VAASAN AMMATTIKORKEAKOULU
VASA YRKESHÖGSKOLA
VAASA POLYTECHNIC

Parameter Control in Genetic Algorithms

Ghodrat Moghadampour (mg@puv.fi)
Principal Lecturer
Vaasa Polytechnic
Vaasa
Finland

Outline

Solutions for Optimization Problems

- Optimization problems can be solved using three different approaches: **exact algorithms**, **approximation algorithms** and **heuristic algorithms.**
- **Exact algorithms** solve the given optimization problem to optimality; they guarantee to find the globally optimal solution for every instance of the problem.
- Exact algorithms can be implemented in different ways, of which the most trivial one is **exhaustive search**.

Solutions for Optimization Problems

- An algorithms is called ε-**approximation** algorithm, where , if and only if, for all instances of the given optimization problem, for the final solution *x* returned by the algorithm we have:

$$\left| \frac{f(x) - f(x_{Opt})}{f(x_{Opt})} \right| \leq \varepsilon$$

Heuristic Algorithms

- A **heuristic** algorithm provides near optimal results for hard optimization problems in a reasonable time, without any guarantee on the feasibility of the problem and the quality of the obtained solution.
- Heuristics are either **metaheuristics**, modifiable to a broad variety of problems, or **problem-specific**, developed for a specific problem area.
- A heuristic algorithm can be **constructive** or **improvement** method.
- A constructive heuristic method starts usually from an “empty solution” and fixes its components successively in either a **deterministic** or **probabilistic** way.

Heuristic Algorithms

- While the solution found by a **deterministic heuristic** algorithm remains the same from run to run, repeated runs of a **probabilistic deterministic** algorithm can yield different final solutions.
- **Improvement heuristics** start from a feasible solution and iteratively improve that by local reorganizations in its structure.

What Are Genetic Algorithms?

- Genetic Algorithms are a class of algorithms that imitate natural selection and genetics.
- They were derived from processes of molecular biology and the evolution of life.
- They are robust search algorithms that do not require knowledge of the objective function to be optimized and can search through large spaces quickly.

What Are Genetic Algorithms?

- Instead of DNA (DeoxyriboNucleic Acid) or RNA (Ribonucleic Acid) strands, GAs usually process strings of symbols of finite length; these symbols encode the parameters to be optimized.
- Genetic Algorithms maintain always a set of candidate solutions called population.
- Members of the population are chromosomes that are represented originally by bit strings.

What Are Genetic Algorithms?

- There are a few synonyms for members of a population; *individuals*, *genotypes*, *structures*, *strings* or *chromosomes*.
- It is assumed that organisms in GAs are *haploid*, i.e. one-chromosome individuals.
- Units which chromosomes are made of are called *genes*, *features*, *characters*, or *decoders*.

What Are Genetic Algorithms?

- These are arranged in linear succession; every gene controls the inheritance of one or several characters.
- Each place, i.e. *loci*, of the chromosome is reserved for gene of certain character.
- Each gene may be in several states, i.e. *alleles*, also called feature values.

How Do Genetic Algorithms Work?

- The first population is initialized at random and evolves in generations.
- In each generation the population is affected by genetic operators and selection mechanism.
- GA operators – crossover, mutation, and reproduction – are isomorphic with the synonymous biological processes.

