www.editada.org

International Journal of Combinatorial Optimization Problems and Informatics, 15(4), Nov 2024, 163-172. ISSN: 2007-1558. https://doi.org/10.61467/2007.1558.2024.v15i4.516

A System for the Analysis of Historical Electoral Results of the National Electoral Institute of Mexico

Paciano Del Ángel Nava, Wenceslao Eduardo Rodríguez Rodríguez, Adolfo Josué Rodríguez Rodríguez, Juan Carlos Huerta Mendoza, David Tomás Vargas Requena, Ramón Gerardo Tijerina Rodríguez, Sanjuanita Crescencia Ortiz Valadez

Ingeniero en Sistemas Computacionales de la Unidad Académica Multidisciplinaria Reynosa Rodhe-Universidad Autónoma de Tamaulipas. Carretera Reynosa-San Fernando, cruce con Canal Rodhe, Col. Arcoiris, Cd. Reynosa Tamaulipas, C.P. 88779
paci-99@hotmail.com, wrodriguez@docentes.uat.edu.mx, arodriguez@docentes.uat.edu.mx, jchuerta@docentes.uat.edu.mx, dvargas@docentes.uat.edu.mx, rtijerina@docentes.uat.edu.mx, sjortiz@uat.edu.mx

Abstract. This paper presents a computer system capable of analyzing the databases of the results of the elections in Mexico, using genetic algorithms to achieve efficient analyses that will serve for the creation of reports with the information generated, in a shorter time than it can take to perform a manual analysis. The developed system implements the Windows Forms environment with the C# programming language and the SQL Server relational database system. In this research, the variables related to the execution time in the generation of reports will be measured, with respect to the traditional method, in addition to collecting relevant information that the system shows through experimentation.

Keywords: genetic algorithms, database, programming language.

Article Info
Received Aug 26, 2024
Accepted Oct 10, 2024

1 Introduction

Genetic algorithms (GA) were developed by John H. Holland in the early 1960s and are based on Charles Darwin's evolutionary theory of natural selection as an optimization procedure characterized by basic operators: selection, crossbreeding and mutation. These GAs start from an initial population of individuals (solutions) and generate offspring based on the characteristics of the best individuals, according to the evaluation function (aptitude function). The new individuals replace the previous ones in such a way that the population tends to converge towards the global solution of the problem to be optimized (García, 2016; Ponce, 2010; Cuevas, 2016; Holland, 1962). Examples of applications of GAs are route optimization, task optimization, automated management of industrial equipment, machine learning, databases, pattern recognition, grammar generation, robot movement planning, prediction, financial sector systems, among others (Goldberg, 1989; EAG, 2023).

In Mexico, the process of electing representatives is based on democratic elections at the federal and state levels. At the federal level, 500 deputies are elected every 3 years, the president of the republic and 128 senators every 6 years. At the state level, the governor is elected every 6 years, and municipal presidents and deputies of the state congress every 3 years (INE, 2024a). In 1991, the Federal Electoral Institute established the Preliminary Electoral Results Information System to speed up the dissemination of results. In 1994, the Preliminary Election Results Program was implemented to transmit data via telephone. Over the years, PREP has been improved with the incorporation of new technologies, such as the Internet, to disseminate preliminary results and databases for detailed analysis (INE, 2024b).

Until the end of the 20th century, a single political party dominated elections in most regions (Nohlen, 2005). However, starting in 2000, elections became more competitive, marking a change in the predominant political party (Alonso, 2000). This political change has generated the need to analyze electoral results, considering factors such as probabilities, statistical data from previous elections, news about candidates and voting intention surveys (Kuschick, 2002).

