Volume 5, Issue 4, 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

A Review on Genetic Algorithm Operators Based on Representation Scheme

Indresh Kumar Gupta
ABV-Indian Institute of Information Technology and Management,
Gwalior, Madhya Pradesh, India

Abstract— Genetic Algorithm (GA) have been extensively used as search and optimization technique in various domains, including the sciences, commerce and engineering. The primary reasons for their success are their broad applicability ease of use and global perspective. In Genetic Algorithm solutions are represented mostly either in binary form or floating point form based on the type of problem which are under consideration. Based on representation scheme we presents the operator used, which drive the process of algorithm.

Keywords— Genetic Algorithm (GA), Evolutionary, Representation, Binary, Floating Point, Operator

I. Introduction

.Genetic Algorithm (GA) mimics natural evolutionary principles to constitute search and optimization process. It was first given by Fraser[1] and later by Bremermann[2] and Reed[3], it was extensive work done by Holland[4] that promote the GA. Holland is considered as father of GA. because they carried out more work for it. G.A is based on genetic evolution, where the characteristics of individual are expressed using genotypes. Genotypes is basically a chromosome, which can be represented as binary or floating point, The main driving operators of GA are selection crossover and mutation.

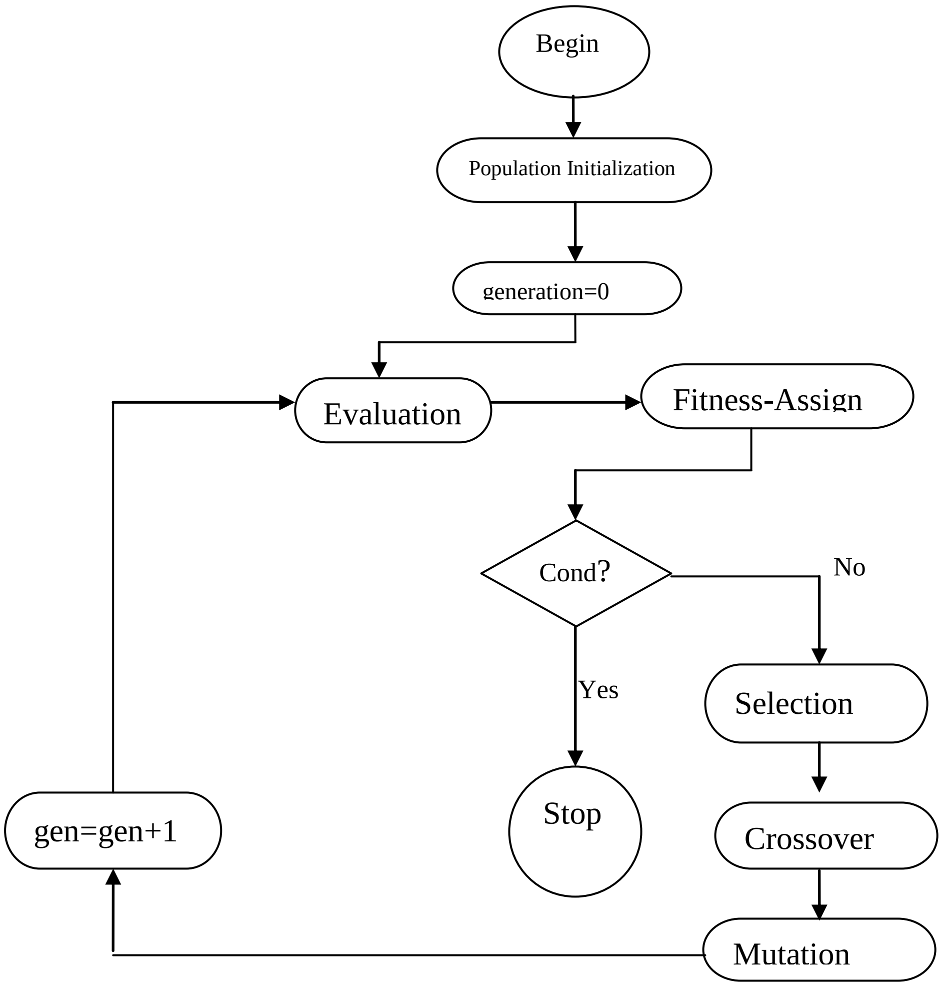


Figure1: flow chart for Genetic Algorithm

As shown in flowchart GA starts its search with a random set of solutions (population), rather than one solution. Once a random population (solutions) is constitute, each is evaluated and fitness is assigned to every solution. The evaluation of a solution means calculating the objective function value and constraint violations. Therefore, a metric must be defined by using the objective function values and constraint violations to assign a relative merit to the solution (called fitness). After that termination condition is checked. If the termination condition is not fulfil, the population (solutions) is modified by three operator selection, crossover, mutation and a new (expected better) population is created. The generation counter is incremented by one to indicate that one generation of the GA is completed.

II. SELECTION

The primary goal of selection operator is to identify the bad solutions, eliminate them and keep the duplicates of the good solutions. There are various selection operators which are as follow

