



**Università
di Genova**

GA - Genetic Algorithms

Method explanation, its fundamentals, pros and cons.

Naghmeh Hajipour (5180329)

pourya Hajihosseini (5280539)

Mohammad biderang (5166097)

Abstract: Nowadays the increasing improvement in processes and systems has caused an increase in data volume. In addition, the nature of some problems adds to the uncertainty. There are many algorithms and methods for solving problems. One of the most widely used of these approaches is the Genetic Algorithm. Genetic Algorithms is a computer science search technique for finding approximate solutions to optimize models, mathematics, and search problems. They are often a good choice for regression-based forecasting techniques. Genetic algorithm modeling is a programming technique that uses genetic evolution as a problem-solving model. The problem to be solved has inputs that are converted to solutions during a modeled process of genetic evolution, then the solutions are evaluated as candidates by the fitness function, and if the exit condition is met, the algorithm ends. First we introduce GA and its fundamentals then we explain about how GA can solve industrial problem and its pros and cons.

Introduce: The genetic algorithms (GA) is a computational model for simulating Darwin's natural selection and biological evolution. In 1975, Michigan University's professor J. Holland published *Adaptation in Natural and Artificial Systems*, which demonstrated a way to search for optimal solutions by simulating natural evolution processes. The two selected parents create new offspring whose good genes are preserved throughout the population's gene pool. Subsequent people replace previous people, and the population continues from generation to generation until the disproportionate people are slowly eliminated. In fact, the basic concept of GA was the method from which real-world biological operations were derived. In the wild, every creature has a certain lifespan, but only certain individuals can survive and continue their gene pool. The two selected parents create new offspring whose good genes are preserved throughout the population's gene pool. Subsequent people replace previous people, and the population continues from generation to generation until the disproportionate people are slowly eliminated. [1]

Literature survey: Thanks to the discovery of the molecule deoxyribonucleic acid (DNA) – (Figure 1) in 1951 by biologists Crick and Watson, the mysterious factors behind evolution are known as the composition and sieving of gene pools that affect reproductive changes and differential mutations. Genotypes as Kehoe points out, populations can be thought of as gene pools, while genes are chemical compounds, parts of long strands of DNA that can be called chromosomes. A person's set of genes is their genotype. [2] [3]



Figure 1: DNA

An unresolved question is what happens if one form of GA evolves to mimic more natural species. CGA is a new framework, apart from developing more applications for modern GAs, (CGA) does not evolve out of a desire to improve the performance of GAs, but rather by imitating more than real life. The new approach in the CGA involves assigning a certain "lifespan" to each individual,

