JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

"Genetic Algorithms"

NAME: SALMA ELMABRI

N° ID: 00947

DR.MOHAMMED SAAD

1. ABSTRACT

Genetic algorithms are a type of optimization algorithm, meaning they are used to find the maximum or minimum of a function. This paper introduced, illustrated, and discussed genetic algorithms for beginning users. Showing what components make up genetic algorithms and how to write them. Advantages and disadvantages, talk about the theory, its elements fitness function, how it operates, the current application.

2. Introduction

The inherent human drive to comprehend and govern the world gives rise to science. Humans have gradually amassed a vast body of knowledge over the course of history that enables to predict a vast array of other natural, social, and cultural phenomena to varying degrees, including the weather, the motions of the planets, solar and lunar eclipses, the course of diseases, the rise and fall of economic growth, the stages of language development in children, and the courses of many different diseases. Even more recently, certain fundamental constraints have been learned for the capacity of prediction. (Melanie, 1999, p2).

Making decisions involves weighing several possibilities and viewpoints in order to decide how to deploy resources to attain a specific objective. Making decisions in project management can be challenging because there are always time and budget constraints. The vast majority of project management problems center on scheduling and planning choices. The quality of the decisions that project managers make throughout the course of a project determines its success to a considerable extent. The project manager's chances of choosing the best choice increase

with the amount of data supplied. However, as the amount of data available grows, decision-making becomes more challenging and sophisticated, necessitating the use of an intelligent decision-making assistance system.

The use of artificial intelligence (AI) in decision-making is often referred to as intelligent decision-making. Genetic algorithms are one of the most important intelligent optimization methods in decision-making (GA). (Chinedu, 2022).

3. What is GA?

A search algorithm based on the principles of natural evolution is known as a genetic algorithm (GA). It cycles through the potential outcomes to choose the best set of variables for optimal or nearly optimal selections. It is an effective optimization system for resolving issues with a great deal of limitations, uncertainty, and potential solutions. Due to its capability to use sequential evolution of two acceptable solutions to build the optimal features, GA is appropriate in situations when quick decisions are required. A GA's search and decision processes can also be guided by knowledge relevant to a certain problem. (Chinedu, 2022).

3.1. GA Theory:

A group of stochastic search algorithms known as evolutionary computing imitate the natural evolution theory first forward by Charles Darwin in 1858. The place of evolutionary algorithms in the field of search techniques is indicated by the following classification:

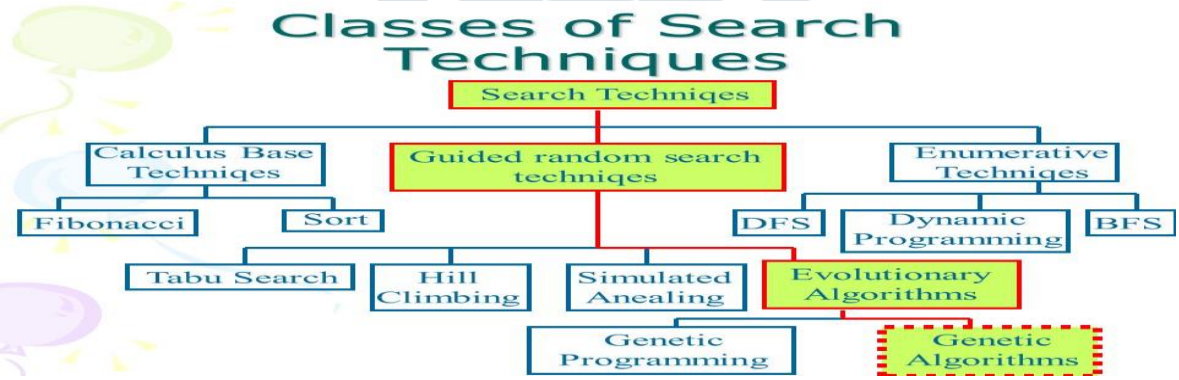

Fig 1. Search techniques.