To address this analysis, the development of a system based on areas of knowledge of computational systems, such as programming and AG, is proposed in this work. This system seeks to help political scientists in the formulation of predictions through the search for patterns in electoral results by means of AG, using the electoral system in Mexico as a reference, with the objective of knowing the probability of winning or failure of a candidate or political party in a future election through the observation of electoral results over time. This work contributes to the generation of new knowledge regarding the use of intelligent algorithms applied to the study of electoral behavior in a specific region (Pavia & Romero, 2022; Alaminos, 2023, Ramírez, 2014). This system will take the results of the elections in a specific period, examining the positions obtained by each political party in each vote. This will allow us to evaluate the probability that a political party can triumph in a future electoral competition. In addition, the possibility of adapting the system to similar electoral systems in other countries is raised. The social relevance of this research lies in its usefulness for political scientists, social analysts and political strategists by providing accurate and detailed information on electoral results. This work also will contribute to the development of statistical systems for agile and intelligent decision-making, taking advantage of current computing technologies.

2 Theoretical foundations

2.1 Programing language

To communicate with a computer, it is necessary to send instructions in a language that the machine can understand. Programming languages are used to express algorithms or instructions using expressions, symbols and characters. The source code represents the instructions written in a programming language, while the machine code is the direct way in which the computer can understand them. A translator (or programming language processor) is responsible for converting the programming language to machine code, allowing it to be executed by the computer (Deitel & Deitel; 2007).

2.2 SAP Crystal Reports

SAP Crystal Reports, developed by SAP SE, is a tool that enables the design and generation of reports. Its version for programmers is integrated with Microsoft Visual Studio, offering a complete development kit (SDK) that allows the creation of reports from various data sources. These reports can be customized visually, improving the organization of information and allowing the definition of formatting criteria. It offers the ability to directly integrate reports into projects, access them from a file repository, manually adjust the formatting and visualize the reports generated within the application. The creation of reports can be carried out by coding or using a graphic designer included in the program (SAP, 2010).

2.3 Database

The term 'database' refers to a collection of organized data represented by entities and interconnections. It is composed of complex, shared structures designed to store and query information. Database models provide a conceptual description of how data works, its relationships, meaning and consistency rules. Among them are the entity-relationship model and the relational model. These databases aim to present an abstract view of the data, detailing how it is stored and the relationships between it, providing a general understanding of the structure and operation of the database (Silberschatz, 2003).

2.4 Genetic Algorithms

Genetic algorithms are a set of adaptive methods inspired by natural selection and genetics. Their purpose is to solve problems by mimicking the behavioral patterns of natural genetics. They start with an initial population of solutions, where those most effective at solving problems survive and reproduce, combining their characteristics to generate new solutions. These algorithms are based on the idea of natural selection, where the strongest individuals have the best chance of reproduction. They are part of evolutionary computation and employ heuristic techniques inspired by the principles of natural evolution. They are commonly applied to optimize parameters and address search problems (Gestal, 2000).

3 Methodology

3.1 Information system methodology

The system has been developed using the C# programming language and use Windows Forms as its graphical interface. It is connected to a database developed in SQL Server, in charge of storing and processing part of the necessary information. In addition, for the generation of reports, SAP Crystal Reports for Visual Studio has been integrated, allowing the creation of reports with a clear and adapted format for the user.

Data entry is carried out manually through an interface specifically designed for users to enter the required information, which is then sent to the database. For this process, it is required to download spreadsheets containing information on election days, generally organized by districts, sections or polling stations. In these sheets, the columns present general information of the region and electoral data by political party or coalition, which requires assigning candidates corresponding to each registered political party or coalition, given this format of information organization.

