LECTURE 8 - Fundamentals of Genetic Algorithms

• Genetic Algorithms (GAs) are adaptive heuristic search algorithm based on the evolutionary ideas of natural selection and genetics.
• Genetic algorithms (GAs) are a part of Evolutionary computing, a rapidly growing area of artificial intelligence. GAs are inspired by Darwin's theory about evolution - "survival of the fittest".
• GAs represent an intelligent exploitation of a random search used to solve optimization problems.
• GAs, although randomized, exploit historical information to direct the search into the region of better performance within the search space.
• In nature, competition among individuals for scanty resources results in the fittest individuals dominating over the weaker ones.

Why Genetic Algorithms?
It is better than conventional AI ; It is more robust.
• Why Genetic Algorithms ?
- Unlike older AI systems, the GA's do not break easily even if the inputs changed slightly, or in the presence of reasonable noise.
- While performing search in large state-space, or multi-modal state-space, or n-dimensional surface, a genetic algorithms offer significant benefits over many other typical search optimization techniques like linear programming, heuristic, depth-first, breath-first.
"Genetic Algorithms are good at taking large, potentially huge search spaces and navigating them, looking for optimal combinations of things, the solutions one might not otherwise find in a lifetime."

Optimization is a process that finds a best, or optimal, solution for a problem. The Optimization problems are centered around three factors :
1. An objective function which is to be minimized or maximized;
Examples
‡ In manufacturing, we want to maximize the profit or minimize the cost.
‡ In designing an automobile panel, we want to maximize the strength.
2. A set of unknowns or variables that affect the objective function,
Examples
‡ In manufacturing, the variables are amount of resources used or the time spent.
‡ In panel design problem, the variables are shape and dimensions of the panel.
3. A set of constraints that allow the unknowns to take on certain values but exclude others;
Examples
‡ In manufacturing, one constrain is, that all "time" variables to be non-negative.

Evolutionary Algorithm (Eas)
Evolutionary Algorithm (EA) is a subset of Evolutionary Computation (EC) which is a subfield of Artificial Intelligence (AI).
Evolutionary Computation(EC) is a general term for several computational techniques. Evolutionary Computation represents powerful search and optimization paradigm influenced by biological mechanisms of evolution : that of natural selection and genetic.
Evolutionary Algorithms (Eas) refers to Evolutionary Computational models using randomness and genetic inspired operations.
Eas involve selection, recombination, random variation and competition of the individuals in a population of adequately represented potential solutions.

The candidate solutions are referred as chromosomes or individuals.
Genetic Algorithms (GAs) represent the main paradigm of Evolutionary Computation.
- **GAs** simulate natural evolution, mimicking processes the nature uses : *Selection, Crosses over, Mutation and Accepting.*
Gas simulate the survival of the fittest among individuals over consecutive generation for solving a problem.

Evolutionary Computing (EC) = Genetic Programming (GP) (1992)+ Evolution strategies (ES)(1965) + Evolutionary programming (EP) (1962)+ Genetic Algorithms (GA)(1970)

Genetic algorithms computing. (Gas) are the main paradigm of evolutionary
GAs are inspired by Darwin's theory about evolution – the "survival of the fittest". In nature, competition among individuals for scanty resources results in the fittest individuals dominating over the weaker ones.
− **Gas** are the ways of solving problems by mimicking processes nature uses; ie., Selection, Crosses over, Mutation and Accepting, to evolve a solution to a problem.
− Gas are adaptive heuristic search based on the evolutionary ideas of natural selection and genetics.
− GAs are intelligent exploitation of random search used in optimization problems.
− GAs, although randomized, exploit historical information to direct the search into the region of better performance within the search space.

Biological Background – Basic Genetics
Every organism has a set of rules, describing how that organism is built. All living organisms consist of cells.
‡ In each cell there is same set of chromosomes. Chromosomes are strings of DNA and serve as a model for the whole organism.
‡ A chromosome consists of genes, blocks of DNA.
‡ Each gene encodes a particular protein that represents a trait (feature), e.g., color of eyes.
‡ Possible settings for a trait (e.g. blue, brown) are called alleles.
‡ Each gene has its own position in the chromosome called its locus.
‡ Complete set of genetic material (all chromosomes) is called a genome.
‡ Particular set of genes in a genome is called genotype.
‡ The physical expression of the genotype (the organism itself after birth) is called the phenotype, its physical and mental characteristics, such as eye color, intelligence etc.
‡ When two organisms mate they share their genes; the resultant offspring may end up having half the genes from one parent and half from the other. This process is called recombination (cross over) .
‡ The new created offspring can then be mutated. Mutation means, that the elements of DNA are a bit changed. This changes are mainly caused by errors in copying genes from parents.
‡ The fitness of an organism is measured by success of the organism in its life (survival).

Pseudo-Code
BEGIN
INITIALISE population with random candidate solution.
EVALUATE each candidate;
REPEAT UNTIL (termination condition) is satisfied DO
1. SELECT parents;
2. RECOMBINE pairs of parents;
3. MUTATE the resulting offspring;

4. SELECT individuals or the next generation;
END.

Working Principles

− Chromosome : a set of genes; a chromosome contains the solution in form of genes.
− Gene : a part of chromosome; a gene contains a part of solution. It determines the solution. e.g. 16743 is a chromosome and 1, 6, 7, 4 and 3 are its genes.
− Individual : same as chromosome.
− Population: of number individuals present with same length of chromosome.
− Fitness : the value assigned to an individual based on how far or close a individual is from the solution; greater the fitness value better the solution it contains.
− Fitness function : a function that assigns fitness value to the individual. It is problem specific.
− Breeding : taking two fit individuals and then intermingling there chromosome to create new two individuals.
− Mutation : changing a random gene in an individual.
− Selection : selecting individuals for creating the next generation.

