Signal processing and identification–AI module, Lecture 2

Evolutionary algorithms – preliminaries, selection of problems

dr inż. Ziemowit Dworakowski
AGH University of Science and Technology, Department of Robotics and Mechatronics

1

Agenda

- Principle of operation of a genetic algorithm
- Schematic of a genetic algorithm
- Basics of configuration

2

Natural evolution– how does it work?

Each individual is defined by its set of genes (a **genotype**) – based on which a set of features is developed (a **phenotype**).

Each individual wants to pass its genes – so it needs to survive its childhood and find a partner (so win a **selection** in competition with other individuals)

The better the genes, the higher chance of survival and breeding

The offspring of said individual will inherit slighlty modified (**mutated**) genes of both its parents (**crossover**)

…etc.

3

Basic genetic algorithm

Scheme of a basic genetic algorithm

Genarate randomly a **population of solutions (individuals)**

↓

generation

Assess their **fitness** – a value of objective function (a **fitness function**) corresponding to each individual

↓

Do a **selection** of a subset of the best individuals (with highest values of fitness)

↓

Mutate and **crossover** this subset creating new (**offspring**) population

Repat until no time is left or population stops improving

4

Genetic algorithm: coding and fitness

Coding (representation) and fitness assessment of an individual

Individual (A prescription to solve a problem)

↓

Genotype (Set of parameter values)

↓

Phenotype (Realization that is possible to assess – to calculate value of the OF)

↓

Fitness assessment (Value of OF corresponding to the individual)

5

Genetic algorithm: selection

Selection (succession)

Neccessary to determine which individuals will have a chance to pass its genes

n-best

Sort individuals by order of their fitness, select n best
(Easy to implement and configure, intuitive)

Rulette selection

...

Tournament selection

...

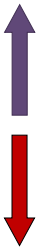
Elite selection

Select **k** best individuals and pass them **without** alteration to the next population (usually k is small, e.g. k = 1)
(Prevents from deterioration of result)

6

Genetic algorithm: selective pressure

Selective pressure tells us *„what chance of breeding have individuals with relatively small values of fitness"*



Higher selection pressure

- **Faster convergence**
- **Decreases diversity**
- **More sensitive to local minima**

Lower selection pressure

- **Slower**
- **Maintains diversity**
- **Less sensitive to local minima**

7

Genetic algorithm: mutation

Mutation

Allows for generation of new (possibly: better) solutions based on previous ones

Individual's genes (Parent's)		Wektor losowy (rather small)		New set of genes (offspring individual)
$[x_1,x_2,x_3,\ldots,x_n]$	+	$S^*[r_1,r_2,r_3,\ldots,r_n]$	=	$[x'_1,x'_2,x'_3,\ldots,x'_n]$

- All the genes are **slightly** modified
- Usually range of mutation (how long is the random vector) is changed using a consant called a **mutation step**

↓

Influences balance between **Exploration** and **Exploitation**

8

Genetic algorithm: exploration and exploitation

Exploration of a parameter space in initial phase of search

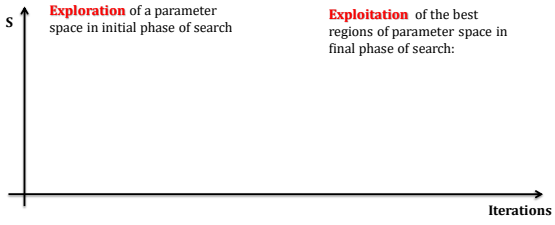
Exploitation of the best regions of parameter space in final phase of search:

Exploration requires:
- Large mutation step
- *(Small selective pressure)*
- *(Maintaining diversity)*

Exploitation requires:
- Small mutation step
- *(Small selective pressure)*
- *(Reduction of diversity)*

9

Exploration and exploitation: mutation step decrease



Exploration of a parameter space in initial phase of search

Exploitation of the best regions of parameter space in final phase of search:

Exploration requires:
- Large mutation step

Exploitation requires:
- Small mutation step

10

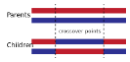
Genetic algorithm: crossover

Crossover

Allows to exchange genetic information between at least two individuals

One-point / multipoint

Select a point(or few of them) in a genome, change source of genes after each point