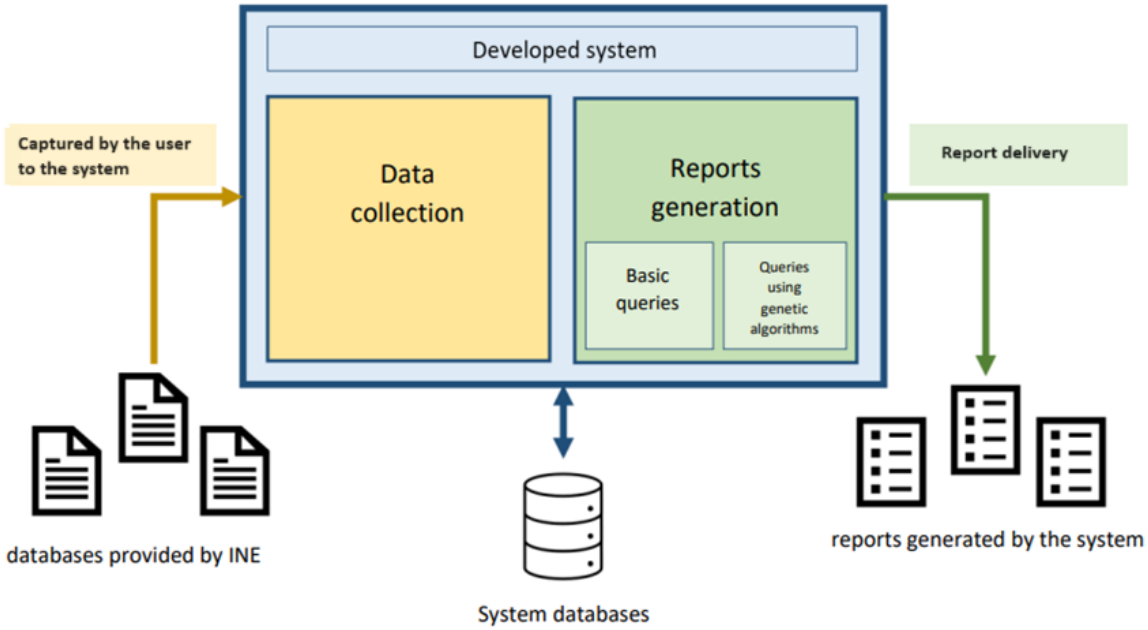

Fig.1. Logical operation of the developed system

3.2 Genetic algorithms methodology

One of the modules implemented is the use of a genetic algorithm to generate a trend from certain values. The genetic algorithm requires the data found in the system, the information is processed in such a way that an array of initial individuals can be created, which we will call population, each individual will have its own genetic information (represented in binary system), then the selection of individuals to be used for the cross is performed, these new individuals will undergo a mutation in their genetic information randomly and through a function that will qualify each individual, will be selected to be part of the next generation. This process will be repeated for several generations until a certain condition is met and the best individual or the most optimal result is obtained.

START *of the algorithm*
Array Initialization Datos_Electorales
Individuals Array Initialization
WHILE Cantidad_Generaciones THEN
Selection of individuals //Probabilistic Tournament
Crossing of individuals //Two-point crossing
Mutation of individuals //1-point mutation
Next Population //The best individuals are selected that will be passed on to the next generation
END WHILE
Returns the Individual with the Best Aptitude
END of the algorithm

3.2.1 Definition of the problem to be optimized

For the analysis of the data in this module, it requires as input the electoral data of the days and political party selected, where it will be stored in a matrix with the percentage obtained and the year it happened.

$$electoralData[i,j] = \begin{bmatrix} year_0 & PercentageofVotes_0 \\ & . \\ & . \\ & . \\ year_i & PercentageofVotes_j \end{bmatrix} \quad (1)$$

This genetic algorithm seeks to minimize the difference between the generated trend versus the electoral data, where it can be represented as follows:

$$Fitness = \sum_{i=0}^{n} \left| electoralData[i,1] - \sum_{j=o}^{m} (individual[j] * electoralData[i,0]^j \right| \quad (2)$$

3.2.2 Initial population.

The structure of the individuals is related to the degree of the equation that is intended to be used to obtain the trend. The values of individuals are randomly generated in the range of -1 to 1 considering floating point numbers; however, the values can change during the process to values in the range of -100 to 100. To avoid problems such as the "Hamming Cliff" during mutation operations, the representation of individuals in reflected binary or Gray code is used. This technique reduces the impact of the Hamming phenomenon, ensuring a more stable evolution of individuals throughout the iterations of the genetic algorithm.

$$Individuals\ [n+1] = [number_0, number_1, number_3, \dots, number_n] : 2 \leq n \leq 5 \quad (3)$$