Mutation Operator

Mutation Point

↓

1 0 0 0 0 1 0 0

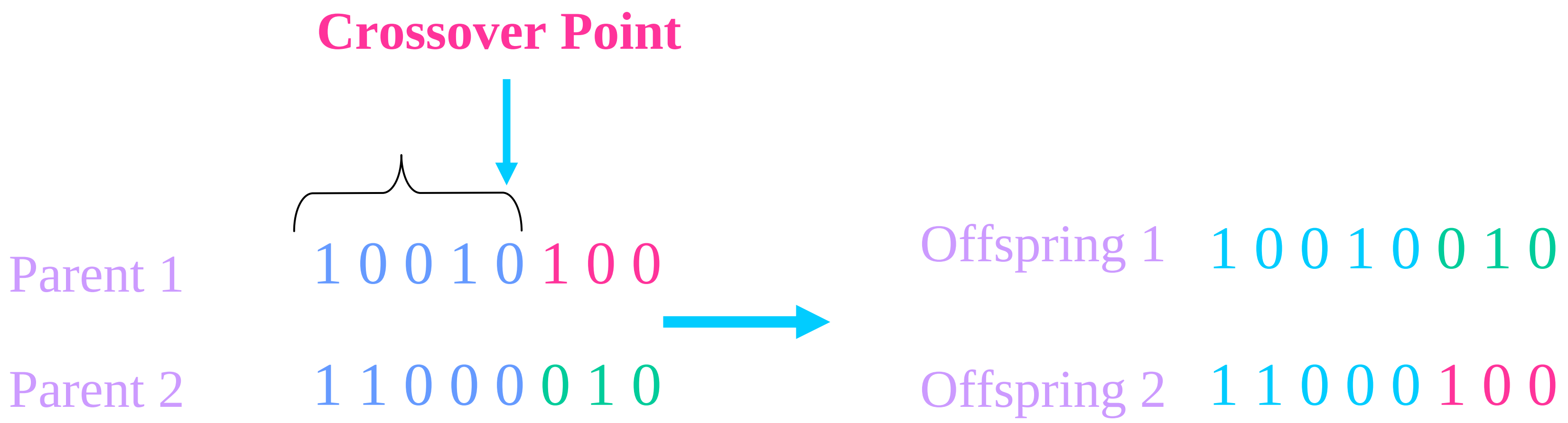
↓

Resuts in:

↓

1 0 1 0 0 1 0 0

Crossover Operator



How Do Genetic Algorithms Work?

- Genetic operators such as crossover and mutation provide information flow among chromosomes while selection promotes survival of the fittest chromosomes.
- A fitness function evaluates individual chromosomes.
- The selection mechanism is usually a combination of the fitness function with some probability.

Fitness-Proportionate Selection

- In fitness-proportionate selection the number of times an individual is selected to reproduce, “expected value”, is that individual’s fitness divided by the average fitness of the population.
- The most common sampling method for implementing fitness-proportionate selection is “roulette wheel” : each individual is assigned a slice of circular “roulette wheel”, the size of the slice being proportional to the individual’s fitness.
- The wheel is spun as many times as the number of individuals, *N*.
- On each spin, the individual under the wheel’s marker is selected to be as a member of parents for the next generation.

Fitness-Proportionate Selection

- This method can be implemented as follows:
  1. Sum the total expected value of individuals in the population. Let this sum be *T*.
  2. Repeat *N* times:
     a. Choose a random integer *r* between 0 and *T*.
     b. Loop through the individuals in the population, summing the expected values, until the sum is greater than or equal to *r*. The individual whose expected value puts the sum over this limit is one selected.

How Do Genetic Algorithms Work?

- The simplest form of GA involves three types of operators: **selection**, **single point crossover**, and **mutation**.
- A simple GA works as follows:

1. Start with a randomly generated population of *n* *l*-bit strings (chromosomes).
2. Calculate the fitness *f(x)* of each bit string *x* in the population.
3. Repeat the following steps for *n* times:

How Do Genetic Algorithms Work?

a) Select a pair of parent bit strings from the current population, the probability of selection being proportional to the fitness value.

b) With probability p_c crossover the pair at a randomly chosen point to form two offspring. In multi-point crossover the crossover probability for the number of points at which a crossover takes place.

c) Mutate the two offspring at each locus with probability p_m and place the resulting bit strings in the new population.

How Do Genetic Algorithms Work?

4. Replace the current population with the new population.
5. Go to step 2.

When to Use Genetic Algorithms?