Procedure

Genetic algorithm begins with a set of solutions (represented by chromosomes) called the population.
− Solutions from one population are taken and used to form a new population. This is motivated by the possibility that the new population will be better than the old one.
− Solutions are selected according to their fitness to form new solutions (offspring); more suitable they are, more chances they have to reproduce.
− This is repeated until some condition (e.g. number of populations or improvement of the best solution) is satisfied.

Encoding

Before a genetic algorithm can be put to work on any problem, a method is needed to encode potential solutions to that problem in a form so that a computer can process.

Binary Encoding

Binary encoding is the most common to represent information contained. In genetic algorithms, it was first used because of its relative simplicity.
− In binary encoding, every chromosome is a string of bits : 0 or 1, like
Chromosome 1: 101100101100101011100101
Chromosome 2: 111111100000110000011111
− Binary encoding gives many possible chromosomes even with a small number of alleles ie possible settings for a trait (features).
− This encoding is often not natural for many problems and sometimes corrections must be made after crossover and/or mutation.

Value Encoding

The Value encoding can be used in problems where values such as real numbers are used. Use of binary encoding for this type of problems would be difficult.
1. In value encoding, every chromosome is a sequence of some values.
2. The Values can be anything connected to the problem, such as : real numbers, characters or objects.

Examples :

Chromosome A 1.2324 5.3243 0.4556 2.3293 2.4545
Chromosome B ABDJEIFJDHDIERJFDLDFLFEGT
Chromosome C (back), (back), (right), (forward), (left)
3. Value encoding is often necessary to develop some new types of crossovers and mutations specific for the problem.

Permutation Encoding

Permutation encoding can be used in ordering problems, such as traveling salesman problem or task ordering problem.
1. In permutation encoding, every chromosome is a string of numbers that represent a position in a sequence.
Chromosome A 1 5 3 2 6 4 7 9 8
Chromosome B 8 5 6 7 2 3 1 4 9
2. Permutation encoding is useful for ordering problems. For some problems, crossover and mutation corrections must be made to leave the chromosome consistent.

Tree Encoding

Tree encoding is used mainly for evolving programs or expressions.
For genetic programming :
− In tree encoding, every chromosome is a tree of some objects, such as functions or commands in programming language.
− Tree encoding is useful for evolving programs or any other structures that can be encoded in trees.
− The crossover and mutation can be done relatively easy way .

Operators of Genetic Algorithm

Genetic operators used in genetic algorithms maintain genetic diversity.
Genetic diversity or variation is a necessity for the process of evolution.
Genetic operators are analogous to those which occur in the natural world:
− Reproduction (or Selection) ;
− Crossover (or Recombination); and
− Mutation.
In addition to these operators, there are some parameters of GA.
One important parameter is Population size.
− Population size says how many chromosomes are in population (in one generation).
− If there are only few chromosomes, then
GA would have a few possibilities to perform crossover and only a small part of search space is explored.
− If there are many chromosomes, then GA slows down.
− Research shows that after some limit, it is not useful to increase population size, because it does not help in solving the problem faster. The population size depends on the type of encoding and the problem.

Reproduction, or Selection

Reproduction is usually the first operator applied on population. From the population, the chromosomes are selected to be parents to crossover and produce offspring.

Crossover

Crossover is a genetic operator that combines (mates) two chromosomes (parents) to produce a new

chromosome (offspring). The idea behind crossover is that the new chromosome may be better than both of the parents if it takes the best characteristics from each of the parents.
Crossover occurs during evolution according to a user-definable crossover probability. Crossover selects genes from parent chromosomes and creates a new offspring.
The Crossover operators are of many types.
− one simple way is, One-Point crossover.
− the others are Two Point, Uniform, Arithmetic, and Heuristic crossovers.
The operators are selected based on the way chromosomes are encoded.

Arithmetic
Arithmetic crossover operator linearly combines two parent chromosome vectors to produce two new offspring according to the equations:
Offspring1 = a * Parent1 + (1- a) * Parent2
Offspring2 = (1 – a) * Parent1 + a * Parent2
where a is a random weighting factor chosen before each crossover operation.

Heuristic
Heuristic crossover operator uses the fitness values of the two parent chromosomes to determine the direction of the search.
The offspring are created according to the equations:
Offspring1 = BestParent + r * (BestParent − WorstParent)
Offspring2 = BestParent
where r is a random number between 0 and 1.

Mutation
After a crossover is performed, mutation takes place. Mutation is a genetic operator used to maintain genetic diversity from one generation of a population of chromosomes to the next.
Mutation occurs during evolution according to a user-definable mutation probability, usually set to fairly low value, say 0.01 a good first choice.
Mutation alters one or more gene values in a chromosome from its initial state. This can result in entirely new gene values being added to the gene pool. With the new gene values, the genetic algorithm may be able to arrive at better solution than was previously possible.
Mutation is an important part of the genetic search, helps to prevent the population from stagnating at any local optima. Mutation is intended to prevent the search falling into a local optimum of the state space.