

Genetic Programming Model for Hadronic Collisions

^{1,2} Moaaz A. Moussa

¹ Ain Shams University, Faculty of Education, Physics Department, Roxi, Cairo, Egypt.

² Buraydah Colleges, Physics Department, Al-Qassim, Buraydah, King Abdulaziz Road, East Qassim University, P.O.Box 31717, KSA.

moaz2030@yahoo.com

¹ Mahmoud Y. El-Bakry

Tabuk University, Faculty of Sciences, Physics Department, Tabuk, KSA.

^{3,4} A. Radi, ^{3,5} E. El-dahshan, ¹ M. Tantawy

³ Ain Shams University, Faculty of Sciences, Physics Department, Abbassia, Cairo, Egypt.

⁴ The British University in Egypt (BUE).

⁵ Egyptian E-Learning University- 33 El-mesah St., El-Dokki- Giza- Postal Code 12611, Egypt.

Abstract— High Energy Physics (HEP) is in need of powerful efficient techniques for various analysis tasks. Genetic Programming (GP) is a powerful technique that can be used for solving these tasks. In this paper, Genetic programming (GP) has been used to discover a function that calculates charged particles multiplicity distribution of created pions from antiproton-neutron and proton-neutron interactions at high energies. The predicted distributions from the GP-based model are compared with the available experimental data. The discovered function of GP model has proven matching better for experimental data.

Index Terms— Genetic Programming; Hadronic Collisions; High Energy Physics.

1 INTRODUCTION

One of the fundamental interactions in high-energy physics (HEP) is the antiproton-neutron ($\bar{p}-n$) and proton-neutron ($p-n$) interactions particularly above the pion production threshold. Extremely high energy collisions are required to get the fundamental particles close enough to study and understand the interactions between them [1–6]. Different models are provided for the hadron structure [7–10], such as the three-fireball model [11], quark models [12–14], fragmentation model [15–17] and many others.

The application of artificial intelligence (or the machine learning) such as genetic programming (GP) has a strong presence in the high energy physics [18–22]. The effort to understand the interactions of fundamental particles require complex data analysis for which machine learning (ML) algorithms are vital. Machine learning (ML) algorithms are becoming useful as alternate approaches to conventional techniques [23]. The more complex behavior of the high and ultrahigh energy interactions due to the nonlinear relationship between the interaction parameters and the output often becomes complicated. In this sense, ML techniques such as artificial neural network [24], genetic algorithm [25] and genetic programming [26] can be used as alternative tool for the simulation of these interactions [18–22, 27–32].

The motivation of using a GP approach is its ability to evolve a model based entirely on prior data without the need of making underlying assumptions. Another motivation for

applying such machine learning approach (e.g. GP) is simply the lack of knowledge (in most cases) about the mathematical dependence of the quantity of interest on the relevant measured variables [33].

In the present work, we illustrate the GP technique to model the multiplicity distribution of charged pions for different beams in hadronic collisions. The history of studies of these interactions is therefore very long and extremely interesting from both the experimental and theoretical points of view [34, 35]. Making use of the capability of the evolutionary computation, the present work uses the genetic programming to model the charged particles multiplicity distribution for ($\bar{p}-n$) and ($p-n$) interactions at different lab momenta. Also, GP has been used to discover a function that calculates multiplicity distribution for different beams. The rest of the paper is organized as follows: Sec. 2 gives a review to the basics of the GP technique. Section 3 explains how genetic programming is used on modeling the hadron-hadron collisions. Finally, the results and conclusion are provided.

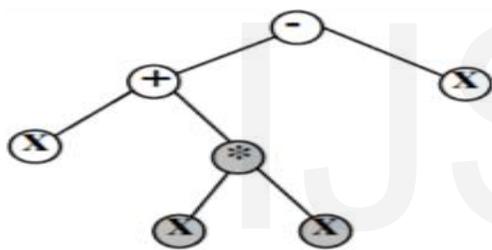
2 GENETIC PROGRAMMING OVERVIEW

Genetic programming is an extension to Genetic Algorithms (GA). GA is an optimization and search technique based on the principles of genetics and natural selection. A GA allows a population composed of many individuals (chromosome) to evolve under specified selection rules to a state that maximizes the fitness (i.e. minimizes the cost function). The

GP is similar to genetic algorithms but unlike the latter its solution is a computer program or an equation as against a set of numbers in the GA. A good explanation of various concepts related to GP can be found in Koza (1992) [26, 36].