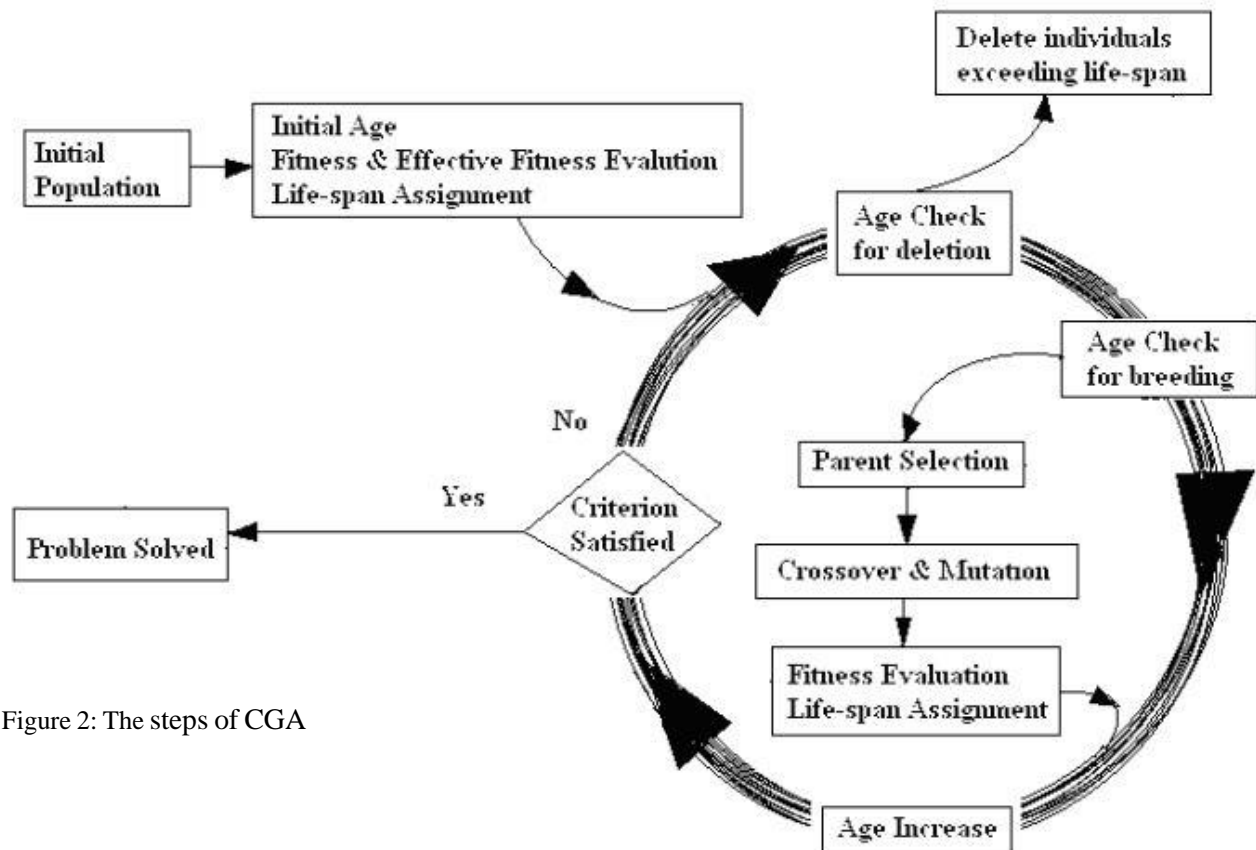


Figure 2: The steps of CGA

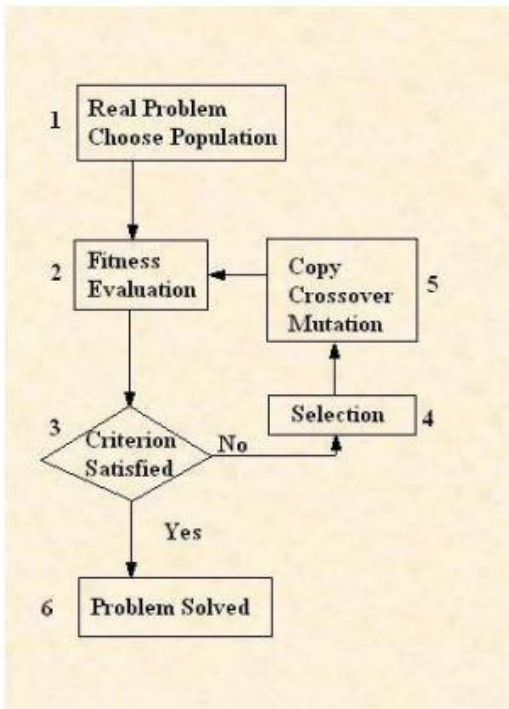
The entire population is first assessed for age and older people are excluded. Next, the remaining people check their age to see if they are old enough to be considered for breeding. Those who are old enough are considered to be the parents' in gene pool. There are still questions about what other items are effective in real life that can be examined as a feature in GA, Figure 2 depicts the steps of CGA.

Some additional feature is also discussed involving multiple populations which have to compete for a limited resource such as water, an illness parameter, accidental death, food shortages, food shortages and etc. [4]

Fundamental: When we are faced with innumerable situations in an ever-changing environment of vital species, simulation software can in turn help to find the most optimal solution. In 1975, John Holland defined the concept of genetic algorithm in his paper [5]. According to Mitchell 1998, the survival of the fittest is a vital trait in which the most suitable members of a population determined by fitness by a suitable function have the best chance of reproduction and thus successfully pass on their genes to the next generation. While all members of the population are considered for reproduction. Selection is done

randomly with a fixed percentage of the most deserving members of the population being passed on to the next generation. [6]

figure3: illustrates the framework of GA with explanation given as follows.



The procedure of GA (as shown in Figure 3) consists of 6 essential steps:

- Set initial population size for the real problem and create initial random population.
- Calculate the fitness value for each member of the population based on its evaluation against the current problem.
- Check all the fitness with criterion satisfied function.
- If „No“,
- Select parents by fitness and solutions with higher fitness value are most likely to be parent during reproduction.
- Crossover & mutation to produce the new offspring then replace the weak individuals by the new solution, thus one generation is complete.
- Back to step 2.
- If „Yes“,
- Problem has been solved.

