JFSA

Journal of Futuristic Sciences and Applications
Vol. 2(2), Jul-Dec 2019, pp. 6-11
doi: 10.51976/jfsa.221902
www.gla.ac.in/journals/jfsa

Evaluation of Genetic Algorithm Optimization by Comparitive Analysis of Error Values

Mohan Gupta and Rahul Sharma***

ABSTRACT

In this paper to test the performance of GA a comparative study has been carried out between PI and PD Controller. The two different objective functions have been chosen that is integral absolute error and integral total absolute error on different generations the controller has been run as a result the minimum value of error has been found in controller in which the objective function was ITAE and a detailed evaluation of Genetic Algorithm has been carried out through out the paper. Optimization and search issues may be solved using genetic algorithms, which are seen as a search process in computers. Global search heuristics are another name for them. Many of these methods are derived from concepts found in evolutionary biology such as mutation, selection, and cross-breeding. For programmes, these algorithms provide a way to automatically enhance their settings. The simulations are tallied to ensure that GA delivers the system promising outcomes.

Keywords: *PI Controller; PD Controller; Error Values; IAE; ITAE; Optimization; Genetic Algorithm.*

1.0 Introduction

The Genetic algorithm is a population genetics-based adaptive heuristic search method. The first genetic algorithm was devised by John Holland in the early 1970s. Using a genetic algorithm is similar to using natural selection and genetics as a search strategy. The initial stage of a genetic algorithm is to create a population of solutions. A chromosome is the answer. The population has grown steadily over time. All of the chromosomes in the population have their fitness evaluated after each generation, and the ones that will be passed down to future generations are randomly selected according to their scores. Children are born as a consequence of haphazard matings when unselective mating is practised. In the process of conceiving a child, random mutations and cross-overs occur.Because current generation chromosomes are more likely to be selected, they may have a higher average fitness value than older generation chromosomes. The evolutionary process continues until the ultimate condition is fulfilled. Genetic algorithms' output is referred to as chromosomes or strings . Lists or strings of chromosomes are often used to describe them. Lists and strings are common building blocks for evolutionary algorithms. An separate set of probabilistic calculations is the basis for genetic algorithms, as opposed to local search approaches. Simulated natural selection is used here to show how the best people are selected from each generation.An individual is the traditional term

**Corresponding author; Assistant Professor, Mechanical Engineering, United College of Engineering and Research, Prayagraj, Uttar Pradesh, India (E-mail: mohanguptaucer@gmail.com)*
***Researcher, Tallinn University of Technology, Estonia, Europe (E-mail: rahul.sharma@taltech.ee)*

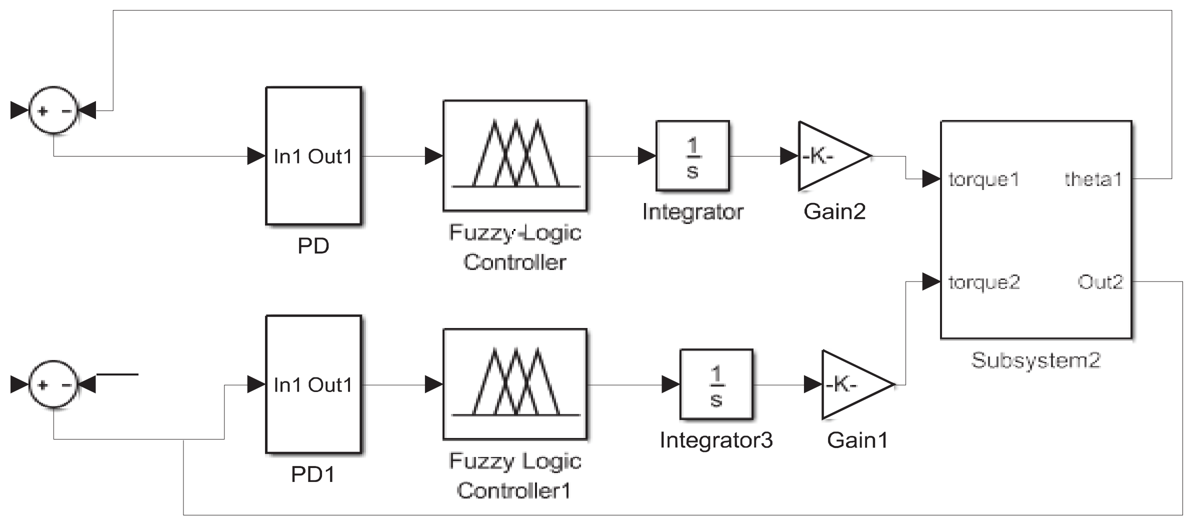
for a solution to a problem under consideration. A population is a grouping of persons that is taken into account. A single chromosomal string encodes all of an individual's info.Each allele in an individual chromosome corresponds to a discrete unit of quantified information, such as a bit, decimal point or alphabetic character. To interchange solutions with the nominal object space, & is an alternate data format required coding and decoding. Evolutionary algorithms include genetic algorithms. Problems that do not have an efficient solution may be solved via evolutionary algorithms. Optimization issues have been addressed with the help of a genetic algorithm.