Consider evolutionary algorithms as a subset of soft computing if we see intelligence as the capacity of an organism to adapt itself to a constantly changing environment:

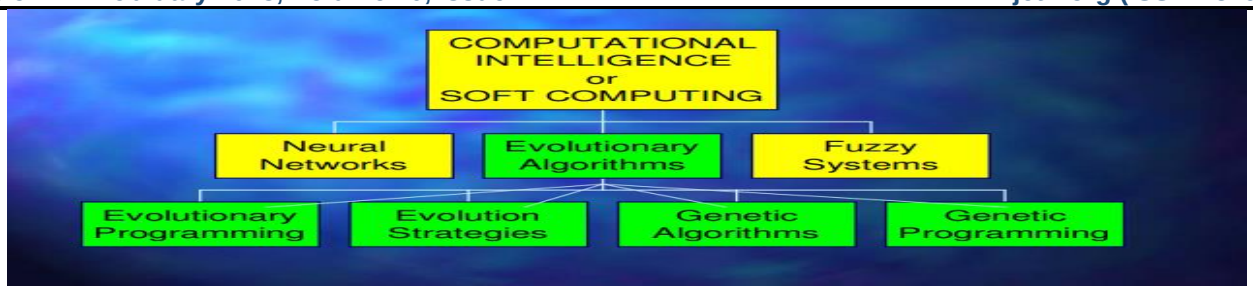

Fig 2. Artificial Intelligence techniques.

These algorithms are constructed from numerous iterations of the fundamental evolutionary cycle:

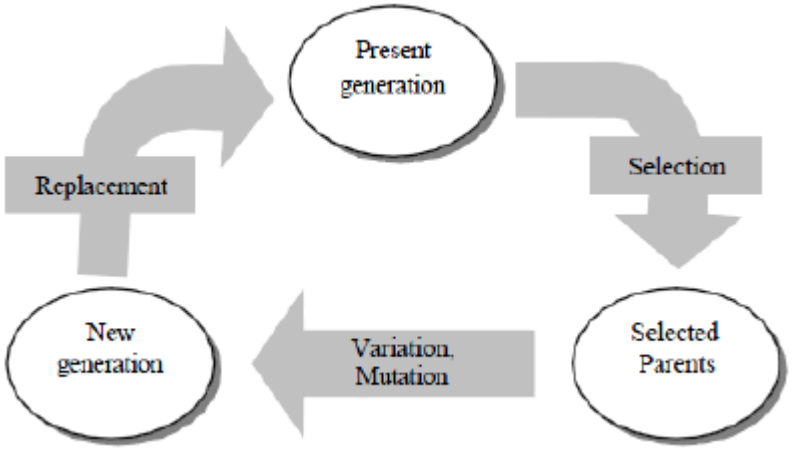

Fig 3. Basic Evolution Cycle

The same basic cycle is incorporated into several Evolutionary Computing versions through various presentation models or particular arrangements of the Variation, Mutation, Selection, and Replacement techniques. The equilibrium between two opposing activities is the intriguing aspect of implementation. The Variation and Mutation operators aim to increase population variety, while the Selection operation aims to decrease population diversity (set of potential solutions). The convergence rate and solution quality are influenced by this fact. Evolutionary algorithms should be studied as an optimization tool to find solutions to problems like: • Rate of convergence • Strength of the developed solution • Computer specifications Evolutionary algorithms' overall structure has not yet been the subject of a general analysis framework, however some particular modifications or implementations may be the subject of theoretical or empirical research. The theoretical approach looks for mathematical facts about algorithms that hold true over a broad range of application domains. The empirical approach, on the other hand, makes an effort to evaluate how well an implementation performs in a particular application area. Both approaches have benefits and drawbacks, and in actual application they should be combined to create and fine-tune the particular instance of an algorithm. (Mehrdad, 2018).

3.2. Elements of GA:

Most "GA" techniques include at least the following components: populations of chromosomes, fitness-based selection, and crossover to create new offspring, and random mutation of new offspring. The fourth component of GAs proposed by Holland, inversion, is rarely employed in current implementations, and its benefits, if any, are not clearly proven. In a GA population, chromosomes are often represented as bit strings. There are two potential alleles at each chromosomal locus: 0 and 1. You might imagine each chromosome as a location in the search area for potential answers. Chromosome populations are processed by the GA, which successively replaces one population with another. The GA frequently calls for a fitness function that rates each chromosome in the present population according to its fitness. A chromosome's fitness is determined by how well it handles the current issue. (Melanie, 1999, p7).

