A REVIEW OF THE PUBLISHED RESEARCH CONCERNING THE GENETIC ALGORITHM

Jayshri Kushwah
Research Scholar
Department of Mathematics
Kalinga University, Naya Raipur.
jayshrikush01@gmail.com

Dr. Rahul Dwivedi
Associate Professor
Department of Mathematics
Kalinga University, Naya Raipur.

Abstract

A technique known as a genetic algorithm is used to estimate computer models using techniques that have been adopted from the biological area of genetics. "Genes" are potential model behaviors that have been encoded in order to employ this approach. After every generation, the present models are evaluated for fitness and given the go-ahead to mate and produce offspring. Gene exchange, crossovers, and mutations may take place during mating. The existing population is eliminated, and the descendants make up the next generation. We shall discuss a variety of issues that may be resolved using a genetic algorithm technique in this article.

Keywords— *Genetic Algorithm, Crossover operators, Mutation Operators*

INTRODUCTION

In order to tackle search and optimization issues, genetic algorithms are adaptable techniques. They are based on how biological creatures use their genetic makeup. Natural populations change over many generations in accordance with the ideas of natural selection and survival of the fittest, which were originally articulated by Charles Darwin in The Origin of Species. If they have been properly programmed, genetic algorithms may solve issues in the actual world by simulating this process. For instance, genetic algorithms may be used to build bridges with the best strength-to-weight ratio or to find the most efficient way to cut forms out of fabric. They may also be used for load balancing on a multi-processor computer system or for online process control, such as in a chemical plant.

Holland initially outlined the fundamental ideas of genetic algorithms in 1975. Genetic algorithms mimic the natural world's processes that are vital to evolution. The specific biological processes that are necessary for evolution and those that play a little or no part in it are still up for debate, but the fundamentals are well known. People in a population compete with one another for resources like food, water, and shelter in the natural world. Additionally, individuals of the same group often compete with one another for a partner. The most resilient and fertile people will produce a disproportionately greater number of children. Individuals who do poorly will have few children, if any at all. As a result, each succeeding generation will see an increase in the number of people carrying the genes of the well adapted individuals.

Sometimes super fit children, whose fitness is higher than that of either parent, may be produced by the fusion of positive traits from several ancestors. In this approach, organisms develop to become ever more adapted to their surroundings. An exact analogy to natural behavior is used by genetic algorithms. They work with a group of people, each of whom stands for a potential response to a certain issue. Each person receives a fitness rating based on how well they solved the challenge. Through interbreeding with other members of the population, the highly fit people are offered chances to procreate. This results in brand-new people who inherit certain traits from both of their parents as offspring. The population's least fit individuals are less likely to be chosen for reproduction and eventually disappear.

LITERATURE REVIEW

Along with genetic programming, evolution methods, and evolutionary programming, the genetic algorithm is a member of the family of evolutionary algorithms. An extensive family of stochastic optimization methods may be thought of as including evolutionary algorithms. An evolutionary algorithm maintains a population of potential answers to the current issue. A series of stochastic operators are then repeatedly applied to the population to develop it. Mutation, recombination, and selection—or a mix of these—commonly make up the set of operators. Let's talk about some of the publications that have been written in the past about genetic algorithms.

Bryant examined the outcomes of using a variety of crossover and mutation operators developed for the traveling salesman problem in this article, and came to the conclusion that operators that employ heuristic data or a matrix representation of the graph provide the best outcomes. Utilizing the survival of the fittest principle, genetic algorithms are an evolutionary method that employs crossover and mutation operators to solve optimization issues. They have been used effectively to a number of issues, including the traveling salesman issue. The goal of the traveling salesman issue is to determine a route across every node in a weighted graph that minimizes the overall weight. A decent solution to the NP-hard traveling salesman issue would be helpful since it has numerous real-world applications.

Enrietch explained in this study "Evolutionary algorithms" that stochastic optimization approaches based on the principles of natural evolution are what evolutionary algorithms are. The basic operation of EAs and a description of the major families into which they may be classified are presented along with an overview of these approaches. The analysis of the various EA components is followed by several examples of how they may be implemented. Finally, it gave a brief overview of the many uses for these strategies. Evolutionary programming, evolutionary methods, genetic programming, and genetic algorithms are some examples of methodologies. These paradigms' representation systems, reproduction operators, and selection techniques all vary fundamentally from one another.

For the solution of scheduling issues that arise in the real world, Madureira devised a genetic algorithm and a coordination mechanism. Delivering goods on schedule and ensuring effective production management are challenging issues because of how often dynamic circumstances change. The purpose of scheduling is to establish an optimal allocation plan that optimizes a certain performance metric by allocating a group of machines to accomplish a set of tasks within a predetermined time frame. The order crossover operator and natural representation are utilized to encode the answers for the implementation problems. They used the inversion mechanism as the operator for mutation. Last but not least, Madureia et al. used a collection of static scheduling methods and a genetic algorithm to solve the dynamic scheduling issue, demonstrating the viability of the genetic algorithm in the Job-Shop scheduling problem.

