Pseudo - rapidity Distributions of Charged Particles for \( p - p(\bar{p}) \) Collisions Using Genetic Programming

*ahmed Y. El-Bakry
Tabuk University, Faculty of Science, Department of Physics, Tabuk, KSA.
E.mail: Kemo_h3s@hotmail.com
A. Radi
Ain Shams University, Faculty of Science, Department of Physics, Abbassia, Cairo, Egypt
Center of Theoretical Physics (CTP), the British University (BUE) in Egypt, El-Shorok, Egypt.
E.mail: Amr1165@yahoo.com
*Shaimaa F. Abdel-Hamied
Ain Shams University, Faculty of Education, Department of Physics, Roki, Cairo, Egypt.

Abstract— Genetic programming (GP) is one of a number of machine learning techniques in which a computer program is given the elements of possible solutions to the \( p - p(\bar{p}) \) collisions at high and ultrahigh energies. GP can be used to calculate pseudo-rapidity distribution of charged particles \( dN_{ch} / d\eta \) at central mass energies \( \sqrt{s} = 23.6, 53, 200, 546, 900, 1800 \text{ GeV and 2.36 TeV} \). The discovered function obtained from GP shows an excellent fitting to the experimental data. Moreover, it is capable for predicting the pseudo-rapidity at \( \sqrt{s} = 7 \text{ TeV} \) that are not used in the training. Also, \( dN_{ch} / d\eta \) is expected at \( \sqrt{s} = 10 \) and \( 14 \text{ TeV} \) using GP model and with other models. The results showed very accurate fitting to the experimental data recommending it to be a good alternative to other theoretical technique.

Index Terms— Genetic programming (GP), machine learning (ML), proton-proton interaction, pseudo-rapidity distribution.

1 INTRODUCTION

High energy physics (HEP) research focuses on the fundamental particles and forces in the universe. The leptons, like the electrons and muons, and the quarks from which the strongly interacting particles, such as the proton and neutron, are formed constitute the fundamental particles. Extremely high energy collisions are required to get these particles close enough to study and understanding the interactions between them [1-6].

One of the fundamental interactions in high-energy physics is the proton-proton (p-p) interaction particularly above the threshold of pion production [7]. The Large Hadron Collider (LHC) at CERN is designed for colliding proton-proton beams onto \( \sqrt{s} = 14 \text{ TeV} \) [8]. Collisions at these unprecedented high energies will provide opportunities for new physics [8]. The history of studies of this interaction is therefore very long and extremely interesting from both the experimental and theoretical points of view. Different models are provided for the hadron structure [9-12], such as the three-fireball model [13], quark models [14-16], fragmentation model [17-19] and many others.

Parallel to theoretical models, there are numerical solutions have been introduced in High energy such as the application of artificial intelligence (or the machine learning) such as genetic programming (GP) has a strong presence in the high energy physics [20-24]. The effort to understand the interactions of fundamental particles requires complex data analysis for which machine learning (ML) algorithms are vital [25]. In this sense, ML techniques such as artificial neural network [26], genetic algorithm [27] and Genetic programming [28], PYTHIA [29] and PHOJET [30] Monte Carlo models. The PHOJET model combines the ideas based on a dual parton model [31] on soft process of particle production and uses lowest-order perturbative QCD for hard process. PYTHIA on the other hand uses string fragmentation as a process of hadronization and tends to use the perturbative parton-parton scattering for low to high \( P_t \) particle production. In the present work, we illustrate the Genetic programming (GP) that is known as a technique with the capability of generating mathematical equations, which are able to define models for the given training data. Genetic programming was proposed by Koza [32].

The rest of the paper is organized as follows: Section 2 gives a review to the basics of the GP technique. Section 3 explains how genetic programing is used on modeling the \( p - p(\bar{p}) \) collisions. Finally, the results and conclusion are provided.

2 GENETIC PROGRAMMING OVERVIEW

Genetic programming (GP) 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) [32-40].

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 ‘offspring’s by the process of reproduction, mutation and crossover. The creation of offspring’s continues (in an iterative manner) until a specified number of offspring’s in a generation are produced and further until another specified number of generations are created. The GP thus transforms one population of individuals into another one in an iterative manner by the natural genetic operations like reproduction, mutation and crossover.

Each individual contributes with its own genetic information to the building of new ones (offspring) 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).