Random Selection: It is simplest selection approach, in which selection probability of each individual is$1/\mu$ (μ is the population size). No fitness information is used, which implies that best and worst individuals have exactly the same chance of surviving to the next generation.

Proportionate Selection: If all population member fitness is F_{avg} , then i^{th}individual with fitness F_i gets an expected F_i/F_{avg} number of copies [4].

Roulette Wheel Selection: In this approach , a wheel is divided into μ parts , and each individual in the population gets a parts of roulette wheel whose portion is proportional to the fitness value that means larger the fitness value is , larger the portion is . Then wheel is spun μ times and each time when wheel is stop individual corresponding to that portion is selected [5]. In proportional roulette wheel selection, selection probability of individual is directly proportional to fitness value. If the fitness value for i^{th} individual is F_i then its selection probability is $P_i = F_i/\sum_{j=1}^{\mu} F_j$.

.Rank Based Selection: When fitness value of an individual is very high then roulette wheel suffers. If an individual is 75% fittest, then it's got 75% portion of the roulette wheel and selected most of the times. To overcome this rank based selection is performed [6] in which individual is sorted based on their fitness value. The best fitness, worst fitness have value μ and 1 respectively. Based on ranking, selection probability of i^{th} individual is $P_i = \mathrm{rank}(i)/\mu(\mu - 1)$.

Tournament Selection: In this approach a tournament is performed among $t < \mu$ (μ is the population size) individual , the individual who has the highest fitness win the tournament and inserted into the matting pool. If we want size of mating pool be s, then s time tournament is performed. Tournament winner creates the mating pool [7].

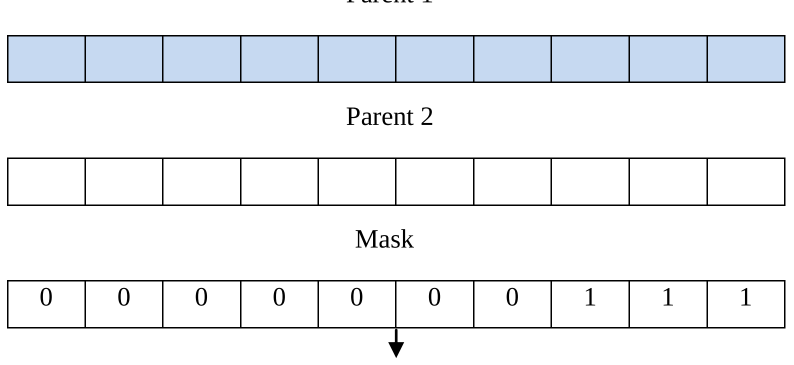
Selection of New Population: In each generation from μ parents λ offspring are produced using crossover and mutation, now individual to be selected for next generation. Two fundamental approach are $(\mu + \lambda)$ and (μ, λ) .In $(\mu + \lambda)$ algorithm produced λ offspring from μ parents, with$1 \leq \mu \leq \lambda < \infty$. From μ parents and λ offspring μ individual are selected to survive in next generation. This strategy implement the elitism to guarantee that the fittest individual survive in next generation. In (μ, λ) strategy μ best individual are selected from λ offspring with$1 \leq \mu < \lambda < \infty$. As elitism is not used, thus this strategy shows a lower selective pressure than for the plus strategies. Elitism refers to the process of ensuring that the best individuals of the current population survive to the next generation [8].