3.2.3 Natural selection.

The selection of individuals is carried out using the tournament method, specifically in its probabilistic variant. In this process, two shuffles of the individuals' positions are performed and stored in two different arrays, each with a length equivalent to the number of individuals in the population. This probabilistic tournament selection structure follows the following procedure:

$$shuf\ f\ le01[n] = [position_0 \dots position_n]$$

$$shuf\ f\ le02[n] = [position_0 \dots position_n] \quad (4)$$

The tournament is carried out between the saved position in the 2 arrays, as shown below:

$$individualWinner = \begin{cases} Individual[shuf\ f\ le01[n]], & fitness(Individual[shu\ f\ le01[n]]) < fitness(Individual[shuf\ f\ le02[n]]) \\ Individual[shuf\ f\ le02[n]], & fitness(Individual[shu\ f\ le01[n]]) > fitness(Individual[shuf\ f\ le02[n]] \end{cases}$$

$$individualWinner = \begin{cases} Individual[shuf\ f\ le01[n]], & fitness(Individual[shu\ f\ le01[n]]) < fitness(Individual[shuf\ f\ le02[n]]) \\ Individual[shuf\ f\ le02[n]], & fitness(Individual[shu\ f\ le01[n]]) > fitness(Individual[shuf\ f\ le02[n]]) \end{cases} \quad (5)$$

The genetic algorithm is configured with a probability of 0.5 to choose the fittest individual; therefore, 2 individuals with the highest fitness are previously selected, which will pass directly as parents to avoid the loss of these during the execution of the tournament. This strategy ensures that the fittest individuals can contribute to the generation of offspring without running the risk of being excluded during the tournament competition.

3.2.4 Crossing phase

For the crossover process, the two-point method is used to generate the offspring. The probability of outcrossing is 0.5 with respect to the number of individuals in the generation. This strategy allows the generation of new individuals from the combination of genetic information of the parents, thus contributing to the genetic diversity of the population. In these step 2 points are selected at random from the coded individual (genotype) where:

$$0 \leq position_1 < position_2 \leq individualLengthBinary \quad (6)$$

This process is carried out with the purpose of exploring and further diversifying the space of potential solutions, to find more optimal solutions or new genetic combinations that may be beneficial in the evolutionary process of the genetic algorithm.

3.2.5 Mutation

The mutation of the individual occurs with a probability of 0.2 where a single point within the coded representation of the individual (genotype) is randomly chosen and its value is inverted, this helps to have a more diverse population that can approach a more optimal solution.

4 Materials

The materials presented in Table 1 are those that were used to carry out the experimentation of this work. Equipment may always differ and when they are similar versions or with an update that does not affect the procedure directly.

Table 1. Equipment and Instrument Description

Equipment/Instrument	Brand	Model	Supplier	Remarks
Microsoft Visual Studio Community 2019	Visual Studio	Version 16.7.7	Microsoft Corporation	Environment Integrated development for the .NET platform
SAP Crystal Reports: developer Visual Studio version	SAP Crystal Reports	Version 13.0.30.3805	SAP SE	SDK for authoring of reports within Visual Studio
Microsoft SQL Server Management Studio 18.8	SQL Server	Version 15.0.18369.0	Microsoft Corporation	Database handler for SQL Server
SQL Server Express 2019	SQL Server	2019	Microsoft Corporation	Data base engine
Microsoft Office Excel	Microsoft Office	Office 360	Microsoft Corporation	Sheet Program calculation required for the reading of the databases provided
Dell™ OptiPlex™ 980	OptiPlex	Mini tower	Dell	Processor: Intel Core i5 660 RAM Memory: 8GB SSD hard drive 240GB