In GP, a random population of individuals (equations or computer programs) is created, the fitness of individuals is evaluated and then the parents are selected out of these individuals. The parents are then made to yield offsprings by following the process of reproduction, mutation and crossover.

The creation of offsprings continues (in an iterative manner) until a specified number of offsprings in a generation are produced and further until another specified number of generations are created. The resulting offsprings at the end of all this process is the solution of the problem. The GP thus transforms one population of individuals into another one in an iterative manner by following the natural genetic operations like reproduction, mutation and crossover. Each individual contributes with its own genetic information to the building of new ones (offsprings) adapted to the environment with higher chances of surviving. This is the basis of genetic algorithms and programming. The representation of a solution for the problem provided by the GP algorithm is a tree (Fig. 1).



“Figure 1. Tree representation of the equation $(X + X^2 - X)$ ”

3 GENETIC PROGRAMMING TECHNIQUE

Genetic programming is a technique that mimics natural evolution and improvement of life through reproduction to find a computer program that solves a particular task. It is inspired by the Darwinian principle — the most fit chromosome duals have the greatest chance of surviving and passing into the next generation [37]. Each chromosome contributes with its own genetic information to the building of new ones (offsprings) adapted to the environment with higher chances of surviving. This is the basis of genetic algorithms and programming [38]. Genetic programming searches the space of computer programs, or the space of functional forms specified by compositions of functions from a function set acting on terminals from the terminal set. The chromosome represents the model of the problem solution using trees. A tree is a model representation that contains nodes and leaves. Nodes are mathematical operators from the specified function set. Leaves are terminals from the specified terminal set [38]. Table 1 shows some typical functions and terminals used in GP.

Trees are manipulated through the basic genetic operators: crossover (sexual recombination operation), mutation (asexual operation), and reproduction.

TABLE 1

TYPICAL FUNCTIONS AND TERMINALS USED IN GP

Functions		Terminals	
Kind	Examples	Kind	Examples
Mathematical	Sin, cos, exp, log	Variables	x, y
Arithmetic	$+, *, /$	Constant values	3, 0.45
Boolean	AND, OR, NOT	0-arity functions	Rand, go-left
Conditional	IF, THEN-ELSE,	Random constant	random
Looping	For, Repeat		

Crossover (Sexual Recombination) Operation: In the crossover or sexual recombination operation, two parental programs are probabilistically selected from the population based on fitness. The two parents participating in crossover are usually of different sizes and shapes. A crossover point is randomly chosen in the first parent and a crossover point is randomly chosen in the second parent. Then the subtree rooted at the crossover point of the first, or receiving, parent is deleted and replaced by the subtree from the second, or contributing, parent. Crossover is the predominant operation in genetic programming (and genetic algorithm) works and is performed with a high probability.

Mutation Operation: In the mutation operation, a single parental program is probabilistically selected from the population based on fitness. A mutation point is randomly chosen, the subtree rooted at that point is deleted, and a new subtree is grown there using the same random growth process that was used to generate the initial population. This asexual mutation operation is typically performed sparingly (with a low probability of, say, 1% during each generation of the run).

Reproduction Operation: The reproduction operation copies a single chromosome, probabilistically.

In order to apply the genetic programming technique to a problem, one must first perform the preparatory steps and the executional steps [39]. The preparatory steps are the problem-specific and domain-specific steps that are performed by the human user prior to launching a run of the problem-solving method. The executional steps are automatically executed during a run of the problem-solving method. The five major preparatory steps for the basic version of genetic programming require a human user to specify:

- I) the set of terminals,
- II) the set of primitive functions,
- III) the fitness measure,
- IV) certain GP parameters (see Table 2) for controlling the run, and
- V) a termination criterion and method for designating the result of the run.

The fitness function defines the quality of chromosome as a solution to the problem. The dataset is divided into two

parts: one is for training and the second for validation. The training dataset is used to obtain the model and the validation dataset is used to measure the accuracy of the model with data that was not used in training. The fitness function evaluates how accurate the mathematical model.

TABLE 2
 DEFINITION OF THE GP PARAMETERS.