A tree is a model representation that contains nodes and leaves. Nodes are mathematical operators (multiplication, addition, subtraction, and division,…..). Leaves are terminals (the attributes of the dataset and random numbers). Trees are manipulated through genetic operators (reproduction, mutation and crossover). The crossover operator points a tree branch and exchanges it with another branch and obtains new trees. The mutation operator changes the branch for a random new branch. Reproduction means an exact duplication of the program if it is found to be acceptable by the fitness criteria (Fig. 2). The length of the chromosome is variable.

To select individuals for crossover, mutation, reproduction and to determine how good the individuals are at solving the given problem, fitness functions are employed. The fitness function assesses how an individual is fitted to the environment of a domain problem, after calculating the fitness for all individuals. There are many varieties fitness function such as number of hits, relative squared error (RSE), mean squared error (MSE) etc. that can be applied for evaluating performance of generated computer program (solution) [41].

GP evolves computer programs to solve problems by executing the following steps:

Step 1: One or more initial population of individuals is randomly generated with functions and terminals related to the problem domain.

Step 2: The implementation of GP iteratively performs the following steps until the termination criterion has been satisfied:
   i. The fitness value of every individual is estimated according to a selected fitness measure.
   ii. All individuals in the population are sorted based on their fitness values.
   iii. The next generation is produced using the genetic operations (reproduction, crossover and mutation).
   iv. The termination criterion is checked. If it is not satisfied, the next iteration is performed; if satisfied go to step 3.

Step 3: The result may be a solution to the problem domain.

3 THE PROPOSED GP MODEL

We have proposed GP model trained using experimental data to simulate and predict the pseudo rapidity distribution for $p - p$ interaction at $\sqrt{s} = 23.6, 53, 200, 546, 900, 1800$ GeV, 2.36, 7, 10, and 14 TeV [42 - 53].

The proposed GP model for the pseudo-rapidity of charged particles has the total center of mass energy $\sqrt{s}$ and the pseudorapidity $\eta$ as inputs. The output is the corresponding pseudo-rapidity distributions $dN_{ch}/d\eta$, (see Figure 3).

Our representation, the fitness function is calculated as a negative value of the total absolute performance error of the discovered function on the experimental data set, i.e., a lower error must correspond to a higher fitness. The total performance error can be defined for all the experimental data ($i = 1$, ...,
\[ E = \sum_{j=1}^{n} \left| X_j - Y_j \right|^2 , \quad (1) \]

Where, \( X_j \) represents the experimental data for element \( j \) and \( Y_j \) represents the calculated results for element \( j \). The running process stops when the error \( E \) is reduced to an acceptable level.

4 RESULTS

GP was run for 100 generations with a maximum population size of 1500. The operators (and selection probability) were: crossover with probability 0.9 and mutation with probability 0.01. The function set is \( (+, -, /, \log_2, \ln, \exp, \sin, \cos, \sqrt{\cdot}) \) and the terminal set is \( (\text{random constant from 0 to 15, } \sqrt{s}, \eta) \). The “full” initialization method was used with an initial maximum depth of 27, and tournament selection with a tournament size of 6 as in Table 1.

<table>
<thead>
<tr>
<th>GP parameters</th>
<th>GP model</th>
</tr>
</thead>
<tbody>
<tr>
<td>GENERATIONS</td>
<td>100</td>
</tr>
<tr>
<td>POPULATIONS</td>
<td>1500</td>
</tr>
<tr>
<td>FUNCTION SET</td>
<td>(+,-,\cdot,/,\log_2,\ln,\sin,\cos,\sqrt{\cdot})</td>
</tr>
<tr>
<td>TERMINAL SET</td>
<td>(\sqrt{s}) AND (\eta)</td>
</tr>
<tr>
<td>SELECTION METHOD</td>
<td>TOURNAMENT</td>
</tr>
<tr>
<td>MUTATION RATE</td>
<td>0.01</td>
</tr>
<tr>
<td>CROSSOVER RATE</td>
<td>0.9</td>
</tr>
<tr>
<td>FITNESS FUNCTION</td>
<td>MSE</td>
</tr>
</tbody>
</table>