3.3. Encoding:

The issue is crucial to encoding. There are numerous encoding strategies in the literature on GAs. Binary encoding is one of the most popular and straightforward encoding techniques, in which people are represented as a series of bits (zero or one). Although straightforward, binary encoding is frequently not a natural solution for a variety of issues, and modifications could be necessary after carrying out specific genetic procedures. Permutation encoding, which is frequently employed in ordering issues, is another illustration of an encoding technique. An individual is represented by a string of numbers in this encoding, where each number denotes a specific place in a series. Repetition is not permitted, unlike the binary encoding. (uomustansiriyah.edu.iq)

3.4. Initialization:

The next step is to establish the population size after choosing the encoding method that best portrays the issue (the number of individuals). Because people must learn from one another and, more importantly, to prevent premature convergence to suboptimal solutions, which is a problem GAs can experience when there is a lack of diversity in the population, diversity in the population is essential. There are several ways to preserve diversity in the original population, including:

1. Consistently random: The problem's search space is used to generate random individuals using a uniform distribution.

2. Grid initialization: To seed the starting population, specific regions of the search space are chosen. The choice of these components typically depends on the problem.

3. Non-clustering rule: Limiting the newly generated solutions is another technique to guarantee diversity. There is a rule that specifies how many genes a newly generated individual must differ from all previously added individuals in the population by.

4. Heuristic: To create the initial population in GA, deterministic ad-hoc heuristics can be utilized, including best-first search and hill climbing. The GA will provide a solution that is at least as excellent as the seeded solution, which is the driving force behind this. (uomustansiriyah.edu.iq)

3.5. Fitness function:

Akin to the habitat to which organisms (the candidate solutions) adapt is the fitness function. It can mean the difference between finding the best answer and finding none at all because it is the sole phase in the algorithm that predicts how the chromosomes will change over time. According to Kinnear, editor of Advances in Genetic Programming, "fitness function is the sole chance that you have to communicate your intentions to the strong process that genetic programming represents. Make sure it expresses your intentions clearly. In other words, "you can never take too much care in crafting" it.

The fitness function must accurately score the chromosomes based on a range of fitness values in order to distinguish between solutions that are partially complete and those that are more complete. This requires the fitness function to be more sensitive than simply identifying "good" chromosomes from "bad" chromosomes. Kinnear refers to this as "half credit" being given. Because this will affect the direction in which the entire population moves, it is crucial to think carefully about which partial solutions should be preferred over other partial answers. (Jenna, 2014).

4. GA Operators

4.1. Flow Chart:

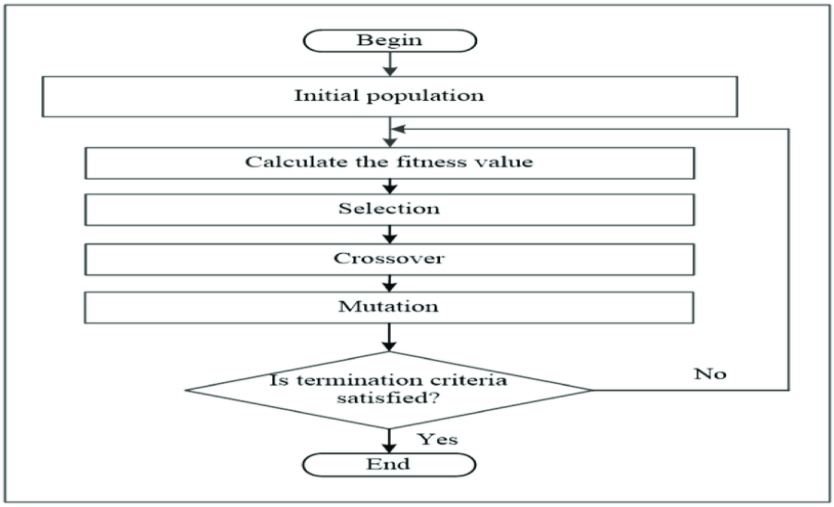

Figure 4. Flow chart of a genetic algorithm (GA). (researchgate.net)

4.2. Selection:

The process that decides which people should be kept around and permitted to procreate while others should be eliminated is referred to as the selection operator. Numerous selection approaches are available in the literature, however they all largely depend on the fitness function. The proper selection operator, i.e., one that avoids picking poor answers and permits the survival of good solutions to the problem under consideration, must be chosen, although this depends on the particular situation. Common selection methods include, for instance:

1. Random selection: In this sort of selection, which is thought to be the simplest selection operator, candidates are chosen at random regardless of how fit they are, giving both the best and the poorest candidates an equal chance of selection. Random selection has the lowest selective pressure towards fitter people when compared to other selection mechanisms.