2.0 PD and PI Optimization

In this paper there is a comparison between linear controllers that is PD and PI As controller design approaches draw influence from natural selection, GAs have been shown to be an ideal method for adjusting the controller.

An IAE has been attached to the controller's input, which is used as an objective function in the Genetic Algorithm. We designed IAE by combining the absolute block with the integrator, and we have attached these two blocks to the error and thus get IAE, which integrates error over time and helps us to give the least value of error. With the help of the Q-learning algorithm, we have developed a method for optimising closed-loop system cost functions. The user may choose a tuning vector based on their control objectives and the available information. Three types of tuning parameters exist E, D, and U make up the parameter vector when utilising an FPI or FPD linear controller. When the precondition parameters are fixed, a fuzzy rule conclusion vector. Triangle membership function positions create a vector with known conclusions.

Figure 1: Block Diagram of PID Controller



From the premises and conclusions, this parameter is a combination of all of them. A number of contenders vie for each parameter's attention. Each candidate has a Q-value assigned to them by the QLA. As time goes on, the Q-value changes accordingly. Choosing an adequate set of settings for the learning process is critical to maximising future reinforcements. As a result, since the amounts are initially unknown, the fuzzy controller must test and explore possible actions. Longer exploration periods are more typical than you may think. Since fuzzy rules may be read and tuning parameters

Doi: 10.51976/jfsa.221902

have physical value, adding information to the initial FPI or FPD controller may dramatically cut the learning time. Prior knowledge may be divided into three main circumstances.

3.0 Multi Objective Optimization

In this paper we are considering both the integral absolute error and the total integral absolute error , now in order to explain multi objective function Consider the case when K goals are noncomparable and the decision maker has no clear preference for the objectives compared to each other. Minimizing a negative with a positive produces a maximisation kind of objective without compromising generality. This is how a multi-objective decision issue with k-objective minimization is defined: It is possible to find the minimization of K objective functions by selecting an n-dimensional choice variable vector in space X and dividing it by 1 in terms of 1, y and xn by 1 in terms of 1 in terms of xn. This is known as the minimising of K objective functions. There are several constraints that restrict the solution space, such as gj(x*)14 and limitations on the choice variables.In many real-world situations, competing objectives must be taken into account. As a result, optimising x for a single goal usually results in subpar performance for all other goals. A multi-objective solution that maximises each objective function separately is thus virtually unachievable. Multi-objective issues may be addressed logically by evaluating many solutions, each of which achieves the objectives at an acceptable level without being overtaken by any other solution. XY may only be said to be superior than x if all goal functions aim to minimise the total number of available solutions (all viable solutions) (i.e., y, K). An optimal Pareto solution is the only one that exists in the problem space. It is difficult to change the Pareto optimum response without impacting the other Pareto optimum answers."

All feasible non-dominant solutions in X are referred to as the Pareto optimum set, and the Pareto front is the objective function values in objective space that correspond to a certain Pareto optimal set. The Pareto fonts. There may be an unlimited number of Pareto optimal solutions. A multi-objective optimization technique's ultimate goal is to find Pareto optimal solutions. Many multi-objective challenges are so large that it is almost impossible to determine the whole Pareto optimal set. There are numerous combinatorial optimization scenarios where proving optimality is computationally impractical. Multi-objective optimization issues may be realistically solved using the Pareto set, which is the most well-known collection of solutions These problems need a multi-objective approach. Three competing objectives should be addressed by an optimization strategy: As a rule of thumb, the most well-known Pareto front should be as near to the actual Pareto front as feasible. 2. Pareto optimum set should ideally include the best-known Pareto set. For a fair depiction of the trade-offs, the best-known Pareto set should include solutions that are evenly dispersed and diversified over the Pareto front. A Pareto front should include the whole range of Pareto fronts. Finding solutions at the extremes of objective function space is necessary for this. It is advisable to concentrate (intensify) the search in one area of the Pareto front for a certain computational time restriction. The second objective, on the other hand, calls for an even distribution of search efforts throughout the Pareto front. Extending the Pareto front to both ends and investigating additional extreme solutions are the third and final goals of the project.