Sandstrom recommended using the Genetic algorithm, which determines job priorities and offsets to ensure that real-time timing restrictions are met. In a real-time system, assigning a temporal restriction to a job is not an easy issue. They demonstrated how the properties of periodic tasks running on a typical preemptive real-time operating system may be translated to temporal restrictions. They utilized the genetic algorithm because, in situations when it is not feasible to satisfy all of the criteria, the genetic algorithm may provide a solution that fulfills a portion of the temporal restrictions. Natural selection's technique, the genetic algorithm, steadily improves how people are assigned to temporal restrictions in a population. It passed several tests and had positive results.

The assessment of literature benchmarks often used to evaluate optimization processes specialized to multidimensional, continuous optimization tasks is provided by Molga in this study on "Test functions for optimization needs". Multiple-extreme functions, which are used as the quality test for competing optimization techniques, have received particular focus. The effectiveness of optimization techniques is commonly assessed using well accepted literature standards. Such test functions can be divided into several classes, all of which are continuous. The first class is unimodal, convex, multidimensional; the second is multimodal, two-dimensional, with a sparse number of local extremes; the third is multimodal, two-dimensional, with a large number of local extremes; and the last is multimodal, multidimensional, with a large number of local extremes. When class first comprises both good functions and bad situations, the convergence to a single global extreme is weak or delayed. Class second is used to assess the effectiveness of common optimization techniques in an unfavorable environment, meaning one with few local extremes and a single global extreme. Class second is sandwiched between classes one and third. It is advised to examine the effectiveness of intelligent resistance optimization strategies in classes three and four.

Omar suggested utilizing a genetic algorithm to solve the problem of job-shop scheduling. The results of various well-known priority criteria, such as lowest processing time and longest processing time, were included in the original population that was produced at random. From then, unless a set of predetermined stopping conditions were met, the population would reproduce, cross over, and mutate to establish a new population for the next generation. The number of generations is utilized as a stopping point throughout the study. The crucial block neighborhood is utilized in crossover and mutation, and the distance is calculated to assist assess the schedules. The application of the crucial block neighborhood and the distance measured may provide the same results as other techniques, according to the results.

In his research titled "Genetic Algorithm

Approach to Operating System Process Scheduling Problem," Dr. Rakesh Kumar made a suggestion. Operating systems' scheduling has a crucial impact in the throughput and overall performance of the system. The effectiveness of the system depends on an effective schedule. Scheduling is thought to be an NP-hard issue.To provide effective process scheduling, the power of genetic algorithms is leveraged. To assign and schedule the task to the CPU, an effective scheduler is needed.

In his article titled "A Genetic Algorithm on Single Machine Scheduling Problem to Minimize Total Weighted Completion Time," H. Nazif made the suggestion. He covers the scheduling of a single machine family with several workloads in this study. Each work has a processing time and a positive weight that go along with it. These jobs are divided into families, and setup time is needed between these families. He suggests an evolutionary approach for this issue that uses an optimized crossover operator created by an undirected bipartite graph to produce an ideal schedule that minimizes the overall weighted completion time of the tasks while taking into account the setup durations for sequence independent families.

According to Snehal Kamalapur's work, "Efficient CPU Scheduling: A Genetic Algorithm based Approach," CPU scheduling has a significant impact on the performance and throughput of operating systems. It is thought that the scheduling is an NP issue. Performance of the system is improved through effective scheduling. In this essay, she describes and assesses a technique for scheduling processes. She covered the use of genetic algorithms in her research to enable effective process scheduling. And compare the suggested algorithm's effectiveness and performance to those of existing deterministic algorithms in a method that optimizes some performance via simulation.

S. Ramya proposed a novel method for window constrained scheduling that is appropriate for weakly-hard real-time systems in his work titled "Window Constrained Scheduling of Processes in Real Time CPU Using Multi Objective Genetic Algorithm." The original technique, known as Virtual Deadline Scheduling, aims to ensure that m out of k deadlines are met for real-time operations like recurring CPU workloads. VDS is able to create a workable, resource-full schedule that is window limited. However, VDS must update the relevant virtual deadline whenever it serves a task or changes to a new request period. The algorithm's bottleneck is this update, which raises the time complexity. Furthermore, the quantity of context shifts rises as VDS attempts to address the issue of latency. There are two criteria that conflict: context switching and delay. A trade off between the context switching and the latency may be accomplished by employing Multi Objective Genetic Algorithm. The updation issue, which was an extra burden in the original VDS method, is something we address in our algorithm design.

By combining multiple distinct local directed searches into the genetic algorithm process, Birch provides four alternative variants of computationally effective genetic algorithms in this study. These local searches are based on the Newton-Rampson technique, the steepest descent method, the derivative-free directional search method, and the SD/DFDS approach. Through a Monte

Carlo simulation research with a split-plot design, certain benchmark functions—such as a low-dimensional function compared to a high-dimensional function and a somewhat uneven function compared to a highly uneven function—are used to demonstrate the improvement of these suggested approaches. Additionally, a real-world instance of the multi-response optimization issue is utilized to highlight how these suggested approaches outperform both the conventional Genetic algorithm and the method employed by the Design-Expert statistical software program. According to our findings, the Genetic algorithm may often be made more accurate and efficient in terms of computing by including a local directed search.