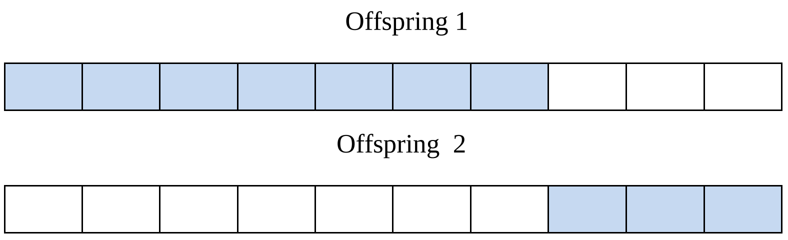
III. CROSSOVER

Crossover operator is used to create the new solutions (i.e. offsprings). Crossover operator can be split into three fundamental categories in light of the number of parent used. These outcomes in three fundamental classes of crossover operators: Asexual: where an offspring is produced from one parent. Sexual: where one or two offspring is produced using two parents. Multi-recombination: where one or more offspring is produced using more than two parents. Based on the representation scheme used in the algorithm crossover operator further classified into two main categories: (1) binary representation crossover (2) floating point representation crossover.

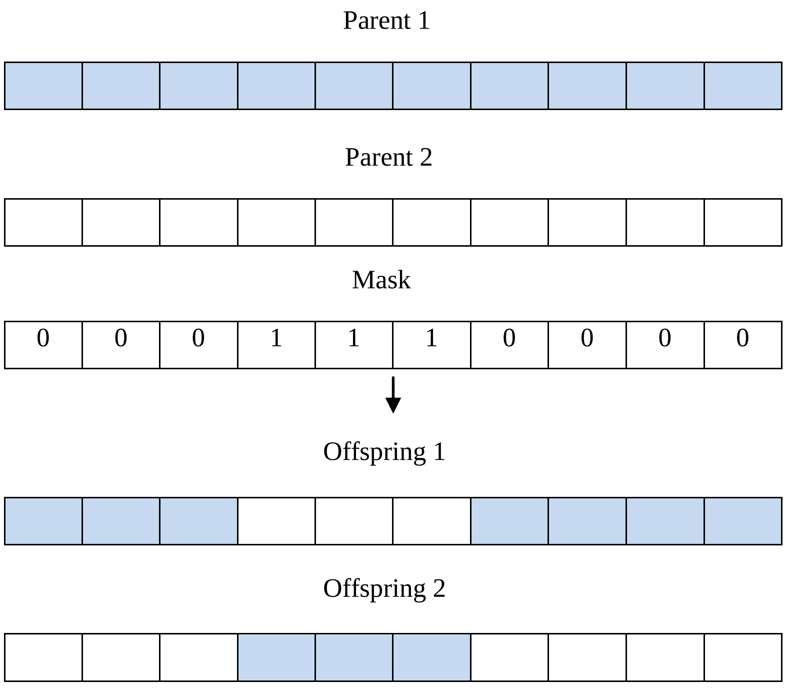
Binary Representation: Mostly binary representation uses the sexual, where two parents is used to create the offspring. Suppose $X_1(k)$ and $X_2(k)$ represent the two chosen parent and $m(k)$ is a mask that indicates which bits of the parents should be swapped to generate the offspring $X_1'(k)$ and$X_2'(k)$. Binary representation crossover operators are as follows

One-Point Crossover: In this a segments of genes be swapped between the parents to create their offspring. A one point crossover operator was developed that randomly select a crossover point, and bit strings after that point are swapped between the two parents [4].