4.0 Results

The table below shows the comparison between absolute error and total absolute error in the three different generations. when the iteration has been carried out to 100 generation the minimum value of error is obtained and in terms of comparative study between IAE and ITAE. IAE outperformed as the number of iterations increases the value of error will goes on decreasing the below table shows this result when system is not stable.

Table 1: Comparison of IAE Values

IAE	ITAE	Generations
9.32	58.40	20
8.96	75.45	50
6.64	43.21	100

Figure 1: Comparison of Error Values for Unstable System

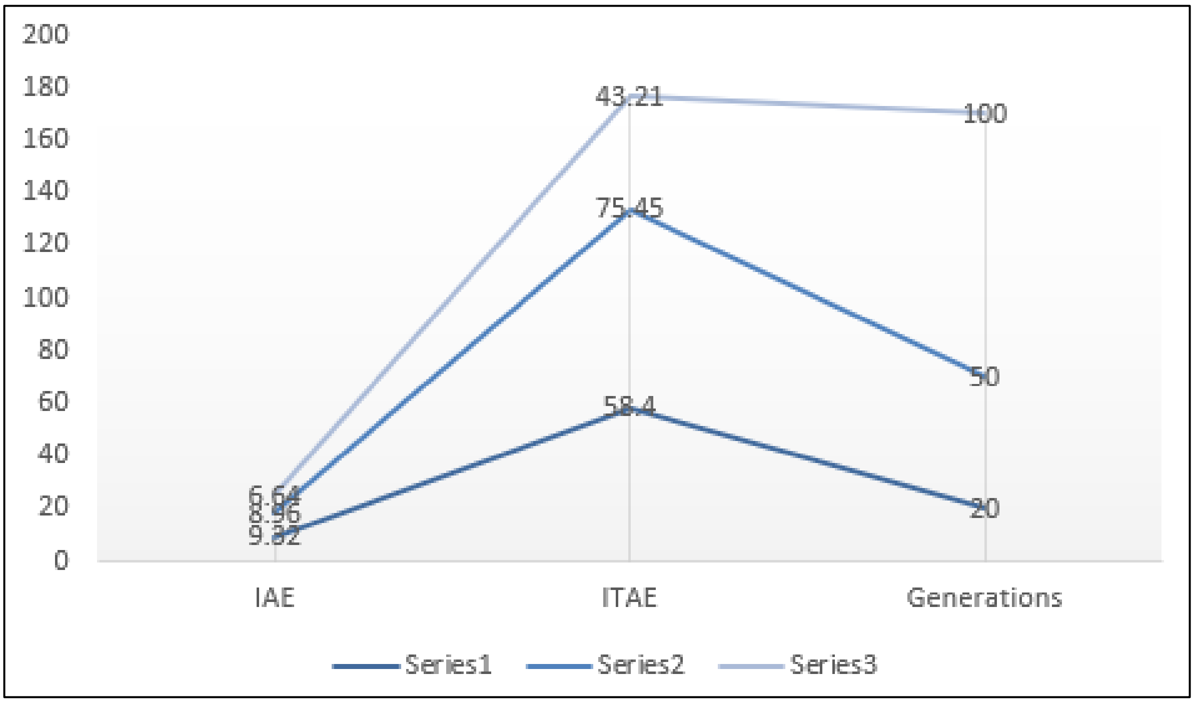


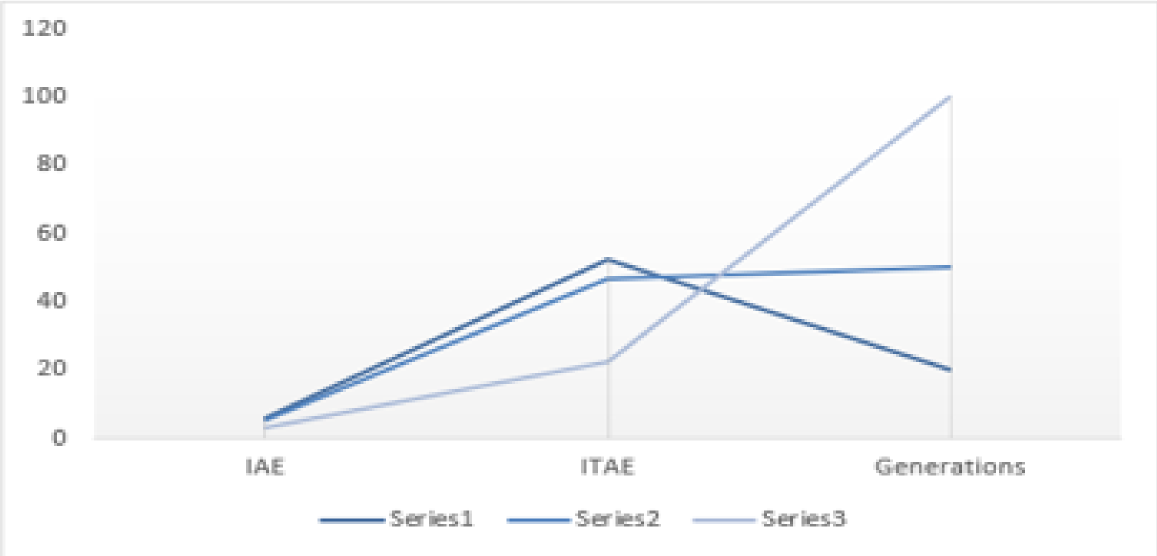
Table 2: Comparison of ITAE Values

IAE	ITAE	Generations
6.22	52.49	20
5.32	46.65	50
3.28	22.43	100

The tabular form result is being obtained for stable system .in which IAE objective function outperformed ITAE objective function. The least value of error obtained from IAE is 3.28 that has been obtained when genetic algorithm has been run for 100 generation and in this correspondence 22.43 is the ITAE value that has been obtained the graph below represent the tabular form data result in graphical form.

Doi: 10.51976/jfsa.221902

Figure 2: Comparison of Error Values for Stable System



5.0 Conclusion

Genetic algorithm is a probabilistic addressing optimization problem which is patterned on a genetic assessments process\sin biology and is targeted as an efficient method to discover a worldwide best solution for various forms of issue. This method is particularly relevant in several artificial intelligence approaches in this Paper Genetic Algorithm shows improvement in the error values as the generations increases also it shows better outcome in ITAE in comparison with IAE

References

1. Zhou, Q., Leng, G. & Feng, L. Predictability of state-level flood damage in the conterminous United States: the role of hazard, exposure and vulnerability. Scientific reports. 7, 5354 (2017).