When faced with a problem, the parameters need to be encoded, the problem must be defined as a function so that a decision can be made to terminate the search. Each individual possible solution is like a chromosome in the real life species. As mentioned, GA consists of the following components:

- **Population of chromosomes:** The chromosome is typically represented by a form of binary bit string in a GA population. Each chromosome is a point in the search space for all possible solutions. Chromosome populations occur continuously by replacing one population with another. If chromosomes are not flexible, the search is limited and it is impossible to find the right solution. chromosomes are also referred to as genotypes or individuals
- **Selection according to fitness:** The fitness level of each chromosome is assessed by a score calculated through a specific fitness function. GA search starts with relatively weak performance information and results in good results. Since fit for each chromosome is defined as the ability of one chromosome to solve a problem. One of the common applications of GAs is optimization, which aims to identify a set of parameters that maximize or minimize a complex multipara metric function. The operator selects the parents according to their readiness [7]. Solutions that have a higher fit value are more likely to generate new solutions during reproduction. All selection functions are stochastically designed. This helps maintain population diversity.
- **Crossover to produce new offspring:** This operator is the key of Genetic Algorithms“ power. There are several forms of crossover: e.g. n-point crossover and uniform crossover [8] [9], considering a point crossover, for example, randomly selects a point in two bit strings and exchanges the sequence to that place between the two parents to produce two offspring. The crossover operator mimics the law of biological recombination. He does. Without crossover, species reproduction can have

populations containing the genes of successful members of one or other parents (and also be affected by mutations), however, no member has both. By crossover, beneficial mutations in both parents can occur immediately after reproduction. The usefulness of putting two beneficial components together can also be disruptive and break up a good individual, but overall, the selective pressure in GAs is positive. But it is almost inevitable that it will disrupt good solutions, in such cases mimetic algorithms tend to be used for the final stages [10] [11] and the offspring are naturally adapted to the environment, like native species.

- Random mutation of new offspring: our definition, is one point mutation method. One decimal figure between 0 and 1 is random chosen to each individual. If that figure is below the value of mutation rate, a location is randomly chosen and mutation takes place. In this case, every individual has only 1 location to change from 0 to 1 or 1 to 0.

Application: Computing the Hubble Constant: Astronomer Edwin Hubble was the first to notice the expansion of the universe. His observations helped to formulate what is now known as Hubble's law, which states that the velocity v , in which two galaxies recede from one another, is proportional to the distance D between them through the Hubble constant H .

$$v=HD$$

This relationship, shown in Figure 3, shows that the expansion of space is uniform on a large scale, where the mutual gravitational attraction between the galaxies is negligible.

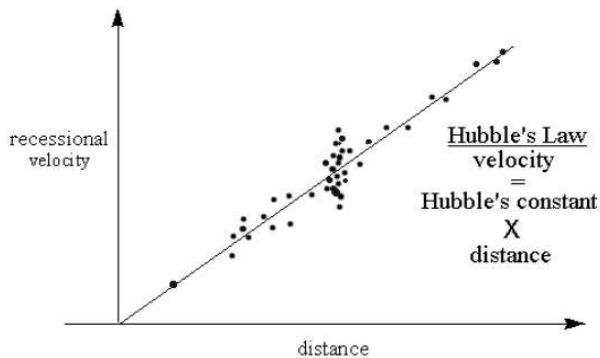


Figure 4: Hubble's Law. The slope of the above line is the Hubble constant, the proportionality constant that relates the distance of receding galaxies to their velocities [12]

While there is some degree of uncertainty in the value of H , it is thought to be in the range of 45-90 km/sec/Mpc [13], with the best current data suggesting a value near 70, km/sec/Mpc. In this example, the best-fit value for the Hubble constant is found using data taken from a sampling of twelve galaxies. Since v and D have a linear relationship, the problem simply consists of finding the best-fit slope of a line, given the x -data and y -data (the distances and velocities, respectively). However, the observation data for this problem aren't v and D , so additional calculations must be carried out in order to obtain values for recessional velocity and distance, which are to be used in the least-squares merit function.

$$(1) \rightarrow D=10^{(m-M+5)/5}$$

Table 1

Object	m
Uma1-2	14.7
Uma1-3	14.6
Uma1-1	14.5
CrBor2	15.5
CrBor1	15.4
Boot2	16.8
Boot3	16.7
Coma2	12.5
Coma3	12.7
Uma2-1	16.6
Uma2-3	16.8
Uma2-2	16.8

