Evolutionary Computation

Lecture Notes_3

Evolutionary programming, Genetic programming

Evolutionary programming is one of the four major evolutionary algorithm paradigms. It is similar to genetic programming, but the structure of the program to be optimized is fixed, while its numerical parameters are allowed to evolve.

It was first used by Lawrence J. Fogel in the US in 1960 in order to use simulated evolution as a learning process aiming to generate artificial intelligence. Fogel used finite-state machines as predictors and evolved them. Currently evolutionary programming is a wide evolutionary computing dialect with no fixed structure or (representation), in contrast with some of the other dialects. It has become harder to distinguish from evolutionary strategies.

Its main variation operator is mutation; members of the population are viewed as part of a specific species rather than members of the same species therefore each parent generates an offspring, using a (μ + μ)[further explanation needed] survivor selection.

Application areas of Evolutionary programming

- traffic routing and planning
- pharmaceutical design
- epidemiology
- cancer detection
- military planning
- control systems
- system identification
- signal processing
- learning in games

Main components of evolutionary programming

Evolutionary Programming

- Evolutionary Computation.
- Particle Swarm Optimization.
- Genetic Programming.
- Heuristics.
- Evolution Strategy.

- Genetic Algorithm.
- Evolutionary Algorithm.

Genetic programming

Genetic programming (GP) is a technique of evolving programs, starting from a population of unfit (usually random) programs, fit for a particular task by applying operations analogous to natural genetic processes to the population of programs.

The operations are: selection of the fittest programs for reproduction (crossover) and mutation according to a predefined fitness measure, usually proficiency at the desired task. The crossover operation involves swapping random parts of selected pairs (parents) to produce new and different offspring that become part of the new generation of programs. Mutation involves substitution of some random part of a program with some other random part of a program. Some programs not selected for reproduction are copied from the current generation to the new generation. Then the selection and other operations are recursively applied to the new generation of programs.

Typically, members of each new generation are on average more fit than the members of the previous generation, and the best-of-generation program is often better than the best-of-generation programs from previous generations. Termination of the evolution usually occurs when some individual program reaches a predefined proficiency or fitness level.

It may and often does happen that a particular run of the algorithm results in premature convergence to some local maximum which is not a globally optimal or even good solution. Multiple runs (dozens to hundreds) are usually necessary to produce a very good result. It may also be necessary to have a large starting population size and variability of the individuals to avoid pathologies.

Swarm intelligence

Swarm intelligence (SI) is the collective behavior of decentralized, self-organized systems, natural or artificial. The concept is employed in work on artificial intelligence. The expression was introduced by Gerardo Beni and Jing Wang in 1989, in the context of cellular robotic systems.

Swarm intelligence (SI) is one of the computational intelligence techniques which are used to solve complex problem. SI involves collective study of the individual's behavior of population interact with one another locally. Especially for biological systems nature often act as an inspiration. Simple rules are followed by agents and no centralized control structure exists in order to predict the behavior of individual agents. The random iteration of certain degree between the agents provides an "intelligent" behavior which is then unknown to individual agents.

SI systems consist typically of a population of simple agents or boids interacting locally with one another and with their environment. The inspiration often comes from nature, especially biological systems. The agents follow very simple rules, and although there is no centralized control structure dictating how individual agents should behave, local, and to a certain degree random, interactions between such agents lead to the emergence of "intelligent" global behavior, unknown to the individual agents. Examples of swarm intelligence in natural systems include ant colonies, bee colonies, bird flocking, hawks hunting, animal herding, bacterial growth, fish schooling and microbial intelligence.

The application of swarm principles to robots is called swarm robotics while swarm intelligence refers to the more general set of algorithms. Swarm prediction has been used in the context of forecasting problems. Similar approaches to those proposed for swarm robotics are considered for genetically modified organisms in synthetic collective intelligence.

Besides the applications to conventional optimization problems, SI can be employed in library materials acquisition, communications, medical dataset classification, dynamic control, heating system planning, moving objects tracking, and prediction.

Some of popular SI algorithms included Particle Swarm Optimization (PSO), Artificial Bee Colony (ABC) and Ant Colony Optimization (ACO)

Advantages:

1) Flexible: The colony respond to internal disturbances and external challenges.

2) Robust: Tasks are completed even if some agents fail.

3) Scalable: From a few agents to millions

4) Decentralized: There is no central control in the colony.

5) Self-organized: The solutions are emergent rather than pre-defined.

6) Adaptation: The swarm system can not only adjust to predetermined stimuli but also to new stimuli.

7) Speed: Changes in the network can be propagated very fast.

8) Modularity: Agents act independently of other network layers.

9) Parallelism: Agents' operations are inherently parallel.