**Introduction to Evolutionary Algorithms**

An optimization is a mathematical tool that selects the best solution from available alternatives. Several real-world mathematically defined problems use the optimization con cept, such as scheduling, engineering, mathematics, com merce, networks, and economics. Within the last decades, solving optimization problems has caught researchers’ attention. Metaheuristic optimization algorithms (MOA) are commonly utilized to solve those problems. MOA can be classified based on a search strategy (local search and global search), the number of candidate solutions (single solution and population-based), and hybridization (hybrid and memetic). There are several kinds of MOA, such as **Evolutionary Algorithms (EAs).**

EAs are the most well-known global-search, population based, and memetic MOAs. EAs are heuristic methods inspired by mechanisms that rely on biological evolution, such as reproduction, mutation, and natural selection. In EA, the search space X is a set of chromosomes (i.e., DNA strings) considered candidate solutions for a specific problem. Their fitness is evaluated by objective function (f).

Numerous EAs are developed, such as Genetic Algorithms (GAs), Genetic Programming (GP), Evolutionary Programming (EP), Deferential Evolution, and Evolution Strategy. The GAs mimic natural selection (i.e., survival of the fittest) and the biological reproduction processes of the fittest individual. The optimal solution (i.e., the fittest individual) is developed from one generation to the next without depending on strict mathematical formulation. Therefore, the optimal solution consists of the best components (i.e., genes) of the fittest individual in previous generations. The simplest form of GA works on a population consisting of individuals (i.e., fixed bit stings).

GA selects a parent pool from the population based on selection criteria to prepare the next generation. The crossover and mutation operators supply the population with new candidates. The crossover operator produces new children (i.e., offspring) by exchanging partial bit strings and inverting bits between two distinct parents. The mutation operator may flip some genes of the new children. GA evaluates each individual using a fitness function. In the last generation, the fittest individual is considered the optimal solution.

**Terminologies and definitions:**

**The fundamental terminologies of GAs are:**

**• Population:** A population is a set of candidate solutions.

**• Chromosome/individual**: A chromosome/individual is a candidate solution. Each chromosome consists of a set of genes and their alleles. A gene is one element position of a chromosome, which is a single bit or short block of adjacent bits [15]. An allele is the gene’s value of a par ticular chromosome

• **Initialization**: Initialization is the first process in GA responsible for preparing the initial population. The GA f ills the population with random candidate solutions (i.e., individuals).

• **Evaluation**: An evaluation process is responsible for determining the fitness level of an individual. GA utilizes a problem-dependent fitness function. This operation is triggered once a new individual is produced.

• **Selection**: A selection process is essential in GA to select the parents for the crossover operation. The simplest selection technique is based on the fitness value, where the better solutions have the highest probability of being selected than the worse ones.

• **Crossover**: A crossover operation is a recombination process responsible for producing new offspring.

• **Mutation**: A mutation operation is a random deformation of the individual with a specific probability.

• **Replacement**: A replacement operation is responsible for preparing the population for the next generation. The basic technique selects the fittest individuals of the cur rent generation (i.e., parents and new offspring) to pre pare the next generation.

* **Stop criteria**: Stop criteria are specified to determine when to stop the GA and select the optimal solution.

Typically, at least one of the following criteria is specified:

* Reach the maximum number of generations.
* Find an individual in the population with a fitness value lower/higher than a threshold.

**Genetic algorithm basics and operations:**

A GA starts with initializing a population of size N. The fitness value of each individual in the population is evaluated using a f itness function. Then, the process enters a loop for a specified number of generations (maxGan) where GA uses the current generation (currentGen) to generate the next generation. A set of individuals is selected to form the parents pool. The parent pool helps in producing new offspring using crossover and muta tion techniques. The search process is terminated when a stop criterion is satisfied (i.e., reaching the maxGan generation or f inding an individual who satisfied a stop criterion).

**Population diversity**. A population with a low diversity level leads to a GA like a local search algorithm with an additional overhead from maintaining many similar solutions [16]. Premature convergence refers to a popula tion containing similar individuals before exploring the search space. A diverse population helps the GA explore different regions of the search space, thus reducing the probability of being stuck in the local optimum of a bad f itness degree.

**Population size**. The population size is fixed; thus, it significantly impacts GA performance. The probability of covering promising regions of the search space decreases as the search space dimensionality increases. Thus, selecting a small population size reduces population diversity quickly after applying the crossover operation. On the other hand, selecting a large population waste computational resources.

The representation of the individual in the population depends mainly on the problem. An individual represents by using a bit string (i.e., simplest and most popular encoding) or non-binary representation. Generally, the individual con sists of a set of genes, and each gene has an allele.