5 Analysis of results

For the purposes of this paper, the historical results of the elections in Mexico will represent the population, being the environment with which we are going to work and take the data to carry out the analysis processes by the developed system. Because of the magnitude of the data and the population where this work is developed, the historical results are taken as a sample of the elections in the state of Tamaulipas (Mexico), Specifically, the electoral information from the cities of Reynosa, Victoria CIty and Tampico, for the city council elections in 2007, 2010, 2013, 2016, 2018 and 2021. With the developed system, it is possible to carry out the tests that allow preliminary results to be obtained for study. Information is obtained of the election results in spreadsheet format, where it is advisable to organize at least to the most recent election day in the region that is intended to be studied. The information can be entered into the system through an Excel spreadsheet format., this format is compatible with most of the files that are used in the provide with the electoral results; matches are registered beforehand politicians and coalitions that competed on election days. Once the information the relationship between candidates and political parties is recorded to process the information correctly. The historical analysis consists of determining the trend based on the percentages obtained by one political party for each election day held in a region that is registered in the system. The trend is obtained by means of a polynomial equation where the user defines a grade from 2 to 5, which by means of a genetic algorithm will look for the optimal values that allow minimizing the margin of error of the polynomial equation and allows to perform a prediction. The tests were carried out with the political parties "Revolutionary Party", "National Party" and "National Movement" considering their coalitions, for the municipalities of Reynosa, Victoria City and Tampico with the polynomial equation from second to fifth grade. The best results were obtained with the third-degree polynomial equation, where the margin of error is smaller compared to the other tests and the trends make sense with the data entered. An example is presented in Figures 2 and 3, where a notable margin of error is generated when there is a high variation in any of the values and the trend does not manage to cover all the values, which, despite increasing the degree of the equation, the margin of error and the forecast for the future become illogical compared to a lower degree, where the results make sense with the data that has been entered.