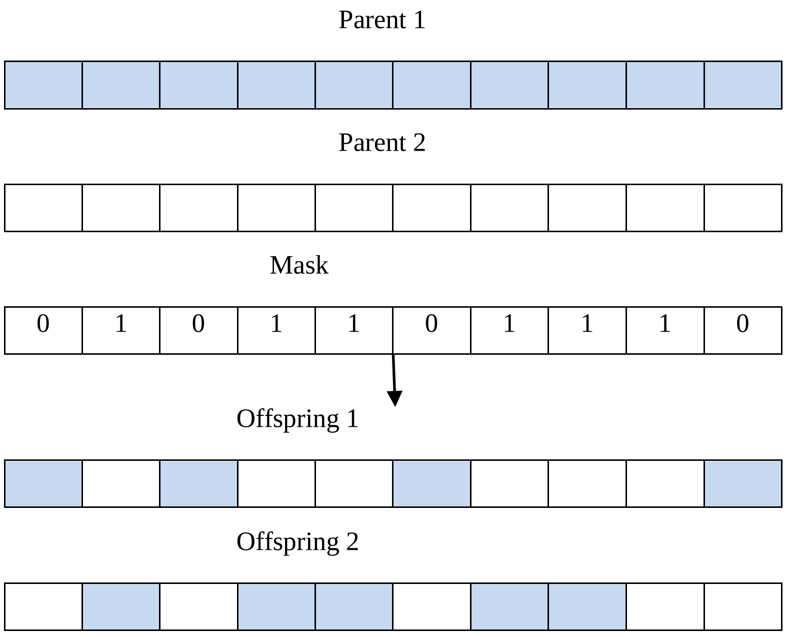




Two Point Crossover: In this bit string between two randomly selected positions is swapped. Position is selected using mask. This operator can be generalized to n point crossover [3][9][10].



Uniform Crossover: For this the n- dimensional mask is created randomly [11][12]. Here p is the bit swapping probability. In the event that$p = 0.5$, then every bit has an equivalent opportunity to be swapped. Uniform cross over is illustrated below.



Floating Point Representation: For this linear crossover, directional heuristic crossover, arithmetic crossover, blend crossover, geometrical crossover, simulated binary crossover are mostly used. Suppose n is number of the decision variables.

Linear Crossover: Wright [13] give the linear crossover , from the parents $X_1(k)$ and $X_2(k)$, three offspring are produced as $(X_1(k)+X_2(k))$,$(1.5X_1(k)-0.5X_2(k))$and $(-0.5X_1(k)+1.5X_2(k))$. The two best solutions are chosen as the offspring.

Heuristic Crossover: This crossover is proposed by Wright [13], In which one offspring is created using two parents using,

$$X'_{ij}(k) = U(0,1)\left(X_{1j}(k) - X_{2j}(K)\right) + X_{2j}(k) \qquad (1)$$

Constraint is that parent $X_2(k)$ is not worse than parent$X_1(k)$.

Arithmetic Crossover: Michalewicz[14] give the arithmetic crossover ,based on multi-parent recombination approach that takes weighted average over two or more parents . One offspring is generated as ,

$$X'_{ij}(k) = \sum_{l=1}^{\mu} \gamma_l X_{lj}(k) \qquad (2)$$

Where $\sum_{l=1}^{\mu} \gamma_l = 1$. A special case of this operator is gotten for $\rho = 2$ in which case,

$$X'_{ij}(k) = (1-\gamma)X_{1j}(k) + \gamma X_{2j}(k) \qquad (3)$$

With $\gamma \epsilon [0,1]$. In the event$\gamma = 0.5$, the impact is that every component of the offspring is basically the average of the corresponding components of the parents

Blend Crossover: Eshelman and Shaffer [15] gives this crossover based on weighted average of (3) known as blend crossover(BLX-α) . One offspring is generated as,

$$X'_{ij}(K) = (1-\gamma_j)X_{1j}(k) + \gamma_j X_{2j}(k) \qquad (4)$$

With $\gamma_j = (1+2\alpha)$, $\alpha \epsilon U(0,1)$ experimentally found that blend crossover work well with $\alpha = 0.5$.

