4. Genetic Algorithms

Idea of evolutionary computing was introduced in the year 1960s by I. Rechenberg in his work "Evolution strategies". His idea was then developed by other researchers.

Genetic Algorithms (GAs) were invented by John Holland in early 1970's to mimic some of the processes observed in natural evolution.

Later in 1992 John Koza used GAs to evolve programs to perform certain tasks. He called his method "Genetic Programming" (GP).

GAs simulate natural evolution, a combination of selection, recombination and mutation to evolve a solution to a problem.

GAs simulate the survival of the fittest, among individuals over consecutive generation for solving a problem. Each generation consists of a population of character strings that are analogous to the chromosome in our DNA (Deoxyribonucleic acid). DNA contains the genetic instructions used in the development and functioning of all known living organisms.

- **What are 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**

  "Genetic Algorithms are good at taking large, potentially huge search spaces and navigating them, looking for optimal combinations of things, solutions you might not otherwise find in a lifetime." - Salvatore Mangano Computer Design, May 1995.

  - GA is better than conventional AI, in that it is more robust.
  - Unlike older AI systems, GAs do not break easily even if the inputs changed slightly, or in the presence of reasonable noise.
  - In searching a large state-space, multi-modal state-space, or n-dimensional surface, a GA may offer significant benefits over more typical search of optimization techniques, like - linear programming, heuristic, depth-first, breath-first.

- **Mechanics of Biological Evolution**

Genetic Algorithms are a way of solving problems by mimicking processes the nature uses - Selection, Crosses over, Mutation and Accepting to evolve a solution to a problem.

- Every **organism** has a set of **rules**, describing how that organism is built, and encoded in the **genes** of an organism.
- The genes are connected together into long strings called **chromosomes**.
- Each gene represents a specific **trait** (feature) of the organism and has several different settings, e.g. setting for a hair color gene may be black or brown.
- The genes and their settings are referred as an organism's **genotype.**
- 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 parent. This process is called **crossover** (recombination).
- The newly created offspring can then be mutated. A gene may be **mutated** and expressed in the organism as a completely new trait. Mutation means, that the elements of DNA are a bit changed. This change is mainly caused by errors in copying genes from parents.
- The **fitness** of an organism is measured by success of the organism in its life.

4.1 Artificial Evolution and Search Optimization

The problem of finding solutions to problems is itself a problem with no general solution. Solving problems usually mean looking for solutions, which will be the best among others.

- In engineering and mathematics finding the solution to a problem is often thought as a process of optimization.
- Here the process is : first formulate the problems as mathematical models expressed in terms of functions; then to find a solution, discover the parameters that optimize the model or the function components that provide optimal system performance.

The well-established search / optimization techniques are usually classified in to three broad categories : Enumerative, Calculus-based, and Guided random search techniques. A taxonomy of Evolution & Search Optimization classes is illustrated in the next slide.