ADVANTAGES, LIMITATIONS AND APPLICATIONS OF GENETIC ALGORITHM

A. Advantages of Genetic Algorithm

The advantages of genetic algorithm are:

1. Parallelism.
2. Liability.
3. Solution space is wider.
4. The fitness landscape is complex.
5. Easy to discover global optimum.
6. The problem has multi objective function.
7. Only uses function evaluations.
8. Easily modified for different problems.
9. Handles noisy functions well.
10. Handles large, poorly understood search spaces easily.
11. Good for multi-modal problems, returns a group of solutions.
12. Very vigorous to difficulties in the evaluation of the objective function.
13. They are more resistant to becoming trapped in local optima.
14. They perform very well for large-scale optimization problems.
15. Can be employed for a wide variety of optimization problems.

B. Limitations of Genetic Algorithm

The limitation of genetic algorithm includes:

1. The problem of identifying fitness function.
2. Definition of representation for the problem.
3. Premature convergence occurs.
4. The problem of choosing the various parameters like the size of the population, mutation rate, cross over rate, the selection method and its strength.
5. Cannot use gradients.
6. Cannot easily incorporate problem specific information.
7. No effective termination point.
9. Not effective for smooth unimodal functions.
10. Needs to be coupled with a local search technique.
11. Have trouble finding the exact global optimum.

C. Applications of Genetic Algorithm

Genetic algorithms have been used for difficult problems, for machine learning and also for evolving simple programs. They have been also used for some art works, for evolving pictures and music. A few applications of Genetic algorithm are as follows:

- Strategy planning.
- Robot trajectory planning.
- TSP and sequence scheduling.
- Function Optimizations.
- Control–gas pipeline, missile evasion.
- Aircraft design and communication networks.
- Scheduling–manufacturing.
- Machine Learning–Designing

neural networks, both architecture and weights, improving classification algorithms.

CONCLUSION

The fundamental idea behind genetic algorithms is to imitate the natural processes involved in evolution, particularly those that adhere to Charles Darwin's original ideas on the survival of the fittest. As a result, they constitute an intelligent application of a chance search inside a predetermined search area. Numerous studies, experiments, and applications of genetic algorithms have been made in the engineering industry. In this essay, we looked at a variety of issues that genetic algorithms may help tackle.

REFERENCES

[1]. Goldberg, D. E. (1989). Genetic Algorithms in Search, Optimization, and Machine Learning. Boston: Addison- Wesley.

[2]. Davis, L. (1991). Handbook of Genetic Algorithm. Von Nostrand Reinhold, Newyork.

[3]. Mitchell, Melanie (1996). An Introduction to Genetic Algorithms. MIT Press.

[4]. Bryant, Kylie (2000). Genetic Algorithms and the Traveling Salesman Problem, in Proceedings of 1st GNT Regional Conference on Mathematics, Statistics and Applications.

[5]. Coello, Carlos (2000). An updated survey of GA-based multiobjective optimization techniques, ACM Computing Surveys, vol. 32, no. 2, pp. 109-143.

[6]. Hanne, Thomas (2000). Global multiobjective optimization using evolutionary algorithms. Journal of Heuristics, vol. 6, no. 3, pp. 347-360.

[7]. Madureira, A., Ramos, C., & Silva, S. Carmo, (2002). A Coordination Mechanism for Real World Scheduling Problems using Genetic Algorithms. Evolutionary Computation, in Proceedings of the 2002 CEC, vol. 1, pp. 175 –180.

[8]. Molga, M. & Smutnicki, C. (2005). Test functions for optimization needs, in Proceedings of 4th Conference on Genetic Algorithms.

[9]. Omar, M., Baharum, A., & Hasan, Y. Abu (2006). A Job-Shop Scheduling Problem (JSSP) Using Genetic Algorithm ,in Proceedings of 2nd IMT-GT Regional Conference on Mathematics,

[10]. Inazawa, H. & Kitakaze, K. (2006). Locus-Shift Operator for Function Optimization in Genetic Algorithms. Complex Systems Publications, Inc.

[11]. Snehal Kamalapur (2006).Efficient CPU Scheduling: A Genetic Algorithm based Approach. IEEE, pp.206- 207

[12]. Dehuri, S. et al. (2008). Application of elitist multi-objective genetic algorithm for classification rule generation, Applied Soft Computing, pp. 477–487.

[13]. Sivanandam, S. N. & Deepa, S. N. (2008). Introduction to Genetic Algorithms. Springer.

[14]. Birch, J. B. & Wan, W. (2009). An Improved Genetic Algorithm Using a Directional Search. Tech report presented at Virginia Polytechnic Institute and State University, Blacksburg.

[15]. H. Nazif(2009). A Genetic Algorithm on Single Machine Scheduling Problem toMinimise Total Weighted Completion Time. European Journal of Scientific Research,Vol.35 No.3, pp.444-452

[16]. Dr. Rakesh Chawla(2010). Genetic Algorithm approach to Operating system process scheduling problem. International Journal of ngineering Science and Technology, pp. 4247-4252