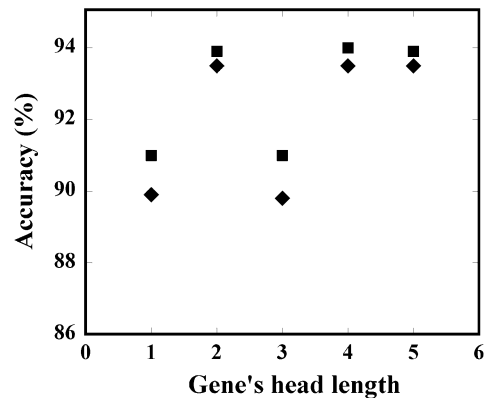


Fig. 7. Classification accuracy for training (squares) and test (diamonds) data samples (high background data sample).

minutes were needed in order to reach convergence of the search process, on a computer with a 2.8 GHz processor.

V. CONCLUSION

A first application of the GEP algorithm for event selection in high energy physics data analysis was presented. The results of this study indicate GEP to be a potentially powerful technique for automatic identification of selection rules that can be interpreted as cuts applied on the input variables.

The most powerful cuts are found and optimised very quickly for both low and high background data samples. The classification accuracy is over 90% in all cases.

The cuts that do not significantly improve the classification power are not included in the solution found by GEP. This suggests that GEP can be also used as a tool for identifying the significant cuts for a certain analysis, helping in reducing the systematic error of the final result.

The depth of the study presented here was limited by the capabilities of the software used. While APS 3.0 is a nice tool for Windows based applications which need to process relatively small amounts of data, it is not flexible enough for more complex analysis with particle physics data. No such flexible software implementation of GEP is available so far.

This lack of software resources and the promising results of this first application of GEP to particle physics data has motivated us to start our own software development. The project is underway. It is our intention to make this software freely available to the scientific community for evaluation and exploitation.

Once this software tool is available, the analysis presented here will be extended using other fitness functions, more input functions and more data, as well as performing comparisons with other statistical learning methods.

ACKNOWLEDGMENT

The author would like to thank Prof. S. Watts from Brunel University for useful discussions.

REFERENCES

- [1] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: Univ. of Michigan Press, 1975.
- [2] J. R. Koza, *Genetic Programming: On the Programming of the Computers by Means of Natural Selection*. Cambridge, MA: MIT Press, 1992.
- [3] C. Ferreira, "Gene expression programming: A new adaptive algorithm for solving problems," *Complex Systems*, vol. 13, no. 2, pp. 87–129, 2001.
- [4] C. Ferreira, *Gene Expression Programming: Mathematical Modelling by an Artificial Intelligence*. Portugal: Angra do Heroismo, 2002.
- [5] C. Zhou, W. Xiao, P. C. Nelson, and T. M. Tirpak, "Evolving accurate and compact classification rules with gene expression programming," *IEEE Trans. Evol. Comput.*, vol. 7, no. 6, pp. 519–531, Dec. 2003.
- [6] C. Zhou, P. C. Nelson, W. Xiao, and T. M. Tirpak, "Discovery of classification rules by using gene expression programming," in *Proc. Int. Conf. Artificial Intelligence*, Las Vegas, NV, 2002, pp. 1355–1361.
- [7] Q. Li., Z. Cai, S. Jiang, and L. Zhu, "Gene expression programming in prediction," in *Proc. World Congress Intelligent Control and Automation*, Hangzhou, China, 2004, pp. 2171–2175.
- [8] Z. Xie, "Using gene expression programming to construct sentence ranking functions for text summarization," in *Proc. 20th Int. Conf. Computational Linguistics*, Geneva, Switzerland, 2004, pp. 23–27.
- [9] H. Cao, J. Yu, and L. Kang, "An evolutionary approach for modeling the equivalent circuit for electrochemical impedance spectroscopy," in *Proc. Congress Evolutionary Computation*, Canberra, Australia, 2003, pp. 1819–1825.
- [10] M. Saltan and T. Serdal, "Comparative analysis of using artificial neural networks and gene expression programming in backcalculation of pavement layer thickness," *Indian J. Eng. Mater. Sci.*, vol. 12, Feb. 2005.
- [11] Z. Cai, Q. Li., S. Jiang, and Y. Zhao, "Symbolic regression based on GEP and its application in predicting amount of gas emitted from coal face," in *Proc. 2004 Int. Symp. Safety Science and Technology*, Shanghai, China, pp. 637–641.
- [12] D. G. Ireland, "A genetic algorithm analysis of N^* resonances in $p(\gamma, K^+) \Lambda$ reactions," *Nucl. Phys. A*, vol. A740, pp. 147–167, 2004.

- [13] B. C. Allanach, Genetic Algorithms and Experimental Discrimination of SUSY Models [Online]. Available: <http://arXiv.org/hep-ph/0406277>, 2004.
- [14] S. Abdullin, "Genetic algorithm for SUSY trigger optimisation in CMS detector at LHC," *Nucl. Instrum. Methods Phys. Res. A*, vol. A502, pp. 693–695, 2003.
- [15] Y. Azusa, Genetic Algorithm for SU(N) Gauge Theory on a Lattice [Online]. Available: <http://arXiv.org/hep-lat/9808001>, 1998.
- [16] K. Cranmer and R. S. Bowman, PhysicsGP: A Genetic Programming Approach to Event Selection [Online]. Available: <http://arXiv.org/physics/0402030>
- [17] J. M. Link, Application of Genetic Programming to High Energy Physics Event Selection [Online]. Available: <http://arXiv.org/hep-ex/0503007>, 2005.
- [18] J. M. Link, Search for $\Lambda_c^+ \rightarrow p K^+ \pi^-$ and $D_s^+ \rightarrow K^+ K^+ \pi^-$ Using Genetic Programming Event Selection [Online]. Available: <http://arXiv.org/hep-ex/0507103>, 2005.
- [19] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA: Addison-Wesley, 1989.
- [20] Automatic Problem Solver - APS 3.0 [Online]. Available: <http://www.gepssoft.com>
- [21] B. Aubert, "The BaBar detector," *Nucl. Instrum. Methods Phys. Res. A*, vol. A479, pp. 1–116, 2002.
- [22] T. Sjostrand, "High-energy-physics event generation with PYTHIA 5.7 and JETSET 7.4," *Comput. Phys. Commun.*, vol. 82, pp. 74–89, 1994.
- [23] D. Lange, "The EvtGen particle decay simulation package," *Nucl. Instrum. Methods Phys. Res. A*, vol. A462, pp. 152–155, 2001.
- [24] S. Agostinelli, "GEANT4—A simulation toolkit," *Nucl. Instrum. Methods Phys. Res. A*, vol. A506, pp. 250–303, 2003.
- [25] B. Aubert, Search for Strange Pentaquark Production in $e^+ e^-$ Annihilations at $\sqrt{s} = 10.58$ GeV and in $\Upsilon(4S)$ Decays [Online]. Available: <http://arXiv.org/hep-ex/0408064>, 2004.