Uniform

Copy each gene randomly from any parent

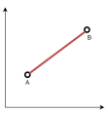
Image source: Wikipedia

11

Genetic algorithm: crossover

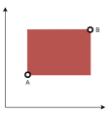
Crossover

Allows to exchange genetic information between at least two individuals



Line crossover

Offspring is generated along the line between two parents



Box crossover

Offspring is generated in a range defined by parent's genes

Image source: Heesch M, *Memetic algorithm for training of artificial neural networks in structural health monitoring*, Bachelor engineering thesis, AGH University of Science and Technology, 2016

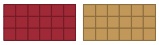
12

Genetic algorithm: crossover

Crossover

Allows to exchange genetic information between at least two individuals

parents



„Special" (modular) crossover

Variant of a many-point crossover, genes are copied from different parents in connected modules that each code disjoint features of a phenotype

offspring

Image source: Heesch M, *Memetic algorithm for training of artificial neural networks in structural health monitoring*, Bachelor engineering thesis, AGH University of Science and Technology, 2016

13

Diversity is good

- If the whole population is placed in one small area of parameter space, it can effectively be treated as just one individual („n-best", n = 1)

↓

- Leaving a local minimum is very challenging
- Information of a previously explored areas of parameter space is lost

How to maintain diversity of a population?

- We can add new, randomly generated individuals
- We can spread the population using mutations
- **We can pick a proper (not too large) selection pressure**
- **We can include diversity in fitness assessment**

14

General remarks

*In a simplification, 10 individuals in 100 iterations **could** do similar work as 100 individuals for 10 iterations*

BUT:

- In „difficult spaces" – highly dimensional or noisy we need large population to provide any progress
- Good exploitation requires usually many iterations

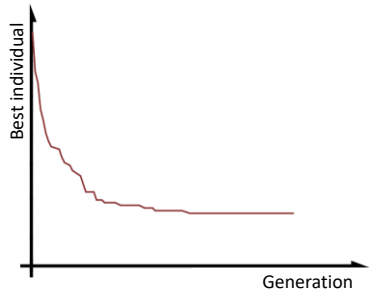
Genetic operators (mutation and crossover) should allow for obtaining any possible genome values (any available point in parameter space)

-**Non-deterministic method** – **Consecutive runs of the algorithm might provide different results**

- We need to test it many times and obtain **statistical information**
- We should use a **histogram of solutions**
- We should observe various **convergence curves**