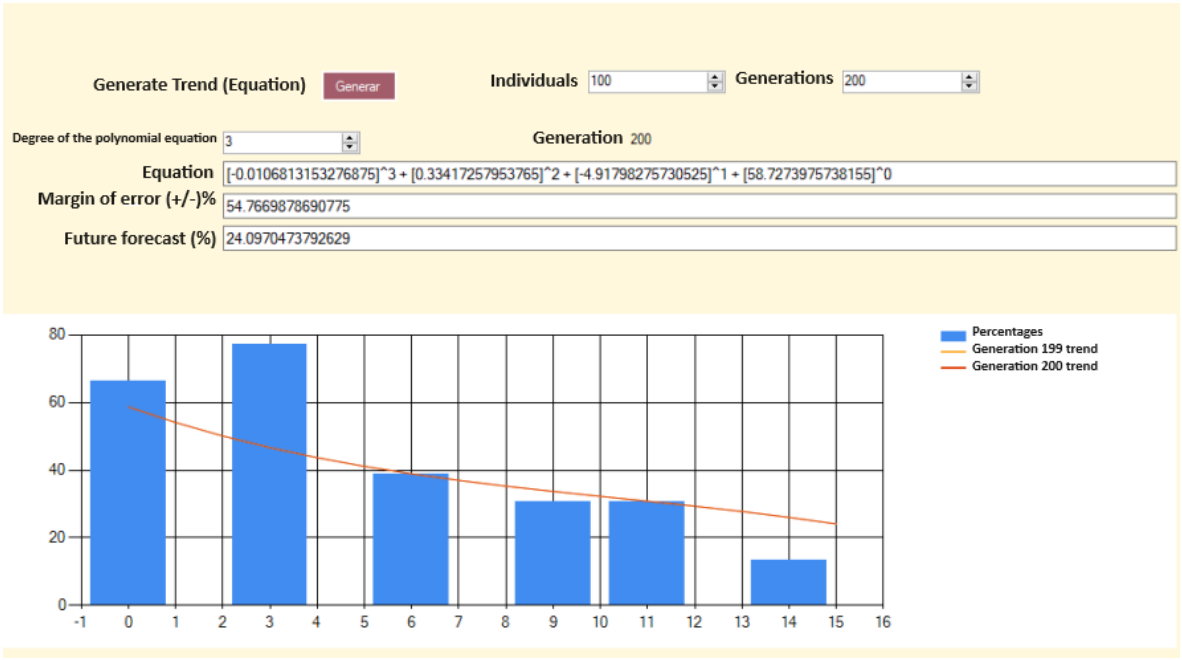

Figure 2. Analysis with third-degree polynomial equation

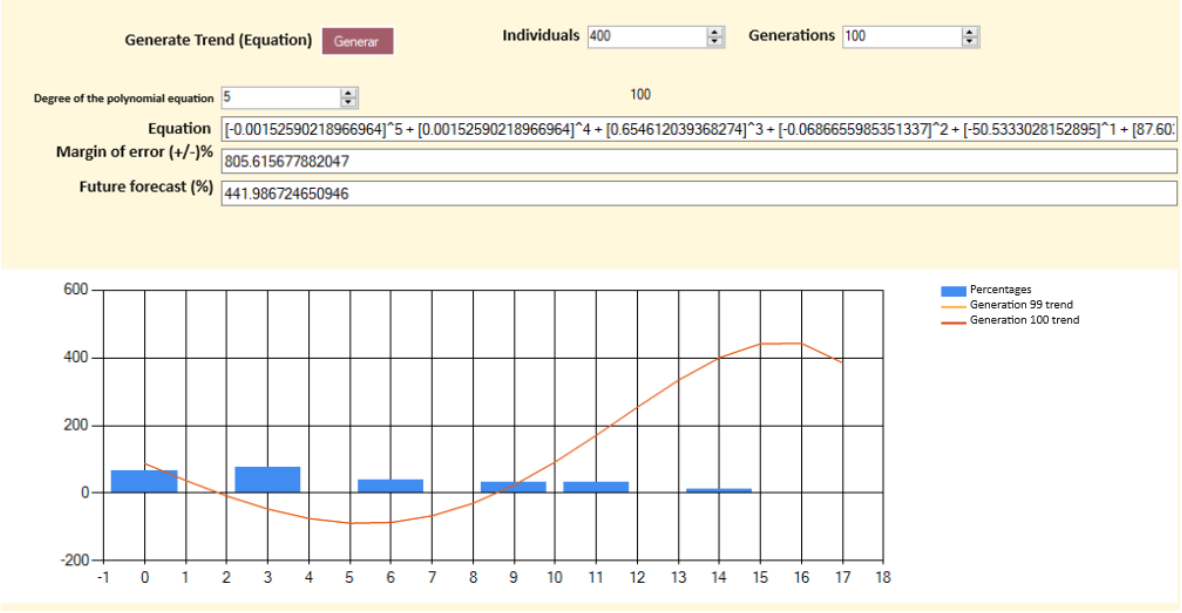

Figure 3. Analysis with fifth-degree polynomial equation

Another of the cases that were observed are those in which the political party has participated in few contests because it is of recent creation, as can be seen in the case of the Figure 4 that the party has only participated on 3 occasions and the value of the prediction is very high for the values that are held. At the time of decreasing the degree of the equation polynomial, an equation is observed that allows logical values to be obtained even though the margin of error increases. The result of the tests carried out with this type of analysis is shown in Tables 2, 3, 4 and 5, where the percentage of margin of error per party is recorded (%) and the execution time of the genetic algorithm (seconds); In addition, the parameters entered the genetic algorithm.

Table 2. Test results considering 100 individuals, 100 generations, and a second-degree polynomial equation.

Region of study	**Percentage margin of error per match (%) / Genetic algorithm execution time (Seconds)**					
	Revolutionary Party		**National Party**		**National Movement**	
Reynosa	14.05 %	19.74 s	32.53 %	17.83 s	41.42 %	7.30 s
Victoria City	76.93 %	16.81 s	28.81 %	16.28 s	33.38 %	7.18 s
Tampico	41.06 %	17.11 s	17.44 %	18.27 s	35.47 %	7.93 s

Table 3. Test results considering 200 individuals, 100 generations, and a third-degree polynomial equation.

Region of study	**Percentage margin of error per match (%) / Genetic algorithm execution time (Seconds)**					
	Revolutionary Party		**National Party**		**National Movement**	
Reynosa	115.61 %	28.94 s	28.97 %	28.95 s	25.70 %	14.07 s
Victoria City	36.74 %	26.17 s	35.98 %	26.03 s	20.25 %	12.50 s
Tampico	70.01 %	26.49 s	39.08 %	28.26 s	5.26 %	15.38 s

Table 4. Test results considering 300 individuals, 100 generations, and a fourth-degree polynomial equation.