Training data are the total center of mass energy ranging over about two orders of magnitude, from the ISR (\( \sqrt{s} = 23.6 \) GeV) to the Tevatron (CDF data, \( \sqrt{s} = 1.8 \) and 3.36 TeV). The simulation results contain seven trained energies (\( \sqrt{s} = 23.6, 53, 200, 546, 900, 1800 \) GeV and 2.36 TeV) to assure the simulation capability of the proposed GP model. Also, the discovered function (in the Appendix) has been trained to associate the input patterns \( (\sqrt{s}, \eta) \) to the target output patterns \( (dNch/d\eta) \) for the above energies. The discovered function has been used to predict \( \sqrt{s} = 7 \) TeV and expect for \( \sqrt{s} = 10 \) and 14 TeV. Figures 4, 5, and 6 illustrate the experimental, trained and predicted pseudo-rapidity distribution of charged particles. The GP model performed almost exact fitting to the experimental data. The expectation values of energies (10 and 14 TeV) for \( dNch/d\eta \) using GP compared with other conventional theoretical technique (Extrapolation, PHOJET and PYTHIA).
5 CONCLUSION
Genetic programming (GP) has been run to model the $p - p(\bar{p})$ interaction at high and ultrahigh energies. GP
discovered a function that describes pseudo-rapidity distribution of pions $p - p(\bar{p})$ from interaction at different high energies. GP results have been compared to extrapolation, PYTHIA and PHOJET. It is observed that at both $\sqrt{s} = 10$ and 14 TeV, the predictions from extrapolation are high than the GP model. The GP results are close to the PYTHIA. The values from GP are higher than those from PHOJET model. In general all the three distributions have almost similar shape with the GP model.

The discovered function of GP model shows good match to the experimental data. Moreover, the discovered function is capable of predicting experimental data for pseudo-rapidity distribution that are not used in the training session. However, the overall similarity of the results proves that the GP model works correctly finding, in automatic way, powerful simulation and prediction. This is a remarkable result for GP, taking into account that it uses only a list of functions and variables as input, without any knowledge about the process. Finally, we conclude that GP have become one of the important research areas in high-energy physics.

**APPENDIX**

\[
\frac{dN_{ch}}{d\eta} = [C_2 - C_1]^{1/4}
\]

Where,

\[
C_1 = \sin(0.45X1 - X1) + 0.455\exp(A_2 - A_1)^{1/2}
\]

\[
C_2 = 0.456 \times (X_2 - \exp(X_1))/C_3
\]

\[
C_3 = \sin(\sin(20 - d_1)) + \exp(C_4)
\]

\[
C_4 = \left[\sin(\frac{X_2}{X_1} - X_1) + \exp(0.455 \times X_1)\right] \times 0.496
\]

\[
d_1 = [0.5(0.25d_2 + X_1) - \log(\sin(\exp(\sin(X_2))))]^{1/2}
\]

\[
d_2 = \cos\left[\frac{X_2 - \exp X_1^{1/2}}{X_2}\right] \times \frac{d_3}{X_2}
\]

\[
d_3 = \cos(\sin X_1) - (0.05 + X_2^{1/2}) \times \sin(\exp(20 + 0.97))^{1/2}
\]

\[
A_2 = \frac{[X_2 - \exp X_1]^{0.48404}}{H_1}
\]

\[
H_1 = \cos[\sin(\frac{X_2}{X_1} - X_1) + \exp(0.45562 X_1) + H_2]
\]

\[
H_2 = \exp[0.49572 \times \sin(\sin X_1) + \exp(0.45562 X_1))]
\]

\[
h = X_1 \times (\exp(\sin X 2 - 1))
\]

\[
A_1 = \frac{\log[U_1 + \sin(\frac{X_2}{X_1} - X_1)] + \exp(0.45562 X_1)]}{X_2}
\]

\[
U_1 = \sin[u] + \exp(0.45562 f_5)]
\]

\[
u = 0.45562 X_1 - \cos(\log(\sin(X_1))) + 1.3712
\]

\[
f_5 = \sqrt{[F - \log((X_2 - 1.4 \times 10^{-3}) + 20)]/X_2}
\]

\[
F = (0.48404 X_2 - 0.48404 \exp(X_1)) \times [f_1 + 0.49575 \exp f_2]
\]

\[
f_1 = \cos[\sin(\frac{X_2}{X_1} - X_1)] + \exp(0.45562 X_1)
\]

\[
f_2 = \sin([X_1 \times (\exp[\sin(X 2 - 1)]) - X_1] + \exp(0.45562 X_1))
\]

Where, $X_1, X_2$ are $\sqrt{s}$, $\eta$ respectively

**REFERENCES**


[51] F. Abe et al., 1990 Phys. Rev. D 41 2330

[52] Khachatryan V. et al., 2010 JHEP, 02 041.