Geometrical Crossover: Michalewicz [16] establish the two parent geometrical crossover to generate single offspring as follows

$$X'_{ij}(k) = (X_{1j}X_{2j})^{0.5} \qquad (5)$$

The geometrical crossover can be generalized to multi-parent crossover as follows

$$X'_{ij}(k) = \left(X_{1j}^{\alpha_1} X_{2j}^{\alpha_2} \dots \dots . X_{\mu j}^{\alpha_\mu}\right) \qquad (6)$$

Where μ is the number of parents and $\sum_{l=1}^{\mu} \alpha_l = 1$.

Simulated Binary Crossover : Deb and Agrawal [18] gives this operator to recreate the conduct of one-point crossover operator for binary representation. Two parents $X_1(k)$ and $X_2(k)$ are used to produce two offspring , where j=1,2,……n,

$$X'_{1j}(k) = 0.5\left[(1+\gamma_j)X_{1j}(k) + (1-\gamma_j)X_{2j}(k)\right] \qquad (7)$$

$$X'_{2j}(k) = 0.5\left[(1-\gamma_j)X_{1j}(k) + (1+\gamma_j)X_{2j}(k)\right] \qquad (8)$$

Where

$$\gamma_j = \begin{cases} (2r_j)^{\frac{1}{\eta+1}} & \text{if } r_j \le 0.5 \\ \left(\frac{1}{2(1-r_j)}\right)^{\frac{1}{\eta+1}} & \text{else} \end{cases} \qquad (9)$$

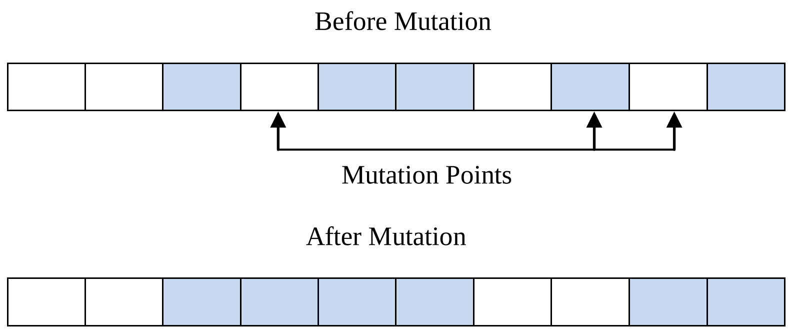
With $r_j \sim U(0,1)$, and $\eta > 0$ is the distribution index .$\eta = 1$ is recommended by Deb and Agrawal.

IV. Mutation

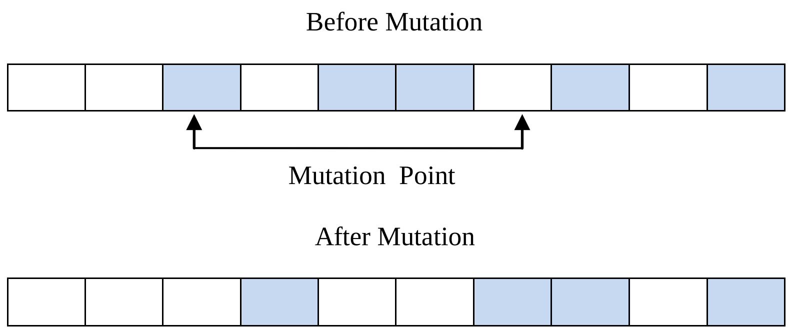
The objective of mutation is to bring new genetic material into an existing individual i.e. to add diversity to the genetic traits of the population. Mutation is utilized as a part of backing of crossover that the full scope of allele is available for each gene. With a certain probability P_m , to every gene of the offspring , $X'_i(k)$ to generate the mutated offspring $X''_i(k)$.The mutation probability, also known as mutation rate , is generally a small estimation, P_m, to establish that good solution are not distorted too much. For mutation probability p_m Individual will be mutated with probability $(X'_i(k) \text{is mutated}) = 1-(1-P_m)^n$. Taking into account the representation plan utilized as a part of the Algorithm mutation operator is classified into two fundamental categories (1)Binary representation mutation (2)Floating point representation mutation.

