Topological Genetic Algorithms
Initial Idea and Conceptual Design

Vladimir Filipović

University of Belgrade, Faculty of Mathematics

DND 2024

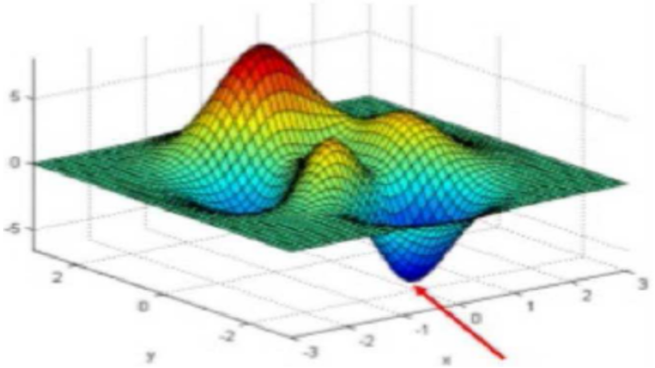
19th October 2024

Contents

Optimization

- Optimization is a field at the intersection of Mathematics, Computer Science and Operations Research
- Optimization is frequently used within designing and modeling complex systems
- Optimization studies methods to find the best solution(s) according to a criterion

Optimization problems



- Following elements are known:
  - search space S
  - solution space X, $X \subset S$
  - objective function f, $f : S \to R$
- In minimum optimization problem, the goal is to calculate $x^* \in X$, such that $f(x^*) = min\{f(x) | x \in X\}$

Metaheuristics

- Metaheuristics are **generalized** computational intelligence methods that can be successfully adopted to **various** problem domains
- They are trying to obtain the **optimal** solution, or the solution that is **close** to optimal one
- Metaheuristics are characterized with **approximation** and **non-determinism**
- Basic metaheuristics concepts are **abstractly represented** - they should be **adapted** to problem domain, otherwise they should won't obtain enough good solution
- Metaheuristics can be **population-based** or **single-solution**

Genetic algorithms

- Genetic algorithms (GA) are **population metaheuristics** that imitate some spontaneous optimization processes in the **natural selection** and **reproduction**
- At each iteration (generation) GA manipulates a **population** of **encoded solutions** (individuals), starting from either randomly or heuristically generated one
- Individuals from the current population are evaluated using a **fitness function** to determine their qualities
- Good individuals are selected to produce the new ones (offspring), applying operators inspired from genetics (**crossover** and **mutation**), and they replace some of the individuals from the current population

GA Pseudo-code

Algorithm 1: Genetic Algorithm

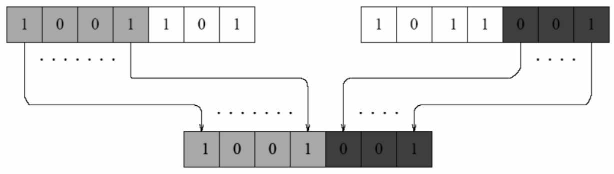
Data: population size n

Result: solution of the optimization problem

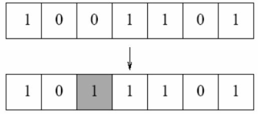
```
Pop ← initialize_population() ;
for i ← 1 to n do
    evaluate_fitness(P_i);
while stopping_condition_not_met() do
    parents ← select_for_mating(P) ;
    offspring ← crossover(parents) ;
    offspring ← mutation(offspring) ;
    evaluate_fitness(offspring);
    N ← selection(P, offspring);
    P ← replacement(P, N);
return best_fitted_individual(P) ;
```

Representation and GA operators

- GA usually works with the **binary encoding** of the individuals, e.g. binary string is the representation of the individual
- Crossover produce offspring from parents usually exchange some parts of the genetic code of the parents



- Mutation usually randomly flips small part of the indidvidual's representation



Exploitation and exploration in GA

- GA, like other metaheuristics, have inherent conflicts between **exploitation** (intensification) and **exploration** (diversification)
  - Exploitation: during the solution searching, algorithm uses information obtained in the past (about previously visited points in search space) to determine smaller regions promising for further search
  - Exploration: procedure that obtains new information – it visits new regions in the search space in order to find promising points or sub-regions
- Balance between exploitation and exploration depends of the characteristics of the optimization problem that is solved
- Setting adequate balance for specific problem is critical for the success of the optimization process

Topological Metaheuristics

- The main motivation for integrating topology and metaheuristics comes from the notion that metaheuristics might use the **topological regularities** inside the solution space to better maneuver through it
- This can become especially useful when the solution space becomes extremely large
  - In such situation, classical metaheuristics might use too much resources in order to search the solution space
  - Although this sounds like it could lead to premature convergence to local optima, we stress that our conceptual design essentially generalizes and encompasses the classical metaheuristic algorithms
- Proposed metaheuristics, during its execution, will **gradually converge** to its classical variants

Fitness Landscape

- Fitness landscape analysis, which includes analysis of local optima positions, is very important for design of such metaheuristics
- If some topological regularity in fitness landscape is detected, that regularity can be exploited and used for designing metaheuristic that will perform better than the alternatives
- Topology-based models and techniques already achieved good results in revealing **hidden structures** and detecting **new regularities**, so it can be expected that it will be helpful in this domain
- The most important topology (more precisely, algebraic topology) concepts in this domain are **simplicial complexes**, **homology groups** and **persistent homologies**.

"Canonical" GA and abstract simplicial complexes

- When looking at the individuals within "Canonical" GA population from topological point of view, it can be observed that they can be modeled as 0-simplices
- In other words, GA population forms a point cloud
- Previously described "classical" mutation in GA works only on top of 0-simplices and further uses 1-simplex neighborhoods for transitions during mutation
- That mutation changes individual representation to some other that is edge-connected with respect to given distance function

Example: 1-simplex structure

- Assume that:
  - the solutions have a binary representation with a length of 5
  - the distance between two solutions is given by the Hamming distance
  - maximal possible of the distance value is set to 1
  - there are two such individuals with representations 1$\underline{0}$11110 and 1$\underline{1}$11110
- For the previously defined distance function and individuals, we have two 0-simplices that are connected in the 1-simplex
- Therefore, a transfer (within the mutation) from the first solution to the second is possible.

Topology Genetic Algorithm

- Topology Genetic Algorithm (TGA) is designed as a generalization of GA that builds on m-simplex data, where for special case $m = 0$ that algorithm becomes a classical GA
- The main difference in TGA, in comparison to classical GA, are crossover and mutation operators
  - Within TGA, we try to create new individual within the solution space, so that individual forms a m-simplex with other $n - 1$ individuals from the current population
  - In GA, crossover and mutation are by the randomness. In TGA, the randomness is restricted with respect to parameter m
  - This means that for $m > 0$, the set of possible representations for the new individual, which is randomly chosen, now becomes smaller
  - If, for a given individual(s) and current simplex size m, the operator execution is not possible, the simplex size m is reduced by 1.

Projected Characteristics of the TGA

- The overall effect that we expect TGA will have on the search process in comparison to GA is increased preservation of the same or similar topological regularity through time (if this regularity exists)
- We also think that the expansion of already existing simplices, especially large ones, is well motivated - because the existence of regular formation of local optima itself is an indicator that more new local optima may be found around that formation
- Another important observation is that since TGA falls back to classical GA, we can expect that TGA will be generally applicable, i.e. if the topological regularity is low and cannot be exploited, TGA should work at least as good as classical GA (though performance might get deteriorated in the worst case)

Thank for the attention!

vladimir.filipovic@matf.bg.ac.rs