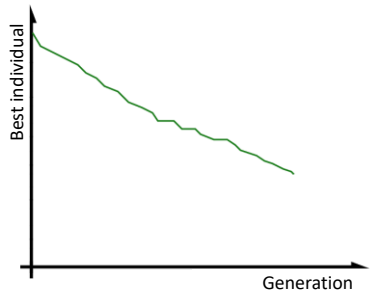
15

GA testing – convergence curve



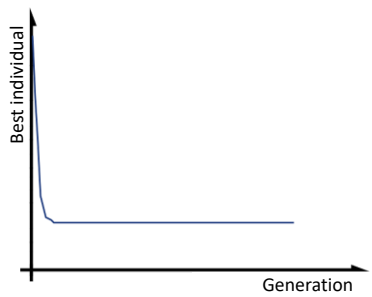
16

GA testing – convergence curve



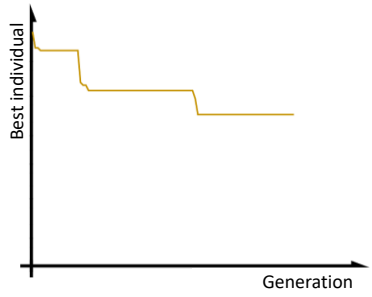
17

GA testing – convergence curve



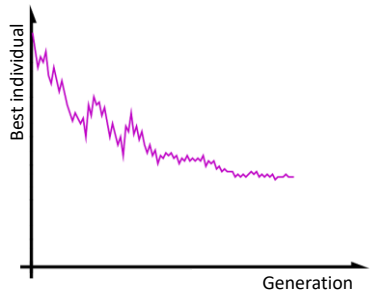
18

GA testing – convergence curve



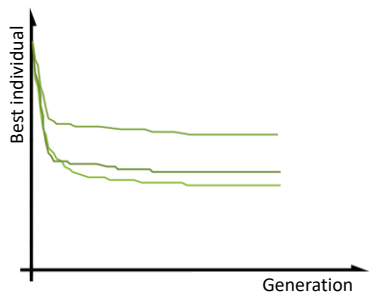
19

GA testing – convergence curve



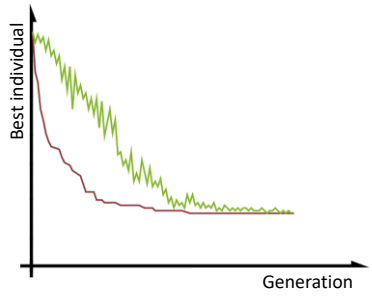
20

GA testing – convergence curve



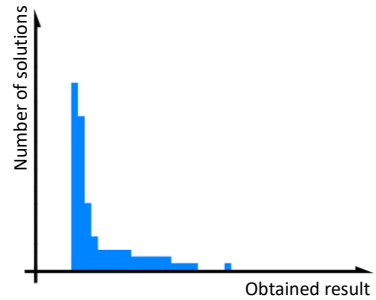
21

GA testing – convergence curve



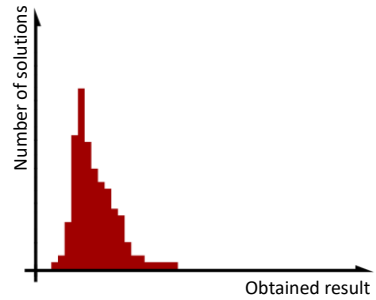
22

GA testing – histogram of solutions



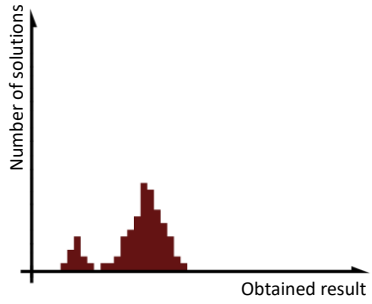
23

GA testing – histogram of solutions



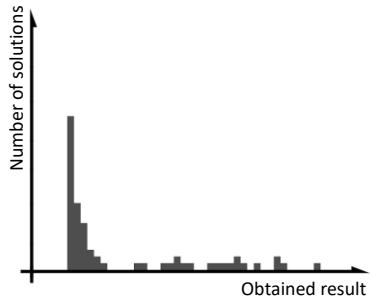
24

GA testing – histogram of solutions



25

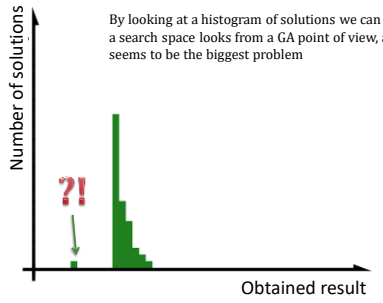
GA testing – histogram of solutions



26

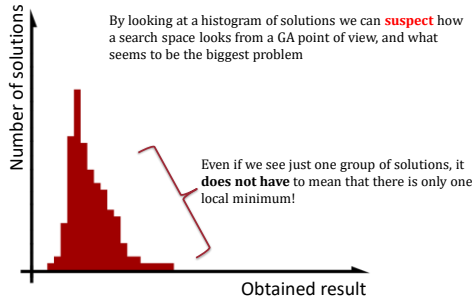
GA testing – histogram of solutions

By looking at a histogram of solutions we can **suspect** how a search space looks from a GA point of view, and what seems to be the biggest problem



27

GA testing – histogram of solutions



By looking at a histogram of solutions we can **suspect** how a search space looks from a GA point of view, and what seems to be the biggest problem

Even if we see just one group of solutions, it **does not have** to mean that there is only one local minimum!

28

Optimization algorithms - comparison

Algorithm	Implementation, configuration	Sensitivity to local minima	Sensitivity to dimensionality	Non-continuous OF	„flat" OF	Convergence speed	Overall efficiency
Grid search	😐	🙂	☹	🙂	🙂	☹	🙂
Random	🙂	🙂	☹	🙂	🙂	☹	☹
1+1	🙂	😐	🙂	🙂	🙂	🙂	🙂
Gradient descent	🙂	☹	🙂	☹	☹	😐	😐
Multistart gradient descent	😐	😐	🙂	☹	☹	😐	😐
Gradient with momentum	😐	😐	🙂	☹	😐	😐	🙂
Newton	😐	☹	😐	☹	😐	🙂	🙂

29

Signal processing and identification–AI module, Lecture 2

Evolutionary algorithms – preliminaries, selection of problems

1) How does a GA work?
2) What is the difference between genotype and phenotype?
3) What is an n-best selection?
4) What is an elite selection? Why do we use it?
5) Why and how does the mutation work?
6) What is exploration and exploitation?
7) Crossover methods
8) How to test GA?
9) How does GA fare in comparison with other methods?

30