2. Proportional selection: One of the most well-liked selection operators is proportional selection. Contrary to the random selection operator, proportional selection favors fitter people; that is, it makes choices based on fitness. Each candidate has a selection probability that is inversely correlated with its fitness. As a result, those with the highest fitness scores are more likely to be chosen for reproduction. Since a result, as a high selection pressure is exerted in such a selection operator towards fitter people in the population, better individuals are anticipated to evolve over time. The two methods for applying proportional selection are stochastic universal selection and roulette wheel selection.

3. Tournament selection: Several tournaments are held among a small group of people who have been randomly chosen from the population in a tournament selection. The most physically fit competitor advances to the next round of each competition. (uomustansiriyah.edu.iq)

4.3. Crossover:

Modification >> this entails using a variety of genetic operators to alter the population's members individually. The following two common genetic operators can be employed to change people:

The crossover operator: Reproduction and biological crossover are both equivalent to the crossover operator. By choosing members of the parental generation and exchanging their genes, new solutions (offspring) are produced. Crossover enables the search process to investigate new areas of the solution space and produce high caliber candidates for the following

generation. In the literature on evolutionary computing, a variety of crossover operators have been reported, most of which are solution representation-dependent.

1. One-Point Crossover (1P): Given two parent solutions, 1P is the pioneer crossover operator. It begins by creating a random position between 1 and the person's size-1. Each parent is split in half by this position, which acts as an exchange point. The two first segments of the parents are exchanged to produce two new offspring.

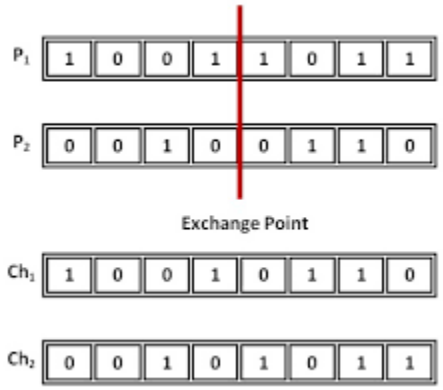

Fig.5 shows an example of 1P crossover.

2. Two-Point Crossover (2P): The crossover operator 2P is a generalized version of the 1P crossover operator. Unlike the one-point crossover, this operator starts by creating two random cutting points between 1 and the individual's size-1 when given two parent solutions. These roles act as places of exchange, dividing each parent into three halves. By switching the parental segments between the two cutting locations, two new offspring are produced.

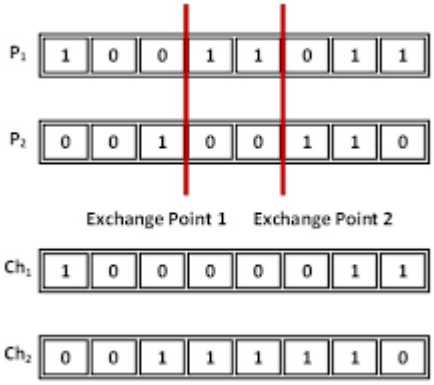

Fig.6 shows an example of 2P crossover

3. Half Uniform Crossover (HUX): HUX, as originally proposed in reference [150], calls for the swapping of half of the non-matching genes. To do this, the Hamming distance, or the number of different genes, is determined and then cut in half. The calculation yields the number of bits that are unmatched genes between the two parents, and consequently the number of genes that need to be swapped.

4. Order Crossover (OX): A straightforward permutation crossover operator, order crossover.

5. Matched Crossover with Some Match (PMX): The most popular crossover operator for issues using a permutation-based representation is PMX, also known as Mapped Crossover.

6. Cycle Crossover (CX): a different crossover operator used for permutation-based representation, is another option. (uomustansiriyah.edu.iq)

4.3. Mutation:

One of a GA's most crucial components, the mutation operator is connected to the search space exploration. Individuals are randomly transformed through mutation, maintaining and introducing variation into succeeding generations. Several mutation operators, including: are described in the literature on GAs.