GP parameters	Definition
Population size	The number of chromosomes in a population Generation Size — The number of iterations of the main selection/operation loop
Maximum tree size	The maximum depth of an expression tree. (This is necessary since crossovers tend to increase the average size of a population, and this inadvertently increases the run time of each generation.)
Mutation rate	How often mutations occur
Reproduction rate	How often reproductions occur
Crossover rate	How often crossovers occur

4 RESULTS AND DISCUSSION

The GP is implemented using the experimental data to simulate multiplicity distribution of charged particles P (n_{ch}) for ($\bar{p}-n$) and ($p-n$) collisions at the given range of energies. The GP model was constructed with training sets and the accuracy was verified by the test sets. In order to generate the GP model we have implemented the GP steps (Fitness evaluation, reproduction, crossover and mutation) that were mentioned in Section 3. Table 3 lists the values of the control parameters and the set of function genes that are used in modeling the multiplicity distribution. Our discovered function is generated using the obtained control GP parameters as follows,

$$P(n_{ch}) = (\text{sqrt}(\log(X + (5/Y)))) / (\text{sqrt}(\exp(X) / ((Y/7) + (3/2)))) + ((\text{sqrt}(Y) / (1 + 2)) + ((2/5) + (2/3))));$$

where the actual parameters are, X = number of charged particles (n_{ch}), Y = lab momentum (P_L). After simplification and putting the corresponding values, the final form of the discovered equation becomes

$$P(n_{ch}) = (\text{sqrt}(\log(n_{ch} + (5/P_L)))) / ((\text{sqrt}(\exp(n_{ch})) / ((P_L/7) + (3/2)))) + ((\text{sqrt}(P_L) / (1 + 2)) + ((2/5) + (2/3))));$$

This discovered function has been used to predict the multiplicity distribution of pions for antiproton- neutron ($\bar{p}-n$)

and proton-neutron ($p-n$) interactions.

Simulation results based on GP model, for modeling the multiplicity distribution of pions for antiproton-neutron ($\bar{p}-n$) and proton-neutron ($p-n$) interactions at $P_L = 50, 80 \text{ GeV}/c$ for $\bar{p}-n$ and $P_L = 100, 200 \text{ GeV}/c$ for $p-n$ (the training cases) are given in Fig. 2 (a, b, c, d) respectively. While Fig. 3 describes the predicted results of $P_L = 400 \text{ GeV}/c$ for $p-n$ interaction, we notice that the curves (for training cases and prediction case) obtained by the trained GP model show a best fitting to the experimental data in the five cases. Then, the GP model is able to exactly model for multiplicity distribution at lab momenta for different beams in h-h collisions. If the large dataset is used in training, the best GP model is obtained.