- There is no rigorous theory on when a GA outperforms other search methods.
- Local search for numerical applications seems to be difficult for GAs.
- GAs are suggested to be used as a preprocessor to perform the initial search, before turning the search process over to a system which employs domain knowledge to guide the local search.
- Like natural genetic systems, individual structures are not the focus of attention for GAs.
- After identifying high performance regions of the search by a GA, it may be useful to invoke a local search routine to optimize the members of the final population.

When to Use Genetic Algorithms?

- Local search requires the utilization of schemata of higher order and longer defining length than those suggested by the schema theorem.
- For problems whose domains of parameters are unlimited, the number of parameters is quite large, and high precision is required, binary solution vectors should be long. However, for such problems the performance of GAs is quite poor.

When to Use Genetic Algorithms?

- However, many researchers intuitively share the idea that in the following situations a GA will have a good chance of being competitive with or even surpassing methods that do not use domain-specific knowledge in their search procedure:
  - The search space is large,
  - The search space is known not to be perfectly smooth and unimodal (A finite sequence which first increases and then decreases).
  - The search space is not well understood.
  - The fitness function is noisy.
  - If the task does not require a global optimum to be found.

When to Use Genetic Algorithms?

- Otherwise, a GA might converge on a local optimum in a rather small search space.
- GA's performance depend very much on details such as the method for encoding candidate solutions, the operators, the parameter strings, and the particular criterion for success.

GA Parameters

- Controlling values of various parameters of an evolutionary algorithm is one of the most challenging area of research fields.
- It has a potential of adjusting the algorithm to the problem while solving the problem.
- Applying any heuristic search algorithm to a particular problem requires a specific representation and evaluation (fitness) function suitable for the problem.
- On the other hand, defining an evolutionary algorithm requires 1) selection of its components, such as variation operators (mutation and recombination) that suit the representation, 2) selection mechanisms for selecting parents and offspring and 3) an initial population.

VAASAN AMMATTIKORKEAKOULU
VASA YRKESHÖGSKOLA
VAASA POLYTECHNIC

GA Parameters

- Each of these components may have parameters whose values greatly determine the efficiency of the algorithm and whether the algorithm will find a near-optimum solution.
- Some of such parameters are the probability of mutation, the tournament size of selection, and the population size.
- Finding the right parameter values is a time-consuming task and it has been the subject of many researches.
- Different techniques can be used to set parameters in an evolutionary algorithm.
- These techniques can be classified based on different criteria and various terminologies are used to describe these classifications.

GA Parameters

- The main criteria for classifying parameter setting methods are:
  - 1) what is changed
  - 2) how the change is made
- The first criterion refers to the components of the evolutionary algorithm and consists of six categories:
  - 1) representation
  - 2) evaluation function
  - 3) variation operators (mutation and recombination)
  - 4) selection
  - 5) replacement
  - 6) population

GA Parameters

- The second criterion refers to the parameter setting methods, which can be divided to three main types:
  - 1) deterministic (or fixed) parameter control (also called parameter tuning) in which the parameter-altering transformations takes no input variables related to the progress of search method,
  - 2) adaptive (also called explicitly adaptive) parameter control in which there is some form of feedback from the search,
  - 3) self-adaptive (also called implicitly adaptive) parameter control in which the parameters to be adapted are encoded into the chromosomes and undergo mutation and recombination.

Parameter Tuning

- Parameter tuning refers to the approach that amounts to finding good fixed values for the parameters before the run of the algorithm.
- In this approach an attempt is made to find the optimal and general set of parameters applicable to a wide range of optimization problems.
- Two main approaches were tried to do this:
  - **1. Pre-defined parameter values**
    - A considerable effort can be put into finding parameter values for GA.
    - Parameter values for a traditional GA (bit-representation, one-point crossover, bit-flip mutation and roulette wheel selection), for a set of test functions: population size: 50, probability of crossover: 0.6, probability of mutation: 0.001, generation gap: 100%, scaling window: $n=\infty$ and selection strategy: elitist.