- **Taxonomy of Evolution & Search Optimization Classes**

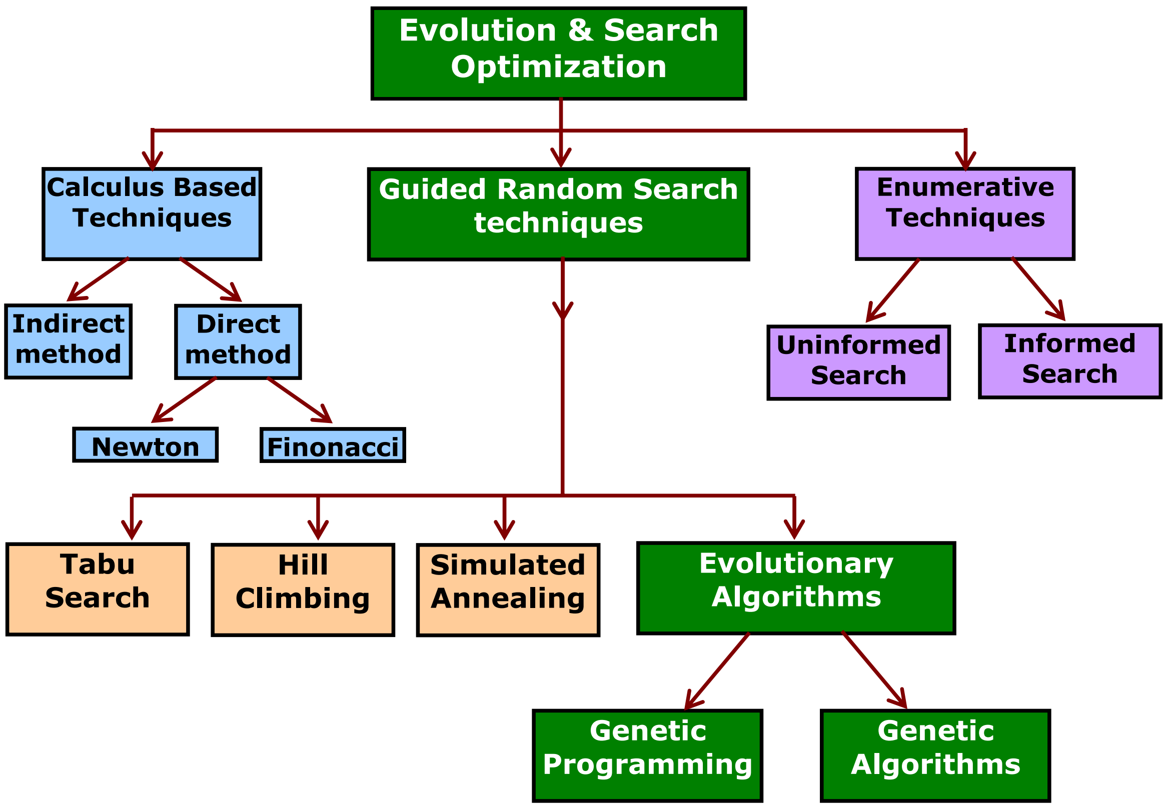

Fig Evolution & Search Optimization Techniques

Each of these techniques are briefly explained in the next three slides.

- **Enumerative Methods**

  These are the traditional search and control strategies. They search for a solution in a problem space within the domain of artificial intelligence. There are many control structures for search. The depth-first search and breadth-first search are the two most common search strategies. Here the search goes through every point related to the function's domain space (finite or discretized), one point at a time. They are very simple to implement but usually require significant computation. These techniques are not suitable for applications with large domain spaces.

  In the field of AI, enumerative methods are subdivide into two categories : uninformed and informed methods.

  - **Uninformed or blind methods :** Such as mini-max algorithm searches all points in the space in a predefined order; this is used in game playing;

  - **Informed methods :** Such as Alpha-Beta and A*, does more sophisticated search using domain specific knowledge in the form of a cost function or heuristic in order to reduce the cost of the search.

- **Calculus based techniques**

  Here a set of necessary and sufficient conditions to be satisfied by the solutions of an optimization problem. They subdivide into direct and indirect methods.

  - **Direct or Numerical methods**, such as Newton or Fibonacci, seek extremes by "hopping" around the search space and assessing the gradient of the new point, which guides the search. This is simply the notion of "hill climbing", which finds the best local point by climbing the steepest permissible gradient. These techniques can be used only on a restricted set of "well behaved" functions.

  - **Indirect methods** search for local extremes by solving the usually non-linear set of equations resulting from setting the gradient of the objective function to zero. The search for possible solutions (function peaks) starts by restricting itself to points with zero slope in all directions.

- **Guided Random Search techniques**

  These are based on enumerative techniques but they use additional information to guide the search. Two major subclasses are simulated annealing and evolutionary algorithms. Both are evolutionary processes.

  - **Simulated annealing** uses a thermodynamic evolution process to search minimum energy states.

  - **Evolutionary algorithms (EAs)** use natural selection principles. This form of search evolves throughout generations, improving the features of potential solutions by means of biological inspired operations. Genetic Algorithms (GAs) are a good example of this technique.

Our main concern is, how does an Evolutionary algorithm :

- implement and carry out search,
- describes the process of search,
- what are the elements required to carry out search, and
- what are the different search strategies.