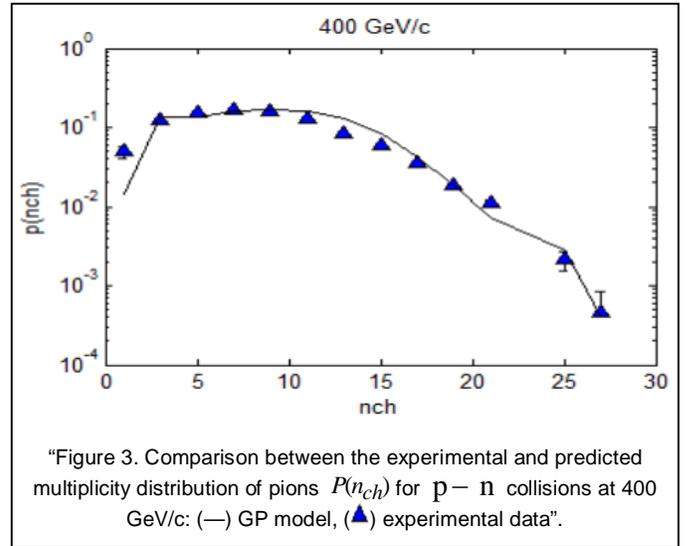
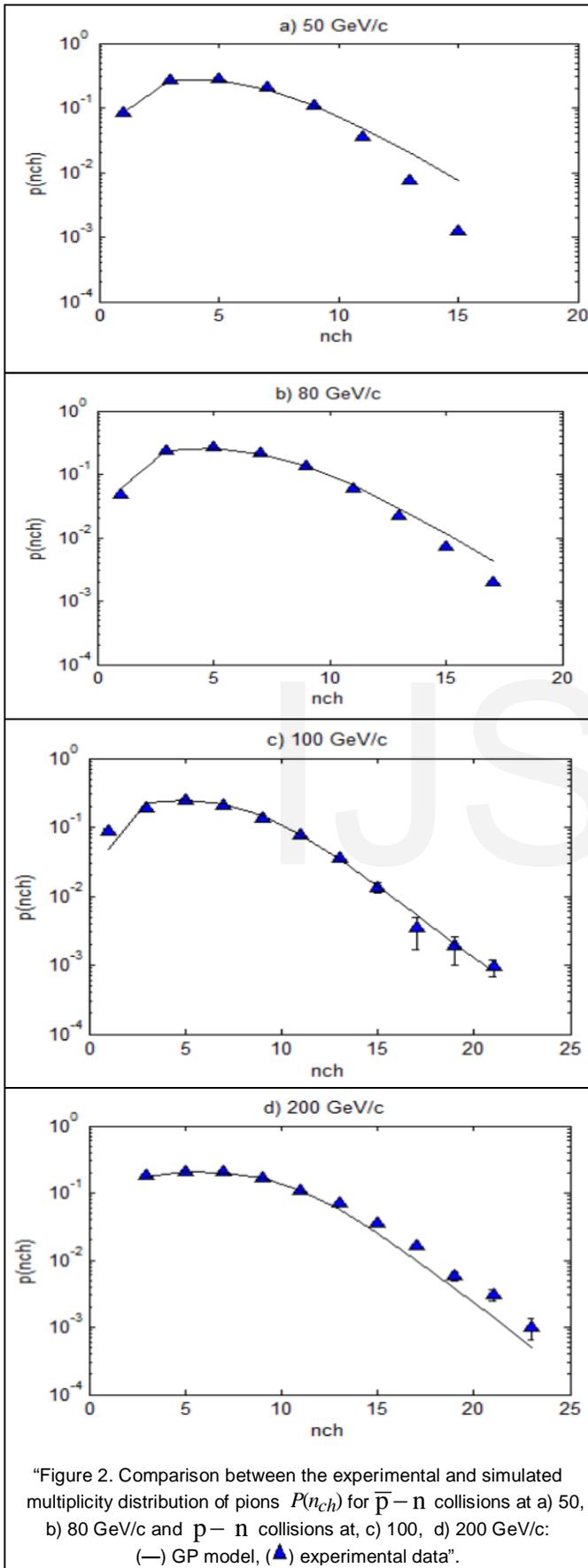
TABLE 3

LISTS THE VALUES OF THE CONTROL PARAMETERS USED IN MULTIPPLICITY DISTRIBUTION

GP Parameters	Values
Generations	1000
Populations	40000
Function set	*,/, -, +, log, sqrt, sin, cos
Terminal Set	{constant, X, Y}
Fitness function	SSE
Selection method	Elites, rank and roulette
Mutation rate	0.01
Crossover rate	0.9

5 CONCLUSION

Genetic programming, with its advantage of discovering mathematical equations, has been shown to be an efficient method for modeling the antiproton-neutron ($\bar{p}-n$) and the proton-neutron ($p-n$) interactions particularly above the pion production threshold. This paper presents an efficient approach for calculating the multiplicity distribution P (n_{ch}) of charged particles, at different values of high energies through the obtained discovered functions. The discovered function shows an excellent match to the experimental data. Moreover, the discovered function is capable of predicting the experimental data that are not used in the training set. The present study has shown that the GP approach can be employed successfully to model the h-h interactions at high energies. Finally, we conclude that GP has become one of important research areas in the field of hadron-hadron collisions.