Region of study	**Percentage margin of error per match (%) / Genetic algorithm execution time (Seconds)**					
	Revolutionary Party		**National Party**		**National Movement**	
Reynosa	73.60 %	52.22 s	46.84 %	50.15 s	28.15 %	26.42 s
Victori City	158.08 %	46.21 s	36.54 %	45.96 s	17.73 %	24.69 s
Tampico	124.45 %	50.51 s	58.64 %	50.96 s	11.88 %	28.29 s

Table 5. Test results considering 400 individuals, 100 generations, and a fifth-degree polynomial equation.

Region of study	**Percentage margin of error per match (%) / Genetic algorithm execution time (Seconds)**					
	Revolutionary Party		**National Party**		**National Movement**	
Reynosa	70.93 %	86.36 s	105.03 %	92.29 s	31.47 %	48.76 s
Victori City	758.26 %	86.76 s	260.67 %	82.24 s	55.57 %	45.45 s
Tampico	85.23 %	89.92 s	377.55 %	90.51 s	27.83 %	53.17 s

The genetic algorithm executed for this analysis was carried out between 100 to 400 individuals with 100 generations, which allows obtaining an average response time of 38.58 seconds from the data that can be seen in the graph. These times are related to the number of individuals, although these values may change as the number of generations increases, as can be seen in Figure 4.

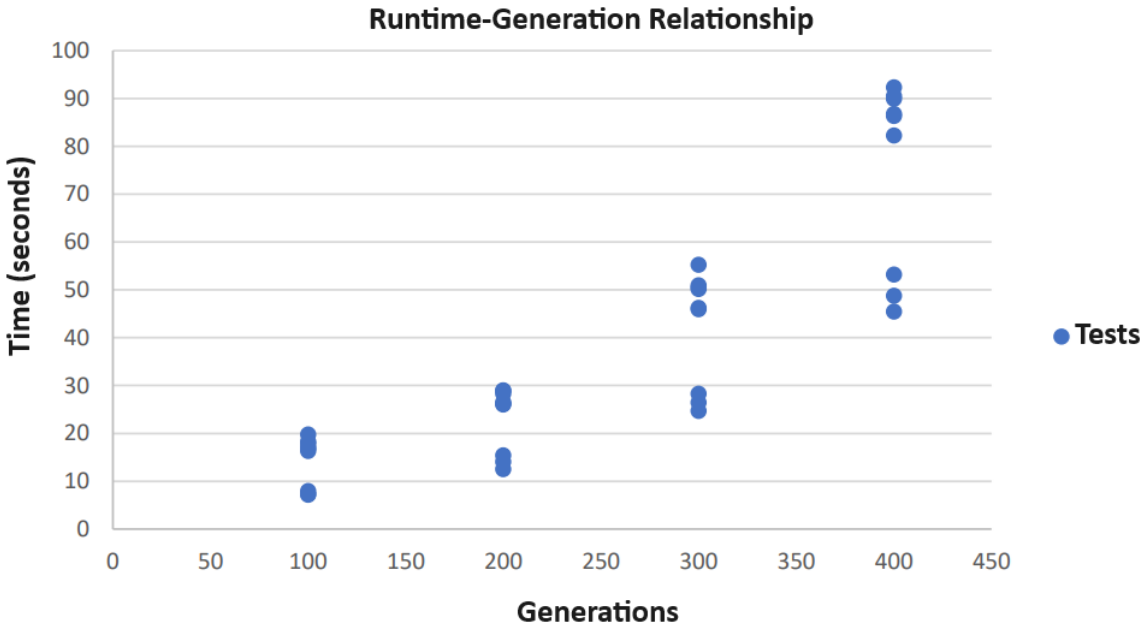

Figure 4. Time/Generations relationship generated by the performance of the computer system.

The proposed system for the analysis of electoral results turns out to be efficient for cases of study in delimited regions, because it is required to enter the information of manually, in addition to requires installation of the software on a Windows 10 or compatible computer. It is recommended to migrate this system to a web application format, to eliminate the requirement of installation on a computer and thus reduce resource consumption by the client, in addition to improving the way of presenting reports on different devices that have a browser, with a responsive web design. In the case of the historical analysis, the genetic algorithm presented good results by matching the trend with respect to the data entered in most cases, however, the use of other algorithms or methods to improve these parameters is not limited.