2. Khosravi, K., Pourghasemi, H. R., Chapi, K. & Bahri, M. Flash flood susceptibility analysis and its mapping using different bivariate models in Iran: a comparison between Shannon's entropy, statistical index, and weighting factor models. Environmental monitoring and assessment. 188, 656 (2016)

3. Chapi, K. et al. Spatial-temporal dynamics of runoff generation areas in a small agricultural watershed in southern Ontario. Journal of Water Resource and Protection. 7, 14 (2015).

4. Pradhan, B. Flood susceptible mapping and risk area delineation using logistic regression, GIS and remote sensing. Journal of Spatial Hydrology. 9 (2010).

5. Tehrany, M. S., Pradhan, B. & Jebur, M. N. Flood susceptibility mapping using a novel ensemble weights-of-evidence and support vector machine models in GIS. Journal of hydrology. 512, 332–343 (2014)

6. Kia, M. B. et al. An artificial neural network model for flood simulation using GIS: Johor River Basin, Malaysia. Environmental Earth Sciences. 67, 251–264 (2012).

7. 5. Nayak, P., Sudheer, K., Rangan, D. & Ramasastri, K. Short-term flood forecasting with a neurofuzzy model. Water Resources Research. 41 (2005).

8. Lee, S. Comparison of landslide susceptibility maps generated through multiple logistic regression for three test areas in Korea. Earth Surface Processes and Landforms. 32, 2133–2148 (2007).

9. 3. García-Davalillo, J. C., Herrera, G., Notti, D., Strozzi, T. & Álvarez-Fernández, I. DInSAR analysis of ALOS PALSAR images for the assessment of very slow landslides: the Tena Valley case study. Landslides. 11, 225–246, https://doi.org/10.1007/s10346-012-0379-8 (2014).

10. Shirzadi, A. et al. A comparative study between popular statistical and machine learning methods for simulating volume of landslides. Catena. 157, 213–226 (2017).

11. Zeng, Y., Zhang, Z. & Kusiak, A. Predictive modeling and optimization of a multi-zone HVAC system with data mining and firefly algorithms. Energy. 86, 393–402 (2015).