1. Random mutation: The first mutation operator was random mutation. This kind of mutation flips the value of a randomly chosen place within an individual. Since it is simple to switch from 0 to 1 and vice versa, this method excels at encoding problems that can be expressed in binary form.



Fig.7. An example of the insert mutation.

2. Insert mutation: In this operator, two places, POS1 and POS2, are randomly chosen from an individual so that 1 POS1 POS2 n, where n is the length of encoding. POS2 is then put immediately before POS1, and all positions from POS1 to POS2 are right shifted. Fig.7. shows an example of the insert mutation.

3. Swap mutation: Similar to the insert mutation operator, an individual is randomly chosen from two positions, 1 POS1 POS2 n, where n is the length of encoding. However, the swap operator only swaps their values rather than adding POS1 before POS2. (uomustansiriyah.edu.iq)



Fig. 8. Shows an example of the swap mutation.

5. GA application

The optimization of issues and solutions is one of the fundamental uses for genetic algorithms, which have many other uses as well. We employ optimization to locate the best answer to every issue. There are various advantages to undertaking optimization using genetic algorithms, which can be referred to as genetic optimization. Here are 10 examples of actual genetic optimization applications:

-Travelling salesman problem (TSP).
-vehicle routing problem (VRP).
-financial markets.
-manufacturing systems.
-mechanical engineering design.
-data clustering and mining.
-image processing.
-neural networks.
-wireless sensor networks.
-medical science. (analyticsindiamag.com)

6. Advantages and Disadvantages of GA

Through its use in scheduling under numerous constraints, time-cost trade-offs, critical path problems, resource leveling, facility architecture, and finance-based scheduling, GA has proven its promise as an optimization tool. The current success in using GA as a decision-making optimization tool could be attributed to its drivers, including its capacity to process multiple solutions at once, quick output, capacity to choose solutions based on the fitness value assigned to each chromosome, capacity to find the global optimum and avoid becoming stuck in a local optimum, capacity to resolve poorly understood problems, and capacity to change fitness

functions depending on the desired solution Despite the advantages of GA mentioned above, there are still several drawbacks that restrict its use as a tool for decision-making optimization. Due to its complexity and the length of time required for its creation, a robust GA optimization model has a high development cost. Additionally, solution coding with GA may use architecture that is problem-specific and not necessarily transferable to other problems. (Chinedu, 2022).

CONCLUSION

Today, big businesses employ genetic algorithms to develop everything from giant aircraft to tiny computer chips to pharmaceuticals and to optimize schedules. Because it can be used to solve both discrete and continuous optimization problems and can look for solutions from a sample population rather than a single point, GA differs from other conventional optimization algorithms in a significant way. This significantly lowers the risk of becoming stuck in a local optimum. However, there are drawbacks to GA, such as the complexity needed for its development and the potential for early convergence when the wrong initial sample population is chosen. GA has proven effective despite the limited restrictions it has.
This paper's contribution is a thorough analysis of GA. a condensed idea of GA.

REFRENCES

- Chinedu Okonkwo, Richard Garza, Tulio Sulbaran, Ph.D., and Awolusi Ibukun, Ph.D. A, EPiC Series in Built Environment Volume 3, 2022, Pages 254–262 ASC2022. 58th Annual Associated Schools of Construction International Conference, Review of Genetic Algorithm as a DecisionMaking Optimization Tool in Project Management, the University of Texas at San Antonio San Antonio, Texas.

-Genetic algorithm (GA), Access date (22.8.2022), www.uomustansiriyah.edu.iq

-Jenna Carr May 16/2014, an Introduction to Genetic Algorithms, Whitman.edu

- Mehrdad Dianati, 2 Insop Song, and Mark Treiber, Electrical & Computer Engineering, Systems Design Engineering, and Mechanical Engineering,2018, An Introduction to Genetic Algorithms and Evolution Strategies, 1,2,3200 Univ. Ave. West, University of Waterloo, Ontario, N2L 3G1, Canada

- Melanie, M.(1999). An Introduction to Genetic Algorithms. Fifth printing. A Bradford Book the MIT Press. Cambridge, Massachusetts • London, England

-Mustafa albadr , masri ayoub , Sabrina tiun , fahad taha al-dhief , October 2020, Genetic Algorithm Based on Natural Selection Theory for Optimization Problems , www.researchgate.net

- Real-life applications of Genetic Optimization, Access date (11.8.2022), www.analyticsindiamag.com