6 Conclusions

The computer system presented in this work meets its proposed objectives if there is the greatest amount of information available in the system, being a very important factor when performing the analysis of the results. For the historical analysis, a genetic algorithm was implemented to generate a polynomial equation that would show the trend of the electoral results and with it have the possibility of making predictions for the future. The margin of error shown by the implementation of the genetic algorithm is considered good in those histories where their values do not have high variations and where there are sufficient values to approximate the trend. The implementation of the genetic algorithm obtained an adequate execution time, without affecting the user experience when generating this type of analysis and allowing to see the evolution of trends over generations. In general, the computer system for the analysis of electoral results meets most of their expectations, being a useful tool for consulting information.

References

Alaminos-Fernández, A. F., & Alaminos, A. (2023). *Métodos y Modelos para la Predicción Electoral. Una Guía Práctica*. Universidad de Alicante. Obets Ciencia Abierta. Alicante: Limencop.

Alonso, J. (2000). Las elecciones federales del 2000, consolidación del panismo. *Espiral*, 7(19), 95-126.

Cuevas, E., Oliva, D., Díaz, M., & Osuna, J. (2016). Algoritmos genéticos. En *Optimización-Algoritmos programados con MATLAB* (pp. 27-48). México: Editorial Alfaomega.

Deitel, H. M., & Deitel, P. J. (2007). *Como programar en C#*. México: Pearson Educación.

Enzyme Advising Group (EAG). (2023). Algoritmos genéticos y su gran cantidad de aplicaciones. retrieved from https://enzyme.biz/blog/algoritmos-geneticos-y-sus-aplicaciones-para-soluciones.

García, A. (2016). Búsqueda informada. En *Inteligencia Artificial. Fundamentos, práctica y aplicaciones* (pp. 126-139). México: Editorial Alfaomega.

Gestal, M. P. (2000). Introducción a los Algoritmos Genéticos. retrieved from https://cursa.ihmc.us/rid=1KNKMJ4LN-11XXFSG-1KV5/Algoritmos%20de%20Terminos.pdf.

Goldberg, D. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Massachusetts: Addison-Wesley Publishing Co.

Holland, J. (1962). Concerning efficient adaptive systems. En *Self-Organizing Systems* (pp. 215-230). Washington, D.C.: Spartan Books.

Instituto Nacional Electoral (INE). (2024). Qué es el PREP? | Instituto Nacional Electoral. retrieved from https://portalanterior.ine.mx/archivos3/portal/historico/contenido/interiores/Detalle_que_es_el_PREP_2009-id-9b039f05125bf110VgnVCM1000000c68000aRCRD/.

Instituto Nacional Electoral (INE). (2024). Sistema Político Electoral - Instituto Nacional Electoral. retrieved from https://www.ine.mx/sobre-el-ine/sistema-politico-electoral/.

Kuschick, M. (2002). México: elecciones y el uso de las encuestas preelectorales. *Revista Mexicana de Ciencias Políticas y Sociales*, 45(184), 103-127.

Nohlen, D. (2005). North America, Central America, and the Caribbean. En *Elections in the Americas: A data handbook* (Vol. I, pp. 339-446). New York: Oxford University Press.

Pavía, J. M., & Romero, R. (2022). Improving estimates accuracy of voter transitions. Two new algorithms for ecological inference based on linear programming. *Sociological Methods & Research*, *0*(0). retrieved from https://doi.org/10.1177/00491241221092725.

Ponce, P. (2010). Algoritmos genéticos. En *Inteligencia Artificial con aplicaciones a la Ingeniería* (pp. 285-303). México: Editorial Alfaomega.

Ramírez, M., & Chávez, E. (2014). Uso de la sociofísica para realizar predicciones electorales utilizando algoritmos genéticos. *Tecsistecatl*, 16.

SAP. (2024). Manual de usuario de SAP Financial Consolidation. retrieved from https://help.sap.com/doc/ee4d60e174631014a7da368fb0e91070/10.1.9/es-ES/fc_101_win_user_es.pdf.

Silberschatz, A. (2003). *Fundamentos de bases de datos* (pp. 1-8, 645-646). Madrid: McGraw-Hill.