Parameter Tuning

1. **2. Parameter definition by meta-algorithm**
   - A GA can be used as a meta-algorithm to optimize values for the same parameters for both on-line and off-line performance of the algorithm.
   - On-line performance refers to monitoring the best solution in each generation, while off-line performance takes all solutions in the population into account.
   - The best set of parameters to optimize the on-line performance of the GA was found to be: population size: 30, probability of crossover: 0.95, probability of mutation: 0.01, generation gap: 100%, scaling window: n=1 and selection strategy: elitist.
   - Respectively the best set of parameters to optimize the off-line performance of the GA was found to be: population size: 80, probability of crossover: 0.45, probability of mutation: 0.01, generation gap: 90%, scaling window: n=1 and selection strategy: non-elitist.

Parameter Control

- However, any static set of parameters seems to be inappropriate and against the contemporary view of an EA, which acknowledges that specific problem or problem types require specific EA setup for satisfactory performance.
- Parameter control refers to the approach, which amounts to starting a run with initial parameter values, which are changed during the run.
- An evolutionary algorithm is normally used to not only to solve the problem but also to adapt the same algorithm to the particular problem.
- Tuning parameters during the run can be implemented mainly in two different ways:
  - 1) using a heuristic feedback mechanism allowing one to base changes on triggers different from elapsing time, such as population diversity measures, relative improvement, absolute solution quality, etc.
  - 2) incorporating parameters into the chromosomes, which leaves changes entirely based on the evolution mechanism.

Parameter Control

- Parameter control methods can be further categorized based on *how* the mechanism of change works and *what* component of the EA is affected by the mechanism.
- Typically the following components of an EA are changed:
  - 1) individual representation
  - 2) evaluation function
  - 3) variation operators and their probabilities
  - 4) selection operator like parent selection or mating selection
  - 5) replacement operator like survival selection or environmental selection
  - 6) characteristics of the population like size, topology, and so on

Parameter Control

- Still each component can be parameterized and the number of parameters is arbitrary.
- Two major classification criteria are:
  - Evolutionary algorithms can be considered as a whole without dividing attention to its different components, e.g. mutation, recombination, selection etc., levels of adaptation and type of update rules.
  - Three levels of adaptation: population, individual and component, together with two types of update mechanisms: absolute and empirical rules can be considered.
  - **Absolute rules** determine how modifications should be made beforehand. However, **empirical update rules** modify parameter values by self-adaptation, namely competition among them.

Parameter Control

- The above classification can be extended by considering the environment level of adaptation categories without paying any attention to what parts of the EA are adapted. Hence, dividing types of update mechanisms into deterministic, adaptive and self-adaptive parameter control:
  - A) **Deterministic (**fixed): the value of a strategy parameter is altered by some deterministic rule, such as a time-varying schedule, without using any feedback from the search.
  - B) **Adaptive** (also. explicitly adaptive): makes use of some form of feedback from the search in order to determine the direction and/or magnitude of the change to the strategy parameter.
  - C) **Self-adaptive** (also implicitly adaptive): parameters to be adapted are encoded into the chromosomes and undergo mutation and recombination. The better values of encoded parameters lead to better individuals, which, in turn, are more likely to survive and produce offspring and hence propagate these better parameter values.

Summary to GA Parameter Update

- GA Parameters can be modified using the following approaches:
  - Parameter Tuning (Fixed parameter values for a specific problem type; ”obsolete”)
  - Parameter Control
    - Adaptive parameter control (feedback from the run)
    - Seld-adaptive parameter control (GA updates the parameter values)

VAASAN AMMATTIKORKEAKOULU
VASA YRKESHÖGSKOLA
VAASA POLYTECHNIC

References

- GA Archives

VAASAN AMMATTIKORKEAKOULU
VASA YRKESHÖGSKOLA
VAASA POLYTECHNIC

Demo

- Demo

VAASAN AMMATTIKORKEAKOULU
VASA YRKESHÖGSKOLA
VAASA POLYTECHNIC

Thank You ☺