REFERENCES

- [1] B. Alessandro et al. (ALICE Collaboration), J. Phys. G: Nucl. Part. Phys. 32, 1295 (2006).
- [2] F. Carminati et al. (ALICE Collaboration), J. Phys. G: Nucl. Part. Phys. 30, 1517 (2004)
- [3] D.G. d’Enterria et al. (CMS Collaboration), J. Phys. G: Nucl. Part. Phys. 34, 2307 (2007)
- [4] G.L. Bayatian et al. (CMS Collaboration), J. Phys. G: Nucl. Part. Phys. 34, 995 (2007)
- [5] G. Aad et al. (ATLAS Collaboration), arXiv:0901.0512v4 [hep-ex]
- [6] A. Augusto Alves et al. (LHCb Collaboration), J. Instrum. 3, S08005 (2008)
- [7] M. Tantawy, M. El Mashad, M.Y. Elbakry, Indian J. Phys. Pt-A 72, 110 (1998)
- [8] E. Fermi, Prog. Theor. Phys. 5, 570 (1951)
- [9] E. Fermi, Phys. Rev. 81, 683 (1950)
- [10] J. Ranft, Phys. Lett. B 31, 529 (1970)
- [11] Cai Xu, Chao Wei-qin, Meng Ta-chung, Phys. Rev. D 33, 1287 (1986)
- [12] Y. Nambu, Sci. Am. 235, 45 (1976)
- [13] M. Gyulassy, Prog. Part. Nucl. Phys. 15, 403 (1985)
- [14] L.S. Kisslinger, Nucl. Phys. A 446, 479 (1985)
- [15] M. Jacob, R. Slansky, Phys. Rev. D 5, 1847 (1972)
- [16] R. Hwa, Phys. Rev. D 1, 1790 (1970)
- [17] R. Hwa, Phys. Rev. Lett. 26, 1143 (1971)
- [18] L. Teodorescu, D. Sherwood, Comput. Phys. Commun. 178, 409 (2008)
- [19] L. Teodorescu, IEEE T. Nucl. Sci. 53, 2221 (2006)
- [20] J.M. Link, Nucl. Instrum. Meth. A 551, 504 (2005)
- [21] S. Yaseen El-Bakry, Amr Radi, Int. J. Mod. Phys. C 18, 351 (2007)
- [22] E. El-dahshan, A. Radi, M.Y. El-Bakry, Int. J. Mod. Phys. C 20, 1817 (2009)
- [23] S. Whiteson, D. Whiteson, Eng. Appl. Artif. Intel. 22, 1203 (2009); Moaaz A. Moussa, Mahmoud Y. El-Bakry, A. Radi, E. El-Dahshan, M. Tantawy, “Artificial Neural Network Modeling in Hadrons Collisions”, International Journal of Scientific and Engineering Research Vol. 3, Issue 8, August (2012)
- [24] S. Haykin, Neural Networks: A Comprehensive Foundation (IEEE Press and Macmillan College Publishing Company, New York, NY, (1994)

- [25] J.H. Holland, *Adaptation in Natural and Artificial Systems* (University of Michigan Press, Ann Arbor, 1975)
- [26] J.R. Koza, *Genetic Programming: On the Programming of Computers by means of Natural Selection* (The MIT Press, Cambridge, MA, 1992)
- [27] K. Cranmer, R.S. Bowman, *Comput. Phys. Commun.* 167, 165 (2005)
- [28] M.Y. El-Bakry, A. Radi, *Int. J. Mod. Phys. C* 18, 329 (2007)
- [29] M.Y. El-Bakry, K.A. El-Metwally, *Chaos Soliton. Fract.* 16, 279 (2003)
- [30] K.A. El-Metwally, T.I. Haweel, M.Y. El-Bakry, *Int. J. Mod. Phys. C* 11, 619 (2000)
- [31] E. El-dahshan, A. Radi, M.Y. El-Bakry, *Int. J. Mod. Phys. C* 19, 1787 (2008)
- [32] A.E. Eiben, J.E. Smith, *Introduction to Evolutionary Algorithms* (Springer, Berlin, 2003)
- [33] H. Etemadi, A.A.A. Rostamy, H.F. Dehkordi, *Expert Syst. Appl.* 36, 3199 (2009)
- [34] E. El-Dahshan, A. Radi, M. Y. El-Bakry and M. El Mashad, *Artificial neural network in heavy ion collision, 6th Nuclear and Particles Physics Conference, Luxor, Egypt, 17-21Pp., November (2007).*
- [35] M. Y. El-Bakry, *Chaos, Solitons and Fractals* 18, 995 (2003).
- [36] M. Wolter, *Phys. Part. Nucl.* 38, 255 (2007)
- [37] T. Malisiewicz, *Genetic programming and non-linear regression*, 2 December, 2003.
- [38] J. C. Werner, *High Purity Neutral Pions Selection Using Genetic Programming Discriminate Function*, University of Manchester Internal Report for Research Project GridPP-UK/PPARC reference PP/C502006/1
<http://www.hep.manchester.ac.uk/u/jamwer/>.
- [39] S. Sette and L. Boullart, *Eng. Appl. Artificial Intelligence* 14, 727 (2001).

IJSER