Table 2

Object	λ_k	λ_h
Uma1-2	4134	4172
Uma1-3	4134	4170
Uma1-1	4136	4172
CrBor2	4216	4254
CrBor1	4218	4256
Boot2	4452	4492
Boot3	4452	4488
Coma2	4024	4058
Coma3	4022	4058
Uma2-1	4472	4514
Uma2-3	4472	4510
Uma2-2	4476	4512

table1[14]: Galaxies and their Apparent Visual Magnitudes, shows twelve galaxies and these particular galaxies have an absolute magnitude of -22. From the data in Table 1, the distance of each of these galaxies can be obtained in parsecs units.

table2 [14]: Galaxies and their corresponding Ca K and H spectral lines, Table 2 lists these values in Angstrom. The light that leaves a retreating object is transmitted by the expansion of space to the red end of the visible spectrum. For small enough transitions, $\Delta\lambda$,

The quantity, M, in equation (1) denotes absolute magnitude. The corresponding recessional velocities were obtained in the following manner.

$$(2) \rightarrow v = \frac{\Delta\lambda}{\lambda} c,$$

Where c is the speed of light, and λ_0 denotes the stationary wavelength, measured in Angstroms. For the Ca K and H lines, these are 3933.67 and 3968.847 Angstroms, respectively. For each galaxy in Table 3, equation (2) is used in computing the two recessional velocities associated with each spectral line. The accepted velocities used in obtaining the Hubble constant are taken as the average of these two values.

$$(3) \rightarrow v_k + v_h / 2$$

ctrl(1) - number of individuals in a population (default is 100)

ctrl(2) - number of generations over which solution is to evolve (default is 500)

ctrl(3) - number of significant digits (i.e., number of genes) retained in chromosomal encoding (default is 6)

ctrl(4) - crossover probability; must be ≤ 1.0 (default is 0.85).

ctrl(5) - mutation mode; 1/2/3/4/5 (default is 2)

ctrl(6) - initial mutation rate; should be small (default is 0.005) (Note: the mutation rate is the probability that any one gene locus will mutate in any one generation.)

ctrl(7) - minimum mutation rate; must be ≥ 0.0 (default is 0.0005)

ctrl(8) - maximum mutation rate; must be ≤ 1.0 (default is 0.25)

ctrl(9) - relative fitness differential; range from 0 (none) to 1 (maximum). (default is 1.)

ctrl(10) - reproduction plan; 1/2/3=Full generational replacement/Steady-state-replace-random/Steady-state-replace-worst (default is 3)

ctrl(11) - elitism flag; 0/1=off/on (default is 0) (Applies only to reproduction plans 1 and 2)

ctrl(12) - printed output 0/1/2=None/Minimal/Verbose (default is 0)

Figure 4 : Control parameters in the genetic algorithm program Pikaia[15]

The Pikaia subroutine contains twelve configurable parameters. These and the default values are listed in Figure 4. For this particular problem, all ctrl array elements except the first two are stored in their default values. The number of the population was set at fifty, and the number of generations was set at one hundred. As can be expected, the probability of successful convergence is optimal. In general, much depends on the values of these two input parameters.

Of course, the inherent randomness involved in the algorithm's exploration and exploration of the search space does not always guarantee that larger populations and generations will provide better solutions than smaller ones. However, for simple problems like this, the recursive solution does not appear to change significantly with changes in the values of the input parameters, in particular population size and number of generations.

Repeated executions of the evolutionary run with increasingly greater values of the two, aforementioned parameters reveals a convergence in the fitness of the returned solution, where greater values offer no improvement. With the distance and velocity values obtained from equations (1) and (3), a chi-square measure of fitness is performed using the obtained distance and velocity values

$$\chi^2 = \left(\frac{HD - V}{\sigma} \right)^2$$

(5)-

This is carried out in a separate subroutine as shown below.

Sum=0.

Do i=1, numb

Sum = sum + ((H*D (i) - V (i))/err (i)) **2

End do

The integer numb denotes the number of data points, in this case, fifteen. The arrays D (i) and V(i) contain the fifteen values of the distances and velocities, respectively, and err(i) denote the error estimates of each data point, assumed here to be constant. The fitness is then defined as the inverse of the sum, **(6)- fit= 1/sum**

The solution to be returned by the algorithm is defined in the call program as an array, x (n), bounded in region [0,1]. The n argument is the number of parameters that define a solution. For the current Hubble constant finding problem, only one parameter is fitted, and hence n = 1. In general, for an n-dimensional array, x (n), the search space in any dimension can be multiplied by. With a real number, M.