**Fitness function:** The fitness function is an essential component of any GA used to measure the fitness value of individuals. A fitness function depends on a single objective function or multi objective function. The objective function is a function that measures the performance concerning a set of parameters (i.e., alleles of the individual). In contrast, the fitness func tion measures the reproduction probability of each individ ual depending on the objective function

**Genetic operators**

**Selection operator:** The selection operator prepares the parents’ pool for mat ing. Thus, the selection operator guides the GA to the optimal solution by preferring the fittest individuals over.

The selection techniques are two types: fitness proportionate selection and ordinal selection.

1. Fitness proportionate selection techniques depend on the fitness degree of the individual.

• Roulette-wheel selection: A selection prob ability is computed for each individual.

• Stochastic universal selection:. The stochastic universal selection uses M equally spaced point ers, where M is the number of parents required to be selected. GA orders the population and selects a single random number ( P1 ∈[0, 1/M]). Then, the M parents (i.e., individuals) are chosen starting with P1 and spaced by 1/M.In roulette-wheel selection, the weaker individuals have a lower probabil ity of being selected, while the stochastic universal selection reduces the unfair nature of the roulette wheel selection technique.

• Elitist selection. The fittest M individuals are selected to fill the parents pool.

2. Ordinal Selection.

• Tournament selection: Individuals are chosen randomly from the population. Then, the fittest selected individual wins the tournament. The com mon value of s is 2. Thus, the tournament selection is repeated M times to select M parents.

• Truncation selection: The individuals are ordered according to fitness value. Then, a particular portion (p) of the fittest individuals are selected and reproduced 1/p times.

**Crossover operator**:

A crossover operator is a powerful tool for producing new offspring and improving the quality of individuals by swap ping genes between the parents. Several techniques are utilized to perform the crossover.

• One-point crossover is the simplest crossover tech nique (Fig. 4). One cross point is chosen randomly, and swapped the two tails of the parents to produce two new children.

• Two-point crossover: Two random cross points are selected, and the parents’ genes between the two points are exchanged to produce two new children.

• K-point crossover. One-point and two-point crossover techniques are special cases of K-point crossover. Thus, the concept of one-point and two-point crossover is extended to K-point crossover, where K > 2 cross points are selected randomly.

• Uniform crossover:. Uniform crossover uses a fixed mixing ratio ( Pc ) between two parents. For each gene ( gi ) in the child, a random number ( r ∈[0,1] ) is selected. In case r > Pc , then gene gi inherits the allele of the first parent. Otherwise, gene gi inherits the allele of the second parent.

• Uniform order-based crossover:. The uni form order-based crossover uses a random binary tem plate to produce children. The first child inherits the genes from the first parent according to the ones in the template; otherwise, it inherits the genes from the second parent. The second child inherits the inverse genes of the f irst child from both parents.

**Mutation operator**

Crossover operators negatively impact population diversity since the new children have identical alleles to their parents. The mutation operator maintains the population diversity. The main idea is to change the allele of the child randomly. The mutation operator is controlled by a mutation prob ability ( Pm ) that is kept as low as possible to avoid the GA behaving like a random search. Below, the classical mutation techniques are mentioned:

• Single-point mutation selects one random gene and changes it to a random value with a probability Pm.

• Bitwise inversion mutation inverters the whole bit string bit by bit with probability Pm.

• Random variable mutation selects a random number ( K > 1 ) of genes and changes it to random values with a probability Pm.

• Boundary mutation is utilized with integer and float alleles. A random gene is selected with probability Pm and randomly changed its value to the lower or upper bound.

• Uniform mutation changes the value of the chosen gene (with probability Pm ) to a uniform random value selected between the upper and lower bounds for that gene.

Simple GA

* Initialize Population
* Calculate the fitness
* While Stopping criterion not met

Select parents

Perform crossover

Apply mutation

Calculate fitness

Why use GA? Why not just apply Deep Learning to everything?

* Easy and faster to code
* Gives many solutions – can avoid local extrema
* Can use parallelization

What are the shortcomings?

* They are slow so not suitable for real-time applications
* Fitness function may not be easily designed

Crossover frequency is a factor

1. All the time – All offsprings made by crossover
2. Never – Copy parents

Perhaps its reasonable to copy some chromosomes into the next generation.

Mutation frequency is also a factor.

1. Never – No additional change in offsprings – and it does take longer
2. Too often – huge variability preventing convergence
3. Rarely(around 1 percent?): Additional diversity contributing to good solutions

How to select parents?

1. When fitness values are very different :

Rank Selection, Rank all chromosomes based on fitness values

1. When fitness values are not so different:

Roulette wheel selection

**Multi-Objective Optimization**

Multi-objective optimization involves optimizing a number of objectives simultaneously. The problem becomes challenging when the objectives are of conflict to each other, that is, the optimal solution of an objective function is different from that of the other. In solving such problems, with or without the presence of constraints, these problems give rise to a set of trade-off optimal solutions, popularly known as Pareto-optimal solutions. Due to the multiplicity in solutions, these problems were proposed to be solved suitably using evolutionary algorithms which use a population approach in its search procedure.