Binary Representations Mutation: For this representation mostly two mutation operators is used.

Uniform (Random) Mutation: In this bit position are picked up randomly and the corresponding bit values are negated [4] as shown below.



Inorder Mutation: Randomly two mutation points are chosen and only the bits between these mutation points experience random mutation as shown below.



Floating-Point Representation: For this, mutation is performed using normally distributed number [18] as given below.

$$X''_{ij} = X'_{ij} + \rho N_n(0,1) \qquad (10)$$

Where $N_n(0,1)$ is standard normal distribution with mean=0 and variance= 1 and ρ is standard deviation of normal distribution.

V. CONCLUSION

We have presented the working principle of genetic algorithm, which is based on evolutionary computing principle. As the main driving operators of GA are selection, crossover, and mutation besides this GA depends on representation scheme of solutions that specifies the quality of solution for a problem. Mostly the solutions are represented either in binary form or floating point form. Based on representation scheme of the solution, Operators are presented in this paper.

REFERENCES

[1] Eraser, A. (1957). Simulation of genetic systems by automatic digital computers. I. Introduction. *Australian Journal of Biological Sciences*, *10*, 484-491.
[2] Bremermann, H. J. (1962). Optimization through evolution and recombination. *Self-organizing systems*, *93*, 106.
[3] Reed, J., Toombs, R., & Barricelli, N. A. (1967). Simulation of biological evolution and machine learning: I. Selection of self-reproducing numeric patterns by data processing machines, effects of hereditary control, mutation type and crossing. *Journal of theoretical biology*, *17*(3), 319-342.
[4] Holland, J. H. (1975). Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. U Michigan Press.
[5] Goldberg, D. E., & Deb, K. (1991). A comparative analysis of selection schemes used in genetic algorithms. *Foundations of genetic algorithms*, *1*, 69-93.
[6] Jebari, K., & Madiafi, M. (2013). Selection Methods for Genetic Algorithms. *Int. J. Emerg. Sci*, *3*(4), 333-344.
[7] Miller, B. L., & Goldberg, D. E. (1995). Genetic algorithms, tournament selection, and the effects of noise. *Complex Systems*, *9*(3), 193-212.
[8] Alabsi, F., & Naoum, R. (2012). Comparison of Selection Methods and Crossover Operations using Steady State Genetic Based Intrusion Detection System. *Journal of Emerging Trends in Computing and Information Sciences*, *3*(7), 1053-1058.
[9] De Jong, K. A. (1975). Analysis of the behavior of a class of genetic adaptive systems.
[10] Eshelman, L. J., Caruana, R., & Schaffer, J. D. (1989, January). Biases in the crossover landscape. In Proceedings of the 3rd International Conference on Genetic Algorithms, George Mason University, Fairfax, Virginia, USA, June 1989 (pp. 10-19).
[11] Ackley, D. H. (1987). A connectionist machine for genetic hillclimbing.
[12] Syswerda, G. (1989). Uniform crossover in genetic algorithms.

[13] Wright, A. H. (1991). Genetic algorithms for real parameter optimization. *Foundations of genetic algorithms*, *1*, 205-218.
[14] t I, M. Z. (1992). Genetic Algorithms+ data Structures= Evolutionary Programs.
[15] Eshelman, L. J., & Schaffer, J. D. (1992). Real--Coded Genetic Algorithms and Interval-Schemata.
[16] Michalewicz, Z., Nazhiyath, G., & Michalewicz, M. (1996). A Note on Usefulness of Geometrical Crossover for Numerical Optimization Problems. *Evolutionary programming*, *5*(1), 305-312.
[17] Deb, K., & Agrawal, R. B. (1995). Simulated binary crossover for continuous search space. *Complex systems*, *9*(2), 115-148.
[18] Haupt, R. L., & Haupt, S. E. (2004). *Practical genetic algorithms*. John Wiley & Sons.