(7)- A = x (n_i)*M, Thus, for dimension i, the search for a global optimization is limited to the interval [0, M], with A denoting the normalized ith parameter defining a solution. So while these are the elements of the x (n) array that are encoded and manipulated by the "breeding" process operation, proportionality measurements need to be scaled accordingly. In order for the return answer to have an appropriate large order, the variable H is expressed in the fitting subroutine in terms of a one-dimensional array x (1).

(8)- H = x (1)*100, The best-fit value of H was found to be

H = 72.029 km/s/Mpc

Note that a solution expressed in these units requires that the distances calculated in Equation (1) be converted from parsk to megaparsk before performing the fitting method.

Pros and cons: GA like any other approach, has a number of advantages and disadvantages. As mentioned earlier, genetic algorithms is in the category of search algorithms. However, they are very different from other search algorithms. Genetic algorithms, instead of dealing directly with some problems, work by displaying the coding of problem sets, creating a population of points in a search space to find the answers. They also optimize the target without knowing the "gradient" information associated with the subject "Objective Function".

Genetic algorithms use "probabilistic" mechanisms to "transition" from one state in the problem space to another. Conventional search algorithms, on the other hand, use target-specific information for this purpose. Such an important feature of genetic algorithms is that they become "all-purpose" search algorithms. Genetic algorithms are also used to search for irregular and irregular search spaces. In general, genetic algorithms are used to solve problems in applications such as function optimization, "parameter estimation", and "machine learning". [16]

Pros	Cons
Easy to understand	Time- consuming
Ability to deal with complex problems	Computational complexity
Parallelism	Stochastic
Good for noisy environment	Computationally expensive
Supporting Multi-objective optimization	-
Ability to optimize continuous and discrete problems	-

Conclusions: The use of GA for the optimization of large-scale problems is computationally intensive task, The GA is a probabilistic solution to optimize the problems that are modeled on a genetic evaluation process in biologically and are focused as an effective algorithm to find a global optimum solution for many types of problems. The GA is used in different artificial intelligence applications like object-oriented systems, robotics, and futuristic emerging technologies. In GA method, with a unique search, the most optimal answer is selected and the first possible answer is avoided as the final optimal result.

References:

- [1] Chiras, D. D. 2006. Environmental Science. Seventh edition. America: Jones & Bartlett Publishers.
- [2] Ladd, S. R. 1996. Genetic Algorithms in C++. New York: M&T Books
- [3] Kehoe, A. B. 1998. Humans: An Introduction to Four-field Anthropology. London: Routledge.
- [4] Alan Kwan, a new multiple population genetic algorithm 2010
- [5] Holland, John H, 1975. Adaptation in Natural and Artificial Systems. America University of Michigan Press, Ann Arbor.
- [6] Mitchell, M. 1998. An introduction to Genetic Algorithms. Cambridge: The MIT Press.
- [7] Goldberg, D. E. 1989. Genetic Algorithms in Search, Optimization and Machine Learning. Canada: Addison Wesley Longman, Inc
- [8] Reeves, C. R. 1994. Genetic Algorithms and neighbourhood Search. Fogarty, T. C. (ed) Evolutionary Computing: AISB Workshop, Leeds, U.K., April 11-13, 1994 : Selected Papers. UK: Springer.
- [9] Eshelman, L. J. 1991. The CHC Adaptive Search Algorithms: How to Safe Search When Engaging in Nontraditional Genetic Recombination. Rawlins, G. J. E. (ed) Foundations of Genetic Algorithms. USA: Morgan Kaufmann Publishers, Inc.
- [10] [6] Grefenstette, J. J., De Jong, K. A., and Spears, W. M. 1993. CompetitionBased Learning. Chipman, S. F. Meyrowitz, A. L. (eds) Foundations of Knowledge Acquisition: Machine Learning. Publisher: Springer
- [11] Spears, W. M. and De Jong, K. A, etc. 1993. An overview of evolutionary computation. In Computer science, Volume 667/1993.
- [12] http://astrosun2.astro.cornell.edu/academics/courses//astro201/hubbles_law.htm
- [13] <http://astrosun2.astro.cornell.edu/academics>
- [14] K. Duckett, A Laboratory Textbook for Introductory Astronomy, Fifth Edition, Contemporary, (1998)
- [15] <https://www.hao.ucar.edu/Public/models/pikaia/pikaia.html>
- [16] <https://electricalvoice.com/genetic-algorithm-advantages-disadvantages/>